ENUMERATION OF PERMUTATIONS INDEXING LOCAL COMPLETE INTERSECTION SCHUBERT VARIETIES

A Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

with a

Major in Mathematics

in the

College of Graduate Studies

University of Idaho

by

Masaki Ikeda

Major Professor
Alexander Woo, Ph.D.

Committee
Jennifer Johnson-Leung, Ph.D.
Terence Soule, Ph.D.
Stefan Tohaneanu, Ph.D.

Department Administrator
Christopher Williams, Ph.D.

June 2016

Authorization to Submit Dissertation

This dissertation of Masaki Ikeda, submitted for the degree of Doctor of Philosophy with a Major in Mathematics and titled "Enumeration of permutations indexing local complete intersection Schubert varieties," has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: ______________________ ____________
Alexander Woo, Ph.D. Date

Committee Members: ______________________ ____________
Jennifer Johnson-Leung, Ph.D. Date

______________________ ____________
Terence Soule, Ph.D. Date

______________________ ____________
Stefan Tohaneanu, Ph.D. Date

Department
Administrator: ______________________ ____________
Christopher Williams, Ph.D. Date

Abstract

We find the generating function for the permutation class $\mathcal{A}' = \text{Av}(52341, 53241, 52431, 35142, 42513, 351624)$. Partial motivation for this work comes from algebraic geometry. In particular, certain classes of Schubert varieties are indexed by permutations in some permutation classes. For example, as shown by Lakshmibai and Sandhya [29], a smooth Schubert variety is indexed by a permutation in the class $\text{Av}(3412, 4231)$, and a Schubert variety defined by inclusions (dbi for short) has an index in $\mathcal{A} = \text{Av}(4231, 35142, 42513, 351624)$, as shown by Gasharov and Reiner [20]. In addition to those results, Úlfarsson and Woo [38] showed Schubert varieties which are local complete intersections (lci for short) are indexed by permutations in $\mathcal{A}'$.

The enumeration of the permutations indexing smooth Schubert varieties was initially found by Haiman [22], and then also discussed by Bousquet-Mélou and Butler [14]. Furthermore, Albert and Brignall [7] discovered the enumeration of Schubert varieties defined by inclusions. This dissertation completes the enumeration of the class $\mathcal{A}'$ by extending the method used by Albert and Brignall to enumerate $\mathcal{A}$ in [7].

Acknowledgements

I couldn't have come this far academically without support provided by my family, friends, and former/current professors. The biggest appreciation goes to my advisor, Alexander Woo. He not only guided me to the right direction with his intelligence and knowledge, but also became a great connection to the community of permutation patterns. I could tell that he always cared about me and my career as a mathematician. One day, I hope to become a great inspiration to my own students.

I am truly thankful that I chose the University of Idaho for my graduate career. Teaching assistantship they provided me each year helped me highly, not only financially, but also for me to gain remarkable experiences as an instructor. Thanks to my committee members, Jennifer, Stefan and Terry, and to wonderful staff members, Jana, Melissa, Jaclyn and Stacey, and all other great professors who helped me to become a better scholar every day. Special thanks goes to Monte Boisen, the former department chair who supported me even after his retirement. My stay in Moscow was a truly amazing experience.

I would like to thank Western Oregon University for offering a great position to start my professional career in academia as well as my wonderful undergraduate experience. It is a shame that I had to have so many people wait for such a long time as I worked to finish my thesis. I appreciate all the support from Sue, Debbie and everyone in mathematics department here at WOU. Especially, Mike Ward, who first taught me the beauty of mathematics. Without his courses, I would have never stepped into the path I took.

Permutation patterns community was truly supportive. I met wonderful scholars who were helpful and motivating to complete my thesis. Thanks to Michael Albert, who gave me such kind words to push me through the toughest time as well as his extremely helpful program, PermLab to visualize my study. I am very happy to join such friendly community.

Lastly, I want to thank my host family in Oregon and friends. I am grateful to have so many caring people surrounding me in my life. Jonny Olson, my best friend, was a true motivation and distraction throughout my graduate career. Congratulations to him for achieving Ph.D. degree at Louisiana State University. Thanks to Veronica, Frank, Emily, Josh, Amanda and Caitlin, my second family since 2003, and my best friends, Jeff, Sierra, Ben, Jesse, Jenna, Hannah, Trevor, Morgan, Josh, Don, Mallory, Veronica, Diadra, Karissa and the ones that I shamefully forgot.

Dedication

To my mother, Yuriko, and my father, Katsunobu.

Table of Contents

List of Figures

List of Tables

Chapter 1.

Introduction

1.1. Result

Denote by Av(B) the set of all permutations avoiding every permutation in B. One essential study of permutation classes is to find the generating function for Av(B) with a particular set of permutations B. Our final goal is to prove the following theorem.

Theorem. *The generating function for the class $\mathcal{A}'$ is defined by*

$$f_{\mathcal{A}'} = \frac{\sum_{i=0}^{5} a_i \bar{G}^i}{\sum_{i=0}^{6} b_i \bar{G}^i}$$

where $\bar{G} = G - 1$ and G is the generating function satisfying the equation

$$\bar{G} = 1 + \frac{x\bar{G}}{1 - x\bar{G}^2},$$

and

$$\begin{aligned}
a_0 &= -1 + 14x - 39x^2 + 28x^3 + 9x^4 - 11x^5 + x^6,\\
a_1 &= -12 + 81x - 100x^2 + 15x^3 + 46x^4 - 19x^5,\\
a_2 &= -8 + 35x - 20x^2 - 25x^3 + 31x^4 - 6x^5 - x^6,\\
a_3 &= 7, \qquad a_4 = 1, \qquad a_5 = -2.
\end{aligned}$$

$$\begin{aligned}
b_0 &= -1 + 57x - 125x^2 + 143x^3 - 48x^4 - 64x^5 + 51x^6 - 2x^8,\\
b_1 &= -54 + 260x - 386x^2 + 250x^3 + 81x^4 - 226x^5 + 74x^6 + 15x^7 - 3x^8,\\
b_2 &= -18 + 114x - 104x^2 - 22x^3 + 148x^4 - 123x^5 + 11x^6 + 14x^7 - x^8,\\
b_3 &= 24, \qquad b_4 = -2, \qquad b_5 = -5, \qquad b_6 = 1.
\end{aligned}$$

In this chapter, we introduce history of permutation class study as well as the place of dissertation in the literature. Detailed definitions are given in the next chapter.

1.2. History

The concept of permutation avoidance first appeared in the literature in 1915. In [30], MacMahon proved that the number of permutations of length n which can be partitioned into two decreasing subsequences is the Catalan numbers. Also in 1935, Erdős and Szekeres [17] showed that, given two positive integers a, b and a sequence of n real numbers $x = x_1, \ldots, x_n$ with $n = ab + 1$, x either contains a strictly increasing subsequence of length $a + 1$ or a strictly decreasing subsequence of length $b + 1$. Despite these early results, we consider the study of permutation classes to have begun in 1968 with Knuth's *The Art of Computer Programming* [25]. Knuth proved a permutation π is stack-sortable if and only if π avoids 231, and stack-sortable permutations are also counted by the Catalan numbers. The result of Knuth brought up the notion of permutation avoidance and the study of permutation classes. Within two decades of Knuth's contribution, many enumeration results were discovered by various researchers. In particular, Simion and Schmidt [36] summarized permutation classes avoiding permutations of length 3.

Based on some earlier results, Stanley and Wilf separately conjectured in the late 1980s that, for any permutation π, there exists a constant C_π whose n-th power is an upper bound for the number of permutations of length n avoiding π. In other words, while the growth rate of the set of length n permutations is factorial, having a restriction of avoiding an arbitrary single permutation reduces the growth rate to be exponential. This is known as the Stanley-Wilf conjecture. Some partial results were proved by Bóna [12] and Alon together with Friedgut [9]. A breakthrough was made by Klazar in 2000 [24], when he showed that the Füredi-Hajnal conjecture implies the Stanley-Wilf conjecture. The Stanley-Wilf conjecture remained unproven for almost two decades until Marcus and Tardos proved it in 2004 by proving the Füredi-Hajnal conjecture [31].

Although the proof of Marcus and Tardos gives an upper bound for the growth rate constant C_π depending on the length of π, the precise growth rates for most permutation classes are still unknown. Hence, one of the main problems in present research is to characterize the growth rates of permutation classes. Pratt [35] as well as Spielman together with Bóna [37] showed the existence of permutation classes containing infinite antichains. Since such a class contains uncountably many distinct subclasses, their result implied there exist uncountably many distinct growth rates. Then in [23], Kaiser and Klazar showed the only possible growth rates of any permutation class that is less than 2 are positive solutions to $1 - 2x^k + x^{k+1}$ for some $k \geq 0$. Later, Klazar also showed there are only countably many permutation classes with growth rate less than 2. As an extension of this result, Vatter showed two noteworthy results in [39]: The smallest possible growth rate greater than 2 for a permutation class is the unique positive root of $1 + 2x + x^2 + x^3 - x^4$, which is approximately 2.06599, and the smallest growth rate for which there are uncountably many permutation classes is the unique positive root of $1 + 2x^2 - x^3$, which is approximately 2.20557.

Another notable contribution to the study of growth rates was made by Fox in 2014 [18]. It was believed that the growth rates of permutation classes avoiding a single permutation of length n grows at most quadratically. In [18], Fox showed this is false, but the function $g(k) = \max\{\mathcal{C}_\pi : |\pi| = k\}$ grows mildly exponentially.

While understanding growth rate constants is the primary interest of many researchers, classifying necessary and sufficient conditions for a permutation class to have a specific type of generating function is also an important question. For example, in [5], authors defined the notion of geometric griddable class, and showed every geometrically griddable class has a rational generating function. In 1996, Noonan and Zeilberger conjectured that the generating function of a finitely based permutation class is D-finite [33], *i.e.,* it is the solution to some differential equation with polynomial coefficient. However, Zeilberger himself later conjectured to the contrary that the generating function counting the permutations avoiding 1324 is not D-finite [16]. This, the Noonan-Zeilberger conjecture, was disproved by Garrabrant and Pak in 2015, who give a general method for generating counterexamples [19].

As we briefly mentioned, the generating function for the permutation class avoiding 1324 remains unknown. In fact, this is the only class (up to symmetry) avoiding a single permutation of length 4 that has not been enumerated. To illustrate the difficulty of finding the generating function for this class, Zeilberger stated “Not even God knows $|\mathrm{Av}_{1000}(1324)|$,” where $|\mathrm{Av}_{1000}(1324)|$ is the number of permutations of length 1000 avoiding 1324. On the other hand, Steingrímsson disagrees with Zeilberger by saying “I’m not sure how good Zeilberger’s God is at math, but I believe that some humans will find this number in the not so distant future.” Perhaps, our ultimate goal is to prove Steingrímsson’s claim is correct. The growth rate of this class is also unknown. Currently, the best known lower bound of the growth rate is approximately 9.81, as shown by Bevan [10], and the best known upper bound is approximately 13.74, as shown by Bóna [13].

In order to find the generating functions for more permutation classes as well as for the one of permutations avoiding 1324, researchers have worked on establishing many enumeration techniques which can be applied to specific classes. In particular, with the notion of simple permutation introduced in [2, 4], we can find the generating functions for certain classes by enumerating simple permutations in these classes first. Also, authors of [5] discovered concrete enumeration methods for geometric griddable classes. In [7], Albert and Brignall combine these two ideas to find the generating function for the class avoiding permutations 4231, 35142, 42513 and 351624.

1.3. Place of dissertation in the literature

There are objects known as Schubert varieties in algebraic geometry, and certain types of Schubert varieties are indexed by permutations in some permutation classes. For instance, as shown by Gasharov and Reiner [20], permutations in the class Albert and Brignall enumerated in [7] index Schubert varieties defined by inclusions. Also, Lakshmibai and Sandhya [29] showed that smooth Schubert varieties are indexed by permutations avoiding 4231 and 3412, and Úlfarsson and Woo [38] proved local complete intersection ones are indexed by permutations avoiding 52341, 53241, 52431, 35142, 42513 and 351624. The enumeration of the permutations indexing smooth Schubert varieties was initially found by Haiman [22], and then also discussed by Bousquet-Mélou and Butler [14]. In this dissertation, we enumerate permutations indexing Schubert varieties that are local complete intersections.

As discussed in the previous section, three aspects of the study of permutation classes are growth rates, relations between types of generating functions and classes, and enumeration techniques. With this dissertation, we contribute to all of these aspects. Since we prove our final result by extending the methods Albert and Brignall used in [7], we primarily contribute to the study of enumeration techniques. In the future, we plan to use our result to characterize the permutation classes to which these methods can be applied. In addition, we can see that the generating function we obtain, like the one in [7], is algebraic. Although we don't have enough evidence to establish any conjecture about the types of generating functions related to classes that can be enumerated by methods we use, there may be some possible connections.

In Chapter 2, we define necessary terminology and provide examples to understand the study of permutation classes. Chapter 3 specifically covers two examples of enumeration. These results are stated as lemmas and will be referred in Chapter 6. We spend the entirety of Chapter 4 to describe the methods which Albert and Brignall used in [7] in detail. By extending this idea in Chapter 5 and 6, we complete our final result.

For a more detailed history of the study of permutation classes, we refer the reader to the excellent survey of Vatter [40].

Chapter 2.

Definitions and prerequisites

2.1. Permutations and permutation classes

2.1.1. Permutations

A *permutation* π is a bijective function from $\{1, 2, \ldots, n\}$ to itself for some positive integer n, which is called the *length of* π, denoted by $|\pi|$. We call an integer in the domain of π a *position*, and an integer in the image of π a *value*. In this dissertation, we will write permutations in one-line notation; a permutation π will be written as a sequence of n positive integers $\pi_1 \ldots \pi_n$, indicating for each $i \in \{1, 2, \ldots, n\}$ that $\pi(i) = \pi_i$. (We usually write $\pi(i)$ instead of π_i to refer to the value in the i-th position of π). For example, $\pi = 316254$ is a permutation of length 6, and $\pi(4) = 2$. Let $\mathcal{S}$ and $\mathcal{S}_n$ be the set of all permutations and the set of all permutations of length n, respectively. (For $n = 0$, $\mathcal{S}_0$ is the set containing only the empty permutation, which is denoted by ε). One can easily verify that $|\mathcal{S}_n|$, the number of length n permutations, is $n!$.

Here, we also introduce the *graph of a permutation*, as this idea will help to illustrate some important concepts. Given a permutation of length n, we draw a point at $(i, \pi(i))$ on the Cartesian plane for each i with $1 \leq i \leq n$. Note that we always have a unique point on each of $x = 1, \ldots, x = n$ and $y = 1, \ldots, y = n$ lines. We then draw an $n \times n$ grid containing these points. Figure 2.1 shows the graph of $\pi = 316254$.

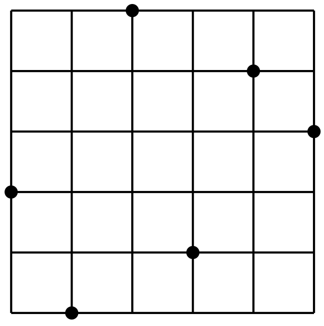
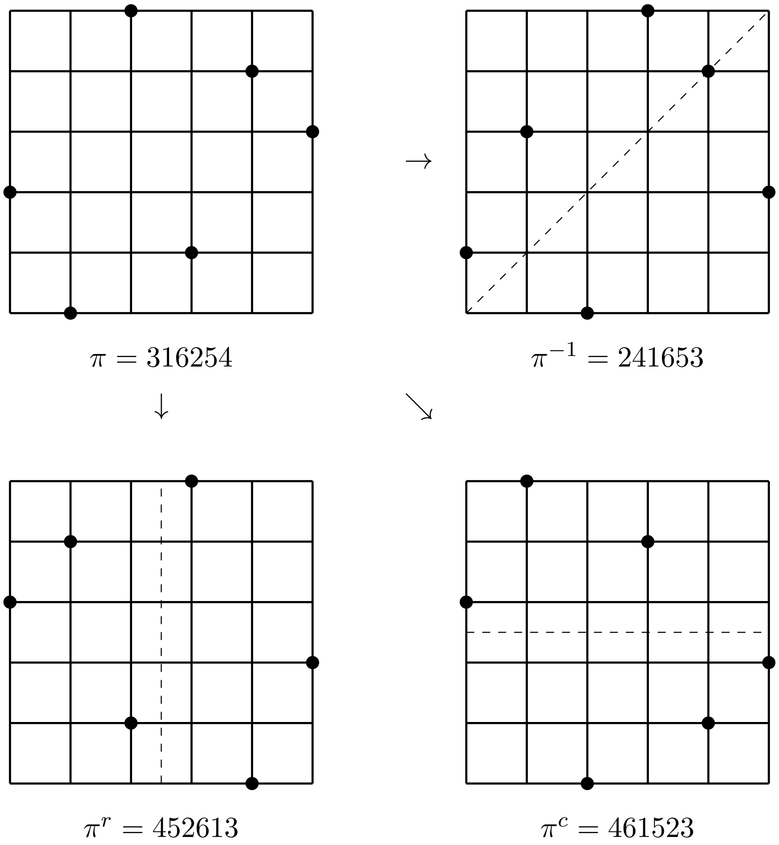


Figure 2.1.: The graph of the permutation $\pi = 316254$.

We frequently abuse notation and think of values as points in the graph. For instance, by saying the value x of π is located below and to the left of y, we mean $x < y$ and $\pi^{-1}(x) < \pi^{-1}(y)$.

2.1.2. Constructions of new permutations

In this section, we introduce several methods to construct other permutations from a given permutation. Let $\pi \in \mathcal{S}_n$. For every i $(1 \leq i \leq n)$, the *inverse of* π is the permutation $\pi^{-1} \in \mathcal{S}_n$ such that $\pi(\pi^{-1}(i)) = \pi^{-1}(\pi(i)) = i$, the *reverse of* π is the permutation $\pi^r \in \mathcal{S}_n$ given by $\pi^r(i) = \pi(n+1-i)$, and the *complement of* π is the permutation $\pi^c \in \mathcal{S}_n$ defined by $\pi^c(i) = n+1-\pi(i)$. Examples of each operation are shown in Figure 2.2 with $\pi = 316254$.



$\pi = 316254$

$\pi^{-1} = 241653$

$\pi^r = 452613$

$\pi^c = 461523$

Figure 2.2.: The graphs of π^{-1}, π^r and π^c.

As we can observe, the inverse, the reverse, and the complement of a permutation π can be obtained by reflecting π respectively over SW-NE diagonal, vertical, and horizontal lines on the graph of π. We denote by $\mathrm{Sym}(\pi)$ the set of permutations obtained by applying compositions of these operations and call this set the *symmetry class of* π. In other words, $\mathrm{Sym}(\pi)$ is the orbit of the dihedral group of order 8 acting on the graph of π. For $\pi = 316254$, one can easily verify that $\mathrm{Sym}(316254) = \{316254, 241653, 452613, 461523, 421635, 356142, 325164, 536124\}$.

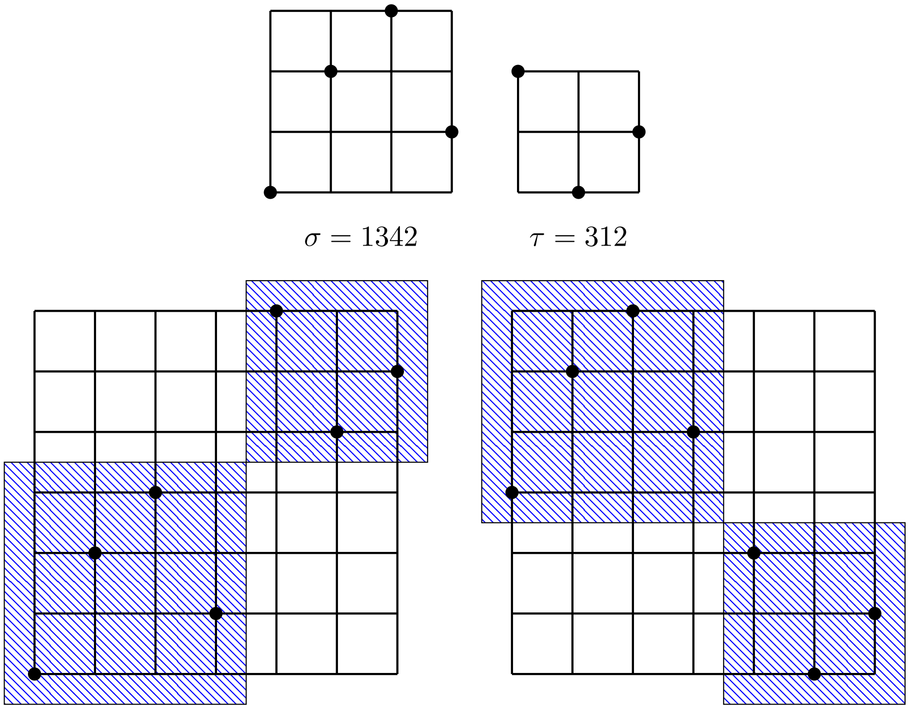
The previous three operations construct a new permutation of the same length from a given permutation. Next, we introduce two ways to "glue" two permutations together to construct a lager permutation. Given $\sigma \in \mathcal{S}_m$ and $\tau \in \mathcal{S}_n$, the *sum of* σ *and* τ is the permutation defined by

$$\sigma \oplus \tau = \sigma(1)\sigma(2)\cdots\sigma(m)\tau'(1)\tau'(2)\cdots\tau'(n)$$

where $\tau'(i) = \tau(i) + m$ for each i with $1 \leq i \leq n$. Similarly, we define the *skew-sum of* σ *and* τ to be the permutation

$$\sigma \ominus \tau = \sigma'(1)'\sigma(2)\cdots\sigma'(m)\tau(1)\tau(2)\cdots\tau(n)$$

where $\sigma'(i) = \sigma(i) + n$ for each i with $1 \leq i \leq n$. Figure 2.3 shows the sum and skew-sum of $\sigma = 1342$ and $\tau = 312$. Note that neither of these sum operations is commutative, but they are associative. For instance, while $\sigma \oplus \tau = 1342756$, we have $\tau \oplus \sigma = 3124675$.



$\sigma = 1342$ $\tau = 312$

$\sigma \oplus \tau = 1342756$ $\sigma \ominus \tau = 4675312$

Figure 2.3.: The graphs of $\sigma \oplus \tau$ and $\sigma \ominus \tau$.

If π can be constructed by a sum $\sigma_1 \oplus \sigma_2$ for some nonempty permutations σ_1 and σ_2, then we say π is *sum-decomposable*. Similarly, if $\pi = \sigma_1 \ominus \sigma_2$ for some nonempty permutations σ_1 and σ_2, then we say π is *skew-decomposable*. More importantly, if π is not sum-decomposable (respectively skew-decomposable), *i.e.*, there does not exist two permutations σ_1 and σ_2 such that $\pi = \sigma_1 \oplus \sigma_2$ (respectively $\pi = \sigma_1 \ominus \sigma_2$), then we say π is *sum-indecomposable* (respectively *skew-indecomposable*).

Lastly, there is another method, called inflation, to construct a new permutation that will be important in this dissertation, but it makes more sense to introduce this construction when we discuss simple permutations, so we postpone discussing it until Section 1.3.

2.1.3. Permutation avoidance and permutation classes

We now introduce the concept of permutation avoidance. The *flattening* of a sequence of n distinct positive real numbers is the unique permutation σ of length n where σ has the same relative order as the sequence. For example, the flattening of 364 is $\sigma = 132$. If a permutation π has a subsequence (not necessarily consecutive) whose flattening is σ, we say the *pattern σ is contained in* π and write $\sigma \preceq \pi$, since pattern containment is a partial order. Notice that the permutation $\pi = 316254$ has a subsequence 364, so $\sigma = 132$ is contained in π. More importantly, if σ is not contained in π, we say π *avoids σ pattern* and write $\sigma \not\preceq \pi$. With the same permutation π for example, π avoids $\tau = 4231$ because π does not contain a subsequence of four positive integers whose flattening is τ.

We can visualize the concepts of containment and avoidance with graphs of permutations. In the graph of $\pi = 316254$, if we take the points corresponding to the subsequence 364 and disregard all the others, we can construct the graph of $\sigma = 132$ by removing the lines with no points and squeezing together the lines that remain into a 3×3 grid, as shown in Figure 2.4. On the other hand, there is no subset of points that we can choose to construct $\tau = 4231$ in this way, and hence, $\tau \not\preceq \pi$.

Define a *permutation class* to be a set $\mathcal{C}$ of permutations with the property that, if $\pi \in \mathcal{C}$ and $\sigma \preceq \pi$, then $\sigma \in \mathcal{C}$. The set of permutations β that are minimal (with respect to containment) among those not in $\mathcal{C}$ is called the *basis of $\mathcal{C}$*. Let B be a set of permutations and $\text{Av}(B)$ be the set of permutations avoiding every permutation in B. We call this set the *permutation class of B*. Note that if B is the basis of $\mathcal{C}$, then we have

$$\mathcal{C} = \text{Av}(B) = \{\pi : \beta \not\preceq \pi \text{ for all } \beta \in B\}.$$

A trivial example is $\mathcal{C} = \text{Av}(21)$ (we usually omit superfluous braces). It is a straightforward exercise to see that for each positive integer n, $12 \ldots n$ is the only permutation in $\mathcal{S}_n$ that avoids 21. Thus, $\mathrm{Av}(21) = \{\varepsilon, 1, 12, 123, 1234, \ldots\}$.

We conclude this section by introducing some basic properties of permutation classes. The definition of a permutation class immediately implies the following proposition.

Proposition 2.1 *Let B_1 and B_2 be distinct sets bases of permutation classes. If, for every $\beta_2 \in B_2$, there exists $\beta_1 \in B_1$ such that $\beta_1 \preceq \beta_2$, then $Av(B_1) \subsetneq Av(B_2)$.*

For example, $\mathrm{Av}(132, 213) \subsetneq \mathrm{Av}(132, 3142)$ since $132 \preceq 132$ and $213 \preceq 3142$.

The following result is less obvious but not too difficult to prove.

Proposition 2.2 *Let op be a fixed composition of the inverse, reverse and complement operations. Given a basis B, define $op(B) = \{op(\beta) : \beta \in B\}$. Then $\pi \in Av(B)$ if and only if $op(\pi) \in Av(op(B))$. Hence, $|Av(B)| = |Av(op(B))|$.*

2.2. Generating functions

Given an arbitrary class $\mathcal{C}$, we are interested in discovering a formula called a generating function which expresses the number of length n permutations in $\mathcal{C}$, if possible. In this section, we define the notion of generating function for an integer sequence and discuss a few enumerative results which have been found in recent years.

Suppose we have a sequence of positive integers $\{a_n\}_{n=0}^{\infty}$. The *generating function for* $\{a_n\}_{n=0}^{\infty}$ is a formal power series of the form

$$f = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots .$$

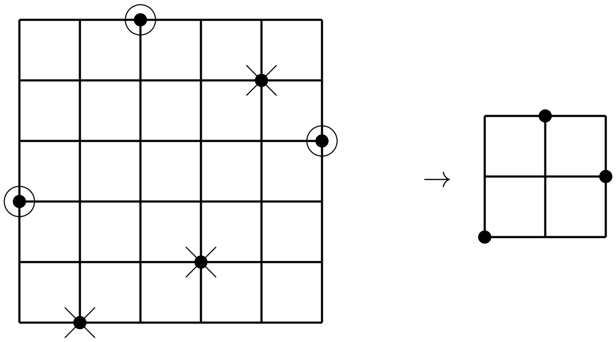


Figure 2.4.: The graph describing $132 \preceq 316254$.

In particular, a generating function for a permutation class $\mathcal{C}$ is a formal power series $f_{\mathcal{C}}$ defined by

$$f_{\mathcal{C}} = \sum_{n=0}^{\infty} s_n(\mathcal{C}) x^n,$$

where $s_n(\mathcal{C}) = |\mathcal{S}_n \cap \mathcal{C}|$ (or simply s_n, if it is clear from context). In other words, for each $i \geq 0$, the coefficient of x^i is the number of length i permutations in $\mathcal{C}$. For instance, the generating function for the class Av(21) is

$$f_{\mathrm{Av}(21)} = \frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + \cdots .$$

It is sometimes convenient (and necessary) to exclude the constant term 1, which corresponds to the empty permutation ε. In this dissertation, we let $\bar{f} = f - 1$.

We will also need the concept of multivariate generating functions. The *multivariate generating function* with i variables for a sequence of positive integers $\{a_{n_1,n_2,\ldots,n_i}\}_{n_1,n_2,\ldots,n_i \geq 0}$ is a formal power series of the form

$$f = \sum_{n_1,n_2,\ldots,n_i \geq 0} a_{n_1,n_2,\ldots,n_i} x_1^{n_1} x_2^{n_2} \cdots x_i^{n_i}.$$

Once we discover the generating function for a permutation class, we are often able to find the growth rate of the class. As the Stanley-Wilf conjecture states, any permutation class has exponential growth. Hence, by applying the Ratio Test and Taylor's Theorem from complex analysis, the growth rate is the reciprocal of the distance from 0 to the closest pole (*i.e.* the radius of convergence). In particular, gr($\mathcal{C}$), the growth rate of a permutation class $\mathcal{C}$, is obtained by

$$\mathrm{gr}(\mathcal{C}) = \lim_{n \to \infty} \frac{|a_{n+1}|}{|a_n|} = \frac{1}{R},$$

where R is the radius of convergence of the generating function for $\mathcal{C}$.

There are many different techniques for finding generating function of permutation classes. We postpone examples of finding generating functions until Chapter 3, where we show how to enumerate two classes, Av(123, 213, 132) and Av(4123, 4213, 4132). In the remainder of this section, we will discuss some historical enumerative results.

Since the enumerative study of permutation classes blossomed in 1980s, generating functions for various classes have been discovered. The first well known result is the following.

Theorem 2.3 *For every permutation β in $\mathcal{S}_3$, the number of permutations of length n in $Av(\beta)$ is the n-th Catalan number. Hence,*

$$f_{\mathrm{Av}(\beta)} = \frac{1-\sqrt{1-4x}}{2x} = 1 + x + 2x^2 + 5x^3 + 14x^4 + 42x^5 + \cdots$$

For $\beta = 123$ and $\beta = 231$, the above result was classically known [26,30]. In [36], Simion and Schmidt summarize this result as well as various results for bases having more than one permutation of length 3, including $\mathrm{Av}(123, 213, 132)$, which we introduce in Chapter 3.

In the 1990s, researchers discovered numerous results with bases containing permutations of length 4 [11,21]. In [11], Bóna not only finds the generating function for $\mathrm{Av}(1342)$, but furthermore finds the exact formula for $s_n(1342)$ for every n by giving a bijection between permutations in $\mathrm{Av}(1342)$ and labeled plane trees of a certain type on n vertices. The class $\mathrm{Av}(3412, 4231)$, the set of permutations indexing smooth Schubert varieties, was also enumerated in the early 1990s by Haiman [22]; this result was also discussed in [14]. Other various results are also discussed in [3, 6], and there are many more.

In addition to these results, researchers have recently established some concrete enumeration techniques which can be applied to certain types of classes. In [15], Brignall summarizes a variety of techniques coming from simple permutations (introduced in the next section). Also, [5] defines so-called geometric grid classes and introduces some techniques that can be used to enumerate this type of class. In Chapter 3, we investigate the method Albert and Brignall use to enumerate the class $\mathcal{A}$ in [7].

2.3. Simple permutations

2.3.1. Definition

We now move onto the discussion of simple permutations. First, we define two kinds of intervals. Given a permutation π of length n, a *segment* of π is a set of consecutive positions $i, i+1, \ldots, j$ in π, and a *range* of π is a set of consecutive values $a, a+1, \ldots, b$ of π. We use the standard notations of intervals for them. Although permutations only contain positive integer values, we may sometimes use open intervals to exclude boundary positions or values. For these intervals, we carry the notion of length from a permutation to denote the number of elements in $[i, j]$, which is simply $j-i+1$. We call a segment and a range that are not $[1, n]$ *proper*.

Now, we define simple permutations. Let $[i, j]$ be a segment of π whose length is m. If the set of values $\{\pi(i), \ldots, \pi(j)\}$ is an interval $[a, b]$, then $[i, j]$ is called a *block of* π. In this case, we may denote by $\pi([i, j])$ the corresponding range to the segment $[i, j]$ forming a block. Every permutation of length n has n singleton blocks as well as the block $[1, n]$. If a permutation π only contains these blocks, π is called *simple*. By convention, $\pi = 1$ is not considered simple. As an example and a non-example of simple permutations, suppose we have $\pi = 25314$ and $\sigma = 4127563$. Notice that σ contains segments $[2, 3]$ and $[4, 6]$ which are mapped to ranges $[1, 2]$ and $[5, 7]$ by π respectively. Thus, they are blocks of length 2 and 3, so σ is not simple. On the contrary, since the only blocks π contains are singletons and the block $[1, 5]$, π is a simple permutation. Given a permutation class $\mathcal{C}$, we denote by $\mathrm{Si}(\mathcal{C})$ the set of simple permutations in $\mathcal{C}$.

2.3.2. Inflation

An alternative definition of simple permutations can be given by introducing the following method to construct a new permutation.

Let π be a permutation of length n and $\sigma_1, \sigma_2, \ldots, \sigma_n$ be n non-empty permutations of various lengths. Denote by i_j the length of σ_j for every j with $1 \leq j \leq n$, and let $i_0 = 0$. Finally, define intervals $I_j = [i_0 + i_1 + \cdots + i_{j-1} + 1, i_0 + i_1 + \cdots + i_j]$ for each j with $1 \leq j \leq n$. The *inflation of* π *by* $\sigma_1, \sigma_2, \ldots, \sigma_n$ is the permutation α of length $i_1 + \cdots + i_n$, denoted by $\pi[\sigma_1, \sigma_2, \ldots, \sigma_n]$, where

1. For every j $(1 \leq j \leq n)$, there exists a constant t such that

$$(\alpha(i_0 + \cdots + i_{j-1} + 1) - t) \cdots (\alpha(i_0 + \cdots + i_j) - t) = \sigma_j.$$

2. For distinct j and k $(1 \leq j, k \leq n)$, if $\ell \in I_j$, $m \in I_k$ and $\pi(\ell) \leq \pi(m)$, then $\alpha(\ell) \leq \alpha(m)$.

In other words, $\alpha = \pi[\sigma_1, \sigma_2, \ldots, \sigma_n]$ is the permutation obtained by replacing each $\pi(j)$ with a block whose flattening is σ_j, so that for every distinct j and k, the relative ordering of $\alpha(\ell)$ and $\alpha(m)$ is the same as the relative ordering of $\pi(\ell)$ and $\pi(m)$. An example of inflation is shown in Figure 2.5, as this concept is best illustrated using graphs.

If the only ways to obtain a permutation π by inflation are $\pi[1, 1, \ldots, 1]$ and $1[\pi]$, then π is simple. Hence, as discussed previously and shown in Figure 2.5, $\sigma = 4127563$ is not simple, whereas $\pi = 25314$ is simple.

2.3.3. The importance of simple permutations

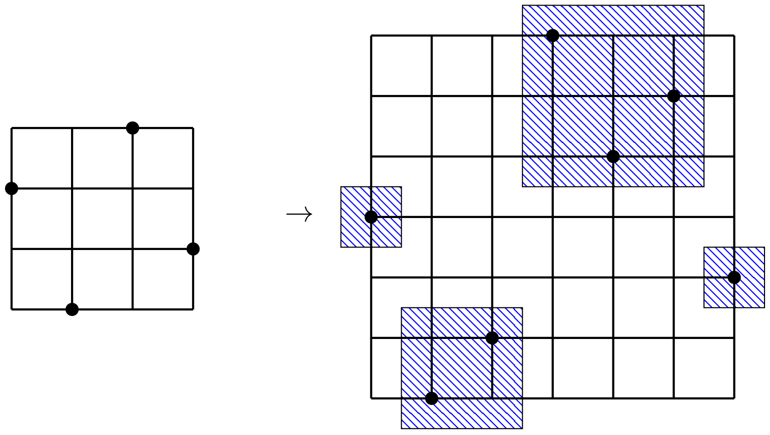
The notion of simple permutations was introduced in [4], where the authors describe how this notion is useful in the study of various permutation classes. The most essential proposition in [4] for our purpose is the following.

Proposition 2.4 *(Albert and Atkinson, 2005) For every permutation α, there exists a simple permutation π of some length n and permutations $\sigma_1, \sigma_2, \ldots, \sigma_n$ such that*

$$\alpha = \pi[\sigma_1, \sigma_2, \ldots, \sigma_n].$$

Furthermore, if $\pi \neq 12, 21$, then the permutation π is uniquely determined by α. If $\pi = 12$ or 21, then σ_1, σ_2 are uniquely determined so long as we require that σ_1 is sum-indecomposable or skew-indecomposable respectively.

In other words, any permutation α can be decomposed as the inflation of a unique simple permutation π by $\sigma_1, \ldots, \sigma_n$. In this case, π is called the *skeleton of* α. As stated, we must ensure that $|\pi| \neq 2$ (so $|\pi| \geq 4$, since there are no simple permutations of length 3) or, equivalently, that α is sum/skew-indecomposable, to say $\sigma_1, \ldots, \sigma_n$ are also uniquely determined. For example, $1234 = 12[1, 123] = 12[12, 12] = 12[123, 1]$. We can enforce uniqueness for sum/skew-decomposable permutations by insisting σ_1 be sum-indecomposable, so $1234 = 12[1, 123]$. Hence with Proposition 2.4 being analogous to the Fundamental Theorem of Arithmetic, simple permutations are to permutations as prime numbers are to integers. This essential idea often helps in discovering more general methods for finding generating functions for certain classes. In particular, we have the following propositions.



3142

$3142[1, 12, 312, 1] = 4127563$

Figure 2.5.: The inflation of 3142 by 1, 12, 312 and 1.

Proposition 2.5 *Let $\mathcal{C}$ be a permutation class, π be a simple permutation of length $n \geq 4$ in $\mathcal{C}$ and $\mathit{ifl}_{\mathcal{C}}(\pi)$ be the set of permutations in $\mathcal{C}$ which can be inflated from π. Suppose there exist subclasses $\mathcal{C}_i$ of $\mathcal{C}$ (for each i, $1 \leq i \leq n$) such that $\pi[\sigma_1, \ldots, \sigma_n] \in \mathcal{C}$ if and only if $\sigma_i \in \mathcal{C}_i$. Then the generating function for $\mathit{ifl}(\pi)$ is*

$$f_{\mathrm{ifl}_{\mathcal{C}}(\pi)} = \prod_{i=1}^{n} \bar{f}_{\mathcal{C}_i}.$$

If $\pi = 12$ (respectively $\pi = 21$) satisfies the above hypothesis, then

$$f_{\mathrm{ifl}_{\mathcal{C}}(12)} = \bar{f}^{\oplus}_{C_1} \cdot \bar{f}_{C_2} \qquad \text{(respectively } f_{\mathrm{ifl}_{\mathcal{C}}(21)} = \bar{f}^{\ominus}_{C_1} \cdot \bar{f}_{C_2})$$

where $\bar{f}^{\oplus}_{C_1}$ (respectively $\bar{f}^{\ominus}_{C_1}$) is the generating function for sum-indecomposable (respectively skew-indecomposable) permutations in C_1, excluding the empty permutation.

Proposition 2.6 *Let $\mathcal{C}$ be a permutation class. If Proposition 2.5 is applicable for all simple permutations in $\mathcal{C}$, then the generating function for $\mathcal{C}$ is*

$$f_{\mathcal{C}} = \sum_{\pi \in \mathrm{Si}(\mathcal{C})} f_{\mathrm{ifl}_{\mathcal{C}}(\pi)}.$$

Proposition 2.5 states that if every set of choices for each σ_i forms a subclass of $\mathcal{C}$ independently, then by the combinatorial meaning of multiplication together with the definition of generating functions, the generating function for $\mathrm{ifl}_{\mathcal{C}}(\pi)$ can be obtained by multiplying each $\bar{f}_{\mathcal{C}_i}$. If the basis of a class $\mathcal{C}$ contains only simple permutations, then $\mathcal{C}_i = \mathcal{C}$ for all i, so the hypothesis for Proposition 2.5 is automatically satisfied.

If Proposition 2.5 is applicable to every simple permutation in $\mathcal{C}$, then by the combinatorial meaning of addition and the definition of generating functions, Proposition 2.6 is an immediate consequence.

In [2], authors also discuss the asymptotic result stated in Theorem 2.7 as well as Theorem 2.8, which is a strong result about generating functions for permutation classes containing finitely many simple permutations.

Theorem 2.7 *(Albert and Atkinson, 2003 [4]) Let p_n be the number of simple permutations of length n. Then*

$$p_n = \frac{n!}{e^2}\left(1 - \frac{4}{n} + \frac{2}{n(n-1)} + O(n^{-3})\right).$$

Theorem 2.8 *(Albert and Atkinson, 2005* [2]*) If a permutation class contains only finitely many simple permutations, then it has a finite basis and an algebraic generating function. Furthermore, if such a class does not contain the permutation* $n(n-1)\cdots 321$ *for some* n*, then it has a rational generating function.*

Theorem 2.8 is not applicable to $\mathcal{A}$ and $\mathcal{A}'$, the classes we enumerate in this dissertation, since they have infinitely many simple permutations. However, the method we use for them heavily depends on the structure of simple permutations in these classes.

2.4. Automata and the transfer matrix method

We conclude this chapter with the discussion of elementary automata theory and the so-called transfer matrix method. Given a digraph with finitely many vertices and edges, the transfer matrix method allows us to find the generating function (according to some weight function on the edges) for paths from a specified vertex to another. We can apply this to the state diagram of an automaton to find the generating function for the language it accepts. This will be the key to finding $f_{\mathcal{A}}$ and $f_{\mathcal{A}'}$.

2.4.1. Definition and example

We start with the definition of a deterministic finite-state automaton. An *alphabet* Σ is a finite set, and we call the elements of the alphabet *letters*. For example, $\Sigma = \{a, b, c\}$ is an alphabet. A string of letters $\alpha_1, \alpha_2, \ldots, \alpha_n$ where $\alpha_i \in \Sigma$ $(1 \leq i \leq n)$ is called a *word*. The string with no letters is called the *empty word*, denoted by λ. The set of all words associated with Σ, including the empty word, is denoted by Σ^*. In particular,

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i, \qquad \text{where } \Sigma^i = \underbrace{\Sigma \times \cdots \times \Sigma}_{i \text{ times}}.$$

With $\Sigma = \{a, b, c\}$, we have $\Sigma^* = \{\lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \ldots\}$. Any subset of Σ^* is called a *language*.

A *deterministic finite-state automaton* is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where:

- Q is a finite set. Elements in Q are called *states* and denoted by q with some subscript.
- Σ is an alphabet.
- $\delta : Q \times \Sigma \to Q$ is a function called the *transition function*.

- $q_0 \in Q$ is called the *initial state.*
- F is a subset of Q. Any state in F is called a *accept state.*

As an example, let $Q = \{A, B, C, J\}$ with A being the initial state, $F = \{C\}$, $\Sigma = \{a, b, c\}$ and δ defined by

$$\delta(q, \alpha) = \begin{cases} A & \text{if } (q, \alpha) = (B, a) \\ B & \text{if } (q, \alpha) = (A, b), (C, b) \\ C & \text{if } (q, \alpha) = (A, c), (B, c), (C, c) \\ J & \text{otherwise} \end{cases}.$$

Together, we have a deterministic finite-state automaton $M = (Q, \Sigma, \delta, A, F)$. We can provide a graphical representation of an automaton called the *state diagram*, which is an edge-labelled directed graph. States are represented by vertices. If $\delta(q_i, \alpha) = q_j$, we draw a directed edge from q_i to q_j which is labeled with α. Such a directed edge is called a *transition* from q_i to q_j. The initial state is indicated by the arrow with no label, and accept states are shown by double circles. For the example above, we have the state diagram shown in Figure 2.6(a).

Associated to each automaton is a language $\mathcal{L}(M)$. Let $w = \alpha_0, \alpha_1, \ldots, \alpha_n$ be a word in Σ^*. The *run* of an automaton on w is a sequence of states $q_0, q_1, \ldots, q_n$ (not necessarily distinct) where q_0 is the initial state, and $q_i = \delta(q_{i-1}, \alpha_i)$ for every i with $1 \leq i \leq n$. A word w is said to be *accepted* by an automaton $(Q, \Sigma, \delta, q_0, F)$ if $q_n \in F$. In other words, an automaton reads letters in w in sequence. As it reads the letter α_i, it moves to the state q_i. If the final state q_n is in F, then automaton accepts w. The language $\mathcal{L}(M)$ is the set of all words accepted by the

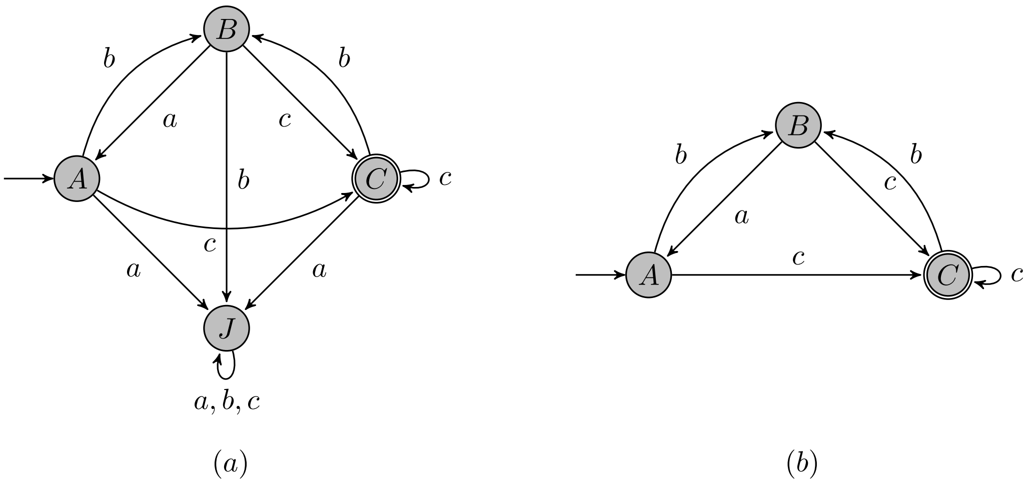


Figure 2.6.: The state diagram of the example automaton.

automaton.

In our example automaton, once we arrive at the state J, the transition function always gives us J thereon and we are no longer able to go to any other states and in particular any accept states. We call such a state a *jail state* and often omit all transitions to jail states as well as jail states themselves from the state diagram. Thus, we may represent the diagram of the above example as Figure 2.6(b) instead.

A language K is said to be *regular* if there exists a deterministic finite-state automaton M such that $\mathcal{L}(M) = K$. Note that in most textbooks, a regular language is defined as an element of the set of languages $\mathcal{R}$ over Σ which is defined recursively as the following.

- $\emptyset, \{\lambda\} \in \mathcal{R}$ and for every $\alpha \in \Sigma$, $\{\alpha\} \in \mathcal{R}$.
- If $K, L \in \mathcal{R}$, then $K \cup L, KL = \{\underbrace{\alpha_1, \ldots, \alpha_m}_{w_K}, \underbrace{\beta_1, \ldots, b_n}_{w_L} : w_K \in K, w_L \in L\}, K^* \in \mathcal{R}$.
- $\mathcal{R}$ is the minimal set satisfying above two conditions.

A classical theorem states that the definition we provide and the definition above are equivalent.

2.4.2. Transfer matrix method

Here, we introduce a useful application of an automaton. With the method we discuss in this section, we are able to find the generating function giving the number of n-letter words that are accepted by a given automaton. To start, we define a weight function on a digraph.

Let D be a finite digraph with a vertex set V and an edge set E. A *weight function on* E is a mapping from E to some commutative ring R. For an arbitrary walk $\Gamma = e_1 e_2 \cdots e_n$, the *weight of* Γ *with respect to* w is defined by $w(e_1)w(e_2)\cdots w(e_n)$. Intuitively, we may think that every time we pass through an edge, we "count" it by multiplying by the assigned weight.

Given a digraph $D = (V, E)$ with $|V| = m$ and a weight function on D, an *adjacency matrix* P *of* D *with respect to* w is the $m \times m$ matrix where row i $(1 \leq i \leq m)$ and column j $(1 \leq i \leq m)$ are labeled by vertices v_i and v_j respectively, and each entry is

$$P_{ij} = \sum_e w(e)$$

where the sum is over all edges from v_i to v_j.

As an example, we take the state diagram in Figure 2.6(a). Let w be the weight function defined by $w(t) = x$ for every transition t. Then we obtain the following adjacency matrix.

$$P = \begin{array}{c} \\ A \\ B \\ C \\ J \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & J \end{array} \\ \begin{bmatrix} 0 & x & x & x \\ x & 0 & x & x \\ 0 & x & x & x \\ 0 & 0 & 0 & 3x \end{bmatrix} \end{array}$$

Again, we are not interested in jail states for our purpose, so we omit the row and column designated for J and express the adjacency matrix of this example as the following.

$$P = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0 & x & x \\ x & 0 & x \\ 0 & x & x \end{bmatrix} \end{array} \qquad (2.1)$$

With this definition of adjacency matrix, we derive the following theorem.

Theorem 2.9 *Let P be the adjacency matrix of some digraph $D = (V, E)$, where $|V| = m$, with respect to a weight function w. For any positive integer n and any i and j with $1 \leq i, j \leq m$, the (i, j)-entry of P^n is*

$$(P^n)_{ij} = \sum_{\Gamma} w(\Gamma)$$

where the sum is over all walks Γ in D of length n from v_i to v_j. (By convention, we define $P^0 = I$ even if P is not invertible.)

Proof. The proof is by induction on n. The base case with $n = 1$ is obvious by the definition of adjacency matrix. Assume the statement is true for some positive integer k. For any i, j with $1 \leq i, j \leq m$, the (i, j)-entry of P^{k+1} is obtained by

$$(P^{k+1})_{ij} = (P^k)_{i1} P_{1j} + (P^k)_{i2} P_{2j} + \cdots + (P^k)_{im} P_{mj} = \sum_{1 \leq \ell \leq m} (P^k)_{i\ell} P_{\ell j}$$

By the inductive hypothesis, for each ℓ with $1 \leq \ell \leq m$, $(P^k)_{i\ell} = \sum_{\Delta} w(\Delta)$ where the sum is over all walks Δ in D of length k from v_i to v_ℓ. By multiplying $P_{\ell j} = \sum_e w(e)$ where the sum is over all edges from v_ℓ to v_j, we have

$$(P^k)_{i\ell} P_{\ell j} = \sum_{\Delta} w(\Delta) \sum_{e} w(e) = \sum_{e} \sum_{\Delta} w(\Delta) w(e) = \sum_{\Gamma_\ell} w(\Gamma_\ell)$$

where the sum is over all walks $\Gamma\ell$ in D of length $k+1$ from v_i to v_j, such that the k-th vertex is v_ℓ. Hence,

$$(P^{k+1})_{ij} = \sum_{1\leq\ell\leq m} (P^k)_{i\ell}P_{\ell j} = \sum_{1\leq\ell\leq m}\sum_{\Gamma_\ell} w(\Gamma_\ell) = \sum_{\Gamma} w(\Gamma)$$

where the sum is over all walks Γ in D of length $k+1$ from v_i to v_j. ■

We provide an example to illustrate the use of this theorem. For the adjacency matrix P (1) for Figure 2.6(b), we compute P^3. We get

$$P^3 = \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{ccc} x^3 & 3x^3 & 4x^3 \\ 2x^3 & 2x^3 & 4x^3 \\ x^3 & 3x^3 & 4x^3 \end{array}\right] \end{array}.$$

We are interested in the entry of row for A and the column C, since A is the initial state and C is the only accept state. We have $4x^3$ in the (A, C)-entry, so we look at the details of where this term comes from. $(P^3)_{AC}$ is the following sum of the nonzero terms:

$$(P^3)_{AC} = P_{AB}P_{BC}P_{CC} + P_{AB}P_{BA}P_{AC} + P_{AC}P_{CB}P_{BC} + P_{AC}P_{CC}P_{CC}. \qquad (2.2)$$

Every time we pass through a transition, we multiply by x. For a walk with three transitions from A to C, we get $x \cdot x \cdot x = x^3$. With four distinct walks from A to C with exactly three transitions, we obtain x^3 four times, and therefore the (A, C)-entry of P^3 is $4x^3$.

Hence, by looking at the (A, C)-entry of P^n, we can find the number of distinct walks from A to C with n transitions. Specifically, this number is the coefficient given our weight function. This is equivalent to the number of distinct n-letter words that are accepted by the automaton of the diagram in Figure 2.6(b). In the case where we have multiple accept states, we consider all the entries in the row corresponding to the initial state and a column corresponding to an accept state and sum them up.

Our next goal is to find the generating function $\sum s_n x^n$, where s_n is the number of distinct n length walks from one vertex to another by applying this technique. Suppose we have a digraph $D = (V, E)$ and the weight function w defined by $w(e) = x$ for every edge $e \in E$. Because the coefficient of x^n in the entry $(P^n)_{ij}$ describes the number of distinct walks of length n from v_i to v_j, the generating function for this is simply given by

$$f = \sum_{n=0}^{\infty}(P^n)_{ij}, \quad \text{or equivalently,} \quad \left(\sum_{n=0}^{\infty} P^n\right)_{ij}.$$

Notice that

$$\begin{aligned} I &= I + (P - P) + (P^2 - P^2) + \cdots = (I + P + P^2 + \cdots) - (P + P^2 + P^3 + \cdots) \\ &= (I + P + P^2 + \cdots) - P(I + P + P^2 + \cdots) = (I + P + P^2 + \cdots)(I - P) \end{aligned}$$

This gives us $(I - P)^{-1} = \sum_{n=0}^{\infty} P^n$, so the (i, j)-entry of the matrix $(I - P)^{-1}$ is the desired function.

With the previous example, we obtain

$$(I - P)^{-1} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} \dfrac{-x^2 - x + 1}{-2x^2 - x + 1} & \dfrac{x}{-2x^2 - x + 1} & \dfrac{x^2 + x}{-2x^2 - x + 1} \\ \dfrac{x - x^2}{-2x^2 - x + 1} & \dfrac{1 - x}{-2x^2 - x + 1} & \dfrac{x^2 + x}{-2x^2 - x + 1} \\ \dfrac{x^2}{-2x^2 - x + 1} & \dfrac{x}{-2x^2 - x + 1} & \dfrac{1 - x^2}{-2x^2 - x + 1} \end{bmatrix} \end{array}$$

The (A, C)-entry is

$$\frac{x^2 + x}{-2x^2 - x + 1} = x + 2x^2 + 4x^3 + 8x^4 + \cdots ,$$

which is the generating function for the number of distinct n-letter words that are accepted by the automaton described in Figure 2.6(b).

Finally, we provide another example of the technique described above, which allows us to obtain more information from P than just the number of distinct n-letter words accepted by the associated automaton. In the previous example, let us replace the weight function with the following one.

$$w : T \to \mathbb{Z}[a, b, c] \quad \text{defined by} \quad w(t) = \begin{cases} a & \text{if } t \text{ is from } B \text{ to } A \\ b & \text{if } t \text{ is from } A \text{ to } B \text{ or from } C \text{ to } B \\ c & \text{if } t \text{ is from } A \text{ to } C, \text{ from } B \text{ to } C \text{ or from } C \text{ to } C \end{cases}$$

where T is the set of all transitions and $\mathbb{Z}[a, b, c]$ is the commutative ring generated by elements a, b and c with integer coefficients. With this weight function, the adjacency matrix is

$$P = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & b & c \\ B & a & 0 & c \\ C & 0 & b & c \end{array} .$$

Again, we look at the (A, C)-entry of P^3 as an example. Since the computation of matrix multiplication does not change, we still get the equation (2). Hence, it is

$$(P^3)_{AC} = b \cdot c \cdot c + b \cdot a \cdot c + c \cdot b \cdot c + c \cdot c \cdot c = abc + 2bc^2 + c^3.$$

Here, we need to keep in mind that a, b and c are elements of the commutative ring, not letters for a regular language associated with the automaton. However, the defined weight function "counts" each transition as a variable according to its label, so the word $bacbc \in \Sigma^*$ for example is weighted as $ab^2c^2 \in \mathbb{Z}[a,b,c]$ by the weight function.

Thus, the equation above of three variables tells us that not only are there four 3-letter words accepted by the automaton of Figure 2.6(b), but also that one of them consists of one a, one b and one c, two of them consist of two b's and one c, and the other one contains three c's. Just as before, we can find the multivariate generating function describing the number of $a^{n_1}b^{n_2}c^{n_3}$ words as its coefficients by looking at the (A,C)-entry of $(I-P)^{-1}$. This time, we obtain

$$(I-P)^{-1} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} \quad A \quad & \quad B \quad & \quad C \quad \end{array} \\ \left[\begin{array}{ccc} \dfrac{-bc-c+1}{-ab-cb-c+1} & \dfrac{b}{-ab-cb-c+1} & \dfrac{bc+c}{-ab-cb-c+1} \\ \dfrac{a-ac}{-ab-cb-c+1} & \dfrac{1-c}{-ab-cb-c+1} & \dfrac{ac+c}{-ab-cb-c+1} \\ \dfrac{ab}{-ab-cb-c+1} & \dfrac{b}{-ab-cb-c+1} & \dfrac{1-ab}{-ab-cb-c+1} \end{array}\right] \end{array}$$

The (A,C)-entry is

$$\frac{bc+c}{-ab-cb-c+1} = c + (bc+c^2) + (abc+2bc^2+c^3) + (ab^2c+2abc^2+b^2c^2+3bc^3+c^4) + \cdots$$

In Chapter 4 and 6, we apply this method to find generating functions for simple permutations of length greater than or equal to 4 in $\mathcal{A}$ and $\mathcal{A}'$ respectively.

Chapter 3.

Examples of finding generating functions

In this chapter, we prove enumerative results for two classes, $\mathrm{Av}(123, 213, 132)$ and $\mathrm{Av}(4123, 4213, 4132)$, as lemmas. Both are proved by showing that a generating function for the class satisfies a specific functional equation.

3.1. Enumeration of the class $\mathbf{Av(123, 213, 132)}$

We first state the result.

Lemma 3.1 *(Simion and Schmidt, 1985* [36]*) The numbers of permutations of length n in $Av(123, 213, 132)$ form the Fibonacci sequence. Thus, the generating function for $Av(123, 213, 132)$ is*

$$f_{\mathrm{Av}(123,213,132)} = \frac{1}{1 - x - x^2}.$$

Proof. For convenience, let $\mathcal{C} = \mathrm{Av}(123, 213, 132)$ and $F = f_{\mathrm{Av}(123,213,132)}$. Since clearly, $s_0 = s_1 = 1$, we show this result by proving the number of length n permutations in $\mathcal{C}$ is equal to the sum of the number of length $n-1$ permutations in $\mathcal{C}$ and the number of length $n-2$ permutations in $\mathcal{C}$ for all $n \geq 2$. Let $\pi \in \mathcal{S}_n \cap \mathcal{C}$ $(n \geq 2)$ be arbitrary. Suppose the biggest value n appears after the second position of π, meaning $\pi(i) = n$ for some $i \geq 3$. This is an immediate contradiction, because either $\pi(1) < \pi(2)$ or $\pi(1) > \pi(2)$, and hence the flattening of positions 1, 2 and i is either 123 or 213.

Now suppose $\pi(2) = n$. If $n-1$ shows up after the second position, then the values $\pi(1)$, n and $n-1$ together create a subsequence whose flattening is 132, so we must have $\pi(1) = n-1$. Thus for the case $\pi(2) = n$, we have

$$\pi = (n-1)n\pi(3)\pi(4)\cdots\pi(n).$$

Notice that $\pi = 12 \ominus \sigma = 21[12, \sigma]$ where $\sigma \in \mathcal{S}_{n-2} \cap \mathcal{C}$. Since 12 is skew-indecomposable, Proposition 2.4 guarantees that for each distinct σ, we obtain a unique π. Thus, we have an obvious bijection between $\mathcal{S}_{n-2} \cap \mathcal{C}$ and the set $\{\pi \in \mathcal{C}_n \cap \mathcal{C} : \pi(2) = n\}$, namely $\phi(\sigma) = 12 \ominus \sigma$, so the number of length n permutations in $\mathcal{C}$ such that $\pi(2) = n$ is s_{n-2}.

Similarly, if $\pi(1) = n$, then we have $\pi = 21[1, \sigma] = 21[1, \sigma]$ where $\sigma \in \mathcal{S}_{n-1} \cap \mathcal{C}$. With the same argument above, we have s_{n-1} distinct π of length n in $\mathcal{C}$ with $\pi(1) = n$.

Consequently, combining these observations together, we have the relation

$$s_n = s_{n-1} + s_{n-2} \text{ for } n \geq 2, \qquad s_0 = s_1 = 1,$$

showing that s_n forms the Fibonacci sequence.

We now translate this into a functional equation. We have

$$\begin{aligned} F &= \sum_{n=0}^{\infty} s_n x^n = 1 + x + \sum_{n=2}^{\infty} s_n x^n = 1 + x + \sum_{n=2}^{\infty} (s_{n-1} + s_{n-2}) x^n \\ &= 1 + x + \sum_{n=2}^{\infty} s_{n-1} x^n + \sum_{n=2}^{\infty} s_{n-2} x^n = 1 + x \left(\sum_{n=0}^{\infty} s_n x^n \right) + x^2 \left(\sum_{n=0}^{\infty} s_n x^n \right) \\ &= 1 + xF + xF^2. \end{aligned}$$

Thus, $F = 1 + xF + x^2 F$. Solving this for F, we obtain the desired result. ■

The proof we presented suggests that any permutation in Av(123, 213, 132) can be written as

$$\bigominus_{\sigma \in \{\varepsilon, 1, 12\}} \sigma.$$

As it was previously mentioned, Lemma 3.1 was first proved in [36]. Note that with $\mathrm{op}(\beta) = \beta^r$ where β is a permutation, $\mathrm{op}(\{123, 213, 132\}) = \{321, 312, 231\}$, so by Proposition 2.2, we have the same enumeration result for Av(321, 312, 231). Permutations in this class are called *free permutations* in [34], and authors describe another way to enumerate this class.

3.2. Enumeration of the class Av(4123, 4213, 4132)

3.2.1. Number of permutations in Av(4123, 4213, 4132)

Next, we derive the generating function for the class Av(4123, 4213, 4132). Note that an alternate derivation can be found in [8], particularly, in Section 4.2.

Lemma 3.2 *Let $G = f_{Av(4123,4213,4132)}$. Then G satisfies the equation*

$$G = 1 + \frac{xG}{1 - xG^2}. \tag{3.1}$$

Proof. For convenience, let $\mathcal{C} = \mathrm{Av}(4123, 4213, 4132)$. Given that $s_0 = 1$, Equation 3.1 can be written as

$$\begin{aligned} G &= 1 + xG - xG^2 + xG^3 = 1 + xG + G \cdot G \cdot (G-1)x \\ &= 1 + (s_0x + s_1x^2 + \cdots) + (s_0 + s_1x + \cdots)(s_0 + s_1x + \cdots)(s_1x + s_2x^2 + \cdots)x \end{aligned}$$

Thus, equation 3.1 claims that, for each $n \geq 1$,

$$\begin{aligned} s_n &= s_{n-1} && + \sum_{\substack{0 \leq i,j \leq n-2,\ 1 \leq k \leq n-1 \\ i+j+k=n-1}} s_i s_j s_k \\ &= s_{n-1} && +(s_0s_{n-2} + s_1s_{n-3} + \cdots + s_{n-2}s_0)s_1 \\ &&& +(s_0s_{n-3} + s_1s_{n-4} + \cdots + s_{n-3}s_0)s_2 \\ &&& \qquad \vdots \\ &&& +(s_0s_1 + s_1s_0)s_{n-2} \\ &&& +s_0s_0s_{n-1} \end{aligned} \tag{3.2}$$

We prove the lemma by showing Equation 3.2 is true for all $n \geq 1$.

Let $n \geq 1$ be arbitrary. We claim that for each k with $1 \leq k \leq n-1$, $(s_0s_{n-(k+1)} + s_1s_{n-(k+2)} + \cdots + s_{n-(k+1)}s_0)s_k$ is the number of permutations of length n in $\mathcal{C}$ whose last element is $k+1$. For example, if $n = 5$ and $k = 2$, $(s_0s_2 + s_1s_1 + s_2s_0)s_2$ is the total number of permutations π such that $|\pi| = 5$ and $\pi(5) = 3$.

To prove this claim, let k be arbitrary with $1 \leq k \leq n-1$, and denote by $\mathcal{C}_k$ the set of permutations of length n in $\mathcal{C}$ having $k+1$ in the last position. We also define particular subsets of $\mathcal{C}_k$. First, let K be the range $[1, k]$. Let $x, y \in K$ be values that have the second right-most position and the last position respectively, so $x = \pi(a)$ and $y = \pi(b)$ where $a < b$ are the two largest integers with $\pi(a), \pi(b) \in K$. If $k = 1$, we let $y = 1$ and do not define x. For $k \neq 1$, let $I = (\pi^{-1}(x), \pi^{-1}(y))$, the segment between the positions of x and y, but excluding $\pi^{-1}(x)$ and $\pi^{-1}(y)$ themselves. If $k = 1$, let $I = [1, \pi^{-1}(y))$, the segment between the first position and the one immediately to the left of the position of $y = 1$. Similarly, let $J = (\pi^{-1}(y), \pi^{-1}(k+1))$.

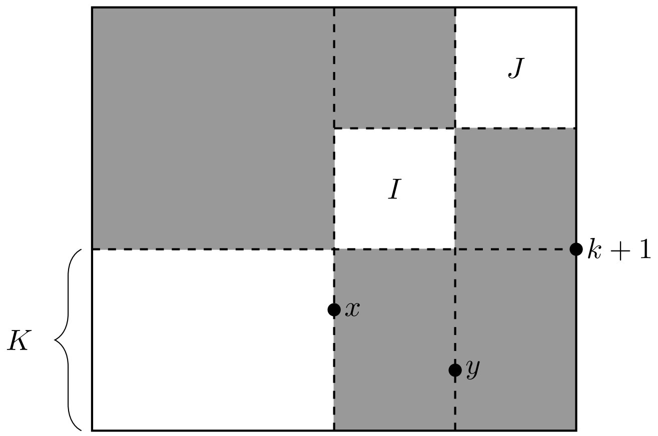


Figure 3.1.: The graphs of a permutation in $\mathcal{C}_{ik}$.

Notice that for all $a \in I$ and $b \in J$, $\pi(a) < \pi(b)$ because, otherwise, π would contain 4132 with y and $k+1$ corresponding to 1 and 2 respectively. Furthermore, positions of every value greater than $k+1$ must belong to either I or J, because if this is not the case, there exists a value $z \geq k+1$ such that $\pi^{-1}(z) < \pi^{-1}(x)$, and $zxy(k+1)$ forms either a 4123 or a 4213 pattern. These observations imply both segments I and J form blocks. The length of these blocks are some nonnegative integers i and j such that $i+j+k = n-1$. For all i with $0 \leq i \leq n-2$, we define $\mathcal{C}_{ik}$ to be the set of permutations in $\mathcal{C}_k$ such that $|I| = i$. Figure 3.1 shows what a permutation in $\mathcal{C}_{ik}$ looks like.

Let i and j be positive integers with $0 \leq i \leq n-2$ and $j = (n-1)-k-i$. We now define a bijection between $\mathcal{C}_{ik}$ and $(\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C})$. Let $\phi : \mathcal{C}_{ik} \to (\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C})$ be a function defined by

$$\phi(\pi) = (\sigma_1, \sigma_2, \sigma_3)$$

for all $\pi \in \mathcal{C}_{ik}$ where

$$\sigma_1 \text{ is the flattening of } \pi(I), \qquad \sigma_2 \text{ is the flattening of } \pi(J),$$
$$\sigma_3 \text{ is the flattening of } \pi(1)\pi(2)\cdots xy.$$

Needless to say, σ_1, σ_2 and σ_3 are in $\mathcal{C}$ simply by the definition of a permutation class. Also, how we defined each segment and range clearly indicates $|\sigma_1| = i$, $|\sigma_2| = j$ and $|\sigma_3| = k$, so $(\sigma_1, \sigma_2, \sigma_3) \in (\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C})$.

We prove that ϕ is a bijection by constructing ϕ^{-1}, the inverse of ϕ, and show it is a function. The inverse map $\phi^{-1} : (\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C}) \to \mathcal{C}_{ik}$ is defined by $\phi^{-1}[(\sigma_1, \sigma_2, \sigma_3)] = \pi$ where

$$\pi(\ell) = \begin{cases} \sigma_3(\ell) & \text{if } 1 \le \ell \le k-1 \\ \sigma_1(\ell-(k-1))+(k+1) & \text{if } k \le \ell \le k-1+i \\ \sigma_3(\ell-i) & \text{if } \ell = k+i \\ \sigma_2(\ell-(k+i))+(k+1+i) & \text{if } k+i+1 \le \ell \le k+i+j \\ k+1 & \text{if } \ell = n \end{cases}.$$

In one-line notation, this is equivalent to

$$\phi^{-1}[(\sigma_1,\sigma_2,\sigma_3)] = \sigma_3(1)\cdots\underbrace{\sigma_3(k-1)}_{x}\underbrace{\sigma_1'(1)\cdots\sigma_1'(i)}_{I}\underbrace{\sigma_3(k)}_{y}\underbrace{\sigma_2'(1)\cdots\sigma_2'(j)}_{J}(k+1),$$

for all $(\sigma_1,\sigma_2,\sigma_3) \in (\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C})$ where $\sigma_1'(\ell) = \sigma_1(\ell)+(k+1)$ and $\sigma_2'(m) = \sigma_2(m)+(k+1+i)$ for all ℓ and m $(1 \le \ell \le i,\ 1 \le m \le j)$.

We show ϕ^{-1} maps into $\mathcal{C}_{ik}$ by contradiction. Suppose there exists $(\sigma_1,\sigma_2,\sigma_3) \in (\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C})$ such that $\pi = \phi^{-1}[(\sigma_1,\sigma_2,\sigma_3)]$ is not in $\mathcal{C}_{ik}$. Since π has $k+1$ in n-th position, and the way π is constructed forces the segment I to have length i, $\pi \notin \mathcal{C}_{ik}$ implies $\beta \preceq \pi$ for some $\beta \in \{4123, 4213, 4132\}$. Because every permutation in the basis has the value 4 in the 1st position, π must have the value v_4 corresponding to the 4 in β before the positions of the other values v_1, v_2 and v_3 corresponding to 1, 2 and 3 respectively. Let m_1, m_2, m_3 and m_4 be the positions of v_1, v_2, v_3 and v_4 respectively. Suppose $1 \le m_4 \le k-1$. Since this implies that the value v_4 is at most k, any position m with $m_4+1 \le m \le n$ such that $\pi(m) < v_4$ must satisfy $m_4+1 \le m \le k-1$ or $m = k+1$. Hence, all of v_1, v_2, v_3 and v_4 are determined by σ_3, but this means $\beta \in \sigma_3$. We can similarly show that we cannot have $k \le m_4 \le k-1+i$ or $k+i+1 \le m_4 \le k+i+j$. In addition, $m_4 \ne k+1$ and $m_4 \ne n$ because there is no position greater than m_4 whose value is less than v_4. Consequently, $\pi = \phi^{-1}[(\sigma_1,\sigma_2,\sigma_3)] \in \mathcal{C}_{ik}$ for every $(\sigma_1,\sigma_2,\sigma_3) \in (\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C})$.

The way ϕ^{-1} is constructed clearly shows that it is the inverse of ϕ, so ϕ is a bijection. Hence, $|\mathcal{C}_{ik}| = |(\mathcal{S}_i \cap \mathcal{C}) \times (\mathcal{S}_j \cap \mathcal{C}) \times (\mathcal{S}_k \cap \mathcal{C})| = s_i s_j s_k$, and

$$\mathcal{C}_k = \bigcup_{0 \le i \le n-2} \mathcal{C}_{ik},$$

which implies

$$|\mathcal{C}_k| = \sum_{i=0}^{n-(k+1)} |\mathcal{C}_{ik}| = (s_0 s_{n-(k+1)} + \cdots + s_{n-(k+1)} s_0) s_k,$$

so this completes the proof of the claim.

Finally, we consider permutations in $\mathcal{C}$ of length n having a 1 in the last position. Notice that every permutation $\pi \in \mathcal{C}$ where $|\pi| = n$ and $\pi(n) = 1$ can be written as $\tau \ominus 1$ where τ is some permutation in $\mathcal{C}$ whose length is $n-1$ (including the case of $\pi = 1$), and for every permutation $\tau \in \mathcal{S}_{n-1} \cap \mathcal{C}$, $\tau \ominus 1 \in \mathcal{C}$. Thus, the number of such permutations is the same as the number of permutations of length $n-1$ in $\mathcal{C}$, which is simply s_{n-1}. Summing up every possible case, we obtain

$$s_n = s_{n-1} + \sum_{k=1}^{n-1} |\mathcal{C}_k|,$$

for all $n \geq 1$, and this is equation 3.2. This proves the generating function G for $\mathcal{C}$ satisfies equation 3.1 and completes the proof. ■

There is another famous combinatorial object whose enumeration is related to the class $\text{Av}(4123, 4213, 4132)$. *Schröder n-paths* are lattice paths in the Cartesian plane using $\langle 0,1\rangle$, $\langle 1,0\rangle$ and $\langle 1,1\rangle$ steps that start at $(0,0)$, end at (n,n) and stay on or above the $x = y$ line. The number of distinct Schröder n-paths is called the *n-th Schröder number.* Figure 3.2 is the list of all Schröder 3-paths. As we can see, there are 22 distinct Schröder 3-paths.

It turns out the generating function for Schröder $(n-1)$-paths with no three consecutive up-steps is known to satisfy the equation (1) as well (as listed as A106228 in [32]). Hence, the number of length n permutations in $\text{Av}(4123, 4213, 4132)$ is

$$s_n = \begin{cases} 1 & \textit{if } n = 0 \\ \substack{\textit{the number of Schröder } (n-1)\textit{-paths} \\ \textit{with no three consecutive up-steps}} & \textit{if } n \geq 1 \end{cases}.$$

The last path shown in Figure 3.2 is the only one with three consecutive up-steps, so there are 21 desired Schröder 3-paths. In fact, there are 21 permutations of length 4 in the class $\text{Av}(4123, 4213, 4132)$, namely every permutation of length 4 except the ones in the basis.

Proving Lemma 3.2 by finding a bijection between $\mathcal{S}_n \cap \text{Av}(4123, 4213, 4132)$ and the set of Schröder $(n-1)$-paths having no triple up-steps for each $n \geq 1$ would be ideal. However, this problem remains unsolved. It is worth noting that if we take any two permutations β_1, β_2 from $\{4123, 4213, 4132\}$ and form a subbasis $\{\beta_1, \beta_2\}$, then $s_n(\beta_1, \beta_2)$ is the $(n-1)$-th Schröder number. Bijective proofs are given in [27, 28], although the bijections given are to Schröder generating trees rather than Schröder paths themselves. Thus, one could possibly prove this result bijectively by showing a permutation of length n containing the other permutation β_3 corresponds to a tree that represents a Schröder path having triple up-steps.

3.2.2. Skew-indecomposable permutations in Av(4123, 4213, 4132)

For the remainder of this chapter, we determine the generating function describing the number of skew-indecomposable permutations of length n in $\text{Av}(4123, 4213, 4132)$ as this will be necessary for the main result in Chapter 6.

As it is discussed at the end of the proof for Lemma 3.2, every permutation $\pi \in \text{Av}(4123, 4213, 4132)$ such that $|\pi| = n \geq 1$ and $\pi(n) = 1$ can be written as $\tau \ominus 1$ for some length $n-1$ permutations τ in the same class. Except when $\pi = 1$, τ is nonempty, so whether τ is skew-indecomposable or not, such a permutation π of length $n \geq 2$ is skew-decomposable.

In addition to the above case, there are some permutations in $\text{Av}(4123, 4213, 4132)$ ending with 12 (*i.e.* $\pi(n-1) = 1$ and $\pi(n) = 2$). These are the permutations for which $\pi(n) = 2$ and the segment J, as defined in the proof of Lemma 3.2, is empty. Such permutations can be written as $\rho \ominus 12$ where ρ is a nonempty permutation in $\text{Av}(4123, 4213, 4132)$ of length $n-2$. Therefore, permutations of length $n \geq 3$ having $\pi(n-1) = 1$ and $\pi(n) = 2$ are also skew-decomposable.

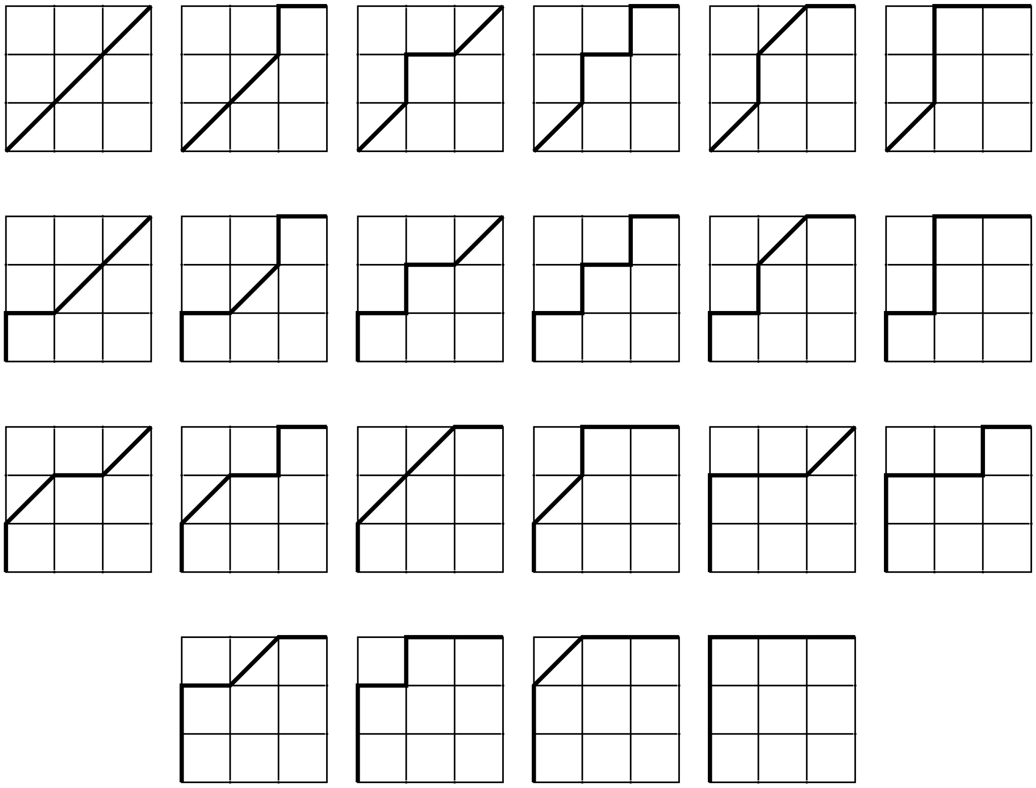


Figure 3.2.: The list of Schröder 3-paths.

We claim that these two are the only ways for a permutation in $\mathrm{Av}(4123, 4213, 4132)$ to be skew-decomposable. Suppose to the contrary that there exists a skew-decomposable permutation π in $\mathrm{Av}(4123, 4213, 4132)$ which can be written as $\pi_1 \ominus \pi_2$ where $\pi_2 \neq 1$ and $\pi_2 \neq 12$. If $\pi_2 = 21$, then this is just the case $\pi_2 = 1$, so the length of π_2 must be strictly greater than 2. If π_2 has the value 2 in the last position, then π_2 must contain 132, but with π_1, this results in π containing 4132, so we achieve a contradiction. Likewise, if π_2 has some value greater than or equal to 3 in the last position, then either $123 \preceq \pi_2$ or $213 \preceq \pi_2$, implying $4123 \preceq \pi$ or $4213 \preceq \pi$, which is again a contradiction. Consequently, a permutation π in $\mathrm{Av}(4123, 4213, 4132)$ is skew-decomposable if and only if $\pi = \tau \ominus 1$ or $\pi = \rho \ominus 12$, where τ and ρ are nonempty permutations in $\mathrm{Av}(4123, 4213, 4132)$, and $|\tau| = n - 1$, $|\rho| = n - 2$.

By excluding these two kinds of permutations, we can obtain the number of skew-indecomposable permutations in $\mathrm{Av}(4123, 4213, 4132)$ as the following.

Lemma 3.3 *Let $\bar{G} = \bar{f}_{Av(4123,4213,4132)}$. The generating function for the number of skew-indecomposable permutations of length n in $Av(4123, 4213, 4132)$, excluding the empty permutation, is*

$$(1 - x - x^2)\bar{G}.$$

Proof. Let t_n be the number of skew-indecomposable permutations of length n ($n \geq 1$) in $\mathrm{Av}(4123, 4213, 4132)$. As we have already discussed, all we have to do is take out the permutations having forms of $\pi = \tau \ominus 1$ or $\pi = \rho \ominus 12$, where τ and ρ are nonempty permutations in $\mathrm{Av}(4123, 4213, 4132)$. The numbers of each are s_{n-1} for $n \geq 2$ and s_{n-2} for $n \geq 3$ respectively, so we have

$$t_n = \begin{cases} s_n & \text{if } n = 1 \\ s_n - s_{n-1} & \text{if } n = 2 \\ s_n - s_{n-1} - s_{n-2} & \text{if } n \geq 3 \end{cases}.$$

Therefore,

$$\begin{aligned} t_1x + t_2x^2 + t_3x^3 + \cdots &= s_1x + (s_2 - s_1)x^2 + (s_3 - s_2 - s_1)x^3 + \cdots \\ &= (s_1x + s_2x^2 + \cdots) - (s_1x + s_2x^2 + \cdots)x - (s_1x + s_2x^2 + \cdots)x^2 \\ &= \bar{G} - \bar{G}x - \bar{G}x^2 = \bar{G}(1 - x - x^2). \end{aligned}$$

■

Chapter 4.

Enumeration of the class $\mathcal{A}$

4.1. Overview

This chapter is entirely based on the paper *Enumerating indices of Schubert varieties defined by inclusions* by Albert and Brignall [7]. We repeat and expand upon the details of that paper here since the methods we use to enumerate $\mathcal{A}'$ are an extension of theirs.

To begin, we give a short overview of their method of enumeration. They first enumerate simple permutations in $\mathcal{A}$. To do this, they characterize the structure of the simple permutations in $\mathcal{A}$ of length greater than or equal to 4. Using this characterization, they define an encoding of simple permutations in $\mathcal{A}$ into words. They then construct an automaton that accepts precisely these words to show that the set of encoded words form a regular language and apply the transfer matrix method to complete the enumeration of simple permutations. The whole class is enumerated by applying Proposition 2.5 and adding in the case of sum and skew decomposable permutations.

Recall $\mathcal{A} = \text{Av}\{4231, 35142, 42513, 351624\}$. Before we start, we note two symmetry properties of the class $\mathcal{A}$, introduce new terminology called the extreme pattern of a permutation, and give the general idea of the enumeration method. Let op_1 and op_2 be the inverse operation and the reverse complement operation, *i.e.* $\text{op}_1(\pi) = \pi^{-1}$ and $\text{op}_2(\pi) = (\pi^r)^c$ for every permutation π. Notice that $\text{op}_1(\{4231, 35142, 42513, 351624\}) = \text{op}_2(\{4231, 35142, 42513, 351624\}) = \{4231, 35142, 42513, 351624\}$. Thus, by Proposition 2.2, if π is in the class $\mathcal{A}$, then π^{-1} and $(\pi^r)^c$ are also in $\mathcal{A}$.

Now, we define the extreme pattern of a permutation. The *extreme pattern* of a permutation is the flattening of the first, the last, the greatest and the least values of a permutation. For instance, the extreme pattern of $\pi = 47128365$ is 2143 due to the subsequence 4185 where 4, 1, 8 and 5 correspond to the first, the least, the greatest and the last value respectively. It is not always the case that the extreme pattern is length 4, as $\sigma = 52413$ has the greatest value 5 in the first position, so its extreme pattern is 312 due to the subsequence 513. However, if π is simple and $|\pi| \geq 4$, then its extreme pattern must be one of 2143, 2413, 3142 and 3412, since a simple permutation cannot begin or end with its greatest or least value.

4.2. Extreme patterns 2413, 3142 and 3412

In the next section, we will establish the structure of the simple permutations π in $\mathcal{A}$ with $|\pi| \geq 4$ and $\pi(2) \neq 1$. Before we do so, we first study some special cases defined by their extreme patterns. We start with the simple permutations having extreme pattern 2413.

Proposition 4.1 *Let π be a simple permutation in $\mathcal{A}$ with extreme pattern 2413. Let b, d, a and c be the first, the greatest, the least and the last values of π respectively. Then the graph of π is N-shaped, that is, values corresponding to positions in $[\pi^{-1}(b), \pi^{-1}(d)]$ are increasing, values corresponding to positions in $[\pi^{-1}(d), \pi^{-1}(a)]$ are decreasing and values corresponding to positions in $[\pi^{-1}(a), \pi^{-1}(c)]$ are increasing.*

For instance, Figure 4.1 shows the graph of $\pi = 25864137$, a length 8 simple permutation of extreme pattern 2413 in $\mathcal{A}$. As we can see, drawn points form an N-shape.

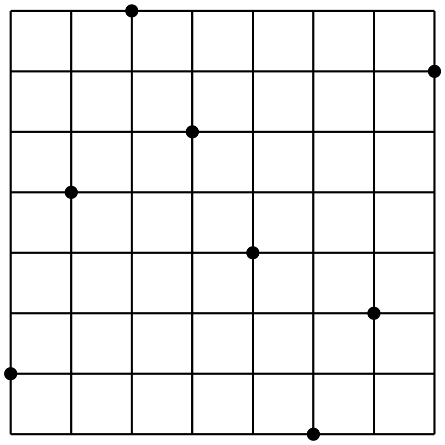


Figure 4.1.: Graph of 25864137.

Proof. Let $\pi \in \mathcal{A}$ be a simple permutation of the extreme pattern 2413. We provide the graph of extreme pattern 2413 in Figure 4.2. A permutation π of extreme pattern 2413 has a graph that can be drawn by filling in more points in the interior regions of this graph. We first claim that, for $\pi \in \mathrm{Si}(\mathcal{A})$, there cannot be any points in the region denoted by B_{31}. In other words, π cannot have a value less than b whose position is in the segment $(\pi^{-1}(b), \pi^{-1}(d))$.

Suppose to the contrary that there is a point in the region B_{31}. Let x be the point in B_{31} of minimum value as shown in Figure 4.3(a). We further examine where other values of π can

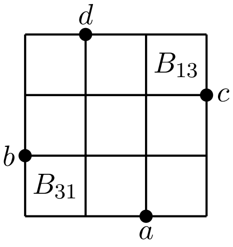


Figure 4.2.: Partial graph of π of extreme pattern 2413.

be located. Notice that if π has a point in any of the dark grey regions, π will contain some permutation in $\{4231, 35142, 42513, 351624\}$, so these regions cannot contain any points. For example, if there exists a point z in the dark grey region $(\pi^{-1}(x), \pi^{-1}(d)) \times (x, b)$, then $bxza$ would form a 4231 pattern. In addition, there are no points in the light grey region because we chose x to be the point of minimum value with position in the segment $(\pi^{-1}(b), \pi^{-1}(d))$. We use this two-color coding to differentiate forbidden regions in future proofs also.

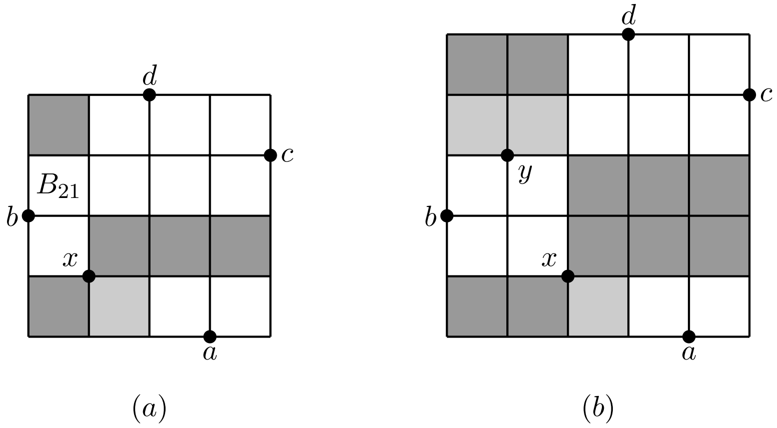


Figure 4.3.: Partial graphs of π with the assumption of having a value in B_{31}.

Since π is a simple permutation, the segment defined by the positions of b and x cannot be a block. Hence, there exists a point in the region denoted by B_{21}. (We will say such a point *splits* the potential block $[\pi^{-1}(b), \pi^{-1}(x)]$.) Choose the point of greatest value in B_{21} and call this value y. Then we obtain the graph shown in Figure 4.3(b). Notice now that $[\pi^{-1}(b), \pi^{-1}(x)]$ must be a block, since there cannot be any points directly above, below, to the right or to the left of the region $[\pi^{-1}(b), \pi^{-1}(x)] \times [x, y]$. Therefore, π is not simple, a contradiction.

Referring back to Figure 4.2, since $\mathcal{A}$ is preserved by reverse complement, we can rotate our previous argument by 180° to show there are no points in the region B_{13}.

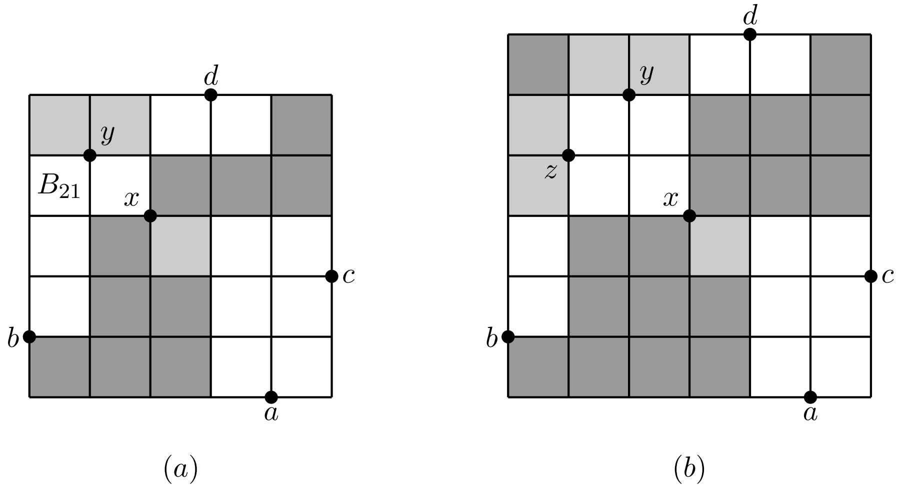


Figure 4.4.: Partial graphs of π with the assumption of having a decreasing sub-segment in $[\pi^{-1}(b), \pi^{-1}(d)]$.

Now, we show that values corresponding to positions in $[\pi^{-1}(b), \pi^{-1}(d)]$ are increasing. We again show this by contradiction using the graph of π. Suppose the values corresponding to the segment $[\pi^{-1}(b), \pi^{-1}(d)]$ are not strictly increasing. This means that there is at least one sub-segment in $[\pi^{-1}(b), \pi^{-1}(d)]$ whose values are decreasing. Let $[\pi^{-1}(y), \pi^{-1}(x)]$ be the decreasing sub-segment with the greatest possible y and the least possible x given y. This provides the graph shown in Figure 4.4(a). Since the segment $[\pi^{-1}(y), \pi^{-1}(x)]$ cannot be a block, we have a point in the region B_{21} of Figure 4.4(a). By choosing the left-most such point and denoting by z the value of this point, we obtain the graph in Figure 4.4(b). Since we have a block $[\pi^{-1}(z), \pi^{-1}(x)]$ which cannot be split, we achieve a contradiction.

The reverse complement of this argument shows values corresponding to positions in $[\pi^{-1}(a), \pi^{-1}(c)]$ are strictly increasing.

Finally, values corresponding to the segment $[\pi^{-1}(d), \pi^{-1}(a)]$ must be decreasing because, otherwise, π would contain 4231 with d and a corresponding to the 4 and 1 respectively. This completes the proof. ■

If we apply the inverse symmetry to the previous proof, we obtain the following result for a simple permutation of extreme pattern 3142.

Proposition 4.2 *Given a simple permutation π in $\mathcal{A}$ with extreme pattern 3142. Let c, a, d and b be the first, the least, the greatest and the last values of π respectively. Then the graph of π is S-shaped, that is, values from the range $[a, b]$ are increasing, values from the range $[b, c]$ are* *decreasing and values from the range* $[c, d]$ *are increasing.*

Lastly, we prove the following proposition.

Proposition 4.3 *No simple permutation in* $\mathcal{A}$ *has extreme pattern 3412.*

Proof. Suppose the statement is false, and let π be a simple permutation in $\mathcal{A}$ whose extreme pattern is 3412. Just as in the proof of Proposition 4.1, we start with the graph of extreme pattern 3412 with c, d, a and b representing the first, the greatest, the least and the last values respectively. As shown in Figure 4.5, the segment $[\pi^{-1}(c), \pi^{-1}(d)]$ would form a block without the presence of a point in either B_{21} or B_{12}.

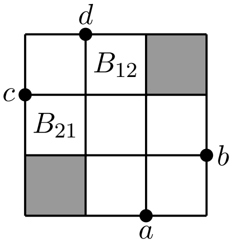


Figure 4.5.: Partial graph of π of extreme pattern 3412.

Assume there is a point in B_{21}, and let x denote the minimum value of any point in B_{21}. This is shown in Figure 4.6(a). Now, we must have a point in B_{11} in Figure 4.6(a) to prevent the segment $[\pi^{-1}(c), \pi^{-1}(x)]$ from being a block. Let y be the greatest value of any point in B_{11}, we obtain the graph in Figure 4.6(b). Now $[\pi^{-1}(c), \pi^{-1}(x)]$ must be a block, which is a contradiction.

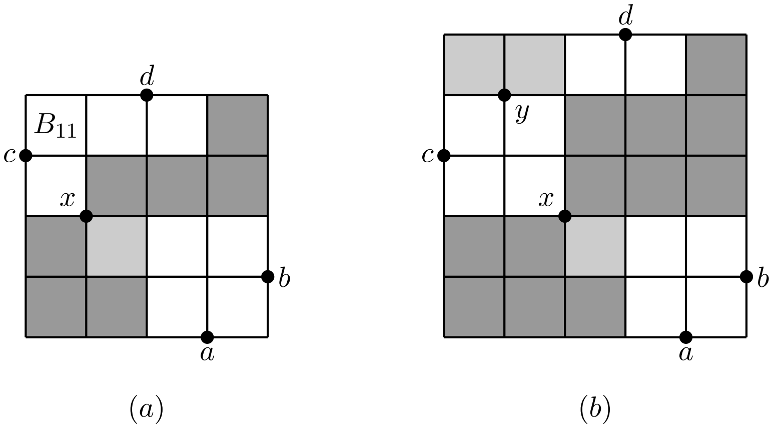


Figure 4.6.: Partial graphs of π with the assumption of having a value in B_{21}.

If we assume the existence of a point in B_{12} in Figure 4.5, we end up with the same result, since $\mathcal{A}$ and the pattern 3412 are preserved by the inverse operation followed by the reverse complement operation. Hence, there is no simple permutation of extreme pattern 3412 in $\mathcal{A}$. ■

4.3. General simple permutations in $\mathcal{A}$.

4.3.1. Structure theorem

We have characterized the structure of simple permutations having extreme patterns 2413, 3142 and 3412. The remaining extreme pattern is 2143. Instead of immediately examining this case, we first discuss the structure of simple permutations π where $|\pi| \geq 4$ and $\pi(2) \neq 1$. Later, we show that the condition $\pi(2) \neq 1$ is equivalent to the condition $\pi(1) = 2$. Thus, every simple permutation in $\mathcal{A}$ has either the value 2 in the first position or the value 1 in the second position, but not both, since a permutation with both is not simple. Observing that any simple permutation of extreme pattern 2413 has 2 in the first position and any permutation of extreme pattern 3142 has 1 in the second position, we can provide the classification of simple permutations as shown in Table 4.1. For the remainder of this chapter, let $H = \{\pi \in \mathrm{Si}(\mathcal{A}) : |\pi| \geq 4 \text{ and } \pi(2) \neq 1\}$.

<table>
<tr><th colspan="6">Simple permutation π in $\mathcal{A}$ with $|\pi| \geq 4$</th></tr>
<tr><td colspan="3">$\pi(1) = 2$</td><td colspan="3">$\pi(2) = 1$</td></tr>
<tr><td colspan="2">Extreme pattern 2413</td><td colspan="2">Extreme pattern 2143</td><td colspan="2">Extreme pattern 3142</td></tr>
</table>

Table 4.1.: Classification of simple permutations in $\mathcal{A}$.

In order to describe the structure of π in H, we need to define four special sum-like operations which we call glue sums that combine two permutations satisfying certain conditions into a longer permutation. They are called the type 1-0 NW glue sum, the type 1-1 NW glue sum, the type 1-0 SE glue sum and the type 1-1 SE glue sum. In Chapter 5, we will define more operations that are similar to these four, and type numbering will be explained there.

We first define the type 1-0 NW glue sum. Let σ and τ be simple permutations in $\mathcal{A}$ of length m and n respectively. Let $i = \sigma^{-1}(m)$ and $j = \tau(1)$. Furthermore, suppose $i \leq m-2$, $\sigma(m) = m-1$, $j \geq 3$ and $\tau(2) = 1$. For σ and τ satisfying these conditions, we define the *type 1-0 NW glue sum*, denoted by $\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau$, as the following permutation.

$$\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m-1)\tau'(3)\tau'(4)\cdots\tau'(n),$$

where $\sigma'(i) = m + (j-3)$ and $\sigma'(k) = \sigma(k)$ for $k \neq i$, and $\tau'(k) = \tau(k) + (m-3)$ for k with $3 \leq k \leq n$.

The type 1-0 NW glue sum identifies the greatest value m in σ and the first value j in τ. These two points are combined into one with the position $\sigma^{-1}(m)$ and the value $j + m - 3$, which is the value j shifted up by $m-3$ as all other points of τ are shifted up by $m+3$. The type 1-0 NW glue sum also eliminates $\sigma(m) = m-1$ and $\tau(2) = 1$. The remaining values in τ are attached just as in the usual sum $\oplus$ except they are shifted up by $m-3$. Notice that $|\sigma \obslash^0_1 \tau| = m + n - 3$, since one pair of values is combined and two values are eliminated.

Any simple permutations in $\mathcal{A}$ of extreme pattern 2413 and simple permutations in $\mathcal{A}$ of extreme pattern 3142 respectively satisfy the conditions for σ and τ required in the definition of $\obslash^0_1$. Figure 4.7 illustrates the type 1-0 NW glue sum with $\sigma_1 = 2753146$ and $\tau_1 = 5162473$.

Next, we define the type 1-1 NW glue sum. Let σ and τ be simple permutations in $\mathcal{A}$ satisfying the same conditions as in the definition of the type 1-0 NW glue sum. The *type 1-1 NW glue sum* denoted by $\sigma \obslash^1_1 \tau$ is defined as the following.

$$\sigma \obslash^1_1 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m)\tau'(3)\tau'(4)\cdots\tau'(n)$$

where $\sigma'(i) = m + (j-2)$ and $\sigma'(k) = \sigma(k)$ for $k \neq i$, and $\tau'(k) = \tau(k) + (m-2)$ for k with $3 \leq k \leq n$. The only differences between $\sigma \obslash^1_1 \tau$ and $\sigma \obslash^0_1 \tau$ are that $\sigma \obslash^1_1 \tau$ has an extra value

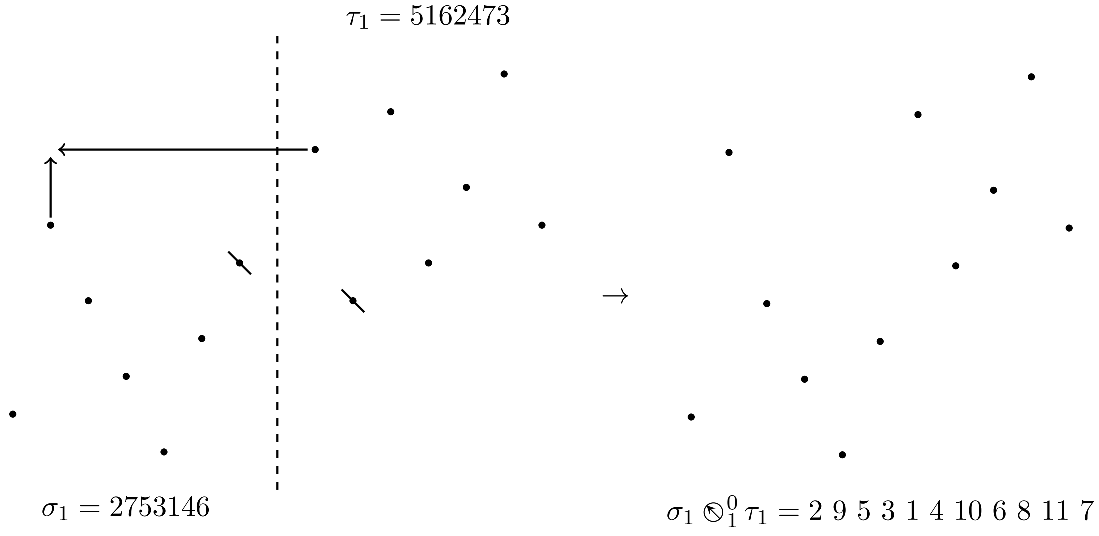

$\tau_1 = 5162473$
$\sigma_1 = 2753146$
$\sigma_1 \obslash^0_1 \tau_1 = 2\ 9\ 5\ 3\ 1\ 4\ 10\ 6\ 8\ 11\ 7$

Figure 4.7.: Illustration of $\sigma_1 \obslash^0_1 \tau_1$.

$\sigma(m) = m-1$ between the subsequences coming from σ and τ (as $m-1$ is not deleted), so $\sigma(i)$ and $\tau(k)$ $(3 \leq k \leq n)$ are shifted up by $m-2$ instead of $m-3$. Therefore, $|\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^1 \tau| = m+n-2$. For $\sigma = 2753146$ and $\tau = 5162473$, we obtain

$$\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^1 \tau = 2\ 10\ 5\ 3\ 1\ 4\ 6\ 11\ 7\ 9\ 12\ 8.$$

Note that, if we restrict to sum simple permutations of extreme pattern 2413 and 3142 together, both NW glue sums are injective operations. To show this for $\mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0$, let σ_1 and σ_2 be simple permutations of extreme pattern 2413 and τ_1 and τ_2 be simple permutations of extreme pattern 3142. Let $\pi_1 = \sigma_1 \mathbin{\bigcirc\mkern-15mu\nwarrow}_0^1 \tau_1$ and $\pi_2 = \sigma_2 \mathbin{\bigcirc\mkern-15mu\nwarrow}_0^1 \tau_2$. Suppose $\pi_1 = \pi_2$. If $|\sigma_1| = |\sigma_2|$, then it is obvious that $\sigma_1 = \sigma_2$ and $\tau_1 = \tau_2$ by definition. So suppose $|\sigma_1| < |\sigma_2|$. Let $|\sigma_1| = m$ and $|\sigma_2| = m+1$. Then $\pi_1(m) = \tau_1'(3) = \sigma_2'(m) = \pi_2(m)$. Because $\tau_1(2) = 1$ by Proposition 4.2, $\tau_1(3)$ must be greater than $\tau_1(n)$ $(n = |\tau_1|)$, because, otherwise, due to Proposition 4.2, we must have $\tau_1(3) = 2$, which is a contradiction to τ_1 being simple. Hence, we obtain $\sigma_2'(m) > \pi_2(m+n-3)$. This is impossible, while $\sigma_2'(m) = \sigma_2(m) < m-1$ and $\pi_2(m+n-3)$ is at least $m-1$ by definition of $\mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0$. We have the same contradiction for the case $|\sigma_2| > m+1$, so $|\sigma_1| = |\sigma_2|$, implying the type 1-0 NW glue sum is injective. The same argument can be applied for the type 1-1 NW glue sum as well.

The type 1-0 and type 1-1 NW glue sums both combine the greatest value in σ and the first value in τ. Next, we define the inverse notions of these two sums, called the type 1-0 and type 1-1 SE glue sums, denoted by $\mathbin{\bigcirc\mkern-15mu\searrow}_1^0$ and $\mathbin{\bigcirc\mkern-15mu\searrow}_1^1$, respectively, with the property that

$$(\sigma \mathbin{\bigcirc\mkern-15mu\searrow}_1^0 \tau)^{-1} = \sigma^{-1} \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau^{-1} \quad \text{and} \quad (\sigma \mathbin{\bigcirc\mkern-15mu\searrow}_1^1 \tau)^{-1} = \sigma^{-1} \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau^{-1}.$$

Specifically, let σ and τ be simple permutations in $\mathcal{A}$ of length m and n respectively. Let $i = \sigma(m)$ and $j = \tau^{-1}(1)$. We require that $i \leq m-2$, $m = \sigma(m-1)$, $j \geq 3$ and $2 = \tau(1)$. For σ and τ satisfying these conditions, we define the *type 1-0 SE glue sum* as the following.

$$\sigma \mathbin{\bigcirc\mkern-15mu\searrow}_1^0 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-2)\tau'(2)\tau'(3)\cdots\tau'(n)$$

where $\tau'(j) = i$ and $\tau'(k) = \tau(k) + (m-3)$ for $k \neq j$. Similarly, the *type 1-1 SE glue sum* is defined as

$$\sigma \mathbin{\bigcirc\mkern-15mu\searrow}_1^1 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-1)\tau'(2)\tau'(3)\cdots\tau'(n)$$

where $\tau'(j) = i$ and $\tau'(k) = \tau(k) + (m-2)$ for $k \neq j$. Both SE glue sums are injective when we sum simple permutations of extreme pattern 3142 and 2413.

Figure 4.8 shows $\sigma_2 \mathbin{\bigcirc\mkern-15mu\searrow}_1^0 \tau_2$ where $\sigma_2 = 5146372$ and $\tau_2 = 2475136$. Notice that $\sigma_2^{-1} = \sigma_1$ and $\tau_2^{-1} = \tau_1$ from the previous example. Indeed, $\sigma_2 \mathbin{\bigcirc\mkern-15mu\searrow}_1^0 \tau_2 = 5\ 1\ 4\ 6\ 3\ 8\ 11\ 9\ 2\ 7\ 10$, which is

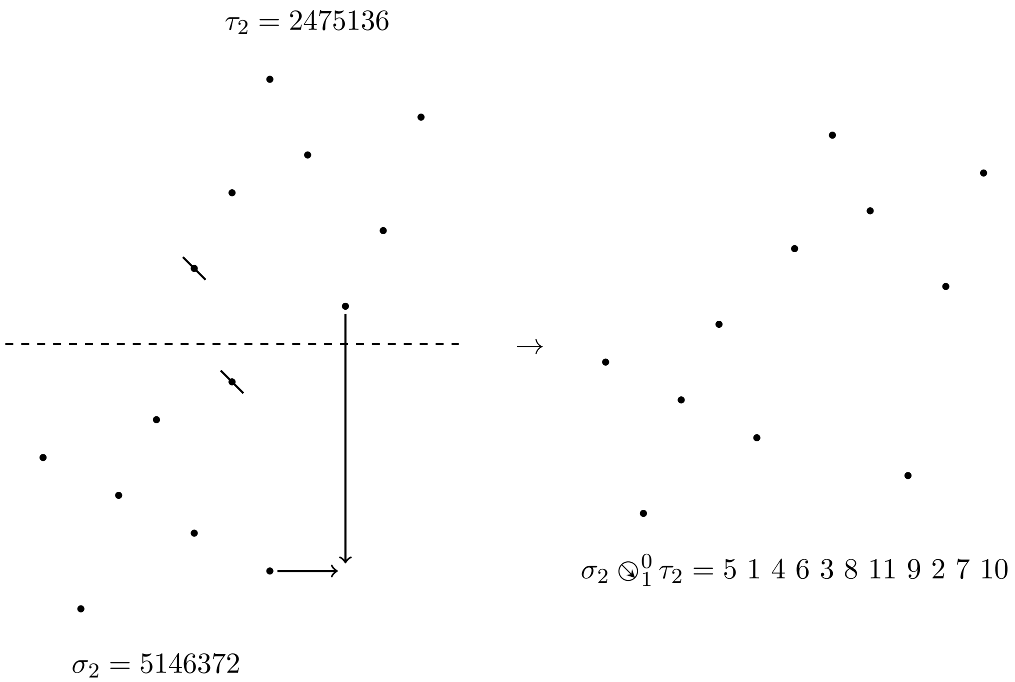

$\tau_2 = 2475136$
$\sigma_2 \mathbin{\textcircled{$\searrow$}}_1^0 \tau_2 = 5\ 1\ 4\ 6\ 3\ 8\ 11\ 9\ 2\ 7\ 10$
$\sigma_2 = 5146372$

Figure 4.8.: Illustration of $\sigma_2 \mathbin{\textcircled{$\searrow$}}_1^0 \tau_2$.

the inverse of $\sigma_1 \mathbin{\textcircled{$\nwarrow$}}_1^0 \tau_1 = 2\ 9\ 5\ 3\ 1\ 4\ 10\ 6\ 8\ 11\ 7$.

Note that glue sums we have defined so far are associative operations only if the lengths of all summands are at least 4. For our purpose, we set the convention that when we sum permutations with multiple glue sums, we always ensure to operate from left to right.

We are now ready to state the theorem for the simple permutations in H.

Theorem 4.4 *Let π be a permutation in H. Then there exist simple permutations in $\mathcal{A}$ of extreme pattern 2413 σ_i (i odd) and simple permutations in $\mathcal{A}$ of extreme pattern 3142 τ_i (i even) such that*

$$\pi = \begin{cases} \sigma_1 \mathbin{\textcircled{$\nwarrow$}}_1^{k_1} \tau_2 \mathbin{\textcircled{$\searrow$}}_1^{k_2} \sigma_3 \mathbin{\textcircled{$\nwarrow$}}_1^{k_3} \tau_4 \mathbin{\textcircled{$\searrow$}}_1^{k_4} \cdots \mathbin{\textcircled{$\searrow$}}_1^{k_{m-1}} \sigma_m & \textit{if } m \textit{ is odd} \qquad (a) \\ \sigma_1 \mathbin{\textcircled{$\nwarrow$}}_1^{k_1} \tau_2 \mathbin{\textcircled{$\searrow$}}_1^{k_2} \sigma_3 \mathbin{\textcircled{$\nwarrow$}}_1^{k_3} \tau_4 \mathbin{\textcircled{$\searrow$}}_1^{k_4} \cdots \mathbin{\textcircled{$\nwarrow$}}_1^{k_{m-1}} \tau_m & \textit{if } m \textit{ is even} \qquad (b) \end{cases} \tag{4.1}$$

where m is a positive integer and $k_\ell \in \{0, 1\}$ $(1 \leq \ell \leq m-1)$. Hence, π has one of the structures illustrated in Figure 4.9. Moreover, every simple permutation of these forms is in H.

In both Equation 4.1(a) and 4.1(b), we always make sure to sum permutations from left to right. As glue sums are injective operations when we sum simple permutations of 2413 and

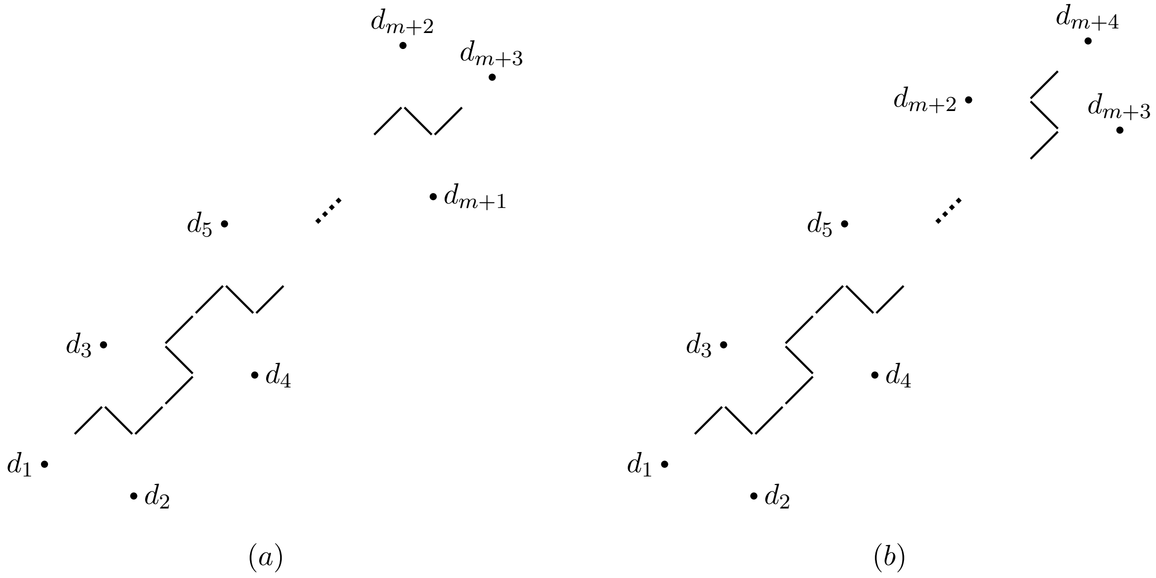


Figure 4.9.: Structure of π with $|\pi| \geq 4$ and $\pi(2) \neq 1$.

3142, every glue sum in Equation 4.1(a) and 4.1(b) are also injective.

We give the description of Figure 4.9. Each point of π is either one of the isolated points denoted by d_i or located on one of the sequences of lines. These isolated points are the ones identified by NW and SE glue sums except the first, the least, the greatest and the last ones. The precise identification of each d_i is given in the proof. We call the sequence of diagonal lines shown in Figure 4.9 a *crenellation* according to its suggested shape. This structure can be arbitrarily long depending on the length of π. Each line segment of the crenellation can contain any number of points or be empty, but points must be placed so that π avoids having blocks. Unlike the lines of the crenellation, the isolated points must be present so long as the structure continues.

Figure 4.10 shows the graph of a simple permutation π of length 21 in $\mathcal{A}$. In particular,

$$\pi = 2573146 \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 514263 \mathbin{\bigcirc\mkern-15mu\searrow}_1^1 246135 \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 6152473 \mathbin{\bigcirc\mkern-15mu\searrow}_1^0 2475316.$$

In order to actually place each point on the crenellation, we usually have to adjust the spacing between points. Thus, the points may not be located at $(i, \pi(i))$ any longer, but this does not change which permutation the graph represents.

Note that every permutation of extreme pattern 2413 is in H. Indeed, an N-shaped structure is a special case of a crenellation, namely one with only 4 isolated points. Thus, simple permutations of extreme pattern 2413 follow the structure described in Figure 4.9(a) with $m = 1$.

4.3.2. Proof of Theorem 4.4 (Part 1)

The proof of Theorem 4.4 is much longer than those of the propositions we discussed earlier, so we break this proof into two propositions.

Proposition 4.5 If π is a simple permutation in H, then π has one of the structures illustrated in Figure 4.9.

Proof. Suppose π is a simple permutation in $\mathcal{A}$ with $|\pi| = n \geq 4$ and $\pi(2) \neq 1$. We define a sequence $d_1, \ldots, d_{m+3}$ of values of π. Let $d_1 = \pi(1)$ and

$$d_i = \begin{cases} \pi(\max\{s : \pi(s) < \pi(d_{i-1})\}) & \text{if } i \text{ is even} \\ \max\{t : \pi^{-1}(t) < \pi^{-1}(d_{i-1})\} & \text{if } i \text{ is odd} \end{cases}$$

for i with $1 \leq i \leq m+3$. In other words, d_i is the right-most value that is less than d_{i-1} if i is even or the greatest value located to the left of d_{i-1} if i is odd. Note that by definition, $d_2 = 1$ for any $\pi \in H$. We will soon show that $d_i \neq d_j$ for $i \neq j$. Therefore, since π has a finite length, we eventually obtain either $d_{m+2} = \pi(n)$ and $d_{m+3} = n$ for some even integer m or $d_{m+2} = n$ and $d_{m+3} = \pi(n)$ for some odd integer m. We let m denote this integer in the remainder of this section.

First, we show $d_i \neq d_j$ for $i \neq j$. By the definition of d_i, it is clear that $d_i \leq d_{i+2} \leq d_{i+4} \leq \cdots$ for odd i and $\pi^{-1}(d_i) \leq \pi^{-1}(d_{i+2}) \leq \pi^{-1}(d_{i+4}) \leq \cdots$ for even i. We claim that, in addition, $d_i \leq d_{i+2} \leq d_{i+4} \leq \cdots$ for even i and $\pi^{-1}(d_i) \leq \pi^{-1}(d_{i+2}) \leq \pi^{-1}(d_{i+4}) \leq \cdots$ for odd

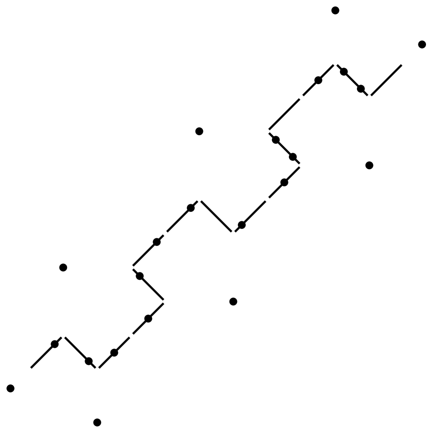


Figure 4.10.: Graph of $\pi = 2\ 5\ 9\ 3\ 1\ 4\ 8\ 6\ 10\ 12\ 17\ 7\ 11\ 16\ 13\ 15\ 19\ 22\ 20\ 18\ 14\ 21$.

i. Suppose $d_i > d_j$ for positive even integers i and j where $i < j$. Then $d_j < d_{i-1}$, but by definition, d_i is the right-most value that is less than d_{i-1}, so we achieve a contradiction. Similarly, if $\pi^{-1}(d_i) > \pi^{-1}(d_j)$ for odd i, j where $i < j$, then $\pi^{-1}(d_j) < \pi^{-1}(d_{i-1})$, which contradicts the fact that d_i is the greatest value located to the left of d_{i-1}. Therefore, $d_i \leq d_{i+2} \leq d_{i+4} \leq \cdots$ and $\pi^{-1}(d_i) \leq \pi^{-1}(d_{i+2}) \leq \pi^{-1}(d_{i+4}) \leq \cdots$ for any positive integer i.

Next, we show $d_i \neq d_j$ where i is odd and j is even. It is immediate by definition that $d_i \neq d_{i+1}$ for any i. For the case $|i-j| > 1$, assume for contradiction that we have $d_i = d_j$ and $i < j$. Then by definition, d_{j-1} must be located to the right of $d_j = d_i$, which cannot be true since $\pi^{-1}(d_i) \leq \pi^{-1}(d_j)$. Likewise, if $i > j$, then $d_{i-1} < d_i = d_j$, which is also a contradiction.

We now claim that, if $d_i \neq n$ for odd i, then $d_i \neq d_{i+2}$. Suppose to the contrary that $d_i = d_{i+2}$. Note that $d_{i+1} \neq \pi(n)$ since $d_{i+1} = \pi(n)$ implies $d_{i+2} = d_i = n$. By definition, d_{i+1} is the right-most value which is less than d_i, and d_i is the greatest whose position is in $[1, \pi^{-1}(d_{i+1})]$. These imply that any position corresponding to a value less than d_i is in $[1, \pi^{-1}(d_{i+1})]$ and any value corresponding to a position less than d_{i+1} is in $[i, d_i]$. Hence, $[1, \pi^{-1}(d_{i+1})]$ is a block with $d_{i+1} \neq \pi(n)$, which is a contradiction.

To show $d_i \neq d_j$ for two distinct odd i and j where $|i-j| > 2$, we assume $d_i = d_j$. Without loss of generality, say $i < j$. With $d_i \leq d_{i+2} \leq d_{i+4} \leq \cdots \leq d_j$, we have $d_i = d_{i+2}$, which cannot be true. Thus, $d_i \neq d_j$ for any two distinct odd integers i and j, so long as $d_i \neq n$. By the inverse argument, $d_i \neq d_j$ for even i and j so long as $d_i \neq \pi(n)$.

Hence, we have shown that $d_i \neq d_j$ for $i \neq j$.

Note that $d_2 \neq \pi(n)$ since $d_2 = 1$ and π is simple. Hence, every simple permutation in H has at least 4 values denoted by d_i for some $i \geq 1$. For a given π, suppose π has $m+3$ values denoted by d_i $(1 \leq i \leq m+3)$. We show that π satisfies one of Equations 4.1 by induction on $m \geq 1$.

For the base case, suppose $m = 1$. Then d_4 is the last value of π. Notice that each d_i $(1 \leq i \leq 4)$ is an extreme point of π, and since the flattening of $d_1 d_3 d_2 d_4$ is 2413, π has extreme pattern 2413, so we are done.

Now, suppose that every $\pi \in H$ with $m+3$ values denoted by d_i $(1 \leq i \leq m+3)$ satisfies Equation 4.1(a) for some positive odd integer m. We first need to show that an arbitrary $\pi \in H$ of length n with $m+4$ values denoted by d_i $(1 \leq i \leq m+4)$ satisfies Equation 4.1(b). Let π be a permutation in H of length n with $m+4$ values denoted by d_i. In this case, $d_{m+4} = n$. Let

p_m be the following.

$$p_m = \pi(\min\{\pi^{-1}(s) : s > d_{m+3}, \pi^{-1}(s) > \pi^{-1}(d_{m+1})\}).$$

In other words, p_m is the left-most value greater than d_{m+3} and located to the right of d_{m+1}. Since $d_{m+4} > d_{m+3}$ and d_{m+4} is located to the right of d_{m+1}, $\pi^{-1}(p_m) \leq \pi^{-1}(d_{m+4})$.

Next, we define q_m and r_m by

$$q_m = \pi(\pi^{-1}(p_m) - 1) \qquad \text{and} \qquad r_m = \max\{\pi(s) < d_{m+3} : s \in [1, \pi^{-1}(q_m)]\}$$

i.e. q_m is the value immediately to the left of p_m, and r_m is greatest value less than d_{m+3} whose position is in the segment $[1, \pi^{-1}(q_m)]$. It is possible that $q_m = d_{m+1}$ or $q_m = r_m$, but not both, since $r_m \geq d_m > d_{m+1}$. Note that $q_m < d_{m+3}$ because, $q_m \geq d_{m+3}$ contradicts the definition of p_m as q_m is located to the left of p_m and greater than d_{m+3}.

We provide Figure 4.11 to show the relations among p_m, q_m, r_m and d_i. The value p_m is the left-most value in the region denoted by R_1. It is possible to have $p_m < d_{m+2}$. The value q_m is immediately to the left of q_m, and as noted previously, it has to be less than d_{m+3}. The position of r_m can also be to the left of d_{m+2}.

We claim that there is no value z with $d_{m+3} < z < d_{m+2}$ whose position is in the segment $[1, \pi^{-1}(q_m)]$. In other words, there is no point in the shaded region denoted by R_2 in 4.11. First, there is no value in the intersected region of R_1 and R_2, because p_m is the left-most value in the region R_1. Suppose we have a value in the region $R_2 \setminus R_1$. That is, there exists a value z in B_{21} or B_{22} of the graph in Figure 4.12(a). If z is in B_{22}, then we have the graph shown in Figure 4.12(b). The segment $[\pi^{-1}(d_{m+2}), \pi^{-1}(x)]$ of Figure 4.12(b) is an unsplittable block, so

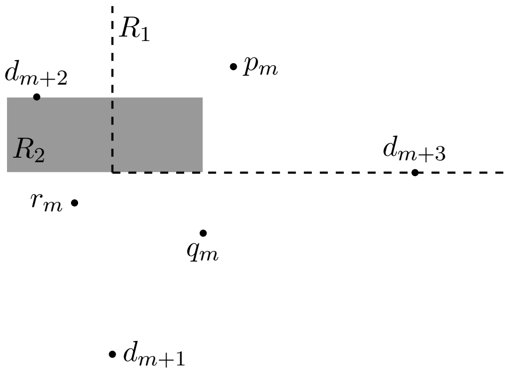


Figure 4.11.: Illustration of relations among p_m, q_m, r_m and d_i.

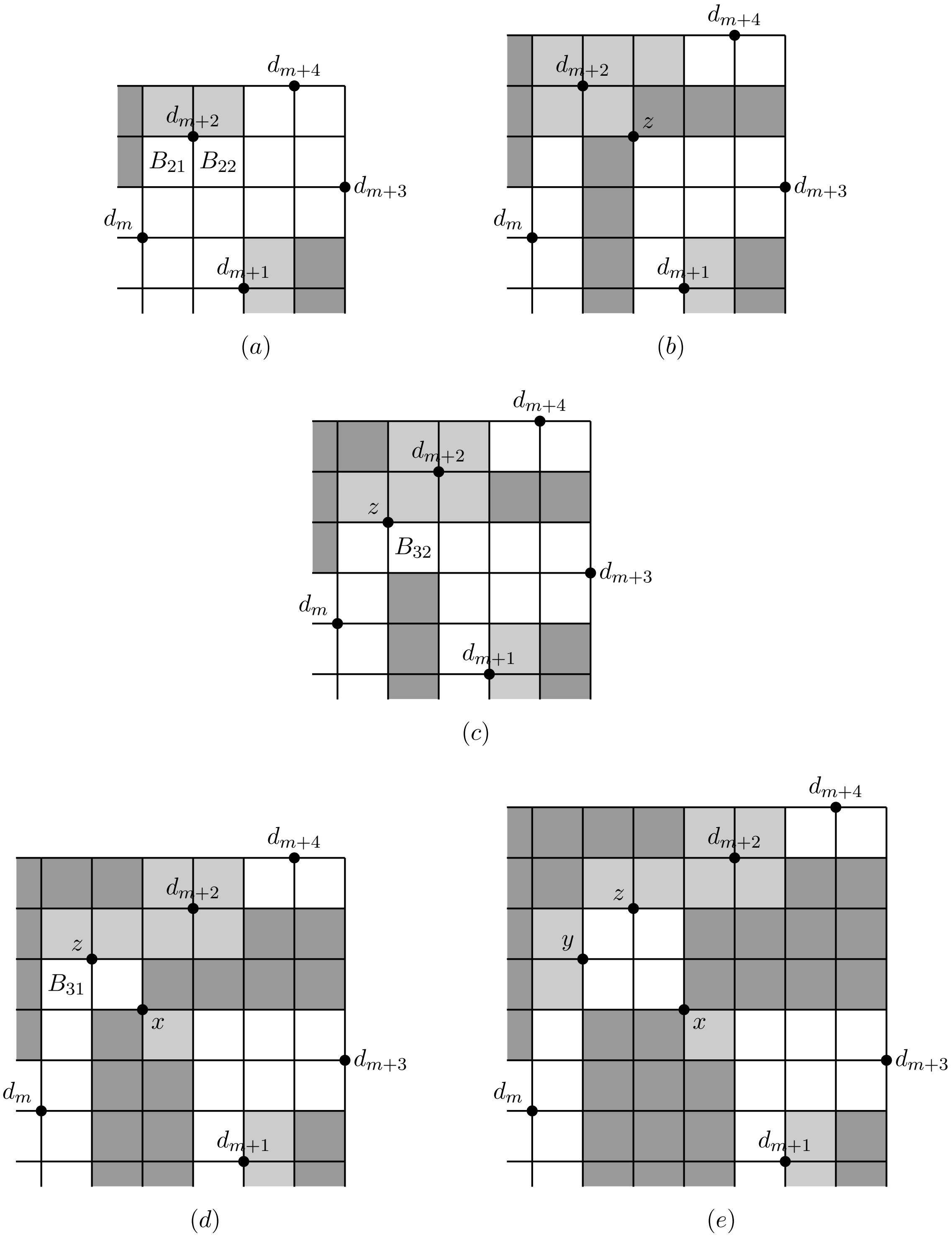


Figure 4.12.: Partial graphs of π to show that there exists no value in R_2.

we achieve an immediate contradiction.

Next, assume z is in the region B_{21} of Figure 4.12(a). Consequently, we have the graph shown in Figure 4.12(c) with the block $[\pi^{-1}(z), \pi^{-1}(d_{m+2})]$ that needs to be split. The only way to do this is by assuming the existence of a point in the region B_{32}. Choose a point with the least value, say x, so we have the graph shown in Figure 4.12(d). To split the block $[\pi^{-1}(z), \pi^{-1}(x)]$ in Figure 4.12(d), choose the left-most value y in the region B_{31}, so we finally have the graph as in Figure 4.12(e). The segment $[\pi^{-1}(y), \pi^{-1}(x)]$ is a block that cannot be split, so we have a contradiction. Therefore, there is no value z with $d_{m+3} < z < d_{m+2}$ and $\pi^{-1}(z) \in [1, \pi^{-1}(q_m)]$.

Denote by π_2 the flattening of the subsequence of π obtained by removing every value except d_{m+2} and d_{m+1} corresponding to a position in $[1, \pi^{-1}(q_m)]$. We claim that π_2 is a simple permutation in $\mathcal{A}$ with $|\pi_2| \geq 4$ and $\pi_2(2) = 1$. From here, we use the hat notation on a variable, such as $\hat{x}$, to refer to the value of π_2 corresponding to the value of π denoted by x. It is obvious that π_2 is still in $\mathcal{A}$ since $\pi_2 \preceq \pi$ and $\pi \in \mathcal{A}$. Also, since $\hat{d}_{m+1}$, $\hat{d}_{m+2}$, $\hat{d}_{m+3}$ and $\hat{d}_{m+4}$ are distinct values of π_2 of π_2, $|\pi_2| \geq 4$. Notice that $\pi_2(1) = \hat{d}_{m+2}$ and $\pi_2(2) = \hat{d}_{m+1} = 1$. Hence, the only thing we need to show is that π_2 is simple.

Suppose π_2 is not simple. This implies that we have a proper non-singleton segment I of π_2 which is a block. There are three cases to consider. First, assume $I = [\pi_2^{-1}(\hat{x}), \pi_2^{-1}(\hat{y})]$ for some values $\hat{x}$ and $\hat{y}$ where $\pi_2^{-1}(\hat{x}) > \pi_2^{-1}(\hat{d}_{m+1})$. Because every point with a position less than $\pi^{-1}(q_m)$ was removed, we must have $\pi^{-1}(q_m) < \pi^{-1}(x)$, so $[\pi^{-1}(x), \pi^{-1}(y)]$ is also a block in π. Since π is simple, π having a block is a contradiction. Next, suppose $I = [\pi_2^{-1}(\hat{d}_{m+1}), \pi_2^{-1}(\hat{y})]$ for some $\hat{y}$. In this case, the value immediately to the right of $\hat{d}_{m+1}$ must be in $\pi(I)$. By construction, this value is $\hat{p}_m$. Because $\hat{d}_{m+1} < \hat{d}_{m+3} < \hat{p}_m$, $\hat{d}_{m+3} \in \pi(I)$, implying $\pi_2^{-1}(\hat{d}_{m+3}) \in I$. However, since $\pi_2^{-1}(\hat{d}_{m+1}) < \pi_2^{-1}(\hat{d}_{m+4}) < \pi_2^{-1}(\hat{d}_{m+3})$, we have $\pi_2^{-1}(\hat{d}_{m+4}) \in I$, which implies $\hat{d}_{m+4} \in \pi(I)$. Since $\hat{d}_{m+4}$ is the greatest value in π_2, $\pi(I) = [\hat{d}_{m+1}, \hat{d}_{m+4}]$, so I is all of π_2, contradiction. Finally, assume $I = [\pi_2^{-1}(\hat{d}_{m+2}), \pi_2^{-1}(\hat{y})]$ for some value $\hat{y}$. Since $\hat{d}_{m+1} \in \pi_2(I)$ and $\hat{d}_{m+3} < \hat{d}_{m+2}$, $\hat{d}_{m+3}$ also has to be in $\pi_2(I)$, so $\pi_2^{-1}(\hat{d}_{m+3}) \in I$. However, since $\hat{d}_{m+3}$ is the last value of π_2, this implies $\hat{y} = \hat{d}_{m+3}$, which means I is all of π_2. Consequently, we conclude that π_2 is simple. Moreover, π_2 has extreme pattern 3142.

Next, let π_1 be the flattening of the subsequence $\pi(1)\pi(2)\cdots\pi(\pi^{-1}(q_m) - 1)d_{m+3}$ of π if $q_m = r_m$. In other words, if q_m is the second greatest value whose position is in the segment $[1, \pi^{-1}(q_m)]$, then π_1 is the flattening of the sequence containing all values up to the one immediately to the left of q_m, and d_{m+3}. Otherwise, let π_1 be the flattening of the subsequence $\pi(1)\pi(2)\cdots q_m d_{m+3}$ of π. We again use the hat notation to refer to the value of π_1 corresponding to the value of π. We claim that π_1 is simple.

First, we show that every value we removed to construct π_1 is greater than or equal to r_m. Suppose to the contrary that there exists a value $z < r_m$ that was removed. Since $p_m > d_{m+3}$ by definition and $d_{m+3} > r_m$, we have $p_m > r_m$, so z cannot be p_m. Also, if $z = q_m$, then $q_m = r_m$ as well, so we have a contradiction. Therefore, we must have $\pi^{-1}(z) > \pi^{-1}(p_m)$. There are four subcases to consider. If $\pi^{-1}(r_m) > \pi^{-1}(d_{m+2})$ and $p_m < d_{m+2}$, then π contains 4231 pattern with $d_{m+2}r_mp_mz$. If $\pi^{-1}(r_m) > \pi^{-1}(d_{m+2})$ and $p_m > d_{m+2}$, then π contains 42513 pattern with $d_{m+2}r_mp_mzd_{m+3}$. If $\pi^{-1}(r_m) < \pi^{-1}(d_{m+2})$ and $p_m < d_{m+2}$, then π contains 35142 pattern with $r_md_{m+2}d_{m+1}p_mz$, and finally, if $\pi^{-1}(r_m) < \pi^{-1}(d_{m+2})$ and $p_m > d_{m+2}$, then π contains 351624 pattern with $r_md_{m+2}d_{m+1}p_mzd_{m+3}$. Hence, every value we removed to construct π_1 must be greater than or equal to r_m.

Now, assume π_1 is not simple. Let I be a proper non-singleton block of π_1. Suppose $\pi_1(I) = [\hat{x}, \hat{y}]$ for some values $\hat{x}$ and $\hat{y}$ where $\hat{y} < \hat{d}_{m+3}$. Note that $\hat{y} < \hat{d}_{m+3}$ implies $\hat{y} \leq \hat{q}_m - 1$ if $q_m = r_m$ and $\hat{y} \leq \hat{q}_m$ if $q_m \neq r_m$. In either case, for all $\hat{z} \in [\hat{x}, \hat{y}]$, $\hat{z} = z$ since every point we removed had a value greater than or equal to r_m. Hence, the block I in π_1 is also a block in π, so we achieve a contradiction. Next, assume $\pi_1(I) = [\hat{x}, \hat{d}_{m+3}]$. Since I is a block, $\hat{d}_{m+3} - 1$ must be in $[\hat{x}, \hat{d}_{m+3}]$. By the way we constructed π_1, the position of $\hat{d}_{m+3} - 1$ must be either to the left of $\hat{d}_{m+2}$ or in between of $\hat{d}_{m+2}$ and $\hat{d}_{m+1}$. In either case, it implies that $\pi_1^{-1}(\hat{d}_{m+1}) \in I$, so $\hat{d}_m \in \pi_1(I)$ as $\hat{d}_{m+1} < \hat{d}_m < \hat{d}_{m+3}$. Hence, $\pi^{-1}(\hat{(d)}_m) \in I$, so $\pi^{-1}(\hat{(d)}_{m+2}) \in I$, which means $\hat{d}_{m+2} \in \pi_1(I)$. However, $\hat{d}_{m+2} > \hat{d}_{m+3}$, so we have a contradiction. Finally, suppose $\pi_1(I) = [\hat{x}, \hat{d}_{m+2}]$. Because $r_m < d_{m+3} < d_{m+2}$, we have $\hat{d}_{m+3} = \hat{d}_{m+2} - 1$. Thus, $\hat{d}_{m+3} \in [\hat{x}, \hat{d}_{m+2}]$. Because the position of $\hat{d}_{m+1}$ is in between $\hat{d}_{m+2}$ and $\hat{d}_{m+3}$, we must have $\hat{d}_{m+1}$ in $[\hat{x}, \hat{d}_{m+2}]$. Now, since $\hat{d}_{m+1} < \hat{d}_m < \hat{d}_{m+2}$, $\hat{d}_m$ must be also in $[\hat{x}, \hat{d}_{m+2}]$, but this again implies $\hat{d}_{m-1} \in [\hat{x}, \hat{d}_{m+2}]$, because $\pi_1^{-1}(\hat{d}_m) < \pi_1^{-1}(\hat{d}_{m-1}) < \pi_1^{-1}(\hat{d}_{m+1})$. Continuing in this way, we must have $\hat{d}_1 = 1$ in $[\hat{x}, \hat{d}_{m+2}]$, but then I is all of π_1. Since I is a proper block, this is a contradiction. Consequently, π_1 is simple.

Since π_1 is a simple permutation of length 4 or more with $\pi(2) \neq 1$, π_1 is in H with $m+3$ values denoted by d_i $(1 \leq i \leq m+3)$, π_1 has the form expressed in Equation 4.1(a) by the induction hypothesis. Let $n_1 = |\pi_1|$ and $n_2 = |\pi_2|$. If $q_m = r_m$, then we apply $\mathbin{\bigcirc\hspace{-0.75em}\nwarrow}_1^1$. The greatest value n_1 of π_1 is shifted upward by $\pi_2(1) - 2$, which is how much d_{m+2} was shifted down to construct π_1 by flattening. The value $\pi_1(n_1)$ is $n_1 - 1$ in π_1, and it is the second greatest value up to the position of itself in π. By definition, this is $r_m = q_m$ in π. Finally, to construct π_2, we removed $n_1 - 2$ values from π, so shifting up each value of π_2, except $\pi_2(1)$ and $\pi_2(2)$, by $n_1 - 2$ will recover the values of π corresponding to π_2. Hence, $\pi_1 \mathbin{\bigcirc\hspace{-0.75em}\nwarrow}_1^1 \pi_2 = \pi_1(1) \cdots d_{m+2} \cdots q_mp_m \cdots d_{m+3}$.

Similarly, if $q_m \neq r_m$, then $\pi = \pi_1(1) \cdots d_{m+2} \cdots q_m p_m \cdots d_{m+3}$. In either case, $\pi \in H$ with $m+4$ values denoted by d_i has the form expressed in Equation 4.1(b), so we are done.

For our purpose, in the process of combining π_1 and π_2 with $\obslash_1^0$, it is more appropriate to think that $\pi_1(n_1)$ and $\pi_2(1)$ are combined into d_{m+3} and d_{m+1} respectively, as these are the values corresponding to them in π_1 and π_2 after the flattening. For $\obslash_1^1$, even though it appears as if $\pi_1(n_1)$ stays where it is, it is still proper to think that $\pi_1(n_1)$ is merged into the right-most value of π_2 to become d_{m+3}, and leaving a copy of itself at where it used to be as the value q_m.

Next, assume m is even, that is, π in H with $m+3$ values denoted by d_i $(1 \leq i \leq m+3)$ satisfies Equation 4.1(b) for some positive even integer m. Then again, we need to show that for $\pi \in H$ of length n with $m+4$ values denoted by d_i $(1 \leq i \leq m+4)$ satisfies Equation 4.1(a). This time, let p_m, q_m and r_m denote the following.

$$p_m = \min\{s : \pi^{-1}(s) > \pi^{-1}(d_{m+3}), s > d_{m+1}\}, \qquad q_i = p_m - 1,$$

$$r_m = \pi(\max\{\pi^{-1}(t) < \pi^{-1}(d_{m+3}) : t \in [1, q_m]\}).$$

The rest of the proof is the the same argument applied to the inverses of all permutations involved. At the end, we acquire Equation 4.1(a) for π, and this completes the proof. ■

Because, for any $\pi \in H$, π is simple and the values corresponding to the segment $[1, \pi^{-1}(d_3)]$ are increasing, we conclude $d_1 = 2$.

The crenellation can be viewed as a repetition of N and S structures. Each N-shape corresponds to a simple permutation σ_i (i odd) of extreme pattern 2413 in Equations 4.1, whereas each S-shape corresponds to a simple permutation τ_i (i even) of extreme pattern 3142. Notice that m in the previous proof indicates the total number of N and S structures. With the condition $\pi(2) \neq 1$, the structure always starts with an N-shape. Therefore, π with $m+3$ values denoted by d_i $(1 \leq i \leq m+3)$ has $(m+1)/2$ N-shapes and $(m-1)/2$ S-shapes if m is odd or $m/2$ N-shapes and $m/2$ S-shapes if m is even. We call m the number of components of π.

4.3.3. Proof of Theorem 4.4 (Part 2)

We now prove that any simple permutation of the structure described in Figure 4.9 is in H. Before we start, we introduce some new terminology. A value of a permutation $\pi(i)$ is called a *left-to-right maximum* if for all j with $1 \leq j \leq i$, $\pi(j) \leq \pi(i)$. In other words, $\pi(i)$ is a left-to-right maximum if it is greater than every value on its left. We define a *right-to-left minimum* analogously. For example, with the permutation

$$\pi = 2\ 5\ 9\ 3\ 1\ 4\ 8\ 6\ 10\ 12\ 17\ 7\ 11\ 16\ 13\ 15\ 19\ 22\ 20\ 18\ 14\ 21$$

which was provided in Figure 4.10, the subsequence of left-to-right maxima is 2 5 9 10 12 17 19 22 and the subsequence of right-to-left minima is 1 4 6 7 11 13 14 21. We denote by LRmax(π) and RLmin(π) the set of left-to-right maxima values of π and the set of right-to-minima values of π respectively.

Proposition 4.6 Let π be a simple permutation whose structure is described in Figure 4.9. Then π is in H.

Proof. Suppose π is a simple permutation of length n whose structure is as described in Figure 4.9 with m components that was explained previously. We show π avoids each permutation in the basis $\{4231, 35142, 42513, 351624\}$. First, we break down the crenellation and set notation for the sets of values in different parts of the crenellation.

Define each d_i $(1 \leq i \leq m+3)$ in the same way as in the previous section. Let N_i (i odd) be the set of values corresponding to the permutation σ_i of extreme pattern 2413 inπ. Similarly, let S_i (i even) denote the set of values corresponding to the permutation τ_i of extreme pattern 31422 in π. Let u_i be the first value of N_i if i is odd or the least value of S_i if i is even. If i is odd,

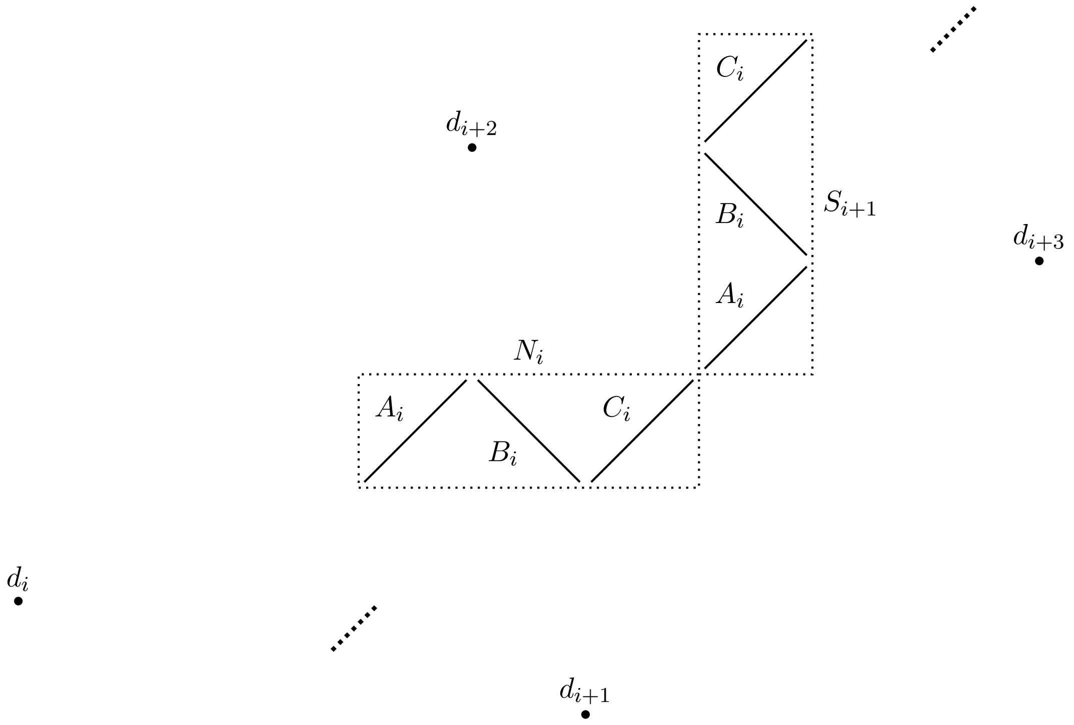


Figure 4.13.: Notations for sets of values in the crenellation.

let A_i, B_i and C_i be the set of values corresponding to positions in $[\pi^{-1}(u_i), \pi^{-1}(d_{i+2}))$, the set of values corresponding to positions in $(\pi^{-1}(d_{i+2}), \pi^{-1}(d_{i+1}))$ and the set of values corresponding to positions in $(\pi^{-1}(d_{i+1}), \pi^{-1}(q_i)]$ respectively. If i is even, let $A_i = [u_i, d_{i+2})$, $B_i = (d_{i+2}, d_{i+1})$ and $C_i = (d_{i+2}, q_i]$ respectively. Figure 4.13 illustrates the definitions of each set of values.

Furthermore, let

$$D = \bigcup_{i=1}^{m+3} \{d_i\}, \quad A_{\text{odd}} = \bigcup_{i \text{ odd}} A_i \quad \text{and} \quad A_{\text{even}} = \bigcup_{i \text{ even}} A_i,$$

and define B_{odd}, B_{even}, C_{odd}, C_{even} analogously. Hence, for i with $1 \le i \le m$, $A_i \cup B_i \cup C_i = N_i$ if i is odd, $A_i \cup B_i \cup C_i = S_i$ if i is even and

$$D \cup \left(\bigcup_{i \text{ odd}} N_i \right) \cup \left(\bigcup_{i \text{ even}} S_i \right) = \{1, \dots, n\}.$$

Now we are ready to show that π avoids 4231. Note that

$$\text{LRmax}(\pi) = \left(\bigcup_{i \text{ odd}}^{m+3} \{d_i\} \right) \cup A_{\text{odd}} \cup C_{\text{even}} \qquad \text{and} \qquad \text{RLmin}(\pi) = \left(\bigcup_{i \text{ even}}^{m+3} \{d_i\} \right) \cup A_{\text{even}} \cup C_{\text{odd}}$$

Hence, if any two values from these two sets play the role of 2 or 3, we cannot find a value corresponding to either 4 or 1. This implies that role of 2 and 3 must be played by two values from $B_{\text{odd}} \cup B_{\text{even}}$. Moreover, since values in each B_i are decreasing, a value playing the role of 2 and a value for 3 must come from distinct B_i and B_j where $i < j$. If $j = i + 1$, we are able to assign d_{i+2} to play the role of either 4 or 1 (depending on whether i is even or odd), but then we cannot find another value for the other. Consequently, we are unable to find a subsequence whose flattening is 4231, so $4231 \not\preceq \pi$.

For 35142, a value corresponding to 4 must come from the set B for the same reason as for the 2 and 3 in 4231. If it is from B_{odd}, say B_i for some odd i, then a value playing the role of 5 must either also come from A_i, from B_i or be d_{i+2}, but there is no value that can be assigned to 1 in between. Similarly, if a value for 4 of 35142 is from B_i for some even i, a value for 2 must be from A_i, from B_i or be d_{i+2}, but then we cannot choose a value for 3. Thus, $35142 \not\preceq \pi$. We can apply the reverse complement argument to show that π avoids 42513 as well.

Lastly, suppose π contains 351624. Suppose a value for 3 comes from B_i for some odd i. Then a value for 1 must be in $B_i \cup C_i \cup \{d_{m+1}\}$. Whichever it is, a value for 2 must be from C_i, but this prevents us from assigning a value to 6. Next, assume a value from B_i for some even i plays the role of 3. If d_{i+2} is for 1, then there is no value for 2, so a value for 1 must come from A_i or B_i. This forces a value corresponding to 5 to come from C_i, but then there is no possible value for 4. Hence, a value for 3 must be from LRmax(π) or RLmin(π). Carrying out similar arguments, we can show that every value comes from LRmax(π) or RLmin(π). Since values in RLmin(π) are right-to-left minima, a value for 3 must be from LRmax(π). So suppose it is from A_i where i is odd. Then a value playing the role of 1 must belong to B_i or it is d_{i+1}. Either way, we are able to assign a value from C_i to the role of 2, but then not for 6. If a value for 3 is from C_i where i is even, then a value for 1 cannot be d_{i+2} as this leaves no choice for 2. Hence, a value playing the role of 1 must be from either A_i or B_i, forcing a value for 5 to be from C_i. However, we now don't have a value for 4, which is a contradiction. Thus, d_i for some odd i must play the role of 3. A value for 5 may come from either C_{i-1} or A_i, or it is d_{i+2}. If it's d_{i+2} or from A_i, a value for 1 must be d_{i+1}, but as before, we cannot find a value for 2. Thus, a value for 5 must be from C_{i-1}, but again, we cannot assign a value for 4, which is another contradiction. Therefore, we achieve a contradiction in every case, implying π avoids 351624.

Since π avoids every permutation in the basis, we have the desired result. ■

Consequently, with Proposition 4.5 and 4.6, we have Theorem 4.4.

4.4. Enumeration

We are now ready to enumerate the class $\mathcal{A}$. As briefly mentioned in Section 4.1, we first find the generating function for the set $H = \{\pi \in \mathrm{Si}(\mathcal{A}) : |\pi| \geq 4 \text{ and } \pi(2) \neq 1\}$. Since the inverses of these permutations give us all the permutations whose second value is 1, doubling the generating function for H gives us the generating function for all simple permutations of length greater than or equal to 4.

4.4.1. Enumeration of simple permutations in $\mathcal{A}$

Let us first state the result.

Theorem 4.7 *Let $f_{Si(\mathcal{A})\setminus\mathcal{S}_2}$ be the generating function for the set of simple permutations in $\mathcal{A}$ excluding $\mathcal{S}_2 = \{12, 21\}$. Then*

$$f_{\mathrm{Si}(\mathcal{A})\setminus\mathcal{S}_2} = \frac{2x^4}{(1-3x)(1+x)}.$$

Once we prove Theorem 4.7, it is trivial to find $f_{\mathrm{Si}(\mathcal{A})}$, the generating function for all simple permutations in $\mathcal{A}$; simply, we add $2x^2$ to the result to include the permutations 12 and

21.

Let $\Sigma = \{a, b, c, d, d_\ell\}$. We take the following six steps to accomplish the proof of Theorem 4.7.

1. Define an encoding function ϕ from H to Σ^*.
2. Define a language $L \subseteq \Sigma^*$.
3. Prove ϕ is a bijection between H and L.
4. Define another language $\overline{L} \subseteq \Sigma^*$ which is related to L.
5. Define an automaton M such that $\mathcal{L}(M) = \overline{L}$.
6. Apply the transfer matrix method to M to enumerate $|\mathcal{L}(M)| = |\overline{L}| = |L| = |H|$.

We start by defining ϕ which maps an arbitrary simple permutation π in H to Σ^*. Let $\pi \in H$ and suppose the number of components of π is m. By looking at the structure of π, we can find which of A_i, B_i, C_i ($1 \le i \le m$) or D each value of π belongs to. The encoding function ϕ reads each value of π in certain order and writes out a unique word w consisting of letters in Σ. Lay out each set of values in the following order.

$$\{d_1, d_2\} \to N_1 \to \{d_3\} \to S_2 \to \{d_4\} \to N_3 \to \{d_5\} \to S_4 \to \cdots$$

This ends as

$$\cdots \to \{d_{m+1}\} \to N_m \to \{d_{m+2}, d_{m+3}\} \text{ if } m \text{ is odd}$$

or

$$\cdots \to \{d_{m+1}\} \to S_m \to \{d_{m+2}, d_{m+3}\} \text{ if } m \text{ is even.}$$

Simply, ϕ writes a, b, c and d for any value in A_i, B_i, C_i and D respectively, except that it writes d_ℓ for d_{m+2}. For N_i ($1 \le i \le m$ and i odd), ϕ encodes values from bottom to top, *i.e.* ϕ first writes the smallest value, then the second smallest value, and so on. On the other hand, ϕ encodes values of S_i ($1 \le i \le m$ and i even) from left to right.

The order of encoding for $\{d_1, d_2\}$ and $\{d_{m+2}, d_{m+3}\}$ actually does not matter, however, when we establish a similar encoding function for the case of $\mathcal{A}'$ in Chapter 6, it becomes important to set a convention. For this reason, ϕ shall decode d_2 first and then d_1 next, both into the letter d. Similarly, ϕ encodes d_{m+3} with d and finally d_{m+2} with d_ℓ.

As an example, we encode π described in Figure 4.10 using the encoding function ϕ. First, ϕ recognizes $d_2 = 1$ and $d_1 = 2$, and encodes both as d's. Moving onto $N_1 = \{3, 4, 5\}$, ϕ encodes

3 as b, 4 as c and 5 as a in this order, since $3 \in B_1$, $4 \in C_1$ and $5 \in A_1$. Next, $d_3 = 9$ is encoded as d, and ϕ continues to $S_2 = \{6, 8, 10\}$. Since ϕ encodes values of S_2 from left to right, it reads values in the order 10, 6, 8. Again, based on where they are placed, 10, 6 and 8 are encoded as c, a and b respectively. Continuing this process up to $d_7 = 22$, the encoded word $w = \phi(\pi)$ comes out as

$$w = d\ d\ b\ c\ a\ d\ b\ a\ c\ d\ c\ a\ d\ b\ a\ b\ d\ b\ a\ b\ d\ d_\ell$$

Arrows in Figure 4.14 shows the order of encoding by ϕ. First arrow reads d_2, d_1, then values in N_1 from bottom to top, and ends with d_3. Once we hit intermediate d's, we move to the next arrow. The last one encodes N_5, then finally d_8 and d_7.

Next, we define a language $L \subseteq \Sigma^*$ with the following conditions for every $w \in L$.

- w must begin with dd and end with dd_ℓ.
- w must not contain aa, bb or cc.
- d_ℓ is only allowed at the very end.
- w cannot begin with dda or end with cdd_ℓ.
- w must not contain da.

We will show ϕ is a bijection between H and L, but before we do so, we define a decoding function ψ from L to $\mathcal{A}$. Suppose w is a word in L. Let w_i $(1 \leq i \leq m)$ be the sub-word of w

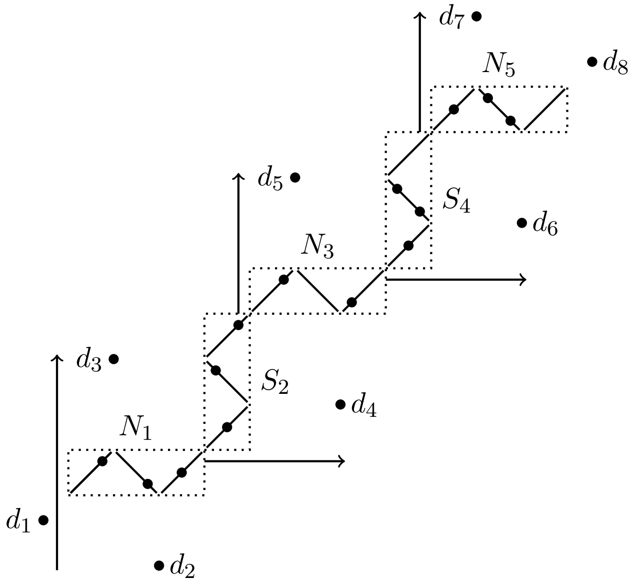


Figure 4.14.: Encoding of $\pi = 2\ 5\ 9\ 3\ 1\ 4\ 8\ 6\ 10\ 12\ 17\ 7\ 11\ 16\ 13\ 15\ 19\ 22\ 20\ 18\ 14\ 21$.

defined as the following.

$$w = \cdots \underbrace{d}_{(i{+}1)\text{-th } d} \underbrace{\cdots \quad \cdots}_{w_i} \underbrace{d}_{(i{+}2)\text{-th } d} \cdots$$

In particular, w_i is the consecutive letters of w from the letter immediately after the $(i+1)$-th d up to the letter immediately before the $(i+2)$-th d. Note that w_i can be empty. Let us divide w as below.

$$w = dd \ \rightarrow\ w_1 \ \rightarrow\ d \ \rightarrow\ w_2 \ \rightarrow\ \cdots \ \rightarrow\ d \ \rightarrow\ w_m \ \rightarrow\ dd_\ell$$

The decoding function ψ reads w from left to right. As ψ reads the k-th letter, it draws a point at the fixed location (x_k, y_k) $(1 \leq k \leq n)$. At the end, ψ constructs the graph of a permutation by drawing points iteratively. We define the decoding function ψ by the following algorithm called DECODE.

```
Algorithm DECODE
INPUT: A word $w$ in $L$.
OUTPUT: A permutation $\pi$ in $H$.

Initialize: Draw first two points and initialize variables.
            Draw points at $(2,1)$ and $(1,2)$. Let $P_a = (2,1)$ and $P_b = P_c = (1,2)$. Let $t = 2$. Let
            $\alpha$ be the third letter in $w$.

  C1se 1: Draw points for $w_i$ ($i$ odd) which corresponds to the set $N_i$.
            If $\alpha$ is in $w_i$ ($i$ odd), then BEGIN
              a. If $\alpha = a$, then draw a point at $(x,y)$ where $P_a^{(x)} < x < P_b^{(x)}$ and $t < y$. Set $P_a$ to
                 be this new point and $t = P_a^{(y)}$. GOTO STEP 1 with setting $\alpha$ to be the next letter.
              b. If $\alpha = b$, then draw a point at $(x,y)$ where $P_a^{(x)} < x < P_b^{(x)}$ and $t < y$. Set $P_b$ to be
                 this new point and $t = P_b^{(y)}$. GOTO STEP 1 with setting $\alpha$ to be the next letter.
              c. If $\alpha = c$, then draw a point at $(x,y)$ where $P_c^{(x)} < x$ and $t < y$. Set $P_c$ to be this
                 new point and $t = P_c^{(y)}$. GOTO STEP 1 with setting $\alpha$ to be the next letter.
            Otherwise, GOTO STEP 2.

  C2se 2: Draw points for $w_i$ ($i$ even) which corresponds to the set $S_i$.
            If $\alpha$ is in $w_i$ ($i$ even), then BEGIN
              a. If $\alpha = a$, then draw a point at $(x,y)$ where $t < x$ and $P_a^{(y)} < y < P_b^{(y)}$. Set $P_a$ to be
                 this new point and $t = P_a^{(x)}$. GOTO STEP 2 with setting $\alpha$ to be the next letter.
              b. If $\alpha = b$, then draw a point at $(x,y)$ where $t < x$ and $P_a^{(y)} < y < P_b^{(y)}$. Set $P_b$ to be
                 this new point and $t = P_b^{(x)}$. GOTO STEP 2 with setting $\alpha$ to be the next letter.
```

```
        c. If $\alpha = c$, then draw a point at $(x, y)$ where $t < x$ and $P_c^{(y)} < y$. Set $P_c$ to be this
           new point and $t = P_c^{(x)}$. GOTO STEP 2 with setting $\alpha$ to be the next letter.
        Otherwise, GOTO STEP 3.
C3se 3: Draw points for $d$'s which correspond to points $d_i$ $(1 \leq i \leq m + 3)$.
        If $\alpha = d$, then BEGIN
        a. If it is the last $d$ (i.e. second letter from the last in $w$), then BEGIN
           i. If it is immediately after $w_i$ with $i$ odd (possibly empty), then draw a point at
              $(x, y)$ where $P_c^{(x)} < x$ and $t < y$. Set $P_c$ to be this new point and $t = P_c^{(y)}$.
              GOTO STEP 4 with setting $\alpha$ to be the next letter.
           ii. If it is immediately after $w_i$ with $i$ even (possibly empty), then draw a point
               at $(x, y)$ where $t < x$ and $P_c^{(y)} < y$. Set $P_c$ to be this new point and $t = P_c^{(x)}$.
               GOTO STEP 4 with setting $\alpha$ to be the next letter.
        b. Otherwise, BEGIN
           i. If it is immediately after $w_i$ with $i$ odd (possibly empty), then draw a point at
              $(x, y)$ where $P_a^{(x)} < x < P_b^{(x)}$ and $t < y$. Set $P_a = (P_c^{(x)}, t)$, $P_b$ and $P_c$ to be this
              new point and $t = P_a^{(x)}$. GOTO STEP 2 with setting $\alpha$ to be the next letter.
           ii. If it is immediately after $w_i$ with $i$ even (possibly empty), then draw a point at
               $(x, y)$ where $t < x$ and $P_y^{(x)} < y < P_b^{(y)}$. Set $P_a = (t, P_c^{(y)})$, $P_b$ and $P_c$ to be this
               new point and $t = P_a^{(y)}$. GOTO STEP 1 with setting $\alpha$ to be the next letter.
C4se 4: Draw a point for $d_\ell$ which correspond to points $d_{m+3}$.
        If $\alpha = d_\ell$, then BEGIN
        a. If $m$ is odd (i.e. the last sub-word $w_m$ corresponds to $N_m$), then draw a point at
           $(x, y)$ where $P_a^{(x)} < x < P_b^{(x)}$ and $t < y$. GOTO STEP 5.
        b. If $m$ is even (i.e. the last sub-word $w_m$ corresponds to $S_m$), then draw a point at
           $(x, y)$ where $t < x$ and $P_a^{(y)} < y < P_b^{(y)}$. GOTO STEP 5.
C5se 5: Let $\pi$ be a permutation obtained by flattening the constructed graph. OUTPUT $\pi$.
```

We visualize how ψ decodes a word w into a permutation. After ψ draws points on $(2, 1)$ and $(1, 2)$ for the first and second d's, it draws points from bottom to top in intervals $I_1 = (1, 2)$ and $I_2 = (2, \infty)$ for each letter up to the next d. For a's and c's, ψ draws points from the left side of I_1 and I_2 respectively towards right and from the right side of I_1 towards left for b's. Additionally, the point for the proceeding d is drawn as if it is for another b, *i.e.* immediately left of the last b. Figure 4.15(a) describes how points are drawn until the third d.

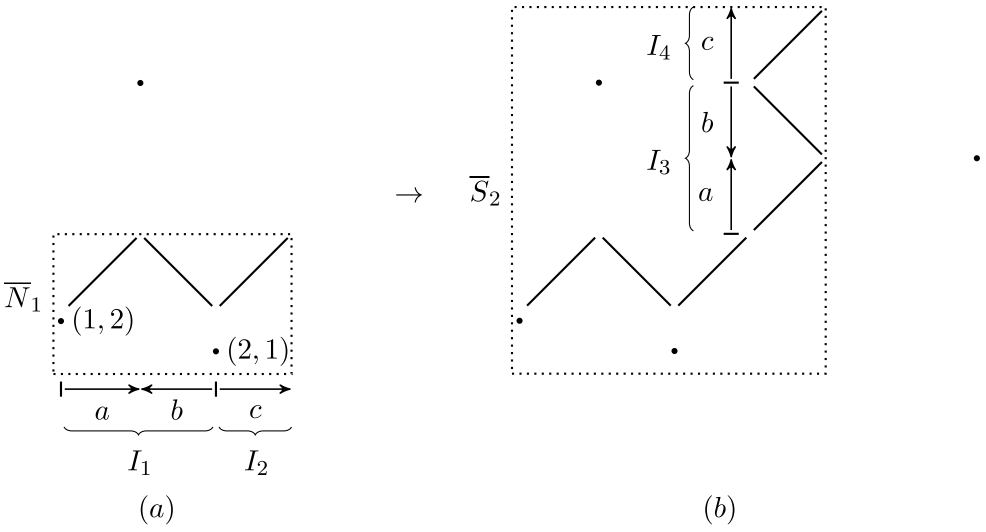


Figure 4.15.: Illustration of the decoding function ψ.

Drawing points for w_2 and the fourth d is shown in Figure 4.15(b). The point with the greatest x-coordinate value (*i.e.* the left-most point) in the section denoted by $\overline{N}_1$ in Figure 4.15(a) determines the x-coordinate of the point corresponding to the first letter in w_2. Also, the point with the greatest y-coordinate in $\overline{N}_1$ and the point corresponding to the third d together determines the next two intervals I_3 and I_4. Points are drawn from left to right. For a's and c's, ψ draws points upward from the bottom of I_3 and I_4 respectively, whereas points for b's are drawn downward from the top of I_3. Again, ψ draws a point for the fourth d as if it is for b. Thus, points for letters in w_2 and proceeding d are drawn as shown in Figure 4.15(b). We continue decoding by repeating this process. The last d and d_ℓ are drawn as if they are cb, both belonging to the last $\overline{N}_i$ or $\overline{S}_i$ section. At the end, we obtain the structure shown in Figure 4.9. Hence, m, the number of sub-words in w is the number of total N and S sets described in the second part of the proof of Theorem 4.4. Each d in w corresponds to d_i ($1 \leq i \leq m+3$) in the proof of Theorem 4.4, and a, b and c are placed to be values in the set A_i, B_i and C_i ($1 \leq i \leq m$) respectively.

Lemma 4.8 *The encoding function ϕ is a bijection between H and L.*

Proof. We first show the image of ϕ is in L. Let π be in H. Our goal is to show that $\phi(\pi) = w$ satisfies the conditions of L. It is clear that w begins with dd and end with dd_ℓ, since the first two values and last two values that are encoded by ϕ are d_2, d_1, d_{m+3} and d_{m+2}.

Now, suppose w contains aa. Two values corresponding to these two a's must come from the same A_i, because, otherwise, they would be separated by at least one d. For i odd, the positions of all values of A_i are consecutive. Therefore, since π is simple, the values of two points in A_i cannot be consecutive, as otherwise, those two points would form a block. Hence, as ϕ encodes N_i from bottom to top, it cannot write aa. On the other hand, for i even, the values of A_i are consecutive, so positions cannot be consecutive. Thus, ϕ also cannot write aa while encoding S_i. Applying the same argument, we see that w does not contain bb and cc either.

If w begins with dda, then the first value encoded in N_1 belongs to A_1. Since ϕ encodes N_1 from bottom to top, this value is 3. However, this forms a block $[1, 2]$ with $\pi([1, 2]) = [2, 3]$. We can show w does not end with cdd_ℓ in the same way.

Finally, suppose w contains da. Further, suppose a comes from A_{i+1} for odd i (so A_{i+1} belongs to S_i). By the definition of p_i from the proof of Proposition 4.5, the first value being encoded by ϕ is p_i so long as S_{i+1} is nonempty. Thus, the letter a of da corresponds to p_i. However, p_i cannot be less than d_{i+3} by definition, so w cannot contain da. If i is odd, we use the same argument with the inverse symmetry applied.

Next, we need to show that the image of ψ is in H. Theorem 4.4 states every simple permutation of the form described in Figure 4.9 is in $\mathcal{A}$. Since the graph of π follows one of the structures in 4.9 by construction, all we need to show is for any $w \in L$, $\psi(w)$ is simple. Suppose $\pi = \psi(w)$ is not simple for some $w \in L$. Hence, there exists a proper non-singleton segment I of π which is a block.

Suppose our non-singleton proper block I contains at least two positions s and t corresponding to d_i and d_j ($1 \leq i, j \leq m + 3$). Notice that I cannot contain positions corresponding to both d_1 and d_{m+3} because, in that case, $I = [1, n]$. Suppose I does not contain 1, which is the position corresponding to d_1. Choose the least $s \in I$ such that $\pi(s) = d_i$ for some i with $1 \leq i \leq m + 3$. First, assume i is even. If $i = m + 1$, then I must contain t ($t = n$ in particular) such that $\pi(t) = d_{i+2} = d_{m+3}$ as well, because d_{m+3} is the only one located to the left of d_{m+1} decoded from a letter d in w. However, this implies that I contains a position u such that $\pi(u) = d_{i-1}$, since $d_i < d_{i-1} < d_{i+2}$. Since $u < s$, this contradicts our assumption. On the other hand, if $i \leq m$, we know there exists $t \in I$ such that $\pi(t) = d_{i+3}$, but then this implies that I contains u such that $\pi(u) = d_{i-1}$ again, and we achieve a contradiction. Now, suppose i is odd. Then there exists t in I such that $\pi(t) = d_{i-1}$, and this means that there exists a position u in I corresponding to d_{i-2}. Since $u < s$, we again have a contradiction. We can apply a similar argument for the case where I does not contain the position corresponding to d_{m+3}, so I cannot contain two or more positions corresponding to isolated points.

Now, assume I contains only one position s such that $\pi(s) = d_i$ for some i with $1 \leq i \leq m+3$. However, this is only possible if $i = 1$ or $i = m+3$. Otherwise, if i is odd, then I contains a value r which is either immediately to the left of d_i or immediately to the right of d_i. Since the algorithm DECODE forces $r < d_{i+1} < d_i$, I must contain t such that $\pi(t) = d_{i+1}$. We achieve a similar result for the case of i being even. So suppose I contains $s = 1$ so that $\pi(1) = d_1$. Since I does not contain a position t such that $\pi(t) = d_3$, I must only contain 1 and any positions whose corresponding points are placed in A_1. However, because dda is not allowed in w, this cannot be the case. Similarly, if I contains s such that $\pi(s) = d_{m+3}$, then other positions in I had to be decoded from c, but since cdd_ℓ is not allowed, we again achieve a contradiction.

Finally, suppose I contains no points in D. This implies that $\pi(I)$ is a subset of B_i or $C_i \cup A_{i+1}$ for some i $(1 \leq i \leq m)$. Because aa, bb and cc are not in w, the only case I can form a block is $I = [s, s+1]$ where the points $(s, \pi(s))$ and $(s+1, \pi(s+1))$ correspond to the last c in w_i and the first a in w_{i+1} respectively. Moreover, they are the last and the first letters in w_i and w_{i+1}. However, this would mean da occurs in w, which is not allowed, so we achieve a contradiction.

Consequently, $\psi(w) = \pi$ cannot have a proper non-singleton segment I forming a block, so π is simple. Therefore, the image of ψ is in H. Due to how we construct ϕ and ψ, it is obvious that $\psi(\phi(\pi)) = \pi$ and $\phi(\psi(w)) = w$ for any $\pi \in H$ and $w \in L$, so ϕ is a bijection. ■

Next, we define another language $\overline{L} \subseteq \Sigma^*$. We let $\overline{L}$ be the set of words that can be constructed by removing the first two d's of an arbitrary w in L. Hence, $\overline{L} = \{w \in \Sigma^* : ddw \in L\}$. There is an obvious bijection from L to $\overline{L}$, namely the one erases the first two d's of w in L. The conditions of $\overline{L} \subseteq \Sigma^*$ are:

- w must end with dd_ℓ.
- w must contain no aa, bb or cc.
- d_ℓ is only allowed at the very end.
- w cannot begin with a or end with cdd_ℓ.
- w must contain no da.

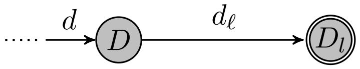
Now, we define an automaton $M = (Q, \Sigma, \delta, A, \{D_\ell\})$ where $Q = \{A, B, C, CD, D, D_\ell\}$, $\Sigma = \{a, b, c, d, d_\ell\}$, and δ is described in Table 4.2. Jail states and transitions to them are omitted. Our last task to complete before the enumeration is to show $\mathcal{L}(M)$, the set of words accepted by M is equal to $\overline{L}$.

	a	b	c	d	d_ℓ
A		B	C	D	
B	A		C	D	
C	A	B		CD	
CD		B	C	D	
D		B	C	D	D_ℓ
D_ℓ					

Table 4.2.: Transitions of M.

Lemma 4.9 *$\mathcal{L}(M) = \overline{L}$.*

Proof. We first show $\mathcal{L}(M) \subseteq \overline{L}$. Let $w \in \mathcal{L}(M)$. We need to show that w does not violate any condition of $\overline{L}$. From the initial state A, transitions that are allowed are b, c and d. Similarly, in order to reach the only accept state D_ℓ, we have to pass through the state D. Since the transition to arrive at D is d, we have



This implies w must begin with b, c or d and end with dd_ℓ. Furthermore, if w ends with cdd_ℓ, then the third transition from the last had to be c. This takes to the state C, and the next transition d takes to the state CD. However, there is no transition to the accept state D_ℓ from CD, so ending with cdd_ℓ is impossible.

Any instance of the letter a in w sends us into state A. Since there is no transition using the letter a from A, w cannot contain aa. By applying similar argument, we can easily show that w does not contain bb, cc or da. Hence, $w \in \overline{L}$.

Next, we show $\overline{L} \subseteq \mathcal{L}(M)$. Suppose w is not in $\mathcal{L}(M)$, that is, w is not accepted by the automaton M. The only ways w cannot be accepted by M are either the run of M on w contains the jail state or the last state is not D_ℓ. The latter implies that the last letter of w is not d_ℓ, so this violates the first condition of $\overline{L}$, and hence, $w \notin \overline{L}$. For the case the run of M on w contains the jail state, we show that every transition to the jail state is due to a failure of w to meet one of the conditions of $\overline{L}$.

Cases (A, a), (B, b) and (C, c): To get to states A, B and C, the previous transitions must be a, b and c respectively. Hence, having these transitions implies that w contains aa, bb and cc respectively, so w violates the second condition.

Cases (CD, a) and (D, a): Since the previous transition is d for both cases, having a next means that w contains da, so w fails to meet the fourth condition.

Cases (A, d_ℓ), (B, d_ℓ), (C, d_ℓ) and (CD, d_ℓ): These transitions imply that w ends with ad_ℓ, bd_ℓ, cd_ℓ and cdd_ℓ respectively. Since w must end with dd_ℓ, but not with cdd_ℓ, none of them are allowed.

Cases (D_ℓ, a), (D_ℓ, b), (D_ℓ, c), (D_ℓ, d) and (D_ℓ, d_ℓ): This causes d_ℓ to appear in the middle of w, so the third condition is not met.

Therefore, any run of M containing the jail state implies w that violates at least one condition of $\overline{L}$. We now have proved $\overline{L} \subseteq \mathcal{L}(M)$, and this completes the proof of $\mathcal{L}(M) = \overline{L}$. ■

Finally, with Lemma 4.8 and 4.9 together, we are ready to prove Theorem 4.7.

Proof of Theorem 4.7. We apply the transfer matrix method to $\mathcal{L}(M)$. With the weight function giving x for all transitions to non-jail states, the adjacency matrix is

$$P = \begin{array}{cc} & \begin{array}{cccccc} A & B & C & CD & D & D_\ell \end{array} \\ \begin{array}{c} A \\ B \\ C \\ CD \\ D \\ D_\ell \end{array} & \left[\begin{array}{cccccc} 0 & x & x & 0 & x & 0 \\ x & 0 & x & 0 & x & 0 \\ x & x & 0 & x & 0 & 0 \\ 0 & x & x & 0 & x & 0 \\ 0 & x & x & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array}$$

By observing the (A, D_ℓ)-entry of $(I - P)^{-1}$, we obtain[1]

$$\frac{x^2}{(1-3x)(1+x)}.$$

This is the generating function for n-letter words in $\mathcal{L}(M)$. By Lemma 4.9, this is also the generating function for n-letter words in $\overline{L}$. Since the number of $(n-2)$-letter words in L is

[1] The computation of $(I - P)^{-1}$ is done by using Mathematica.

equal to the number of n-letter words in $\overline{L}$, the generating function for n-letter words in L is

$$x^2 \cdot \frac{x^2}{(1-3x)(1+x)} = \frac{x^4}{(1-3x)(1+x)},$$

which is, by Lemma 4.8, also the generating function for permutations of length n in the set H. As it was explained previously, doubling the generating function gives us

$$f_{\mathrm{Si}(\mathcal{A})\backslash\mathcal{S}_2} = \frac{2x^4}{(1-3x)(1+x)}.$$

■

4.4.2. Enumeration of the whole class of $\mathcal{A}$

We are close to completing the enumeration of the whole class. From here, we show that every simple permutation in $\mathcal{A}$ satisfies the hypothesis of Proposition 2.5, then apply both Proposition 2.5 and 2.6. Let $K = \mathrm{Si}(\mathcal{A}) - H - \{12, 21\}$. If we show that for all $\pi \in \mathrm{Si}(\mathcal{A})$, π satisfies the hypothesis of Proposition 2.5, we have the equation of generating functions as the following.

$$f_{\mathcal{A}} = 1 + x + \sum_{\pi \in \mathrm{Si}(\mathcal{A})} f_{\mathrm{ifl}(\pi)} = 1 + x + f_{\mathrm{ifl}(12)} + f_{\mathrm{ifl}(21)} + \sum_{\pi \in H} f_{\mathrm{ifl}(\pi)} + \sum_{\pi \in K} f_{\mathrm{ifl}(\pi)}.$$

First, using what we have established, we want to acquire the generating function for length n permutations in $\mathcal{A}$ which can be obtained by inflation of simple permutations in H. By the inverse argument, we get the exact same result for inflation of simple permutations in K, so we will simply multiply 2 as before.

Let $\pi \in H$. As it is explained in the second part of the proof of Theorem 4.4, any value of π belongs to either the set $\mathrm{LRmax}(\pi)$, the set $\mathrm{RLmin}(\pi)$ or the set $B := B_{\mathrm{odd}} \cup B_{\mathrm{even}}$. In order to apply Proposition 2.5, we prove the following lemma.

Lemma 4.10 *The condition $\alpha = \pi[\sigma_1, \ldots, \sigma_n] \in \mathcal{A}$ is equivalent to the condition stating that for all i with $1 \le i \le n$,*

- *if $\pi(i) \in \mathrm{LRmax}(\pi)$, then $\sigma_i \in Av(312)$,*
- *if $\pi(i) \in \mathrm{RLmin}(\pi)$, then $\sigma_i \in Av(231)$, and*
- *if $\pi(i) \in B$, then $\sigma_i \in Av(12)$.*

Proof. Suppose the latter condition is false. That is, at least one of the above three conditions is not met. Assume it is the first one. Then there exists i $(1 \leq i \leq n)$ such that $\pi(i) \in \mathrm{LRmax}(\pi)$ and σ_i contains 312. Now, whichever point $(i, \pi(i))$ is, there is at least one point $(j, \pi(j))$ where $i < j$ and $\pi(i) > \pi(j)$. Namely, this point has value d_k if $\sigma_i \in C_{k-2} \cup A_{k-1} \cup \{d_{k-1}, d_{k+1}\}$. Any point within the block corresponding to σ_j together with the subsequence 312 of α within the block corresponding to σ_i, α contains 4231, so $\alpha \notin \mathcal{A}$. If the second condition is not met, we can apply the reverse complement argument of the previous one to show α contains 4231. If the third condition is false, say $12 \preceq \sigma_i$ for some i and the value $\pi(i)$ is in B_k for some k, then d_{k+1} and d_{k+2} along with σ_i cause α to contain 4231, so again, $\alpha \notin \mathcal{A}$. By the contrapositive argument, $\alpha = \pi[\sigma_1, \ldots, \sigma_n] \in \mathcal{A}$ implies the latter condition.

Next, assume a permutation $\alpha = \pi[\sigma_1, \ldots, \sigma_n]$ where $\pi \in H$ is not in $\mathcal{A}$. Thus, α contains at least one permutation β in the basis. Since π avoids every permutation in the basis, it means there exist $\sigma_{i_1}, \ldots, \sigma_{i_k}$ (for all $j \in \{1, \ldots, k\}$, $1 \leq i_j \leq n$) such that α contains β within the union of subintervals corresponding to $\sigma_{i_1}, \ldots, \sigma_{i_k}$.

Since every permutation in $\{35142, 42513, 351624\}$ is simple, so we only need to consider the case $\beta = 4231$. If there exist $\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}, \sigma_{i_4}$ such that each point of β is contained in intervals corresponding to $\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}, \sigma_{i_4}$ respectively, then again, $\beta \preceq \pi$ which cannot be true. Also, with the same reason as the previous one, if there exists a single σ_i such that $\beta \preceq \sigma_i$, then the latter conditions are false. Hence, there exist two or three subintervals of α such that the containment of β is involved in. So suppose there exist σ_{i_1} and σ_{i_2} which involve β together in α. Then it has to be either $312 \preceq \sigma_{i_1}$ or $231 \preceq \sigma_{i_2}$. Assume $312 \preceq \sigma_{i_1}$ is true. Then notice that $\pi(i_1)$ cannot be in $\mathrm{RLmin}(\pi)$, because there is no point which can play the role of 1. This implies $\pi(i_1) \in \mathrm{LRmax}(\pi)$ and $312 \preceq \sigma_{i_1}$ or $\pi(i_1) \in B$ and $12 \preceq 312 \preceq \sigma_{i_1}$, so the second condition is not met. With the reverse complement argument, we conclude the same for the case of $231 \in \sigma_{i_2}$.

Finally, suppose the containment of β is shared by three subintervals of α, say the ones of σ_{i_1}, σ_{i_2} and σ_{i_3}. Then this must imply $12 \in \sigma_{i_2}$. $\pi(i_2)$ can be in neither $\mathrm{LRmax}(\pi)$ nor $\mathrm{RLmin}(\pi)$. Hence, $\pi(i_2) \in B$, so again, the latter condition is false. With every observation we made and contrapositive argument, the second condition implies $\alpha = \pi[\sigma_1, \ldots, \sigma_n] \in \mathcal{A}$, so those two statements are equivalent. ■

Consequently, every simple permutation in H satisfies the hypothesis of Proposition 2.5. Both Av(312) and Av(231) are counted by the Catalan numbers, as explained in Theorem 2.3. Hence, excluding the empty permutation, we have

$$\bar{f}_{\mathrm{Av}(312)} = \bar{f}_{\mathrm{Av}(231)} = \bar{f}_{\mathrm{cat}} = \frac{1-\sqrt{1-4x}}{2x} - 1 = \frac{1-2x-\sqrt{1-4x}}{2x}.$$

For $\mathrm{Av}(12) \setminus \{\varepsilon\} = \{1, 21, 321, 4321, \ldots\}$, the generating function is the geometric power series minus 1, so

$$\bar{f}_{\mathrm{Av}(12)} = \bar{f}_{\mathrm{geom}} = \frac{1}{1-x} - 1 = \frac{x}{1-x}.$$

Now, we go back to $\mathcal{L}(M)$. Using the transfer matrix method again, we want to find $f_{\mathrm{ifl}(H)} = \sum_{\pi \in H} f_{\mathrm{ifl}(\pi)}$ where $\mathrm{ifl}(H) = \bigcup_{\pi \in H} \mathrm{ifl}(\pi)$. Because each transition embeds the information that which set of $A = A_{\mathrm{odd}} \cup A_{\mathrm{even}}$, B, $C = C_{\mathrm{odd}} \cup C_{\mathrm{even}}$, or D the point corresponding to the letter belongs to, we can effectively define the weight function so that we can obtain the desired generating function $f_{\mathrm{ifl}(H)}$. Define $w(t) = \bar{f}_{\mathrm{cat}}$ if $t = a, c, d$ or d_ℓ, and $w(t) = \bar{f}_{\mathrm{geom}}$ if $t = b$. The adjacency matrix $\hat{P}$ with this weight function w is

$$\hat{P} = \begin{array}{c} \\ A \\ B \\ C \\ CD \\ D \\ D_\ell \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & CD & D & D_\ell \end{array} \\ \begin{bmatrix} 0 & \bar{f}_{\mathrm{geom}} & \bar{f}_{\mathrm{cat}} & 0 & \bar{f}_{\mathrm{cat}} & 0 \\ \bar{f}_{\mathrm{cat}} & 0 & \bar{f}_{\mathrm{cat}} & 0 & \bar{f}_{\mathrm{cat}} & 0 \\ \bar{f}_{\mathrm{cat}} & \bar{f}_{\mathrm{geom}} & 0 & \bar{f}_{\mathrm{cat}} & 0 & 0 \\ 0 & \bar{f}_{\mathrm{geom}} & \bar{f}_{\mathrm{cat}} & 0 & \bar{f}_{\mathrm{cat}} & 0 \\ 0 & \bar{f}_{\mathrm{geom}} & \bar{f}_{\mathrm{cat}} & 0 & \bar{f}_{\mathrm{cat}} & \bar{f}_{\mathrm{cat}} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

By computing the (A, D_ℓ)-entry of $(I - \hat{P})^{-1}$ and multiplying by $(\bar{f}_{\mathrm{cat}})^2$ to include the initial two d's, we obtain the generating function for $\mathrm{ifl}(H)$ by Proposition 2.5 and 2.6. Since $\mathrm{ifl}(K)$ can be obtained by inverting every permutation in $\mathrm{ifl}(H)$, we can multiply 2 to the generating function $f_{\mathrm{ifl}(H)}$ to include this result. By doing so, we arrive at

$$f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}) \setminus \mathcal{S}_2)} = \frac{\left(-2x - \sqrt{1-4x} + 1\right)^4}{8x^2\left(x^2 - \sqrt{1-4x}x + 3x + \sqrt{1-4x} - 1\right)}.$$

We now move onto the case where the skeleton is $\pi = 21$. Recall that, to ensure σ_1 and σ_2 are uniquely determined by α when $\alpha = 21[\sigma_1, \sigma_2]$, we must require that σ_1 to be skew-indecomposable. We claim that $\alpha = 21[\sigma_1, \sigma_2] \in \mathcal{A}$ with skew-indecomposable σ_1 is equivalent to the condition that $\sigma_1 \in \mathrm{Av}(312)$ and σ_1 is skew-indecomposable, and $\sigma_2 \in \mathrm{Av}(231)$. The condition of σ_1 being skew-indecomposable cannot be dropped to enforce the uniqueness of inflation. It is clear that either of $312 \preceq \sigma_1$ or $231 \preceq \sigma_2$ implies $\alpha \notin \mathcal{A}$. So assume that $\alpha \notin \mathcal{A}$. Then

$\beta \in \alpha$ for some $\beta \in \{4231, 35142, 42513, 351624\}$. As before, since 35142, 42513 and 351624 are simple, if $\beta \in \{35142, 42513, 351624\}$, then $\beta \preceq \sigma_1$ or $\beta \preceq \sigma_2$. In either case, $312 \preceq \sigma_1$ or $231 \preceq \sigma_2$, so the second condition is not met. If $\beta = 4231$, then it is immediate that $312 \preceq \sigma_1$ or $231 \preceq \sigma_2$, so the condition $\alpha = 21[\sigma_1, \sigma_2] \in \mathcal{A}$ with skew-indecomposable σ_1 and the condition $\sigma_1 \in \text{Av}(312)$ where σ_1 is skew-indecomposable are equivalent.

By Proposition 2.5, $\bar{f}_{\text{ifl}(21)} = \bar{f}^{\ominus}_{\text{Av}(312)} \cdot \bar{f}_{\text{Av}(231)}$. We need to derive $\bar{f}^{\ominus}_{\text{Av}(312)}$, the generating function for skew-indecomposable permutation in Av(312). Notice that every skew-decomposable permutation π in Av(312) can be written as $\sigma \ominus 1$ where σ is a nonempty permutation avoiding 312. To show this, suppose it is not true. Then $\pi = \sigma \ominus \tau$ where $|\tau| \geq 2$. If $12 \preceq \tau$, then π contains 312. So assume τ avoids 12. Then τ must be strictly decreasing, but then, it is possible to write π as $\sigma' \ominus 1$ for some σ', so if π is skew-decomposable in Av(312), it can be always written as $\sigma \ominus 1$. To find $\bar{f}^{\ominus}_{\text{Av}(312)}$, we need to exclude skew-decomposable permutations. Since the generating function for skew-decomposable permutations is $x\bar{f}_{\text{Av}(312)}$, we obtain $\bar{f}^{\ominus}_{\text{Av}(312)} = \bar{f}_{\text{Av}(312)} - x\bar{f}_{\text{Av}(312)}$. Consequently,

$$f_{\text{ifl}(21)} = f^{\ominus}_{\text{Av}(312)} \cdot f_{\text{Av}(231)} = (\bar{f}_{\text{cat}} - x\bar{f}_{\text{cat}}) \cdot \bar{f}_{\text{cat}} = \frac{\left(1 - 2x - \sqrt{1-4x}\right)^2 (1-x)}{4x^2}.$$

Lastly, for the case $\pi = 12$, it is possible to inflate both 1 and 2 by any permutations σ_1 and σ_2 of $\mathcal{A}$ itself, provided that σ_1 is a sum-indecomposable permutation in $\mathcal{A}$. Three cases for σ_1 being sum-indecomposable are $\sigma_1 = 1$, σ_1 is skew-decomposable, or σ_1 is an inflated permutation of π in $H \cup K$ (possibly with $1, \ldots, 1$). Generating functions for each case are x, $f_{\text{ifl}(21)}$ and $f_{\text{ifl}(\text{Si}(\mathcal{A}) \setminus \mathcal{S}_2)}$ respectively. Thus,

$$f_{\text{ifl}(12)} = (x + f_{\text{ifl}(21)} + f_{\text{ifl}(\text{Si}(\mathcal{A}) \setminus \mathcal{S}_2)}) \cdot \bar{f}_{\mathcal{A}}.$$

Consequently, the generating function for $\mathcal{A}$ satisfies the functional equation

$$\begin{aligned} f_{\mathcal{A}} &= 1 + x + f_{\text{ifl}(12)} + f_{\text{ifl}(21)} + f_{\text{ifl}(\text{Si}(\mathcal{A}) \setminus \mathcal{S}_2)} \\ &= 1 + x + (x + f_{\text{ifl}(21)} + f_{\text{ifl}(\text{Si}(\mathcal{A}) \setminus \mathcal{S}_2)}) \cdot \bar{f}_{\mathcal{A}} + f_{\text{ifl}(21)} + f_{\text{ifl}(\text{Si}(\mathcal{A}) \setminus \mathcal{S}_2)}. \end{aligned}$$

With $\bar{f}_{\mathcal{A}} = f_{\mathcal{A}} - 1$, we solve for $f_{\mathcal{A}}$. Then, we obtain

$$\begin{aligned} f_{\mathcal{A}} &= \frac{1}{1 - x - f_{\text{ifl}(21)} - f_{\text{ifl}(\text{Si}(\mathcal{A}) \setminus \mathcal{S}_2)}} = \frac{2(1 - x^2 - 3x - (1-x)\sqrt{1-4x})}{1 - 3x - \sqrt{1-4x}\,(2x^2 - x + 1)} \\ &= 1 + x + 2x^2 + 6x^3 + 23x^4 + 101x^5 + 477x^6 + 2343x^7 + 11762x^8 + \cdots \end{aligned}$$

Chapter 5.

Structure of general simple permutations in $\mathcal{A}'$

Recall $\mathcal{A}' = \mathrm{Av}(52341, 53241, 52431, 35142, 42513, 351624)$. In this chapter, we establish the theorem analogous to Theorem 4.4. Note that the class $\mathcal{A}'$ is also preserved by the inverse operation and the reverse complement operation.

5.1. Extreme patterns 2413, 3142 and 3412

5.1.1. Structural propositions

We first prove the following.

Proposition 5.1 *No simple permutation in $\mathcal{A}'$ has extreme pattern 3412.*

Proof. Suppose the statement is false. Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 3412. We start with the graph of extreme pattern of π, which is shown in Figure 5.1(a). As before, dark grey indicates a point in the region would create a forbidden pattern, and light grey indicates we have made specific assumptions that there does not exist a point in the region.

There must exist a point in B_{21} or B_{12} to avoid $[\pi^{-1}(c), \pi^{-1}(d)]$ being a block. Just like the proof for Proposition 4.3, one of these two cases can be obtained by the inverse operation followed by the reverse complement operation, so we only give a proof for the case of B_{21}. Letting x be the least value of all possible points in B_{21}, we obtain the graph shown in Figure 5.1(b).

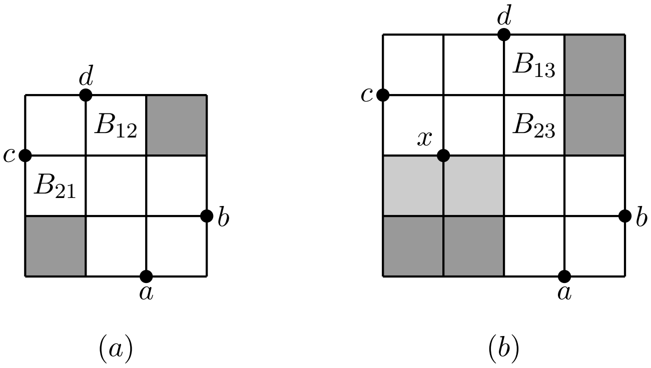


Figure 5.1.: Partial graphs of π of extreme pattern 3412.

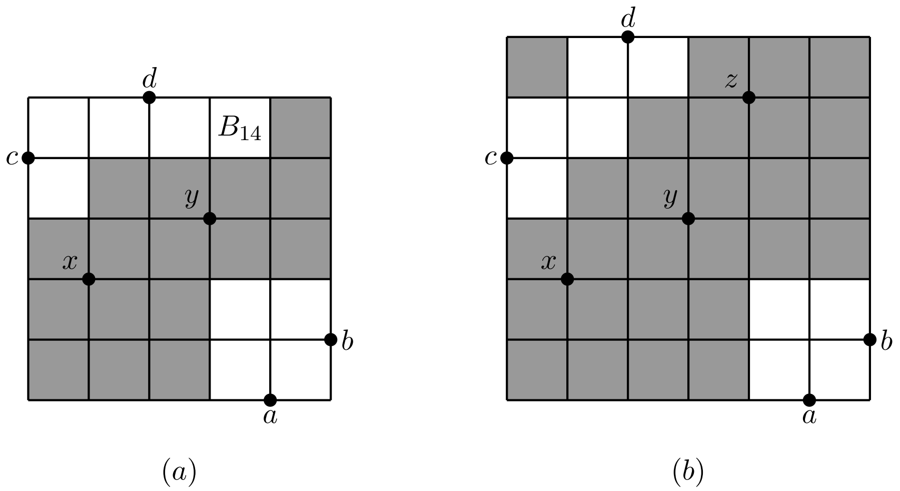


Figure 5.2.: Partial graphs of π with the assumption of having a value in B_{23} in Figure 5.1.

In order to split the block $[\pi^{-1}(c), \pi^{-1}(d)]$, we need to have a point in either B_{13} or B_{23}. Suppose there is a point with some value y in B_{23}. Then we obtain the graph in Figure 5.2(a), forcing it to have a point in B_{14} due to the block $[\pi^{-1}(c), \pi^{-1}(y)]$. However, having a point with a value z in B_{14} results in the graph in Figure 5.2(b), which contains the unsplittable block $[\pi^{-1}(c), \pi^{-1}(z)]$.

Now, assume there exists a point in B_{13} of Figure 5.1(b), and none in B_{23}. Calling the value of left-most point y, we have the graph shown in Figure 5.3(a). The regions indicated by B_{33} and B_{34} are shaded in light grey since having a point in either of these regions is equivalent to having a point in B_{23} of Figure 5.1(b). We must have a point in B_{43} of Figure 5.3(a) to split

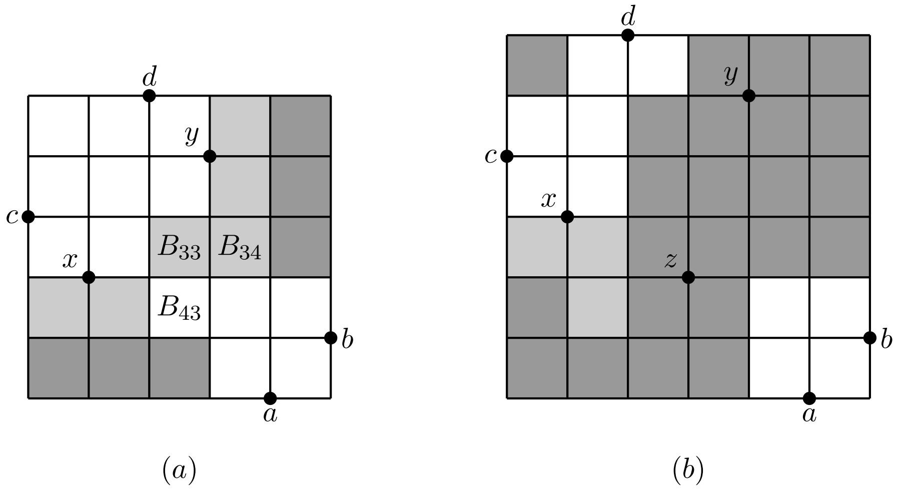


Figure 5.3.: Partial graphs of π with the assumption of having a value in B_{13} in Figure 5.1.

the block $[\pi^{-1}(c), \pi^{-1}(y)]$, but this results in the block $[\pi^{-1}(c), \pi^{-1}(y)]$ shown in Figure 5.3(b) which we cannot split any longer.

Consequently, it is impossible to have a simple permutation in $\mathcal{A}'$ of extreme pattern 3412. ■

Next, we describe the structure of simple permutations in $\mathcal{A}'$ of extreme pattern 2413. Let b, d, a and c be the first, the greatest, the least and the last values of π respectively. We denote by A, B and C each segment $[\pi^{-1}(b), \pi^{-1}(d))$, $[\pi^{-1}(d), \pi^{-1}(a)]$ and $(\pi^{-1}(a), \pi^{-1}(c)]$ respectively. First, we prove the following lemma for values corresponding to positions in B.

Lemma 5.2 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then values corresponding to positions in B have a pattern of the form*

$$\bigominus_{i=1}^{k} \sigma_i, \tag{5.1}$$

where each σ_i is either 1 or 12.

For Equation 5.1, Note $k = 0$ is possible, in which case B consists of only d and a.

The proof is an immediate consequence of 52341, 53241 and 52431 avoidance conditions with the observation we made with Lemma 3.1. Therefore, values corresponding to positions in B form the structure described in Figure 5.4. Except for d and a, every point corresponding to 1 and pair of points corresponding to 12 in Equation 5.1 can be empty.

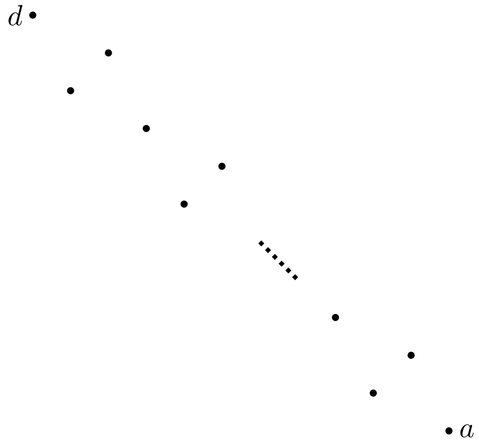


Figure 5.4.: Structure of values corresponding positions in B.

Next, we establish the lemma describing the structure of points whose positions are in the segment A. In order to do this, we need to define a special sum of two permutations. Given $\sigma \in \mathcal{S}_m$ and $\tau \in \mathcal{S}_n$, the *value-interchange sum of* σ *and* τ *with* 1 *shift* is the permutation defined by

$$\sigma \oplus_1 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m)\tau'(1)\tau'(2)\cdots\tau'(n)$$

where

$$\sigma'(i) = \begin{cases} \sigma(i) & \text{if } \sigma(i) \leq m-1 \\ m+1 & \text{if } \sigma(i) = m \end{cases} \qquad \text{for each } i \ (1 \leq i \leq m)$$

and

$$\tau'(j) = \begin{cases} m & \text{if } \tau(j) = 1 \\ \tau(j) + (m+1) & \text{if } \tau(j) \geq 2 \end{cases} \qquad \text{for each } j \ (1 \leq j \leq n)$$

In other words, $\sigma \oplus_1 \tau$ is just like $\sigma \oplus \tau$ except we interchange the positions of the values corresponding the greatest value of σ and the least value of τ. So, algebraically, $\sigma \oplus_1 \tau = (m\ m+1)(\sigma \oplus \tau)$, the product of $(\sigma \oplus \tau)$ with the adjacent transposition $(m\ m+1)$. For example, with $\sigma = 1342$ and $\tau = 312$, $\sigma \oplus_1 \tau = 1352746$ whereas $\sigma \oplus \tau = 1342756$. Note that $\oplus_1$ is not associative if one of the summands is length 1.

Let us now describe a 231-value chain using the sum we discussed above. A *231-value chain* is a sequence of values of the form

$$\bigoplus_{i=1}^{k}{}_1\, \sigma_i$$

where each σ_i is in $\{21, 231\}$. The structure of a 231-value chain is shown in Figure 5.5. The underlined points are optional; if they do not exist, we have added 21 rather than 231. All other points in the figure are required. We will state the lemma describing the structure of points whose positions are in A, but in order to prove it, we first establish a few lemmas.

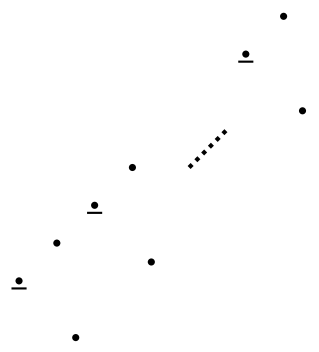


Figure 5.5.: Structure of a 231-value chain.

Lemma 5.3 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then values corresponding to positions in A avoid 321 pattern and 23451 pattern.*

Proof. We first prove that values of π corresponding to positions in A avoid 321 pattern. Suppose to the contrary that values corresponding to positions in A contain 321. Furthermore, assume b corresponds to the 3 of 321. Let z be the least possible value that can play the role of 1 for b and y_0 be the least possible value that can play the role of 2 for given b and z. Hence, the graph of π is as shown in Figure 5.6(a).

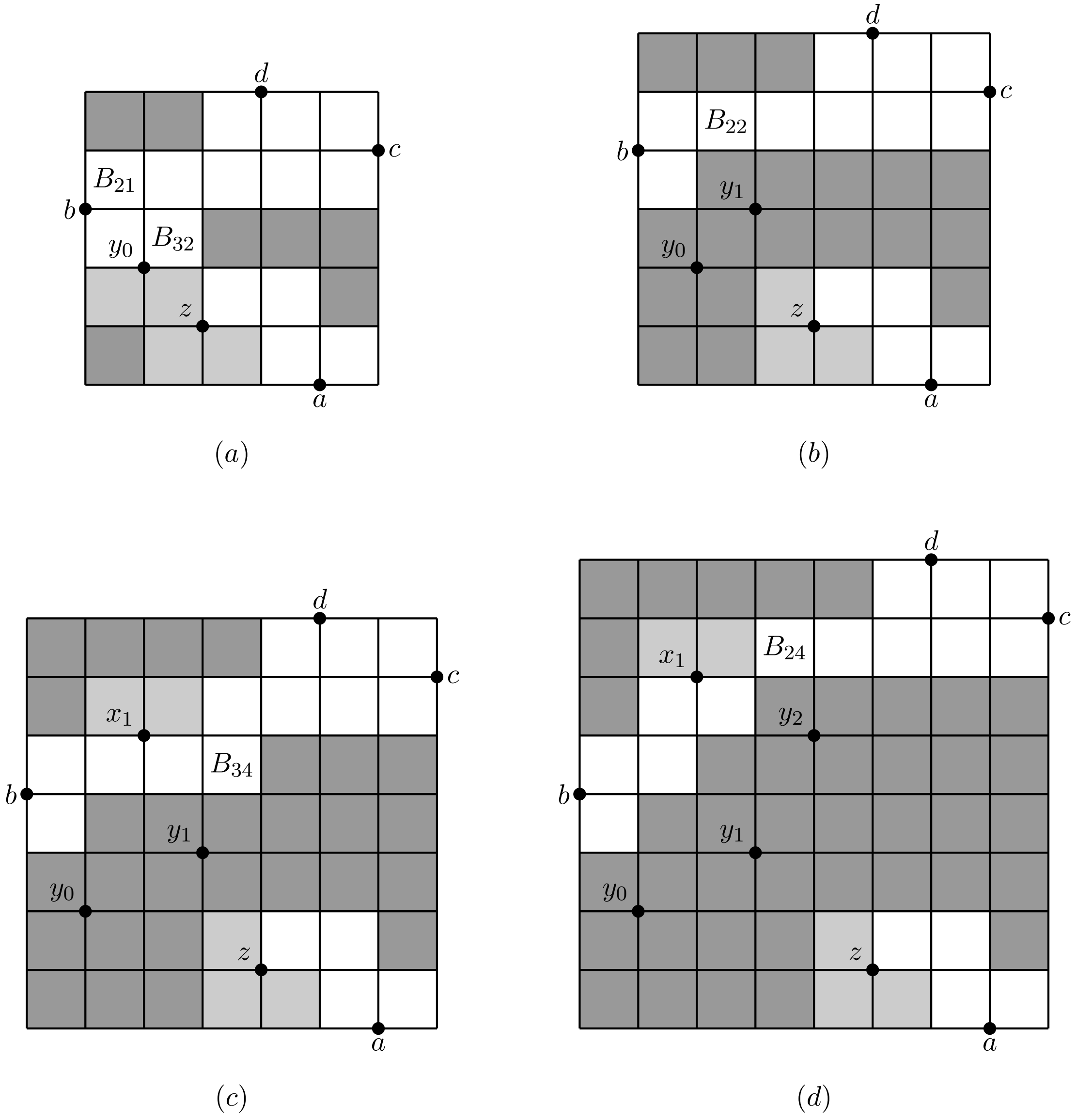


Figure 5.6.: Partial graphs of π with the assumption of b corresponding to the 3 in 321 and having a value in B_{32}.

Since π is simple, we cannot have a block in π. Hence, there exists a point in either B_{21} or B_{32} of Figure 5.6(a) to prevent $[\pi^{-1}(b), \pi^{-1}(y_0)]$ from being a block. Suppose there is a point in the region B_{32} and call its value y_1. Now, we have a block $[\pi^{-1}(y_0), \pi^{-1}(y_1)]$ as shown in Figure 5.6(b), so there must exist a point in B_{22} of 5.6(b). Choose the point of greatest value in B_{22} of 5.6(b) and let x_1 be the value of this point. Then we have the graph shown in Figure 5.6(c). The block defined by the positions of b and y_1 still needs to be split. The only way to do so is by assuming the existence of a point in the region B_{34} of Figure 5.6(c), say y_2. Having y_2 results in the graph shown in Figure 5.6(d).

From here on, attempting to split the block $[\pi^{-1}(b), \pi^{-1}(y_i)]$ can only be done by assuming the existence of points with values x_i and y_{i+1} alternatively. In particular, x_i is the greatest possible value such that $\pi^{-1}(y_{i-1}) < \pi^{-1}(x_i) < \pi^{-1}(y_i)$ and $x_{i-1} < x_i < c$, and y_i is the value such that $\pi^{-1}(y_i) < \pi^{-1}(y_{i+1}) < \pi^{-1}(z)$ and $x_{i-1} < y_{i+1} < x_i$. However, splitting $[\pi^{-1}(b), \pi^{-1}(y_i)]$ by assuming there exists x_i still results in $[\pi^{-1}(b), \pi^{-1}(y_i)]$ being a block. Similarly, splitting it by assuming there exists y_{i+1} constructs another block $[\pi^{-1}(b), \pi^{-1}(y_{i+1})]$. Hence, attempting to split blocks constructs an infinite strict chain of permutations contained in π, but this is impossible since the length of π is finite.

Referring back to Figure 5.6(a), assume there is a point in the region B_{21} this time. Let x_0 be the greatest value possible of all points in B_{21}. We then have a graph shown in Figure 5.7(a). In order to avoid the segment $[\pi^{-1}(b), \pi^{-1}(y_0)]$ being a block, we must have a point in B_{33} shown in Figure 5.7(a). Note that B_{34} is shaded in light grey because having a point here

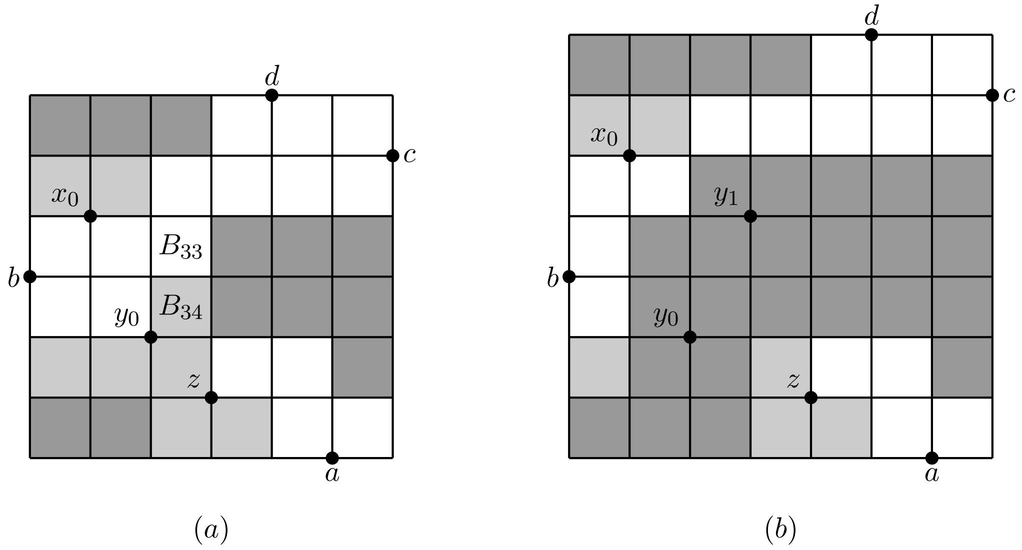


Figure 5.7.: Partial graphs of π with the assumption of b corresponding to the 3 in 321 and having a value in B_{21}.

is equivalent to having a point in B_{32} of Figure 5.6(a), which we have discussed previously. If we have a point y_1 in B_{33}, we have a block $[\pi^{-1}(b), \pi^{-1}(y_1)]$ shown in 5.7(b). The rest of the argument is identical to the one for the previous case. Again, trying to split blocks constructs an infinite strict chain of permutations contained in π, so we achieve a contradiction. Therefore, b cannot correspond to the 3 in 321.

Next, let $\hat{b}$ be the left-most value which can play the role of 3 in 321. Furthermore, let z be the least possible value that can play the role of 1 for $\hat{b}$ and y_0 be the least possible value that can play the role of 2 for given $\hat{b}$ and z. Depending on weather $\hat{b} < c$ or $c < \hat{b}$, we have one of the graphs shown in Figure 5.8. For both cases, achieving a contradiction from here is identical to the case of b playing the role of 3 in 321.

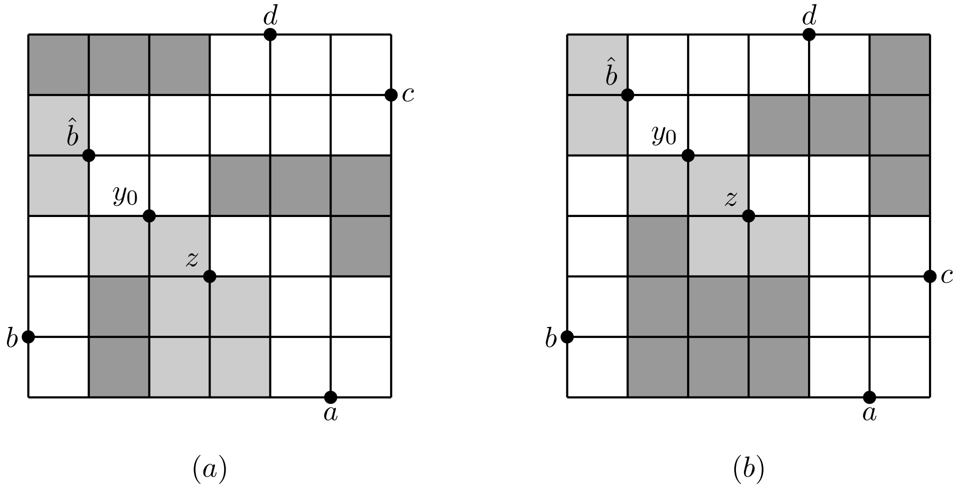


Figure 5.8.: Partial graphs of π with $\hat{b}$ corresponding to the 3 in 321.

Consequently, values corresponding to positions in A avoid 321.

Now, we show that values corresponding to positions in A avoid 23451 pattern as well. Suppose to the contrary that values whose positions are in A contain 23451. Since these values must avoid 321, we obtain Figure 5.9. Since $[\pi^{-1}(x_2), \pi^{-1}(x_3)]$ is a block, we must assume an existence of a point in either B_{73}, B_{83}, B_{46} or B_{47}. However, if there is a point in either B_{73} or B_{83}, then $[\pi^{-1}(x_1), \pi^{-1}(x_2)]$ becomes an unsplittable block. Similarly, assuming a point in either B_{46} or B_{47} implies $[\pi^{-1}(x_3), \pi^{-1}(x_4)]$ is an unsplittable block, so values whose positions are in A cannot form 23451 pattern. We obtain the exact same result even if $a = \pi(1)$ is a part of 23451 pattern.

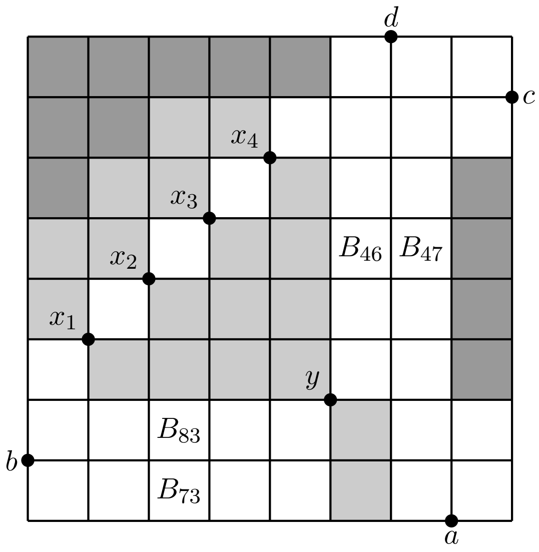


Figure 5.9.: Partial graphs of π with the assumption of values corresponding to positions in A contain 23451.

Hence, we have shown that values corresponding to positions in A of π cannot contain both 321 pattern and 23451 pattern, so we are done. ■

Before we proceed to the next lemma, let us introduce some new terminologies. First, we call a pair of values forming a 21 pattern with consecutive positions a *descent*. Next, for a permutation π, the expression

$$\pi = \bigoplus_{i=1}^{k} \sigma_i$$

where k is the greatest value possible is called the *sum-decomposition of* π. Note that if $k \geq 2$, then π is sum-decomposable. Furthermore, let π' be the flattening of values of π whose positions are in a segment $[a, b]$. Then the above expression for π' is called the *sum-decomposition within* $[a, b]$. In both cases, we call the consecutive positions of values corresponding to each σ_i a *sum block*. Notice that a sum block within the whole segment $[1, n]$ is equivalent to a block, but this is in general not true for a sum block within a proper segment $[a, b]$.

We now state and prove the following lemmas.

Lemma 5.4 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then a sum block within the segment A with a descent must end on a descent.*

Proof. Consider a sum block I within the segment A with at least one descent. Let x and y $(x > y)$ be the last descent in I. Suppose to the contrary that there is a value located to the

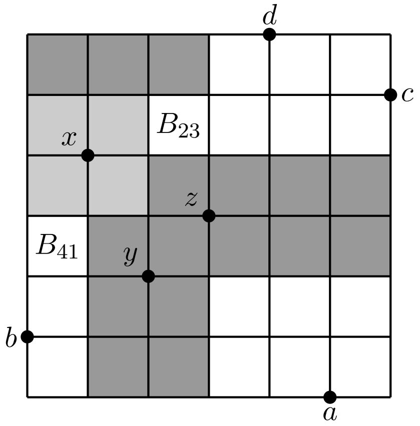


Figure 5.10.: Partial graph of π for Lemma 5.4.

right of y within I. For any such value z, $z < x$ because, otherwise, $\pi^{-1}(z) \notin I$. Also, $y < z$ because 321 pattern is forbidden. Thus, we obtain the graph shown in Figure 5.10. Attempting to split the block $[\pi^{-1}(y), \pi^{-1}(z)]$ by having a value in B_{41} will cause a construction of an infinite strict chain as observed in the proof of Lemma 5.3. Therefore, the only way to split the block $[\pi^{-1}(y), \pi^{-1}(z)]$ is by having a point in B_{23}. However, this implies z is a part of a descent located to the right of x and y, which is a contradiction. Thus, I must end on a descent. ■

Lemma 5.5 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Let $[s,t]$ be a sum block within the segment A. If the values of positions in the segment $[s,t']$ ($t' \leq t$) ends with a descent, then these values form a 231-value chain.*

Proof. We prove the statement by induction on the number of descents in $[s,t']$. First, assume that values of positions in the segment $[s,t']$ has only one descent x and y ($x > y$), *i.e.* $\pi(t') = y$. By Lemma 5.3, values with consecutive positions from s up to $\pi^{-1}(y)$ must form a 21, 231 or 2341 pattern. Suppose to the contrary that we have a 2341 pattern. The graph shown in Figure 5.11 shows this case. As observed, the block formed by the four values for the pattern can only be split by having a value to the right. However, attempting to split this way will immediately cause another block to appear, which is unsplittable. Hence, only a 21 or 231 pattern is permitted in the case which we have a single descent. In either case, we have a 231-value chain.

This time, suppose $[s,t']$ has k descents. Let x and y be the ending descent. Also, let x' be the smallest value with $x' > y$ located to the left of x. At this point, we have a graph as shown in Figure 5.12. Notice that it is possible to have one value in B_{32}, but no more due to 321 and 23451 avoiding conditions. If we don't have a value in B_{52} or B_{62}, then we only have one descent in $[s,t']$, and this is our base case. So suppose we have a value y' in B_{52} or B_{62}. Then by inductive hypothesis, values of positions in the segment $[s,\pi^{-1}(y')]$ form a 231-value

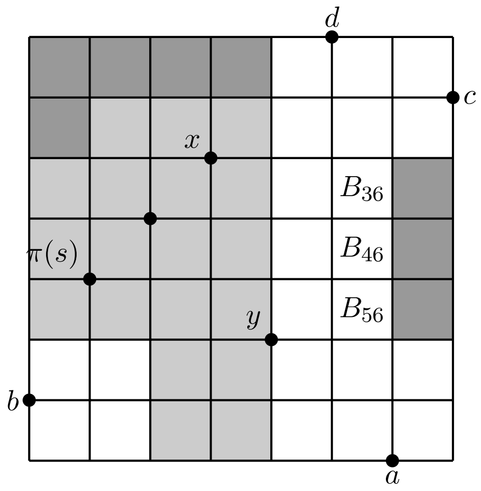


Figure 5.11.: Partial graphs of π with a 2341 pattern and one descent.

chain. With x, y and potentially another value in B_{32} of Figure 5.12, we have a 231-value chain with values whose positions are in $[s, t']$, so we are done.

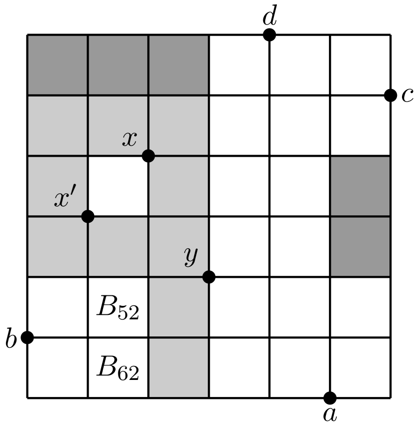


Figure 5.12.: Partial graphs of π for the inductive case.

■

By applying Lemma 5.4 and 5.5, we have the following lemma, which is the description of the structure of points whose positions are in A.

Lemma 5.6 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then values corre-*

sponding to positions in A can be expressed as

$$\bigoplus_{i=1}^{k} \sigma_i, \tag{5.2}$$

where each σ_i is 1 or a 231-value chain.

Proof. By Lemma 5.4, each sum block in A is either a singleton segment or a segment ending on a descent. Also, by Lemma 5.5, every segment ending on a descent is a 231-value chain, so we have the desired result. ■

Next, define a *312-value chain* to be the reverse complement of a 231-value chain, *i.e.* it is a sequence of values of the form

$$\bigoplus_{i=1}^{k}{}_{1}\, \sigma_i$$

where each σ_i is in $\{21, 312\}$. Hence, the structure of a 312-value chain is as shown in Figure 5.13. As before, underlined points can be empty, but no others can be so long as the chain continues.

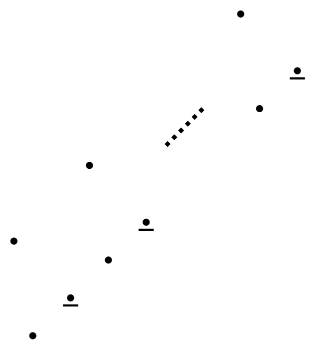


Figure 5.13.: Structure of a 312-value chain.

With the reverse complement property, we obtain the following lemma.

Lemma 5.7 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then values corresponding to positions in C can be expressed as*

$$\bigoplus_{i=1}^{k} \sigma_i, \tag{5.3}$$

where each σ_i is 1 or a 312-value chain.

Finally, we summarize the structure of a simple permutation π in $\mathcal{A}'$ of extreme pattern 2413. By Lemma 5.2, 5.6 and 5.7, we have the following proposition.

Proposition 5.8 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Let b, d, a and c be the first, the greatest, the least and the last values of π respectively. Then values corresponding to positions in $[\pi^{-1}(b), \pi^{-1}(d))$ can be expressed as Equation 5.2, values corresponding to positions in $[\pi^{-1}(d), \pi^{-1}(a)]$ can be expressed as Equation 5.1, and values corresponding to positions in $(\pi^{-1}(a), \pi^{-1}(c)]$ can be expressed as Equation 5.3.*

We can state the structure of a simple permutation π in A' of extreme pattern 3142 with the inverse property, but in order to do so, we need to define the position-interchange sum, which is the inverted value-interchange sum. Given $\sigma \in \mathcal{S}_m$ and $\tau \in \mathcal{S}_n$, The *position-interchange sum of σ and π with 1 shift* is the permutation defined by

$$\sigma \oplus^1 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-1)\tau'(1)\sigma(m)\tau'(2)\cdots\tau'(n)$$

where $\tau'(i) = \tau(i) + m$ for each i with $1 \leq i \leq n$. We then define the *231-position chain* and the *312-position chain* as

$$\bigoplus_{i=1}^{k}{}^{1}\, \sigma_i \qquad \text{where each } \sigma_i \text{ is in } \{21{,}231\}$$

and

$$\bigoplus_{i=1}^{k}{}^{1}\, \sigma_i \qquad \text{where each } \sigma_i \text{ is in } \{21{,}312\}$$

respectively. By using these, the following proposition is the structure of a simple permutation π in $\mathcal{A}'$ of extreme pattern 3142.

Proposition 5.9 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 3142. Let c, a, d and b be the first, the least, the greatest and the last values of π respectively. Then values from the range $[b, c]$ can be expressed as Equation 5.1, and values from the range $[a, b)$ and values from the range $(c, d]$ can be expressed as*

$$\bigoplus_{i=1}^{k} \sigma_i \qquad \textit{where each } \sigma_i \textit{ is 1 or a 312-position chain} \tag{5.4}$$

and

$$\bigoplus_{i=1}^{k} \sigma_i \qquad \textit{where each } \sigma_i \textit{ is 1 or a 231-position chain} \tag{5.5}$$

respectively.

5.1.2. Detailed structures

Here, we discuss more details of how values of a simple permutation in $\mathcal{A}'$ of extreme 2413 can be placed. We state Proposition 5.10 for simple permutations of extreme pattern 2413, and divide into four lemmas to prove the statement. As before, we apply the inverse symmetry to state the analogous proposition for a permutation of extreme pattern 3142. Afterwards, we state propositions that are converses of Proposition 5.10 and 5.16, which will be important in Chapter 6. At the end, we observe how first and last few values of a simple permutation of extreme pattern 2413 can be placed, and what first and last few values of a simple permutation of extreme pattern 3142 can be, so we can establish all the glue sum operations we need to prove the main theorem in the next section.

Consider a simple permutation π in $\mathcal{A}'$ of extreme pattern 2413. For the following proposition and four lemmas, let b, d, a and c be the first, the greatest, the least and the last values of π respectively, and denote by A, B and C each segment $[\pi^{-1}(b), \pi^{-1}(d))$, $[\pi^{-1}(d), \pi^{-1}(a)]$ and $(\pi^{-1}(a), \pi^{-1}(c)]$ respectively.

Proposition 5.10 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then:*

1. *in between two values for $1 \oplus 1$ in Equation 5.2, there exists a value x such that $\pi^{-1}(x) \in B$ or $\pi^{-1}(x) \in C$.*

2. *in between two values for $1 \ominus 1$ in Equation 5.1, there exists a value x such that $\pi^{-1}(x) \in A$ or $\pi^{-1}(x) \in C$.*

3. *in between two values for $1 \oplus 1$ in Equation 5.3, there exists a value x such that $\pi^{-1}(x) \in A$ or $\pi^{-1}(x) \in B$.*

4. *for a 231-value chain α in the segment A, let m and M be the minimum and maximum values of α respectively. Then so long as the chain continues, $\pi^{-1}(x) \in A$ for all $x \in [m, M] \setminus \{M-1\}$, and $\pi^{-1}(M-1) \in B$.*

5. *for a 312-value chain β in the segment C, let m and M be the minimum and maximum values of β respectively. Then so long as the chain continues, $\pi^{-1}(x) \in C$ for all $x \in [m, M] \setminus \{m+1\}$, and $\pi^{-1}(m+1) \in B$.*

6. *in between two values s and t playing roles of 1 and 2 of 12 in Equation 5.1, there exists a value whose position is either in A or C. Further, there can be at most four such values x_1, x_2, y_1 and y_2 with $x_1 < x_2 < y_1 < y_2$ where $\pi^{-1}(x_1), \pi^{-1}(x_2) \in A$ and $\pi^{-1}(y_1), \pi^{-1}(y_2) \in C$.*

7. *the positions of the value $a+1$ must be in A or B and the position of the value $d-1$ must be in B or C.*

Note that, unlike $1 \oplus 1$, in between $\sigma_1 \oplus 1$, $1 \oplus \sigma_1$ and $\sigma_1 \oplus \sigma_2$ in Equation 5.2 with 231-value chains σ_1 and σ_2, there can be a value x such that $\pi^{-1}(x) \in B$ or $\pi^{-1}(x) \in C$, but there does not have to be such a value. Similarly, in between $12 \ominus 1$, $1 \ominus 12$ and $12 \ominus 12$ in Equation 5.1, there can be an optional value x such that $\pi^{-1}(x) \in A$ or $\pi^{-1}(x) \in C$, and in between $\sigma_1 \oplus 1$, $1 \oplus \sigma_1$ and $\sigma_1 \oplus \sigma_2$ in Equation 5.3 with 312-value chains σ_1 and σ_2, there can be an optional value x such that $\pi^{-1}(x) \in A$ or $\pi^{-1}(x) \in B$.

We now prove the following lemma, which is for the first three statements of Proposition 5.10.

Lemma 5.11 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then:*

- *in between two values for $1 \oplus 1$ in Equation 5.2, there exists a value x such that $\pi^{-1}(x) \in B$ or $\pi^{-1}(x) \in C$.*
- *in between two values for $1 \ominus 1$ in Equation 5.1, there exists a value x such that $\pi^{-1}(x) \in A$ or $\pi^{-1}(x) \in C$.*
- *in between two values for $1 \oplus 1$ in Equation 5.3, there exists a value x such that $\pi^{-1}(x) \in A$ or $\pi^{-1}(x) \in B$.*

Proof. Since the proof for each statement is similar, we only prove the first statement. Let s and t be the values for the first 1 and the second 1 of a $1 \oplus 1$ respectively. Since their positions are both in A and consecutive, we cannot have $t = s + 1$. Let $x = s + 1$. Suppose to the contrary that $\pi^{-1}(x) \in A$. Now, due to the structure described in Lemma 5.6, if $\pi^{-1}(x) < \pi^{-1}(s)$, then s and x would be parts of a 231-value chain, so s would not be playing the role of the first 1 of a $1 \oplus 1$. On the other hand, if $\pi^{-1}(x) > \pi^{-1}(t)$, then t and x would be a part of a 231-value chain, so we have a contradiction again. Thus, the position of x must be either in B or C.■

Next, we prove the following.

Lemma 5.12 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then for a 231-value chain α in the segment A, let m and M be the minimum and maximum values of α respectively. Then so long as the chain continues, $\pi^{-1}(x) \in A$ for all $x \in [m, M] \setminus \{M - 1\}$, and $\pi^{-1}(M - 1) \in B$.*

Proof. We first show $\pi^{-1}(M - 1) \in B$. If the position of $M - 1$ is in A, then it must be a part of α due to the structure of π. Further, if $\pi^{-1}(M - 1) > \pi^{-1}(M)$, then as discussed in the proof of Lemma 5.6, $M - 1$ must have its descent pair, but the value of this point would be greater than M, so the position of $M - 1$ must be to the left of M. However, this implies the segment corresponding to α forms a block, so $\pi^{-1}(M - 1) \notin A$. If $\pi^{-1}(M - 1) \in C$, then $M(M - 2)da(M - 1)$ or $M(M - 3)da(M - 1)$ gives us 42513 pattern, so the position of $M - 1$

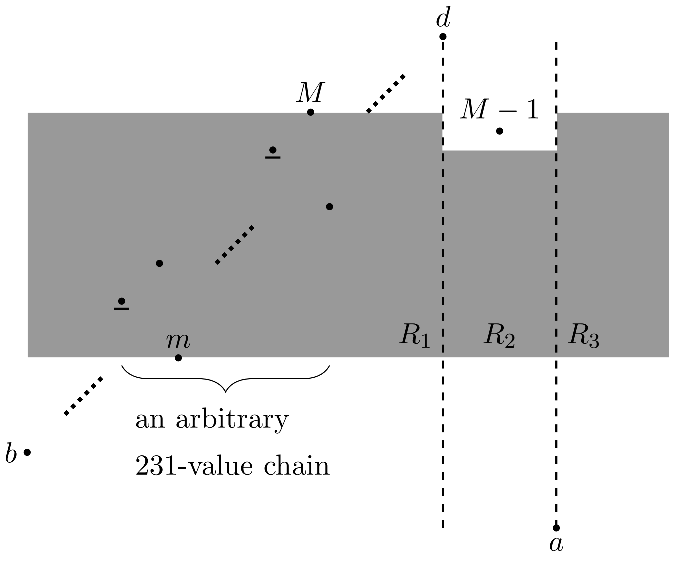


Figure 5.14.: Forbidden regions with a 231-value chain.

must be in B.

Next, we explain why other values in $[m, M]$ must be involved in the chain by using Figure 5.14. We cannot have any point in the region R_1 by 5.6. If we have a point in R_2, then its value, say w, must be between values corresponding to 2 and 1 of a descent 21 or 3 and 1 of 231 in 231-value chain. So suppose w is between values x and y which correspond to 2 and 1 respectively of $\alpha_i = 21$ for some positive integer i. Along with the value corresponding to 1 of α_{i+1}, say z, π contains 52341 or 52431 due to the subsequence $xyzwa$. Thus, we cannot have a point in R_2. We can apply the same argument if w is between values 3 and 1 of 231 in 231-value chain. Finally, assume we have a point w in R_3. Again, w must be between values corresponding to 2 and 1 of 21 or 3 and 1 of 231 in 231-value chain. Letting x and y denote the same as before, $xydaw$ gives 42513 pattern, so we cannot have a point in R_3. ■

With the reverse complement symmetry, the following lemma is an immediate consequence of Lemma 5.12.

Lemma 5.13 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then for a 312-value chain β in the segment C, let m and M be the minimum and maximum values of β respectively. Then so long as the chain continues, $\pi^{-1}(x) \in C$ for all $x \in [m, M] \setminus \{m+1\}$, and $\pi^{-1}(m+1) \in B$.*

Thus, for a 231-value chain with values of positions in A and a 312-value chain with values of positions in C have one value with its position in B. We refer to this value multiple times in later discussion, so it is worth naming this value. For a 231-value chain α in the segment A with the minimum value m and the maximum value M, we call the value $M-1$ the *scissor of a 231-value chain.* Similarly, for a 312-value chain β in the segment C with the minimum value m and the maximum value M, we call the value $m+1$ the *scissor of a 312-value chain.*

We move onto the next lemma describing the positions of values that are in between 1 and 2 of 12 in Equation 5.1.

Lemma 5.14 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then in between two values s and t playing roles of 1 and 2 of 12 in Equation 5.1, there exists a value whose position is either in A or C. Further, there can be at most four such values x_1, x_2, y_1 and y_2 with $x_1 < x_2 < y_1 < y_2$ where $\pi^{-1}(x_1), \pi^{-1}(x_2) \in A$ and $\pi^{-1}(y_1), \pi^{-1}(y_2) \in C$.*

Proof. Given an arbitrary values corresponding to 12 in Equation 5.1, call values playing the roles of 1 and 2 of 12 s and t respectively. It is necessary to have at least one value in $[s, t]$ whose position is in A or C due to the simplicity of π. If we have one value, say x in A and another value, say y in C that are in $[s, t]$, then $y > x$, because, otherwise, 35142 is contained due to $xdsty$. Because each 231-value chain needs the position of its scissor to be in B, it is not possible to have all values of an entire 231-value chain to be in between 1 and 2 of 12 as it requires for B to involve 123, 213 or 132 with the scissor, s and t, resulting in 52341, 53241 or 52431 pattern containment.

Now, suppose we have two values x_1, x_2 in $[s, t]$ with $x_1 < x_2$ whose positions are in A. As we can see in Figure 5.15, splitting the block $[\pi^{-1}(x_1), \pi^{-1}(x_2)]$ is only possible by having

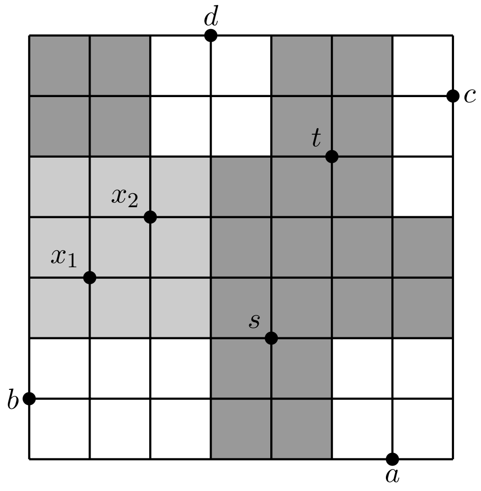


Figure 5.15.: Partial graph of π with the assumption of having two values in $[s, t]$ whose positions are in A.

a point with the value less than s and the position in $[\pi^{-1}(x_1), \pi^{-1}(x_2)]$. This makes x_1 a part of a 231-value chain with $M = x_1$. Since this is the only way to split $[\pi^{-1}(x_1), \pi^{-1}(x_2)]$, it is impossible to have a $1 \oplus 1$ in Equation 5.2 in between s and t. Hence, on the side of A, we can have at most two values x_1 and x_2 with x_1 being the maximum of a 231-value chain. Applying the reverse complement symmetry, we also have at most two values y_1 and y_2 ($y_1 < y_2$) on the side of C with y_2 begin the minimum of a 312-value chain in C. ■

Finally, we prove the lemma for the last statement of Proposition 5.10.

Lemma 5.15 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then the positions of the value $a+1$ must be in A or B and the position of the value $d-1$ must be in B or C.*

Proof. Suppose to the contrary that $\pi^{-1}(a+1) \in C$. Then $a+1$ cannot play the role of 1 in Equation 5.3 because, otherwise, $[\pi^{-1}(a), \pi^{-1}(a+1)]$ would be an unsplittable block. So $a+1$ must be playing the role of the minimum value of a 312-value chain. In either case, the position of $a+2$ must be in B due to Lemma 5.13. However, now we have an unsplittable block $[\pi^{-1}(a+2), \pi^{-1}(\ell)]$ where ℓ is the last value of the 312-value chain that $a+1$ is a part of. Hence, the position of the value $a+1$ cannot be in C. Proving $\pi^{-1}(d-1) \notin A$ can be done by the reverse complement argument. ■

With Lemma 5.11, 5.12, 5.13, 5.14 and 5.15, we have Proposition 5.10.

In Proposition 5.10, we observed the necessary conditions of simple permutations in $\mathcal{A}'$ of extreme pattern 2413. Proposition 5.16 summarize the inverse statement of these propositions for simple permutations in $\mathcal{A}'$ of extreme pattern 3142. Consider a simple permutation π in $\mathcal{A}'$ of extreme pattern 3142. This time, let c, a, d and b be the first, the least, the greatest and the last values of π respectively.

Proposition 5.16 *Let π be a simple permutation in $\mathcal{A}'$ of extreme pattern 3142. Then:*

1. *in between two positions of values for $1 \oplus 1$ in Equation 5.4, there exists a value x in (b, c) or in $(c, d]$ such that $\pi^{-1}(s) \leq \pi^{-1}(x) \leq \pi^{-1}(t)$.*

2. *in between two positions of values for $1 \ominus 1$ in Equation 5.1, there exists a value x in $[a, b)$ or in $(c, d]$ such that $\pi^{-1}(s) \leq \pi^{-1}(x) \leq \pi^{-1}(t)$.*

3. *in between two positions of values for $1 \oplus 1$ in Equation 5.5, there exists a value x in $[a, b)$ or in (b, c) such that $\pi^{-1}(s) \leq \pi^{-1}(x) \leq \pi^{-1}(t)$.*

4. *for a 312-value chain α in the range $[a, b)$, let m and M be the values of α whose positions are the first and the last in α respectively. Then so long as the chain continues, $x \in [a, b)$ for all x with $\pi^{-1}(x) \in [\pi^{-1}(m), \pi^{-1}(M)] \setminus \{\pi^{-1}(M) - 1\}$ and $\pi(\pi^{-1}(M) - 1) \in (b, c)$.*

5. *for a 231-value chain β in the range $(c, d]$, let m and M be the values of β whose positions are the first and the last in β respectively. Then so long as the chain continues, $x \in (c, d]$ for all x with $\pi^{-1}(x) \in [\pi^{-1}(m), \pi^{-1}(M)] \setminus \{\pi^{-1}(m) + 1\}$ and $\pi(\pi^{-1}(m) + 1) \in (b, c)$.*

6. *there exists a value in $[a, b)$ or $(c, d]$ whose position is in between $\pi^{-1}(s)$ and $\pi^{-1}(t)$ where s and t are values playing roles of 1 and 2 of 12 in Equation 5.1. Further, there could be at most four such values x_1, x_2, y_1 and y_2 with $\pi^{-1}(x_1) < \pi^{-1}(x_2) < \pi^{-1}(y_1) < \pi^{-1}(y_2)$ where $x_1, x_2 \in [a, b)$ and $y_1, y_2 \in (c, d]$.*

7. *the value of the position $\pi^{-1}(c) + 1$ must be in $[a, b)$ or $[b, c]$ and the value of the position $\pi^{-1}(b) - 1$ must be in $[b, c]$ or $(c, d]$.*

So far, we have proven that if a permutation π of extreme pattern 2413 is in $\mathrm{Si}(\mathcal{A}')$, then π satisfies every condition listed in Proposition 5.10 as well as the structural conditions described in Proposition 5.8. Proposition 5.19 states that if an arbitrary permutation π of extreme pattern 2413 having a structure described in Proposition 5.8 and 5.10, then $\pi \in \mathrm{Si}(\mathcal{A}')$. In order to prove this statement, we need the following lemma. As usual, let b, d, a and c be the first, the greatest, the least and the last values of π.

Lemma 5.17 There is no value x less than b such that $\pi^{-1}(x) \in C$.

Proof. Suppose to the contrary that there exists a value less than b whose position is in C. Let x be the greatest such value. First, assume x is not a part of a 312-value chain, so we have the graph shown in Figure 5.16(a). The only way to split the block $[\pi^{-1}(a), \pi^{-1}(x)]$ is by having a point in the region denoted by R_{42}. Denoting by y the value of the right-most point in R_{42} of 5.16(a), we obtain the graph in 5.16(b). We attempt to split the block $[\pi^{-1}(y), \pi^{-1}(x)]$ by having a point in R_{33}, but the consequential graph has an unsplittable block $[\pi^{-1}(y), \pi^{-1}(x)]$ as shown in 5.16(c).

Next, suppose x is a part of a 312-value chain. If x plays the role of 2 of 21 in the chain, we have a point with the value z playing the role of 1 of 21 in the region R_{44} of Figure 5.16, resulting in the graph shown in Figure 5.17(a). Note that if x played the role of 3 of 312 in the chain, then z shown in Figure 5.17(a) would be for 1 of 312, and we would have another value for 2 of 312 in the region R_{45}. The rest of the proof is the same for the case of x playing the

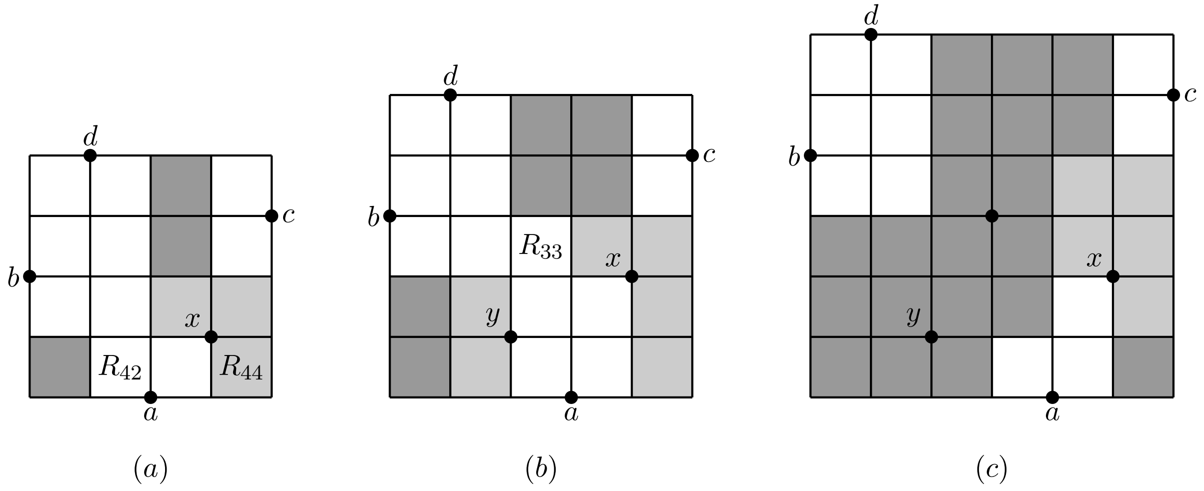


Figure 5.16.: Partial graphs of π with the assumption of having a value x that is not a part of a 312-value chain.

role of 2 of 21, so we only show the case of x for 2 of 21.

Suppose z is the minimum value in the 312-value chain. Then as discussed before, the position of $z + 1$ must be in B. Thus, we have the graph shown in Figure 5.17(b). Splitting the block $[\pi^{-1}(y_1), \pi^{-1}(z)]$ can be only done by having another value y in the region R_{62}, but then we have an unsplittable block $[\pi^{-1}(y), \pi^{-1}(z)]$ as shown in Figure 5.17(c), so we have a contradiction. Now, suppose z is not the minimum value in the 312-value chain. That means we have the minimum value m of the 312-value chain in R_{53} in Figure 5.17(a). With the value $m+1$ whose position is in B, we have the graph as in Figure 5.17(d). The segment $[\pi^{-1}(m+1), \pi^{-1}(z)]$ is a block, and the only way to split it is by having a point with the value y in R_{82}. Consequently, we obtain the graph shown in Figure 5.17(e), but $[\pi^{-1}(y), \pi^{-1}(z)]$ is an unsplittable box. This completes the proof of the lemma. ■

With the reverse complement property of Lemma 5.15, we have the following lemma. With these lemmas, we are ready to prove Proposition 5.19.

Lemma 5.18 *There is no value x greater than a such that $\pi^{-1}(x) \in A$.*

Proposition 5.19 *Let π be a permutation of extreme pattern 2413 where $|\pi| = n$. If π obeys all the conditions of Proposition 5.8 and 5.10, then $\pi \in Si(\mathcal{A}')$.*

Proof. First, we prove π is simple. Suppose π is not simple, and let I be a non-singleton proper block of π. Let b, d, a and c be the first, the greatest, the least and the last values of

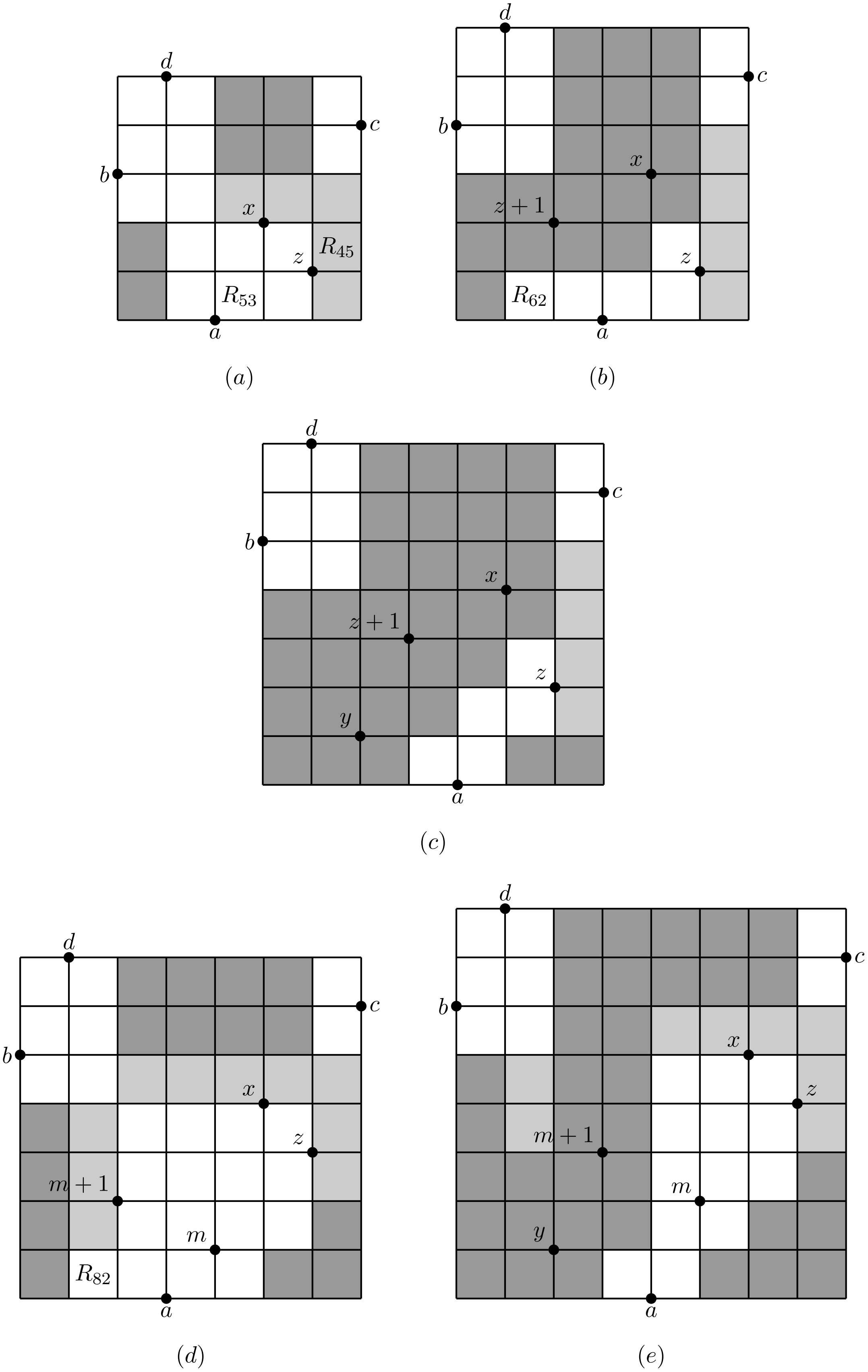


Figure 5.17.: Partial graphs of π with the assumption of having a value x that is a part of a 312-value chain.

π respectively. Additionally, let A, B and C be segments $[\pi^{-1}(b), \pi^{-1}(d))$, $[\pi^{-1}(d), \pi^{-1}(a)]$ and $(\pi^{-1}(a), \pi^{-1}(c)]$ respectively. Obviously, I cannot contain both the positions of b and c because this implies $I = [\pi^{-1}(b), \pi^{-1}(c)] = [1, n]$. The same result can be quickly obtained if I contains any two positions of b, d, a and c. For example, assume I contains the positions of b and d. Since $b < c < d$, we must have $\pi^{-1}(c) \in I$, so again, $I = [1, n]$. Same can be done for any other two of b, d, a and c.

Assume I contains only one position of b, d, a or c. We first assume $\pi^{-1}(d) \in I$. Then $\pi^{-1}(d-1)$ also has to be in I. By the seventh condition listed in Proposition 5.10, we have the assumption $\pi^{-1}(d-1) \notin A$. If $\pi^{-1}(d-1) \in C$, then $\pi^{-1}(a)$ has to be in I, which is a contradiction. So suppose $\pi^{-1}(d-1) \in B$. Note that d and $d-1$ cannot be forming $1 \ominus 1$ due to the second condition listed in Proposition 5.10. Hence, d and $d-1$ are 1 of the initial 1 and 2 of the 12 in the expression $1 \ominus 12 \ominus \cdots$. However, by the sixth condition of Proposition 5.10, we must have a value in between 1 and 2 of 12 whose position is either in A or C, *i.e.* $\pi^{-1}(d-2)$ must be in A or C. Moreover, by Lemma 5.17, every value whose position is in A must be less than c, so $\pi^{-1}(d-2) \in C$. This implies $\pi^{-1}(a) \in I$, which is a contradiction. Hence, I cannot contain the position of d. We have the same result for the position of a by the reverse complement argument.

Thus, suppose I contains $1 = \pi^{-1}(b)$. Because I does not contain the position of d, we have $I \subseteq A$. Now, if b is a part of a 231-value chain, then $[m, M] \subseteq \pi(I)$ where m and M are the minimum and maximum values of the 231-value chain that b is a part of. However, since we know $\pi^{-1}(M-1) \in B$ by the fourth condition of Proposition 5.10, this implies $\pi^{-1}(d) \in I$. So assume b is playing the role of the first 1 in Equation 5.2. If $\pi(2)$ is also playing the role of another 1 in Equation 5.2, then there must be a value x such that $b < x < \pi(2)$ where $\pi^{-1}(x)$ is in B or in C, so again, we have a contradiction of $\pi^{-1}(d) \in I$. Hence, $\pi(2)$ is a part of a 231-value chain, but this implies $[m, M] \subseteq \pi(I)$ where m and M are the minimum and maximum values of the 231-value chain that $\pi(2)$ is a part of, resulting in $\pi^{-1}(d) \in I$ again. Similar result can be obtained for the case of I containing $n = \pi^{-1}(c)$. Consequently, I cannot contain any of $\pi^{-1}(b)$, $\pi^{-1}(d)$, $\pi^{-1}(a)$ or $\pi^{-1}(c)$.

So suppose I does not contain any position of b, d, a or c. This implies $I \subseteq A$, $I \subseteq B$ or $I \subseteq C$. Assume $I \subseteq A$. Then $\pi(I)$ must contain at least two values s and t whose positions are consecutive in A $(\pi^{-1}(s) < \pi^{-1}(t))$. If s and t are playing roles of a $1 \oplus 1$ in Equation 5.2, then there exists a value x such that $s < x < t$ where $\pi^{-1}(x)$ is in B or in C. As before, this implies $\pi^{-1}(d) \in I$, which is a contradiction. If s and t are both parts of the same 231-value chain, then $[m, M] \subseteq \pi(I)$ where m and M are the minimum and maximum values of the 231-value chain that s and t are parts of. Since $\pi^{-1}(M-1) \in B$, this implies $\pi^{-1}(d) \in I$ again, so we have a contradiction. Similar result can be shown if s is playing a role of 1 in Equation 5.2 and t is a part of a 231-value chain, or vice versa. Thus, we cannot have $I \subseteq A$. By the reverse complement argument, $I \subseteq C$ also cannot be true.

Finally, suppose $I \subseteq B$. Then $\pi(I)$ must contain at least two values s and t whose positions are consecutive in B $(\pi^{-1}(s) < \pi^{-1}(t))$. If s and t are playing roles of a $1 \ominus 1$ in Equation 5.1, then there exists a value x whose position is in A or C. Hence, $x \in \pi(I)$, so $\pi^{-1}(x) \in I$. If $\pi^{-1}(x) \in A$, then the position of d would be in I. On the other hand, if $\pi^{-1}(x) \in C$, then the position of a would be in I. Thus, in either way, we achieve a contradiction. Similarly, if s and t are playing roles of 1 and 2 of a 12 in Equation 5.1, then there exists a value x whose position is in A or C. Again either the position of d or a would be in I, which is a contradiction. Similar result can be shown for the case if s is playing a role of 2 of a 12 and t is playing a role of 1 of a 1 in Equation 5.1, and for the case if s is playing a role of 1 of a 1 and t is playing a role of 1 of a 12 in Equation 5.1.

Consequently, we have considered every case of where a non-singleton proper block I can belong to, and we achieved a contradiction in each case. Hence, π is simple.

Next, we prove π avoids every β in $\{52341, 53241, 52431, 35142, 42513, 351624\}$. Suppose to the contrary that $\beta \preceq \pi$ for some $\beta \in \{52341, 53241, 52431, 35142, 42513, 351624\}$. We first let $\beta \in \{52341, 53241, 52431\}$. Notice that $\text{LRmax}(\pi) = \{x : \pi^{-1}(x) \in A \text{ and } x \text{ plays the role of 2 or 3 of a 231-value chain, or 1 of 1 in Equation 5.2}\} \cup \{d\}$ and $\text{RLmin}(\pi) = \{x : \pi^{-1}(x) \in C \text{ and } x \text{ plays the role of 1 or 2 of a 312-value chain, or 1 of 1 in Equation 5.3}\} \cup \{a\}$. If any two values from these two sets play the role of 2, 3 or 4, we cannot find a value corresponding to either 5 or 1. Thus, each value playing the role of 2, 3 and 4 must be in one of the following three sets.

- $S_1 = \{x : \pi^{-1}(x) \in A,\ x \text{ corresponds to 1 of 21 or 231 in a 231-value chain}\}$
- $S_2 = \{x : \pi^{-1}(x) \in B,\ x \neq 1, n\}$
- $S_3 = \{x : \pi^{-1}(x) \in C,\ x \text{ corresponds to 2 of 21 or 3 of 312 in a 312 value chain}\}$

Let $x \in \mathcal{S}_1$. Furthermore, suppose x is a value corresponding to 1 of 231 in some 231-value chain. We show that x cannot play the role of any of 2, 3 and 4. Suppose to the contrary that x plays the role of one of the values of $\beta(2)$, $\beta(3)$ or $\beta(4)$. There are three values y_1, y_2 and y_3 that are greater than x located to the left of x in increasing order, as shown in Figure 5.18, so one of them has to play the role of $\beta(1) = 5$. As we choose one of them to be playing the role of 5, other two cannot be assigned to any other values of β, as these two values are either located

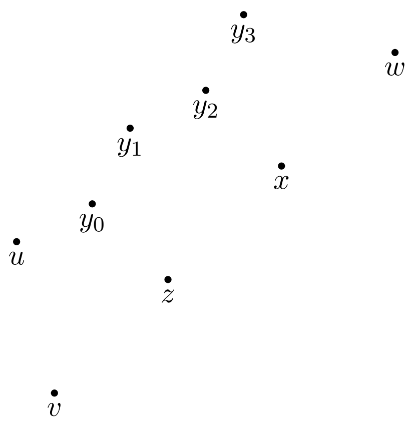


Figure 5.18.: Partial graph of π to show that $52341, 53241, 52431 \not\leq \pi$.

to the left of, or greater than the value for $\beta(1) = 5$.

Suppose x plays the role of $\beta(4)$. If $\beta = 52341$ or 53241, then $\beta(4) = 4$, but then we need two values less than x whose positions are in between $\pi^{-1}(y_i)$ and $\pi^{-1}(x)$ which we don't have for each of y_1, y_2 and y_3. If $\beta = 52431$, then we assign y_1 and its descent pair z to play the roles of 5 and 2 respectively, but we do not have any value for 4. Hence, x cannot correspond to $\beta(4)$.

This time, assume x plays the role of $\beta(3)$. For $\beta = 53241$, we don't have any value corresponding to 3. For $\beta = 52341$, we assign y_1 and z to 5 and 2 respectively as this is the only way to have a value for 2, but since $y_1 = x + 1$, we do not have a value corresponding to 4. Lastly, for $\beta = 52431$, we again assign y_1 and z to 5 and 2 respectively. However, if y_1 and z correspond to 3 and 1 of 231 in the 231-value chain respectively as in Figure 5.18, then the only values greater than z and less than x is y_0 and u shown in Figure 5.18, which are located to the left of x, so we don't have a value for 3. On the other hand, if y_1 and z correspond to 2 and 1 of 21 in the 231-value chain, then the only value in between z and x is u. It is also possible that u and v do not exist, as y_1 and z correspond to the first summand of the 231-value chain expression. In this case, there is no value greater than z and less than x, because $x = z+1$, so we do not have a value for 3 in either way. Therefore, x cannot play the role of $\beta(3)$.

The last case is x playing the role of $\beta(2)$. If y_3 plays the role of $\beta(1) = 5$, then there is one value w that is greater than x and less than y_3 whose position is to the right of x. Note that w can be a value in S_1, *i.e.* a part of the next summand in the 231-value chain expression, or $\pi^{-1}(w) \in B$, if the 231 that y_3 and x are parts of is the last summand in the expression. If $\beta = 52341$ or 52431, we do not have enough values to assign to 3 and 4. If $\beta = 53241$, then w must play the role of 4, but since there is no value less than x whose position is in between

$\pi^{-1}(x)$ and $\pi^{-1}(w)$, we do not have a value corresponding to 2 of β.

Consequently, we achieve a contradiction in every case. If x is a value corresponding to 1 of 21 in a 231-value chain, then we obtain the same result, since the only difference is the absence of y_2 in Figure 5.18. So x cannot play the role of any value of $\beta(2)$, $\beta(3)$ and $\beta(4)$. Since these three values correspond to 2, 3 and 4 for every β, x cannot play the role of any of 2, 3 and 4.

We can apply the reverse complement argument of the above to show that any value in S_3 cannot play the role of any of 2, 3 and 4. Therefore, values playing roles of 2, 3 and 4 must be in S_2, implying their positions are in B. However, since values whose positions are in B must be expressed by Equation 5.1, these values avoid 123, 213 and 231 patterns. If the positions of the values corresponding to 2, 3 and 4 of β are in B, then these values must form either 123, 213 or 231 pattern. Hence, π avoids every $\beta \in \{52341, 53241, 52431\}$.

Now, let $\beta = 35142$. For the same reason that the roles of 2, 3 and 4 of 52341 cannot be played by any value in LRmax(π) and RLmin(π), the role of 4 cannot be taken by any value in LRmax(π) and RLmin(π). Let x be the value playing the role of 4. We first let $x \in S_1$. As before, we may assume x corresponds to 1 of 231 in a 231-value chain, since the case of x corresponding to 1 of 21 in a chain gives the same result. Let us refer to Figure 5.18 again. The only value we can assign to 5 is y_1, as this is the only way to have a value for 1, which is z. Either u or y_0 must play the role of 3. If u plays the role of 3, then we do not have a value for 2 because $u = z + 1$. On the other hand, if y_0 plays the role of 3, then u is the only value in between z and y_0, if u and v exist. Since it is located to the left of y_0, it cannot take the role of 2. Therefore, a value in S_1 cannot be assigned to 4 of β.

Next, assume $x \in S_3$. Thus, we have Figure 5.19. The only way to assign values to both 3 and 2 are by setting w and y_3 to take roles of them respectively, but we now don't have a value

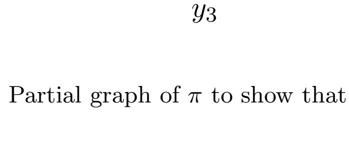


Figure 5.19.: Partial graph of π to show that 35142 $\not\leq \pi$.

corresponding to 5. Consequently, a value in S_3 cannot play the role of 4 of β.

Finally, suppose $x \in S_2$. In this case, the position of a value y playing the role of 5 must be in A or B with $\pi^{-1}(y) < \pi^{-1}(x)$. In order to assign a value to 1, x must correspond to 2 of 12 in Equation 5.1 or the scissor of some 231-value chain. In the former case, the paired value z corresponding to 1 of 12 being assigned to 1 of β. Due to the structure of values whose positions are in B, the position of the value v corresponding to 3 must come from C and the position of the value w corresponding to 2 must come from A. However, with the sixth condition of Proposition 5.10, we ensure that $v < w$, so these assignments are not possible. In case of x corresponding to the scissor of some 231-value chain with the minimum value m and maximum value M, M and its descent paired value must be assigned to 3 and 1 of β respectively. Then we do not have a value corresponding to 2, because except x, the positions of the values in the range $[m, M]$ must be in A. Hence, no value in S_3 can play the role of 4 of β.

Consequently, we have $\beta = 35142 \not\preceq \pi$. We can show that π does not contain $\beta = 42513$ by applying the reverse complement of the argument for 35142. Thus, the remained case is $\beta = 351624$.

Let x be the value playing the role of 6. Note that x cannot be in $\mathrm{RLmin}(\pi)$, since this leaves no values to be assigned to 2 and 4. So assume $x \in S_3$. Referring to Figure 5.19, the value playing the role of 2 must be y_1 or y_2 while the value for 4 must be y_2 or y_3. Therefore, Suppose y_1 plays the role of 2. Then the values for 3 and 5 must come from $\mathrm{LRmax}(\pi)$, S_1 or S_2. However, since x and y_1 are parts of a 312-value chain, the only value that is allowed to have a position outside of C is the splitter, so we don't have enough values to assign for both 3 and 5. We obtain the same result if y_2 plays the roles of 2. Note that, by the reverse complement argument, the value playing the role of 1 cannot be in $\mathrm{LRmax}(\pi)$ or S_1.

Now, suppose $x \in S_2$. Because we cannot assign any value in $\mathrm{LRmax}(\pi)$ or S_1 to 1 of β, we also have to have a value z in S_2 which plays the role of 1. Due to the structure of values with positions in B, the only way to assign values to both 1 and 6 is by setting z and x to correspond to 1 and 2 of the same 12 in Equation 5.1 respectively. Then the values for 3 and 5 must come from $\mathrm{LRmax}(\pi)$ or S_1 and the values for 2 and 4 must come from S_2 or $\mathrm{RLmin}(\pi)$. This is, however, a contradiction, because four values in between of values corresponding to 1 and 2 of 12 in Equation 5.1 must be in increasing order from left to right due to the sixth condition of Proposition 5.10. So this concludes that x cannot be in S_2.

The remained two cases are $x \in S_1$ and $x \in \mathrm{LRmax}(\pi)$. In either case, the value z playing the role of 1 must be in S_1 or LRmax(π) as well, but as discussed previously, the role of 1 cannot be played by a value in S_1 or LRmax(π). Hence, $\beta = 351624 \not\preceq \pi$.

Consequently, π avoids every permutation in $\{52341, 53241, 52431, 35142, 42513, 351624\}$, so π is in $\mathcal{A}'$. With the simplicity we proved earlier, this completes the proof. ■

We state the analogous proposition for permutations of extreme pattern 3142. Proposition 5.19 and 5.20 will be referred in Chapter 6.

Proposition 5.20 *Let π be a permutation of extreme pattern 3142 where $|\pi| = n$. If π has a structure described in Proposition 5.9 and 5.16, then $\pi \in Si(\mathcal{A}')$.*

For the reminder of this section, we discuss how first few values of a simple permutation π of extreme pattern 2413 can be placed. In particular, we divide the arrangement of the values in $\{x : 1 \leq x \leq \pi(1)\}$ into six distinct cases. We will use these six cases in the encoding rule, which we define in Chapter 6. Then we apply the reverse complement property and the inverse property to establish analogous six distinct cases for each of:

- where the values x with $\pi(n) \leq x \leq n$ are located for a simple permutation π of extreme pattern 2413,
- what values x with $1 \leq \pi^{-1}(x) \leq \pi^{-1}(1)$ can be for a simple permutation π of extreme pattern 3142 and
- what values x with $\pi^{-1}(n) \leq \pi^{-1}(x) \leq n$ can be for a simple permutation π of extreme pattern 3142.

As before, let $b = \pi(1)$, $d = n$, $a = 1$ and $c = \pi(n)$ where π is a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Six distinct starting cases are shown in Figure 5.20. The dashed line in each case indicates the position of d.

In every case, $a = 1$ plays the role of the last 1 in Equation 5.1. The difference is what each value x with $2 \leq x \leq b$ is playing the role of. In Case 1, simply, $b = 2$ and it is playing the role of the first 1 in segment A, when the values in segment A are expressed as $1 \oplus \cdots$. Also, in this case, the values of positions in the segment B are written as $\cdots \ominus 1 \ominus 1$. In this case,

$$\pi = 2 \ldots d \ldots 1 \ldots c.$$

In Case 2, b again plays the role of the first 1 when the values of positions in the segment A are formed as $1 \oplus \cdots$. However, the values of positions in the segment B are expressed as

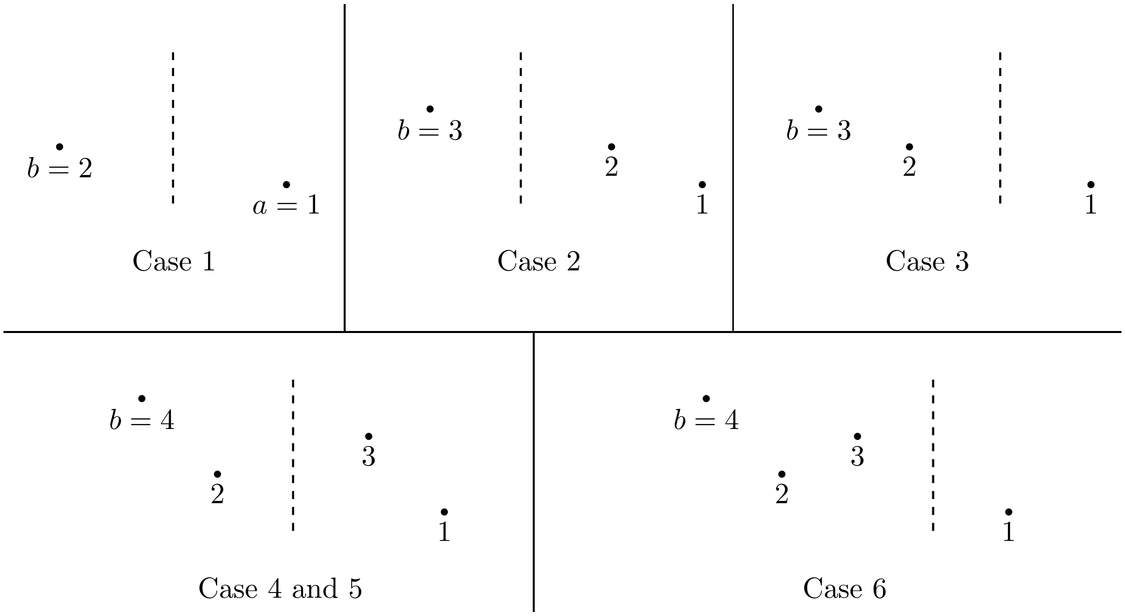


Figure 5.20.: Positions of values x with $1 \leq x \leq \pi(1)$ for π with extreme pattern 2413.

$\cdots \ominus 12 \ominus 1$, and b is in between the two values taking the roles of 1 and 2 of the last 12. Thus, the value 2 is playing the role of 1 of the 12 here, which makes $b = 3$. By Proposition 5.10, there are possibly two more values with positions in segment C which are between the values playing the roles of 1 and 2 of 12, so π should be written as

$$\pi = 3 \ldots d \ldots 2\, z\, 1 \ldots c$$

where $z = 4$, 5 or 6.

Case 3 may appear similar to Case 2, but the role b is taking is completely different. In this case, segment A looks like $\alpha_1 \oplus \cdots$ where α_1 is the first 231-value chain which starts with 231 pattern, *i.e.* $\alpha_1 = 231 \oplus_1 \cdots$. The values $b = 3$ and 2 are playing the roles of 2 and 1 of this 231 respectively. Moreover, the value 5 is playing the role of 3 of this 231, so we have

$$\pi = 3\,5\,2\, \ldots d \ldots 1 \ldots c.$$

If we simply have $\alpha_1 = 231$, then the position of the value 4 must be in the segment B. If the 231-value chain α_1 continues, then $\pi^{-1}(4)$ must be in A.

In Case 4, 5 and 6, segment A starts as $\alpha_1 \oplus \cdots$ where α_1 is the first 231-value chain where $\alpha_1 = 21 \oplus_1 \cdots$. The values $b = 4$ and 2 are playing the roles of 2 and 1 of the first 21 in α_1 respectively. In Case 4 and Case 5, $\alpha_1 = 21$, so the position of the value 3 is in the segment B.

If the values of positions in B is expressed as $\cdots \ominus 1 \ominus 1$, then the value 3 is playing the role of the second 1 from the last. In contrast, if the values of positions in B is written as $\cdots \ominus 12 \ominus 1$, the value 3 is playing the role of 1 of the last 12. Hence, in Case 4 and Case 5, respectively, we have

$$\pi = 4\,2 \ldots d \ldots 3\,1 \ldots c \qquad \text{and} \qquad \pi = 4\,2 \ldots d \ldots 3\,z\,1 \ldots c$$

with $5 \leq z \leq 8$. On the other hand, in Case 6, the 231-value chain α_1 continues after the initial 21, resulting in $\pi^{-1}(3) \in A$. Thus,

$$\pi = 4\,2 \ldots 3 \ldots d \ldots 1 \ldots c.$$

Based on the structure we discussed in Proposition 5.8 and 5.10, these six cases the only possible ways that first several values of simple permutations of extreme pattern 2413 can be placed. By applying the reverse complement, we obtain the possible behaviors of points with the values x with $\pi(n) \leq x \leq n$, where π is a simple permutation of extreme pattern 2413. Analogous cases are shown in Figure 5.21. Also, for a simple permutation π of extreme pattern 3142, the first and the last few values of π are obtained by the inverse property, shown in Figure 5.22 and 5.23 respectively. In Figure 5.21, 5.22 and 5.23, dashed lines indicate $\pi^{-1}(a)$, c and b respectively.

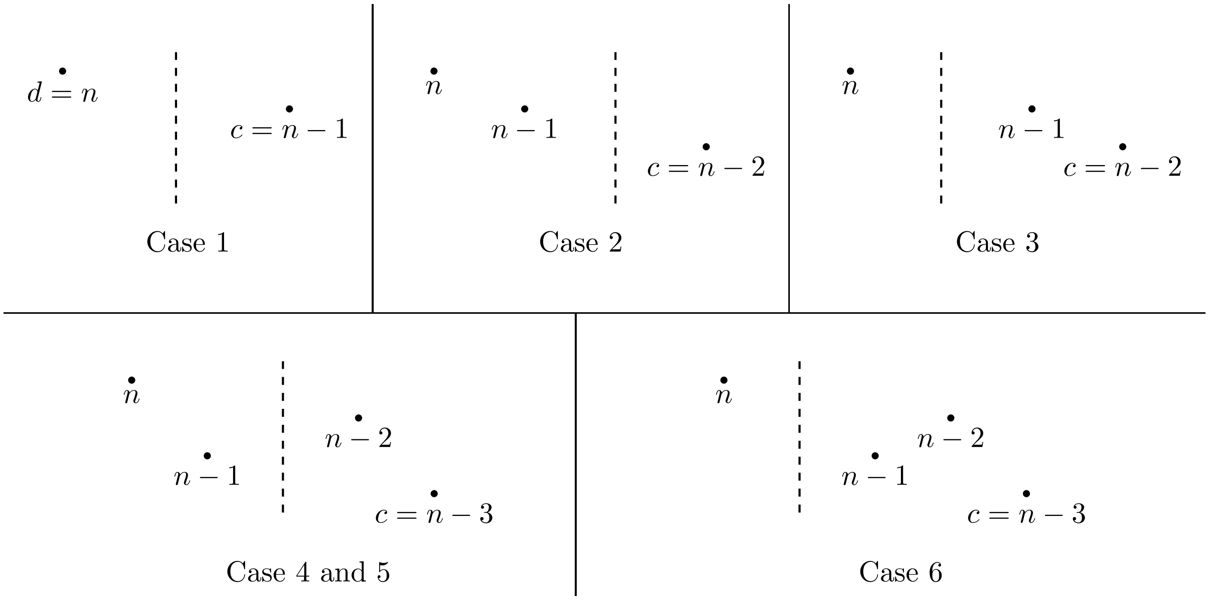


Figure 5.21.: Positions of values x with $\pi(n) \leq x \leq n$ for π with extreme pattern 2413.

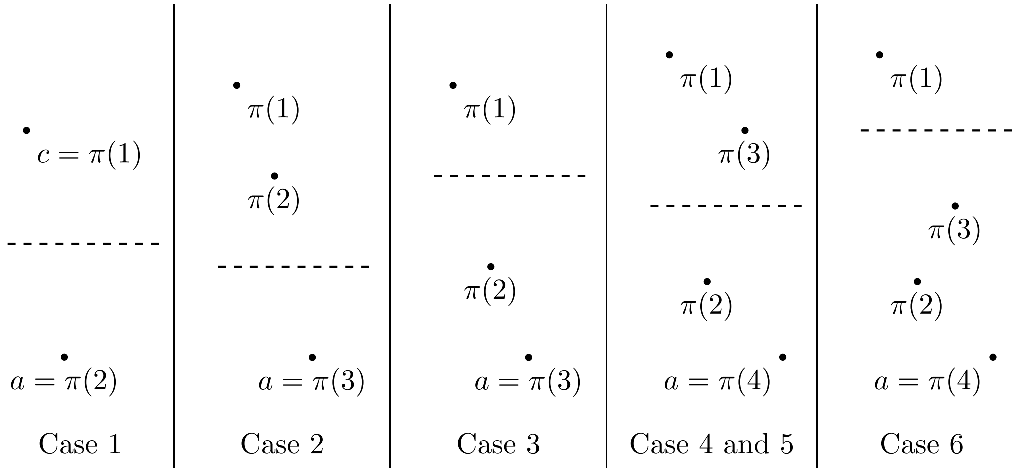


Figure 5.22.: Values of positions s with $1 \leq s \leq \pi^{-1}(1)$ for π with extreme pattern 3142.

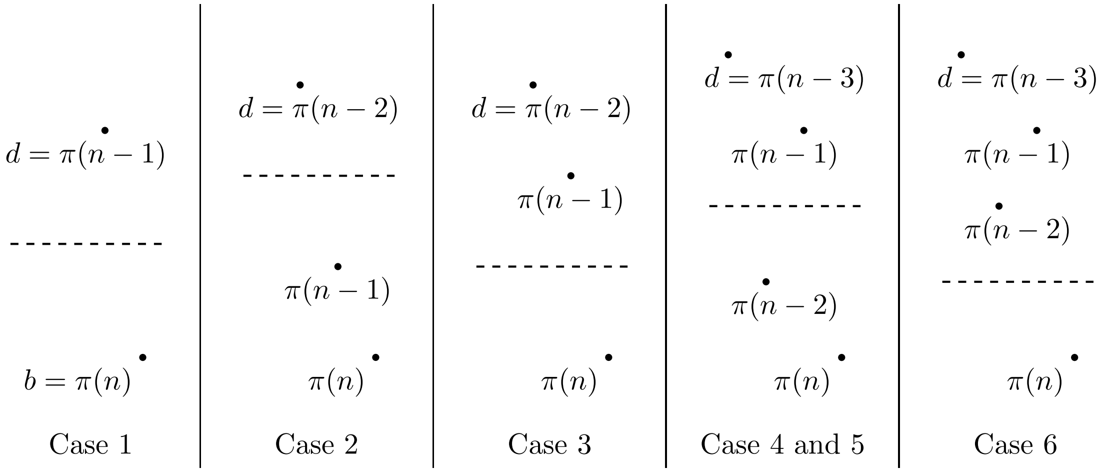


Figure 5.23.: Values of positions s with $1 \leq \pi^{-1}(n) \leq n$ for π with extreme pattern 3142.

5.2. General simple permutations in $\mathcal{A}'$

In the previous section, we looked at the structures of simple permutations of extreme pattern 2413 and the structures of simple permutations of extreme pattern 3142 in great detail. We are ready to discuss the structure of general simple permutations in $\mathcal{A}'$. Recall that in $\mathcal{A}$, we established Theorem 4.4, which states that the half of simple permutations can be written as alternated NW and SE glue sums of simple permutations of extreme pattern 2413 and 3142, starting with one of extreme pattern 2413. In this section, we extend Theorem 4.4 to describe the structure of the half of simple permutations in $\mathcal{A}'$. In order to state the theorem extending Theorem 4.4, we need to define several more glue sums.

5.2.1. Glue sums and the structure theorem

We carry the same definitions for type 1-0 NW glue sum, type 1-0 SE glue sum, type 1-1 NW glue sum and type 1-1 SE glue sum for simple permutations in $\mathcal{A}'$. Recall that, in the definition of $\sigma \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}_1^0 \tau$ and $\sigma \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}_1^1 \tau$, σ ($|\sigma| = m$) and τ ($|\tau| = n$) must satisfy the conditions that $\sigma^{-1}(m) \leq m-2$, $\sigma(m) = m-1$, $\tau(1) \geq 3$ and $\tau(2) = 1$. In the language of the previous section, σ must have the positions of the last two values $m-1$ and m as Case 1 of Figure 5.21 and τ must have the values of the first two positions $\tau(1)$ and $\tau(2)$ as Case 1 of Figure 5.22.

Similarly, for $\sigma \mathbin{\bigcirc\hspace{-0.8em}\searrow}_1^0 \tau$ and $\sigma \mathbin{\bigcirc\hspace{-0.8em}\searrow}_1^1 \tau$, two simple permutations σ ($|\sigma| = m$) and τ ($|\tau| = n$) must satisfy the conditions of $\sigma(m) \leq m-2$, $m = \sigma(m-1)$, $\tau^{-1}(1) \geq 3$ and $2 = \tau(1)$. Thus, σ must have the values of the last two positions $\sigma(m-1)$ and $\sigma(m)$ as Case 1 of Figure 5.23 and τ must have the positions of the first two values 1 and 2 as Case 1 of Figure 5.20.

For every glue sums that we define to combine simple permutations σ and τ in this section, σ and τ have to satisfy specific conditions similar to the ones for type 1-0 and 1-1 glue sums. In particular, σ, the left permutation of the sum, must follow a particular structure described in Figure 5.21 for NW glue sums and 5.23 for SE glue sums. As noted, Type 1-0 and 1-1 are for Case 1 structure. Type 2-0 and 2-1 are for Case 2, type 3-0 is for Case 3, and type 4-0 is for Case 4 and Case 5. According to which structure σ has, τ also has to have a certain structure. The second number indicates whether the last value of σ is left as a copy or not in the process of glue summing. If the second number is 1, the glue sum leaves a copy of the last point of σ when it is merged into the last point of τ, just as it was explained with type 1-1 at the end of the proof of Proposition 4.5.

We now introduce all types of NW glue sums in Table 5.1. Note that every simple permutation of extreme pattern 2413 satisfies the conditions of σ, which is the left side of each sum, and similarly, every simple permutation of extreme pattern 3142 satisfies the conditions of τ, the right side of each sum. Let σ and τ be simple permutations in $\mathcal{A}'$ of length m and n respectively, $i = \sigma^{-1}(m)$ and $j = \tau(1)$. For each NW glue sum, it is still proper to think the last value of σ and the least value of τ are merged into the last value of τ and the least value of σ respectively. Just like type 1-0 and 1-1, each sum combines the point of σ whose value is m and the point of τ whose value is j into one point. This point keeps the position i, and its value is m shifted upward by certain amount. In addition, for type 2-0, 2-1 and 3-0, the point whose value is $m-1$ is also shifted up by the same amount m is shifted. Likewise, for type 4-0, the point whose value is $m-2$ is shifted up together with m by the same amount.

Type	Conditions of σ	Conditions of τ	Notation and Definition
1-0	$i \leq m-2$, $\sigma(m) = m-1$. (Case 1 of Figure 5.21)	$j \geq 3$, $\tau(2) = 1$. (Case 1 of Figure 5.22)	$\sigma \textcircled{\nwarrow}_1^0 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m-1)\tau'(3)\tau'(4)\cdots\tau'(n)$ where $\sigma'(i) = m+(j-3)$ and $\sigma'(k) = \sigma(k)$ for $k \neq i$, and $\tau'(k) = \tau(k)+(m-3)$ for k with $3 \leq k \leq n$.
1-1			$\sigma \textcircled{\nwarrow}_1^1 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m)\tau'(3)\tau'(4)\cdots\tau'(n)$ where $\sigma'(i) = m+(j-2)$ and $\sigma'(k) = \sigma(k)$ for $k \neq i$, and $\tau'(k) = \tau(k)+(m-2)$ for k with $3 \leq k \leq n$.
2-0	$i \leq m-4$, $\sigma(i+2) = m-1$, $\sigma(m) = m-2$. (Case 2 of Figure 5.21)		$\sigma \textcircled{\nwarrow}_2^0 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m-1)\tau'(3)\tau'(4)\cdots\tau'(n)$ where $\sigma'(i) = m+(j-3)$, $\sigma'(i+2) = (m-1)+(j-3)$ and $\sigma'(k) = \sigma(k)$ for $k \neq i, i+2$, and $\tau'(k) = \tau(k)+(m-3)$ for k with $3 \leq k \leq n$.
2-1			$\sigma \textcircled{\nwarrow}_2^1 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m)\tau'(3)\tau'(4)\cdots\tau'(n)$ where $\sigma'(i) = m+(j-2)$, $\sigma'(i+2) = (m-1)+(j-2)$ and $\sigma'(k) = \sigma(k)$ for $k \neq i, i+2$, and $\tau'(k) = \tau(k)+(m-2)$ for k with $3 \leq k \leq n$.
3-0	$i \leq m-5$, $\sigma(m-2) = m-1$, $\sigma(m) = m-2$ and $\sigma(m)$ is a part of the last 312-value chain α. (Case 3 of Figure 5.21)	$j \geq 6$, $\tau(2) = 3$, $\tau(3) = 1$ and 1 is a part of the least 312-position chain β. (Case 3 of Figure 5.22)	$\sigma \textcircled{\nwarrow}_3^0 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m-1)\tau'(\ell+3)\tau'(\ell+4)\cdots\tau'(n)$ where ℓ is the number of values in the 312-position chain β, $\sigma'(i) = m+(j-\ell-3)$, $\sigma'(m-2) = (m-1)+(h-\ell-2)$ where $h = \tau(\ell+1)$ and $\sigma'(k) = \sigma(k)$ for $k \neq i, m-2$, and $\tau'(k) = \tau(k)+(m-\ell-3)$ for k with $\ell+3 \leq k \leq n$.
	Additionally, α must be similar to β.		
4-0	$i \leq m-4$, $\sigma(m-1) = m-1$, $\sigma(m) = m-3$. (Case 4 or 5 of Figure 5.21)	$j \geq 5$, $\tau(2) = j-2$, $\tau(3) = 1$. (Case 2 of Figure 5.22)	$\sigma \textcircled{\nwarrow}_4^0 \tau = \sigma'(1)\sigma'(2)\cdots\sigma'(m-2)\tau'(4)\tau'(5)\cdots\tau'(n)$ where $\sigma'(i) = m+(j-3)$, $\sigma'(s) = (m-2)+(j-3)$ (s position of the value $m-2$) and $\sigma'(k) = \sigma(k)$ for $k \neq i, s$, and $\tau'(k) = \tau(k)+(m-3)$ for k with $4 \leq k \leq n$.

Table 5.1.: Definitions of NW glue sums ($i = \sigma^{-1}(m)$ and $j = \tau(1)$).

The only differences between type 1-0 and type 2-0 are the conditions of σ and whether $m-1$ is shifted up or not. Same can be said for type 1-1 and 2-1.

Before we discuss type 3-0 NW glue sum, we define another terminology. Let α and β be a 312-value chain and a 312-position chain respectively. We say α and β are *similar* if the flattening of $21 \oplus_1 \alpha$ and the flattening of $\beta \oplus^1 21$ are equal to each other. For instance, $\alpha = 21 \oplus_1 312$ is similar to $\beta = 312 \oplus^1 21$ since $21 \oplus_1 \alpha = \beta \oplus^1 21 = 3152746$. Likewise, we say a 231-value chain α and a 231-position chain β are *similar* if the flattening $21 \oplus_1 \alpha$ and the flattening of $\beta \oplus^1 21$ are equal to each other.

As indicated in Table 5.1, for type 3-0 NW glue sum, σ and τ must have a 312-value chain with ℓ values involving $\sigma(m)$ and a 312-position chain with ℓ values involving 1 respectively that are similar to each other. For now, let σ and τ be simple permutations of extreme pattern 2413 and 3142 respectively. We visualize $\sigma \oplus_3^0 \tau$ in Figure 5.24. In σ, we must have a scissor s as indicated in Figure 5.24 because of the fifth condition of Proposition 5.10. Similarly, we have a scissor h as shown in Figure 5.24. Note that the position of h is $\ell+1$, so $h = \tau(\ell+1)$. The 312-value chain of σ together with 1 and s has the same pattern as the pattern of 312-position chain of τ with h and $\tau(n)$. Type 3-0 NW glue sum identifies each boxed point of σ shown in Figure 5.24, and combines them into boxed points of τ from left to right. In this process, $\ell+2$ values of τ that are less than j are combined, so the amount we shift $\sigma(i) = m$ upward is $(j-1)-(\ell+2) = j-\ell-3$. Similarly, $\ell+1$ values of τ below h are combined so the amount we shift $\sigma(m-2) = m-1$ upward is $(h-1)-(\ell+1) = h-\ell-2$. Since $\tau(k)$ $(1 \le k \le \ell+2)$ are combined into corresponding values of σ, the glue sum starts referring to values of τ from $\tau(\ell+3)$. We need to shift $\tau(k)$ $(\ell+3 \le k \le n)$ upward by $m-(\ell+3)$ because this is the number of values in σ that are not merged with values of τ. As a result, type 3-0 NW glue sum creates a chain containing both a 312-value chain and a 231-position chain, as indicated in the box in Figure 5.26. Note that this chain has both a scissor s with respect to its 312-value chain aspect and a scissor $(m-2)+(h-\ell-2)$ with respect to its 231-value chain aspect.

Type 4-0 is rather similar to type 1-0, 1-1, 2-0 and 2-1. It is appropriate to visualize that $\sigma(m-1)$ is merged into either $(4, \tau(4))$ or $(5, \tau(5))$, depending on whether $\tau(4) < j$ or not. If $\tau(4) < j$, then $(m-1, m-1)$ is merged into $(4, \tau(4))$, otherwise, $(m-1, m-1)$ is merged into $(5, \tau(5))$. In addition to the combined point of (i, m) and $(1, j)$, type 4-0 also combines $(s, m-2)$ and $(2, \tau(2))$. The resulting point keeps the position s, and the value is $m-2+(j-3)$. Since three starting values of τ are merged into certain points of σ, values of τ are reserved from the fourth position instead of the third one. Note that $s = i+1$ if σ has the structure of Case 4 in Figure 5.21, and $s = i+2$ if σ has the structure of Case 5 in Figure 5.22.

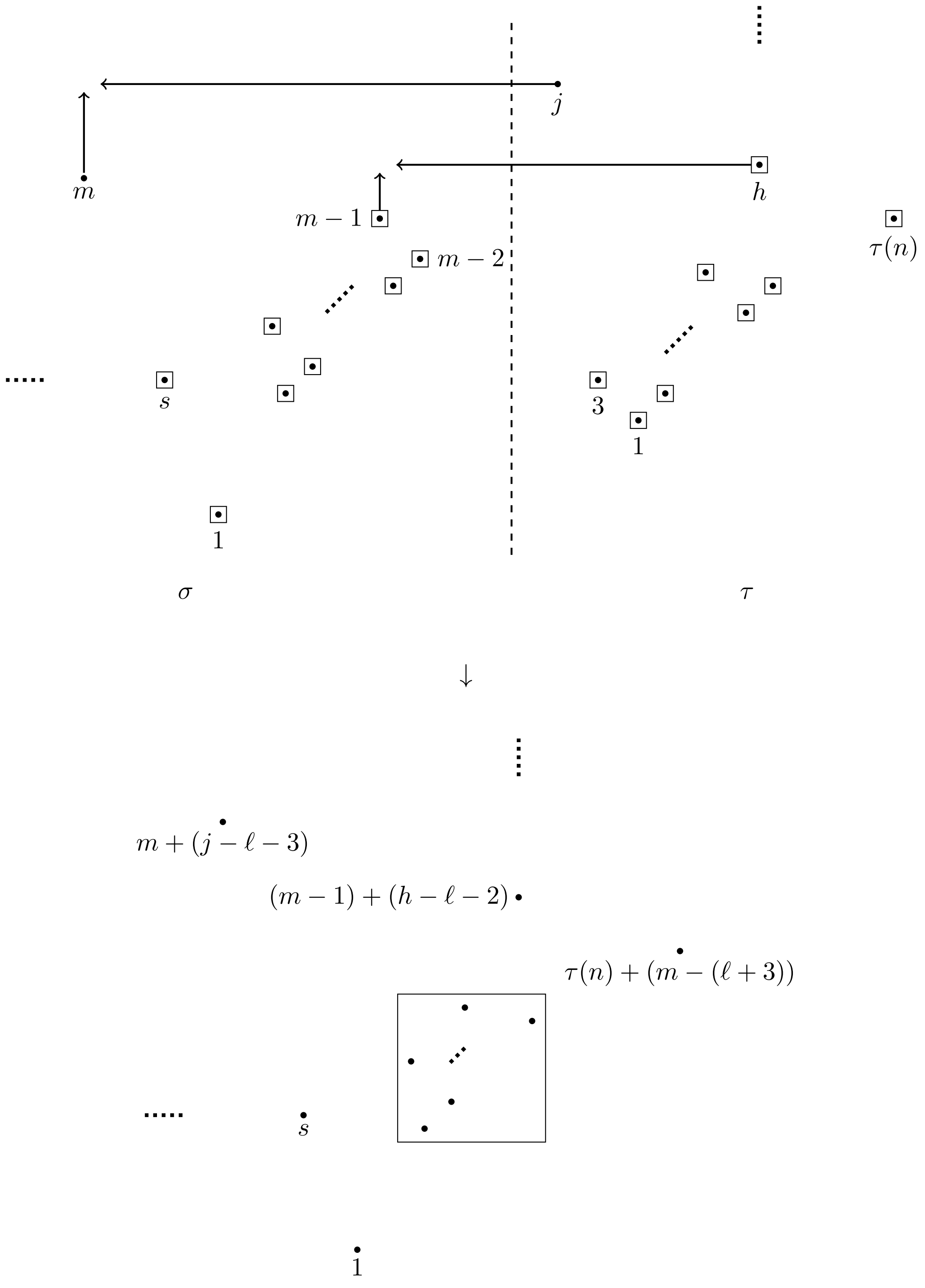


Figure 5.24.: Illustration of $\sigma \obslash^0_3 \tau$.

Type	Conditions of σ	Conditions of τ	Notation and Definition
1-0	$i \leq m-2$, $m = \sigma(m-1)$. (Case 1 of Figure 5.23)	$j \geq 3$, $2 = \tau(1)$. (Case 1 of Figure 5.20)	$\sigma \mathbin{\textcircled{$\searrow$}}_1^0 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-2)\tau'(2)\tau'(3)\cdots\tau'(n)$ where $\tau'(j) = i$ and $\tau'(k) = \tau(k) + (m-3)$ for $k \neq j$.
1-1			$\sigma \mathbin{\textcircled{$\searrow$}}_1^1 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-1)\tau'(2)\tau'(3)\cdots\tau'(n)$ where $\tau'(j) = i$ and $\tau'(k) = \tau(k) + (m-2)$ for $k \neq j$.
2-0	$i \leq m-4$, $i+2 = \sigma(m-1)$, $m = \sigma(m-2)$. (Case 2 of Figure 5.23)		$\sigma \mathbin{\textcircled{$\searrow$}}_2^0 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-3)\tau'(2)\tau'(3)\cdots\tau'(n+1)$ where $\tau'(j) = i+2$, $\tau'(j+1) = i$, and $\tau'(k) = \tau(k) + (m-3)$ for $k \leq j-1$ and $\tau'(k) = \tau(k-1) + (m-3)$ for $j+2 \leq k \leq n$.
2-1			$\sigma \mathbin{\textcircled{$\searrow$}}_2^1 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-3)\sigma(m-2)\tau'(2)\tau'(3)\cdots\tau'(n+1)$ where $\tau'(j) = i+2$, $\tau'(j+1) = i$, and $\tau'(k) = \tau(k) + (m-3)$ for $k \leq j-1$ and $\tau'(k) = \tau(k-1) + (m-3)$ for $j+2 \leq k \leq n$.
3-0	$i \leq m-5$, $m-2 = \sigma(m-1)$, $m = \sigma(m-2)$ and m is a part of the greatest 231-position chain α. (Case 3 of Figure 5.23)	$j \geq 6$, $2 = \tau(3)$, $3 = \tau(1)$ and $\tau(1)$ is a part of the first 231-value chain β. (Case 3 of Figure 5.20)	$\sigma \mathbin{\textcircled{$\searrow$}}_3^0 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-3)\tau'(\ell+1)\tau'(\ell+2)\cdots\tau'(n)$ where ℓ is the number of values in the 231-value chain β, and $\tau(j-1) = \sigma(m-1)$, $\tau(j) = i$, and $\tau'(k) = \tau(k) + (m-\ell-3)$ for k with $\ell+1 \leq k \leq n$ and $k \neq j-1, j$.
	Additionally, α must be similar to β.		
4-0	$i \leq m-4$, $m-1 = \sigma(m-1)$, $m = \sigma(m-3)$. (Case 4 or 5 of Figure 5.23)	$j \geq 5$, $2 = \tau(j-2)$, $3 = \tau(1)$. (Case 2 of Figure 5.20)	$\sigma \mathbin{\textcircled{$\searrow$}}_4^0 \tau = \sigma(1)\sigma(2)\cdots\sigma(m-4)\tau'(2)\tau'(3)\cdots\tau'(n)$ where $\tau'(j-2) = \sigma(m-2)$, $\tau'(j) = i$, and $\tau'(k) = \tau(k) + (m-3)$ for k with $2 \leq k \leq n$ and $k \neq j-2, j$.

Table 5.2.: Definitions of SE glue sums ($i = \sigma(m)$ and $j = \tau^{-1}(1)$).

As before, we define every SE glue sum precisely, so that

$$(\sigma \mathbin{\bigcirc\!\!\!\!\!\!\searrow}_x^y \tau)^{-1} = \sigma^{-1} \mathbin{\bigcirc\!\!\!\!\!\!\nwarrow}_x^y \tau^{-1}$$

for each of $(x, y) \in \{(1,0), (1,1), (2,0), (2,1), (3,0), (4,0)\}$. Thus, each type of SW glue sum is defined as in Table 5.2. Here, σ and τ are simple permutations in $\mathcal{A}'$ of length m and n respectively, $i = \sigma(m)$ and $j = \tau^{-1}$.

As we discussed in Chapter 4, type 1-0 and 1-1 glue sums are injective operations. It is not difficult to show that all other glue sums are also injective. Also, we note that all glues sums are associative only under certain conditions. In particular, both the left and the right summands must have certain lengths for each glue sum to be associative. We set the convention that when we sum multiple permutations with sequence of glue sums, we always sum from left to right.

By using all of the glue sums we defined, we describe the structure of half of the simple permutations in $\mathcal{A}'$, just as we did in Chapter 4. In particular, we find the generating function for the set $H' = \{\pi \in \mathrm{Si}(\mathcal{A}') : |\pi| \geq 4, 2 \leq \pi(1) \leq 4 \text{ and } \pi(2) \neq 1\}$. Notice that this set is the complement of the set of simple permutation π in $\mathcal{A}'$ of length 4 or more such that $\pi(1) \geq 5$ or $\pi(2) = 2$. Later, we will show that the condition of H' is equivalent to $\pi^{-1}(1) \geq 5$ or $\pi(1) = 2$. With what we observed for simple permutations of extreme pattern 2413, we can verify the set of extreme pattern 2413 simple permutations are a subset of H'. Hence, in summary, simple permutations in $\mathcal{A}'$ of length 4 or more can be classified as shown in Table 5.3.

<table>
<tr><th colspan="4">Simple permutation π in $\mathcal{A}'$ with $|\pi| \geq 4$</th></tr>
<tr><td colspan="2">$\pi^{-1}(1) \geq 5$ or $\pi(1) = 2$</td><td colspan="2">$\pi(1) \geq 5$ or $\pi(2) = 1$</td></tr>
<tr><td>Extreme pattern 2413</td><td colspan="2">Extreme pattern 2143</td><td>Extreme pattern 3142</td></tr>
</table>

Table 5.3.: Classification of simple permutations in $\mathcal{A}'$.

The theorem describing the structure of simple permutations in H' is the following.

Theorem 5.21 *Let π be a permutation in H'. Then there uniquely exist simple permutations σ_i (i odd) in $\mathcal{A}'$ of extreme pattern 2413 and simple permutations τ_i (i even) in $\mathcal{A}'$ of extreme* *pattern 3142 such that*

$$\pi = \begin{cases} \sigma_1 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_1}^{y_1} \tau_2 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_2}^{y_2} \sigma_3 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_3}^{y_3} \tau_4 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_4}^{y_4} \cdots \mathbin{\bigcirc\mkern-15mu\searrow}_{x_{m-1}}^{y_{m-1}} \sigma_m & \textit{if } m \textit{ is odd} \qquad (a) \\ \sigma_1 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_1}^{y_1} \tau_2 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_2}^{y_2} \sigma_3 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_3}^{y_3} \tau_4 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_4}^{y_4} \cdots \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_{m-1}}^{y_{m-1}} \tau_m & \textit{if } m \textit{ is even} \qquad (b) \end{cases} \tag{5.6}$$

where m is a positive integer and $(x_\ell, y_\ell) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$ ($1 \leq \ell \leq m-1$). Moreover, every permutation written as Equation 5.1(a) or 5.1(b) is in H'.

5.2.2. Proof of Theorem 5.21 (Part 1)

As we did in Chapter 4, we break the proof of Theorem 5.21 into two propositions and prove each one separately.

Proposition 5.22 *If π is a simple permutation in H', then there uniquely exist simple permutations in $\mathcal{A}'$ of extreme pattern 2413 σ_i (i odd) and simple permutations in $\mathcal{A}'$ of extreme pattern 3142 τ_i (i even) such that*

$$\pi = \begin{cases} \sigma_1 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_1}^{y_1} \tau_2 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_2}^{y_2} \sigma_3 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_3}^{y_3} \tau_4 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_4}^{y_4} \cdots \mathbin{\bigcirc\mkern-15mu\searrow}_{x_{m-1}}^{y_{m-1}} \sigma_m & \textit{if } m \textit{ is odd} \\ \sigma_1 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_1}^{y_1} \tau_2 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_2}^{y_2} \sigma_3 \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_3}^{y_3} \tau_4 \mathbin{\bigcirc\mkern-15mu\searrow}_{x_4}^{y_4} \cdots \mathbin{\bigcirc\mkern-15mu\nwarrow}_{x_{m-1}}^{y_{m-1}} \tau_m & \textit{if } m \textit{ is even} \end{cases}$$

where m is a positive integer and $(x_\ell, y_\ell) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$ ($1 \leq \ell \leq m-1$).

Proof. Note that uniqueness of this decomposition follows from injectivity of the glue sums.

Suppose π is a simple permutation in $\mathcal{A}'$ with $|\pi| \geq 4$ which satisfies either $\pi^{-1}(1) \geq 5$ or $\pi(1) = 2$. We define a sequence of values of π, $d_1, \ldots, d_{m+3}$ in the exact same way as we did in Chapter 4. Therefore, $d_1 = \pi(1)$ and

$$d_i = \begin{cases} \pi(\max\{s : \pi(s) < \pi(d_{i-1})\}) & \text{if } i \text{ is even} \\ \max\{t : \pi^{-1}(t) < \pi^{-1}(d_{i-1})\} & \text{if } i \text{ is odd} \end{cases}$$

for i with $1 \leq i \leq m+3$. Thus, d_i is the right-most value that is less than d_{i-1} if i is even or the greatest value located to the left of d_{i-1} if i is odd. As before, we have $d_{m+2} = \pi(n)$ and $d_{m+3} = n$ for some even integer m or $d_{m+2} = n$ and $d_{m+3} = \pi(n)$ for some odd integer m.

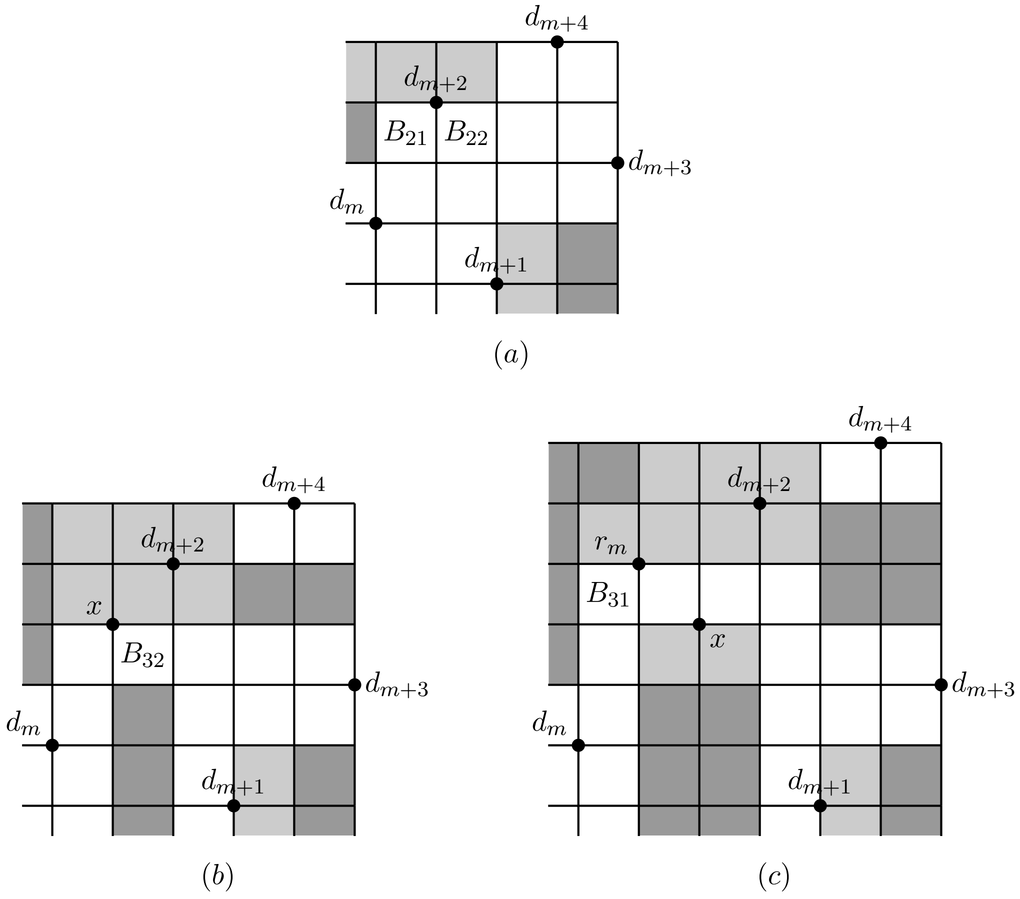
We omit the proof of $d_i \neq d_j$ for $i \neq j$, since it is exactly the same as we did in Chapter 4. Since $d_2 = 1$, we know $d_2 \neq \pi(n)$, so every permutation in H has at least four values denoted by d_i for some $i \geq 1$. As before, we use mathematical induction on $m \geq 1$ to show that π with

$m+1$ values denoted by d_i satisfies either Equation $5.6(a)$ or $5.6(b)$.

The base case is identical to the one in Chapter 4. That is, if $m = 1$, then there are four values d_1, d_2, d_3 and d_4 where $d_4 = \pi(n)$. This implies π is a simple permutation of extreme pattern 2413, so we are done.

Suppose every $\pi \in H'$ with $m+3$ values denoted by d_i $(1 \leq i \leq m+3)$ satisfies Equation $5.6(a)$ for some positive odd integer m. We show that a permutation $\pi \in H'$ of length n with $m+4$ values d_i $(1 \leq i \leq m+4)$ satisfies Equation $5.6(b)$. Let π be a permutation in H' of length n with $m+4$ values denoted by d_i. Thus, $d_{m+4} = n$.

Before we define the values p_m, q_m and r_m, we claim that there is at most one value r'_m besides d_{m+2} such that $r'_m > d_{m+3}$ and $\pi^{-1}(r'_m) < \pi^{-1}(d_{m+1})$. Moreover, if such a point r'_m exists, then the value r'_m must satisfy $r'_m = d_{m+2} - 1$ or $r'_m = d_{m+2} - 2$, and the position $\pi^{-1}(r'_m)$



(a)

(b)

(c)

Figure 5.25.: Partial graphs of π to show that there exists no value in B_{21}.

must satisfy $\pi^{-1}(r'_m) = \pi^{-1}(d_{m+2}) + 1$ or $\pi^{-1}(r'_m) = \pi^{-1}(d_{m+2}) + 2$. So let us start from the graph shown in Figure 5.25(a). We first show that we cannot have a point in the region B_{21}.

As usual, proceed by assuming there is a value in B_{21}, and let x be the greatest such value, so we obtain the graph in Figure 5.25(b). In order to split the block $[\pi^{-1}(x), \pi^{-1}(d_{m+2})]$, we must have a point in the region B_{32}, so let y be the least value there. We achieve the graph shown in Figure 5.25(c), requiring to having a point in either B_{31} or B_{34}. However, having a point in either region implies the infinite chain contradiction that we have observed in previous proofs, such as the one for Lemma 5.3. Therefore, having a point in the region B_{21} of Figure 5.25(a) is prohibited, and we must have the structure of Figure 5.26(a).

Now, let r'_m be the point with the greatest value in the region B_{22} of Figure 5.26(a), so we have the graph as in Figure 5.26(b). Suppose to the contrary there is another point with the value

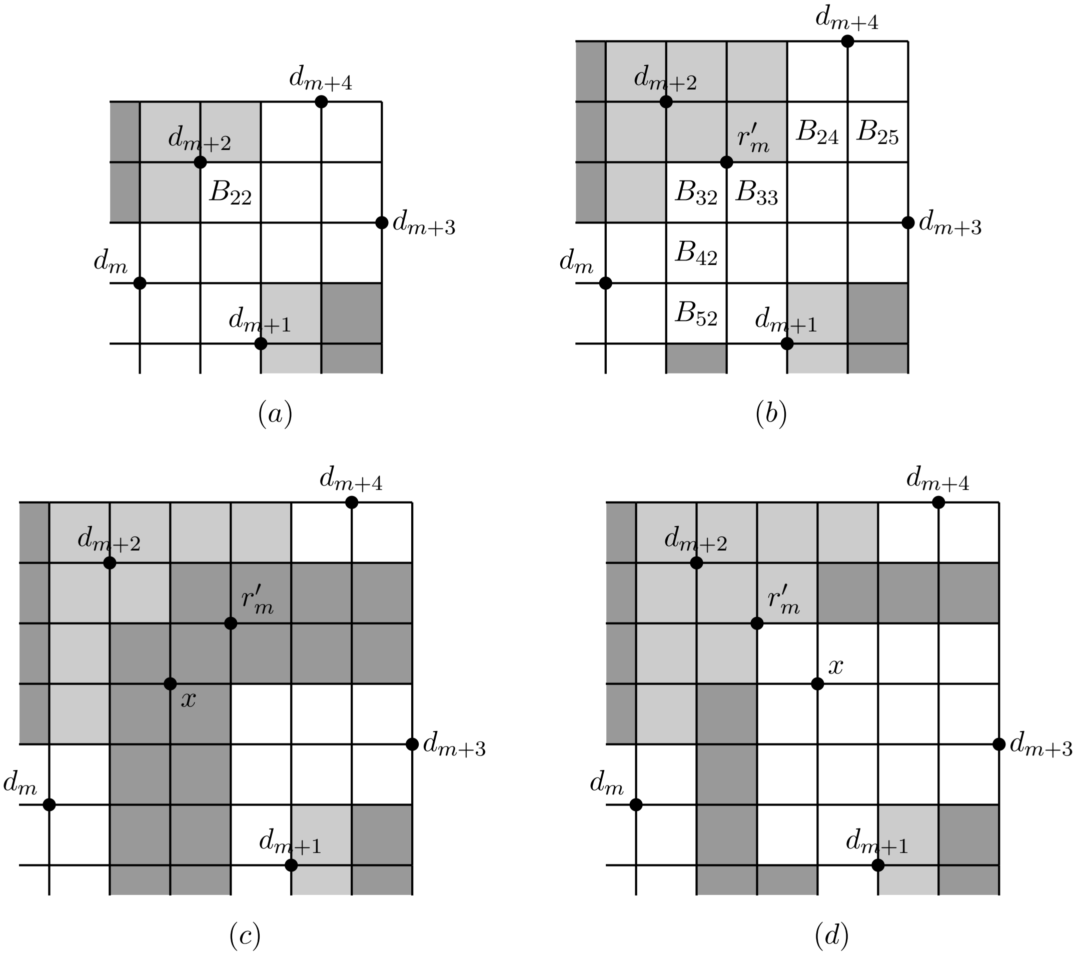


Figure 5.26.: Partial graphs of π to show the possible existence of r'_m.

x such that $x > d_{m+3}$ and $\pi^{-1}(x) < \pi^{-1}(d_{m+1})$. If x is in B_{32} of Figure 5.26(b), then we have the graph shown in Figure 5.26(c) as a result, which has an unsplittable block $[\pi^{-1}(d_{m+2}, \pi^{-1}(r'_m)]$. So we prohibit the region B_{32} of Figure 5.26(b), and assume x is in B_{33} instead. Then we obtain the graph as in Figure 5.26(d), which also has a block $[\pi^{-1}(d_{m+2}, \pi^{-1}(r'_m)]$ that cannot be split. Hence, we can only have one point, besides d_{m+2} that has a value greater than d_{m+3} and is located to the left of d_{m+1}.

When we have r'_m as shown in Figure 5.26(b), having at least one point in either B_{24}, B_{25}, B_{42} or B_{52} is necessary to ensure π is simple. However, due to 52341, 53241 and 52431 patterns avoidance, we can have only one point whose position is in $[\pi^{-1}(d_{m+2}, \pi^{-1}(r'_m)]$ and only one point whose value is in $[r'_m, d_{m+2}]$. It is possible to have both of these two points. Thus, there are three cases, which are

1. $\pi^{-1}(r'_m) = \pi^{-1}(d_{m+2}) + 1$ and $r'_m = d_{m+2} - 2$,
2. $\pi^{-1}(r'_m) = \pi^{-1}(d_{m+2}) + 2$ and $r'_m = d_{m+2} - 1$ and
3. $\pi^{-1}(r'_m) = \pi^{-1}(d_{m+2}) + 2$ and $r'_m = d_{m+2} - 2$.

We are now ready to define the values p_m, q_m and r_m. We define p_m in the same way as before, that is,

$$p_m = \pi(\min\{\pi^{-1}(s) : s > d_{m+3}, \pi^{-1}(s) > \pi^{-1}(d_{m+1})\}).$$

The definition of q_m is much more complex than the one of Chapter 4. We define q_m by the following algorithm called TRACE.

Algorithm TRACE
INPUT: A simple permutation π in H'.
OUTPUT: Value q_m. In addition, value q'_m for Step 3a.

```
Step 1: Define the value $T_1 = u_m$ and $T_2$.
        Let $T_1 = u_m = \pi(\pi^{-1}(p_m) + 1)$, i.e. the value immediately to the right of $p_m$. Let $T_2 = u_m + 1$.
Step 2: Determine if the algorithm continues.
        If $T_1 < p_m < d_{m+2}$ and $\pi^{-1}(T_2) < \pi^{-1}(T_1)$, then GOTO STEP 3. Otherwise, OUTPUT $q_m = \pi(\pi^{-1}(p_m) - 1)$.
Step 3: Find the beginning of the 312-position chain.
        WHILE $\pi(\pi^{-1}(T_2) + 1) < T_1 - 1$ AND $\pi^{-1}(T_2) > \pi^{-1}(d_{m+1})$:
        Newly define $T_1 = \pi(\pi^{-1}(T_2) + 1)$ and $T_2 = T_1 + 1$.
```

```
Step 4: Determine which condition of Loop was violated.
        BEGIN
          a. If $\pi(\pi^{-1}(T_2)+1) \geq T_1 - 1$, then OUTPUT $q_m = \pi(\pi^{-1}(T_2)-1)$.
          b. If $\pi^{-1}(T_2) < \pi^{-1}(d_{m+1})$, then OUTPUT $q_m = u_m$ and $q'_m = T_2$.
```

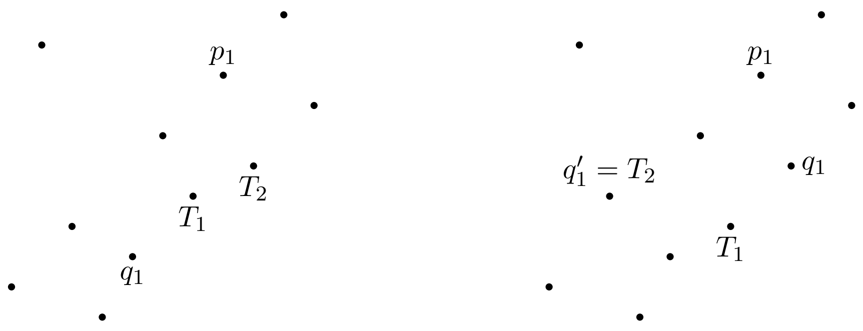
Finally, we let $r_m = \max\{\pi(s) < d_{m+3} : s \in [1, \pi^{-1}(q_m)]\}$. Hence, r_m is the greatest value less than d_{m+3}, whose position is in $[1, \pi^{-1}(q_m)]$.

Before we move on, we explain what the roles of p_m, q_m, q'_m and r_m are. For each NW glue sum, the value q_m is the last value of the left hand side of $(\sigma_1 \oslash^{y_1}_{x_1} \tau_2 \oslash^{y_2}_{x_2} \cdots \oslash^{y_{m-1}}_{x_{m-1}} \sigma_m) \oslash^{y_m}_{x_m} \tau_{m+1}$. Hence, in Table 5.1, q_m is $\sigma'(m-1)$ for $\oslash^0_1$, $\oslash^0_2$ and $\oslash^0_3$, $\sigma'(m)$ for $\oslash^1_1$ and $\oslash^1_2$, and $\sigma'(m-2)$ for $\oslash^0_4$. Informally, we may consider that q_m is where the left permutation and the right permutation are connected. The value p_m is merely used to locate q_m. Except for the case of type 3-0 and some special cases of type 1-0 and type 1-1, q_m is immediately to the left of p_m, so p_m can be viewed as the first value to be glued. The other exceptional cases are explained with more details next. The value r_m is to distinguish the type 1-1 and 2-1 from the type 1-0 and 2-0 respectively.

For a permutation in H', the right-most summand of its glue sum decomposition can either begin with a value greater d_{m+3} or a 312-position chain. The loop in TRACE finds the beginning of this chain. The two exit conditions of this loop correspond to the cases where the 312-position chain ends on its own, or with a scissor on its left, as in the following examples. Suppose we have $\mu_1 = 264135$, $\mu_2 = 2\ 10\ 5\ 1\ 3\ 7\ 4\ 9\ 6\ 8$, $\nu_1 = 71426385$ and $\nu_2 = 831527496$. Notice that these are simple permutations in $\mathcal{A}'$, and furthermore, μ_1 and μ_2 are of extreme pattern 2413 and ν_1 and ν_2 are of extreme pattern 3142. We obtain

$$\mu_1 \oslash^0_1 \nu_1 = 2\ 10\ 4\ 1\ 3\ 7\ 5\ 9\ 6\ 11\ 8 \qquad \text{and} \qquad \mu_2 \oslash^0_3 \nu_2 = 2\ 10\ 5\ 1\ 3\ 7\ 4\ 9\ 6\ 11\ 8$$

The graphs of $\mu_1 \oslash^0_1 \nu_1$ and $\mu_2 \oslash^0_3 \nu_2$ are shown in Figure 5.27. What we have in common in these two sums is that the value corresponding to p_1 is a scissor of a 312-position chain before the sum. In case of $\mu_1 \oslash^0_1 \nu_1$, the values in $[1,5)$ of ν_1 are of the form $1 \oplus 312$. Since 312 alone is a 312-position chain, we have the value 6 as the scissor, and this value corresponds to p_1 in $\mu_1 \oslash^0_1 \nu_1$. When the right summand does not begin with a 312-position chain, the value immediately to the left of p_m is q_m, but when it does as in this example, trace follows this chain to find $q_1 = m'_1(5) = 3$ in Step 4a.



$\mu_1 \obslash_1^0 \nu_1$ $\mu_2 \obslash_3^0 \nu_2$

Figure 5.27.: Graphs of $\mu_1 \obslash_1^0 \nu_1$ and $\mu_2 \obslash_3^0 \nu_2$.

On the other hand, the values in $[1, 6)$ of ν_2 are of the form $\alpha = 312 \oplus^1 21$, a 312-position chain. The value 7 corresponding to p_1 in $\mu_2 \obslash_3^0 \nu_2$ is still a scissor, so we again find the beginning of the 312-position chain using TRACE. However, in this case, we end with T_2 to the left of $1 = d_2 = d_{m+1}$, which means the 312-position chain also has a scissor on its left. We define this scissor to be q_1', and the value immediately to the right of p_1 to be q_1, according to Step 4b. Hence, the value 4 corresponding to q_1' is the scissor of 312-value chain of μ_2 that is similar to the 312-position chain of ν_2. Thus, this whole procedure is done to determine whether two permutations are glued by type 1-0 (or type 1-1) with the values of the right summand starting with $1 \oplus \alpha$ where α is a 312-position chain or they are glued by type 3-0.

In summary, p_m is used to located q_m, q_m is the last value of the left summand, q_m' is the scissor of the last 312-value chain in the left summand, and r_m is used to distinguish the type 1-1 and 2-1 from the type 1-0 and 2-0 respectively.

From here, we divide into four cases based on the existence of r_m', the value greater than d_{m+3} that is located to the left of d_{m+1}, and how q_m is determined by TRACE. These four cases are namely the following, and they correspond to the type of the glue sum used.

Case A: There does not exist r_m' and q_m is output by STEP 2 or STEP 4a.

Case B: There exists $r_m' = d_{m+2} - 1$.

Case C: There does not exist r_m' and q_m is output by STEP 4b.

Case D: There exists $r_m' = d_{m+2} - 2$.

In each case, we slice π into π_1 and π_2 according to the position of q_m. Then we show π_1 is a permutation in H' with $m + 3$ values denoted by d_i, and π_2 is a simple permutation in $\mathcal{A}'$

of extreme pattern 3142. Then by using the appropriate NW glue sum, we show $\pi = \pi_1 \otimes_x^y \pi_2$ $((x,y) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\})$ to complete the proof.

Case A: Denote by π_2 the flattening of the subsequence of π obtained by removing every value whose position is in $[1, \pi^{-1}(q_m)]$, expect d_{m+2} and d_{m+1}. Showing π_2 is a simple permutation in $\mathcal{A}'$ of extreme pattern 3142 is exactly the same as we did in Chapter 4, so we omit it.

Next, let π_1 be the flattening of the subsequence $\pi(1)\pi(2)\cdots\pi(\pi^{-1}(q_m)-1)d_{m+3}$ of π if $q_m = r_m$, and $\pi(1)\pi(2)\cdots q_m d_{m+3}$ of π if $q_m \neq r_m$. We claim that π_1 is simple.

As we did in Chapter 4, we first show that every value we removed to construct π_1 is greater than or equal to r_m. Suppose $q_m = r_m$. Then the positions of values that are removed are in $[\pi^{-1}(q_m), \pi^{-1}(d_{m+3})-1]$. Assume to the contrary that there exists $z < r_m$ such that $\pi^{-1}(z) \in [\pi^{-1}(q_m), \pi^{-1}(d_{m+3})-1]$. Note that $z \neq p_m$ since $p_m > r_m$. If $p_m > d_{m+2}$, then $d_{m+2}r_m p_m z d_{m+3}$ forms a 42513 pattern. So suppose $p_m < d_{m+2}$. Let w be the value whose position is immediately to the right of the position of p_m. For now, assume $w > r_m$. Now, if $w > d_{m+2}$, then π contains a 42513 pattern with $d_{m+2}r_m w z d_{m+3}$. On the other hand, if $w < d_{m+2}$, then π contains either a 52341 pattern or a 52431 pattern with $d_{m+2}q_m p_m w z$. Thus, we have $w < r_m$, which is impossible since $d_{m+2}q_m(w+1)p_m w$ forms a 53241 pattern.

Now, assume $q_m \neq r_m$. Then the positions of values that are removed are in $[\pi^{-1}(p_m), \pi^{-1}(d_{m+3})-1]$. Suppose there exists $z < r_m$ such that $\pi^{-1}(z) \in [\pi^{-1}(p_m), \pi^{-1}(d_{m+3})-1]$. Again, $z \neq p_m$, so we must have $\pi^{-1}(z) > \pi^{-1}(p_m)$. If we assume $\pi^{-1}(r_m) > \pi^{-1}(d_{m+2})$ and $p_m < d_{m+3}$, we achieve a similar contradiction as before. Also, the rest of the proof is identical to the argument in Chapter 4. That is, if $\pi^{-1}(r_m) > \pi^{-1}(d_{m+2})$ and $p_m > d_{m+2}$, then π contains 42513 pattern with $d_{m+2}r_m p_m z d_{m+3}$. If $\pi^{-1}(r_m) < \pi^{-1}(d_{m+2})$ and $p_m < d_{m+2}$, then π contains 35142 pattern with $r_m d_{m+2} d_{m+1} p_m z$, and finally, if $\pi^{-1}(r_m) < \pi^{-1}(d_{m+2})$ and $p_m > d_{m+2}$, then π contains 351624 pattern with $r_m d_{m+2} d_{m+1} p_m z d_{m+3}$.

Hence, every value we removed to construct π_1 must be greater than or equal to r_m. With this, we can show π_1 is simple in the exact same way as in Chapter 4. Therefore, $\pi_1 \in H'$ with $m+3$ values denoted by d_i. Moreover, by the way we constructed π_1 and π_2, π_1 and π_2 respectively have structures of Case 1 in Figure 5.21 and Case 1 in Figure 5.22. If $q_m = r_m$, then $\pi = \pi_1 \otimes_1^1 \pi_2$. Otherwise, $\pi = \pi_1 \otimes_1^0 \pi_2$, so we are done with Case A.

Case B: As before, denote by π_2 the flattening of the subsequence of π obtained by removing every value whose position is in $[1, \pi^{-1}(q_m)]$, except d_{m+2} and d_{m+1}. We can show π_2 is a simple permutation in $\mathcal{A}'$ of extreme pattern 3142 in the same way as Chapter 4. We also define π_1 in the same way as Case A, so it is the flattening of the subsequence $\pi(1)\pi(2)\cdots\pi(\pi^{-1}(q_m)-1)d_{m+3}$ of π if $q_m = r_m$, and $\pi(1)\pi(2)\cdots q_m d_{m+3}$ of π if $q_m \neq r_m$. Showing every value we removed to construct π_1 is greater than or equal to r_m is identical to Case A.

Suppose to the contrary π_1 is not simple. Let us use the hat notation as we did in Chapter 4 to refer to the value of π_1 corresponding to a value of π. Let I be a proper non-singleton block of π_1. We have three cases to consider. First, if $\pi_1(I) = [\hat{x}, \hat{y}]$ for some values $\hat{x}$ and $\hat{y}$ where $\hat{y} \leq \hat{d}_{m+3}$, we use the same argument as Chapter 4. Second, suppose $\pi_1(I) = [\hat{x}, \hat{r}'_m]$. Because $r_m < d_{m+3} < r'_m$, we have $\hat{d}_{m+3} = \hat{r}'_m - 1$. Thus, $\hat{d}_{m+3} \in [\hat{x}, \hat{r}'_m]$. Since $\pi_1 - 1(\hat{r}'_m) < \pi_1^{-1}(\hat{d}_{m+1}) < \pi_1^{-1}(\hat{d}_{m+3})$, we have $\hat{d}_{m+1} \in [\hat{x}, \hat{r}'_m]$. We can then argue that $\hat{d}_m$, and hence $\hat{d}_{m+2}$, are also in $[\hat{x}, \hat{r}'_m]$, which is a contradiction, since $\hat{d}_{m+2} > \hat{r}'_m$. Finally, if $\pi_1(I) = [\hat{x}, \hat{d}_{m+2}]$, then we use the same argument as Chapter 4. (It is impossible to have $\hat{x} = \hat{r}'_m$ because the point in the region B_{42} or B_{52} as shown in Figure 5.19(b) is included in π_1.) Consequently, π_1 is simple.

With $r'_m = \hat{d}_{m+2} - 1$ in π_1, the structure of π_1 is Case 2 in Figure 5.22, whereas π_2 still has the structure of Case 2 in Figure 5.23. By definitions of $\textcircled{\nwarrow}_2^0$ and $\textcircled{\nwarrow}_2^1$, $\pi = \pi_1 \textcircled{\nwarrow}_2^0 \pi_2$ if $q_m = r_m$, and $\pi = \pi_1 \textcircled{\nwarrow}_2^1 \pi_2$ otherwise.

Case C: This time, we let π_2 be the flattening of the subsequence of π obtained by removing every value whose position is in $[1, \pi^{-1}(q'_m - 1) - 2]$, except d_{m+2}, q'_m and d_{m+1}. We show π_2 is a simple permutation in $\mathcal{A}'$ of extreme pattern 3142.

Suppose π_2 is not simple, so there exists a proper non-singleton block I. If $I = [\pi_2^{-1}(\hat{x}), \pi_2^{-1}(\hat{y})]$ for some values $\hat{x}$ and $\hat{y}$ where $\pi_2^{-1}(\hat{x}) > \pi_2^{-1}(\hat{d}_{m+1})$, then $[\pi^{-1}(\hat{x}), \pi^{-1}(\hat{y})]$ is also a block in π. Next, suppose $I = [\pi_2^{-1}(\hat{d}_{m+1}), \pi_2^{-1}(\hat{y})]$ for some $\hat{y}$. By the way q'_m is defined, the value immediately to the right of the value $\hat{d}_{m+1}$ must be greater than $\hat{q}'_m$. Hence, $\pi_2^{-1}(\hat{q}'_m) \in I$, so we have a contradiction. So suppose $I = [\pi_2^{-1}(\hat{q}'_m), \pi_2^{-1}(\hat{y})]$ for some $\hat{y}$. Let $T'_1 = q'_m - 1$ and $T'_2 = \pi_2(\pi_2^{-1}(T'_1) - 1)$. Then $\pi_2^{-1}(T'_1), \pi_2^{-1}(T'_2) \in I$. Next, if $T'_2 \neq \hat{p}_m$, we let $T'_3 = T'_2 - 1$ and $T'_4 = \pi_2(\pi_2^{-1}(T'_3) - 1)$, so we have $\pi_2^{-1}(T'_3), \pi_2^{-1}(T'_4) \in I$. So long as T'_i (i even) is not equal to $\hat{p}_m$, we continue definition T'_i in the same way, so

$$T'_i = \begin{cases} T'_{i-1} - 1 & \text{if } i \text{ is odd} \\ \pi_2(\pi_2^{-1}(T'_{i-1}) - 1) & \text{if } i \text{ is even} \end{cases}$$

The way q'_m is defined guarantees that there exists an even integer j such that $T'_j = \hat{p}_m$ (and thus, $T'_{j-1} = \hat{q}_m$), so $\pi_2^{-1}(\hat{p}_m) \in [\pi_2^{-1}(\hat{q}'_m), \pi_2^{-1}(\hat{y})]$. Since $\hat{p}_m > \hat{d}_{m+3}$, $\pi_2^{-1}(\hat{d}_{m+3}) \in I$, how-

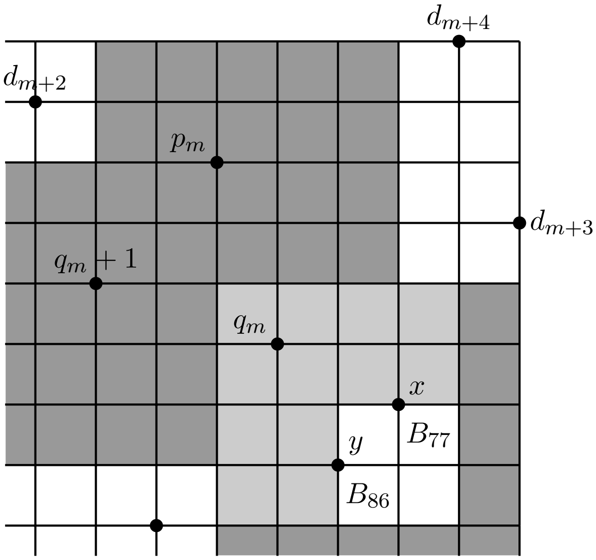


Figure 5.28.: Partial graph of π with the assumption of $\pi^{-1}(x) > \pi^{-1}(q_m)$.

ever, this implies $\pi_2^{-1}(\hat{d}_{m+4}) \in I$, and therefore, $\pi_2^{-1}(\hat{d}_{m+2}) \in I$ as well. Since $\pi_2^{-1}(\hat{d}_{m+2}) < \pi_2^{-1}(\hat{q}'_m)$, we have a contradiction. Lastly, assume $I = [\pi_2^{-1}(\hat{d}_{m+2}), \pi_2^{-1}(\hat{y})]$ for some $\hat{y}$. Because $\pi_2^{-1}(\hat{q}'_m) = \pi_2^{-1}(\hat{d}_{m+2}) + 1$, we must have $\pi_2^{-1}(\hat{q}'_m) \in I$. Since, $\hat{q}'_m < \hat{d}_{m+3}$, we obtain $I = [\pi_2^{-1}(\hat{q}'_m), \pi_2^{-1}(\hat{d}_{m+3})]$, but then I is all of π.

In every case, we achieve a contradiction, so π_2 is simple. In particular, π_2 has extreme pattern 3142 and has the structure of Case 3 in Figure 5.22. By Proposition 5.9, $\hat{q}'_m$ and $\hat{d}_{m+1}$ are involved in the same 312-position chain. In fact, by Proposition 5.16, every value whose position is in $[\pi_2^{-1}(\hat{q}'_m), \pi_2^{-1}(\hat{q}_m)]$, except $\hat{p}_m$ together forms this 312-position chain.

Next, let π_1 be the flattening of $\pi(1)\pi(2)\cdots q_m d_{m+3}$. We first show that every value z removed to construct π_1 is greater than q_m. In order to show this, we first claim that the value $x = q_m - 1$ is located to the left of q_m. Suppose to the contrary that $\pi^{-1}(x) > \pi^{-1}(q_m)$. We divide into two cases. Let y be the value immediately to the right of q_m. Suppose $y > q_m$. Then π contains 42513 due to $d_{m+2}q_m y x d_{m+3}$ if $d_{m+2} < y$, or 53241 due to $d_{m+2}(q_m + 1)q_m y x$ if $y < d_{m+2}$. So assume $y < q_m$. Note that $x \neq y$ because $x = y$ implies that π contains an unsplittable block $[\pi^{-1}(q_m), \pi^{-1}(x)]$. Hence, we obtain the graph shown in Figure 5.28. Splitting the block $[\pi^{-1}(q_m), \pi^{-1}(y)]$ can be only done by having a point in B_{77} or B_{86}. In either case, we obtain an unsplittable block, so we achieve a contradiction. Thus, we must have $\pi^{-1}(x) < \pi^{-1}(q_m)$.

We now show that every value located to the right of q_m is greater than q_m. Suppose to the contrary that there exists a value $z < q_m$ such that $\pi^{-1}(z) > \pi^{-1}(q_m)$. If $\pi^{-1}(x) < \pi^{-1}(d_{m+2})$, then π contains 35142 due to $xd_{m+2}d_{m+1}p_mz$. If $\pi^{-1}(d_{m+2}) < \pi^{-1}(x) < \pi^{-1}(q_m+1)$, then π contains 52341 due to $d_{m+2}x(q_m+1)p_mz$. Finally, if $\pi^{-1}(q_m+1) < \pi^{-1}(x)$, then π contains 53241 due to $d_{m+2}(q_m+1)xp_mz$. Consequently, every value we removed to construct π_1 is greater than q_m.

Showing π_1 is simple is essentially the argument of inverse followed by reverse complement symmetry of π_2. Suppose π_1 is not simple, so there exists a proper non-singleton block I of π_1. If $\pi_1(I) = [\hat{x}, \hat{y}]$ where $\hat{x} < \hat{y} \leq \hat{d}_{m+3} - 1$, then I would be also a block in π. If $\pi_1(I) = [\hat{x}, \hat{d}_{m+3}]$, then $\hat{d}_{m+3} - 1 = \hat{q}_m + 1$ must be in $\pi_1(I)$. However, since $\pi_1^{-1}(\hat{q}_m+1) < \pi_1^{-1}(\hat{p}_m) < \pi_1^{-1}(\hat{d}_{m+3})$, $\hat{p}_m$ also has to be in $\pi_i(I)$. Because $\hat{p}_m > \hat{d}_{m+3}$, we achieve a contradiction. So assume $\pi_1(I) = [\hat{x}, \hat{p}_m]$. Due to the way q'_m in π was assigned, $\hat{q}'_m$ must be in $\pi_1(I)$. However, this implies $\hat{d}_{m+1}$ is also in $\pi_1(I)$, causing $\hat{d}_m \in \pi_1(I)$, and therefore, $\hat{d}_{m+2} \in \pi_1(I)$. Since $\hat{d}_{m+2} > \hat{p}_m$, this is a contradiction. Finally, suppose $\pi_1(I) = [\hat{x}, \hat{d}_{m+2}]$. Then $\hat{p}_m \in \pi_1(I)$, which results in $\hat{x} = \hat{d}_{m+1}$. Since I is all of π_1, we again have a contradiction.

Therefore, π_1 is a simple permutation in H' with $m+3$ values denoted by d_i. Note that the points π_1 and π_2 both contain as results of removal are d_{m+2}, q'_m, d_{m+1}, every value whose position is in $[\pi^{-1}(q'_m - 1) - 1, \pi^{-1}(q_m)]$ (including p_m), and d_{m+3}. Hence, π_1 ends with a 312-value chain α involving q_m, p_m and d_{m+3}. Also, π_2 begins with a 312-position chain β involving q'_m, d_{m+1} and $q'_m - 1$. Then $21 \oplus_1 \alpha = \beta \oplus^1 21$, so α and β are similar. Since π has the structure of Case 3 in Figure 5.21, we have $\pi = \pi_1 \mathbin{\bigcirc\!\!\!\!\!\nwarrow}_3^0 \pi_2$, so we are done with Case C.

Case D: Finally, let π_2 be the flattening of the subsequence of π obtained by removing every value whose position is in $[1, \pi^{-1}(q_m)]$, expect d_{m+2}, r'_m and d_{m+1}. We start by showing π_2 is simple. Assume it is not simple. Let I be a proper non-singleton block of π_2. As usual, if $I = [\pi_2^{-1}(x), \pi_2^{-1}(y)]$ for some values $\hat{x}$ and $\hat{y}$ where $\pi_2^{-1}(x) > \pi_2^{-1}(\hat{d}_{m+1})$, then I would be a block in π as well. Next, suppose $I = [\pi_2^{-1}(\hat{d}_{m+1}), \pi_2^{-1}(\hat{y})]$. Then $\pi_2^{-1}(\hat{p}_m) \in I$, resulting in $\pi_2^{-1}(\hat{d}_{m+3}) \in I$. Hence, $\pi_2^{-1}(\hat{d}_{m+4})$ is in I, and therefore, so is $\pi_2^{-1}(\hat{d}_{m+2})$. Since $\pi_2^{-1}(\hat{d}_{m+2}) < \pi_2^{-1}(\hat{d}_{m+1})$, we have a contradiction. This time, let $I = [\pi_2^{-1}(\hat{r}'_m), \pi_2^{-1}(\hat{y})]$. Since $\pi_2^{-1}(\hat{r}'_m) + 1 = \pi_2^{-1}(\hat{d}_{m+1})$, we know $\pi_2^{-1}(\hat{d}_{m+1}) \in I$. The rest is the same as the previous case, $I = [\pi_2^{-1}(\hat{d}_{m+1}), \pi_2^{-1}(\hat{y})]$. We achieve a contradiction of $\pi_2^{-1}(\hat{d}_{m+2}) \in I$. Lastly, suppose $I = [\pi_2^{-1}(\hat{d}_{m+2}), \pi_2^{-1}(\hat{y})]$. Then $\pi_2^{-1}(\hat{r}'_m)$ is in I. Since $\hat{r}'_m < \hat{p}_m < \hat{d}_{m+2}$, we know $\pi_2^{-1}(\hat{p}_m) \in I$, implying $\pi_2^{-1}(\hat{d}_{m+1}) \in I$. Hence, $\pi_2^{-1}(\hat{d}_{m+3})$ is also in I, but then now I is all of π_2. Altogether, π_2 is simple.

Next, let π_1 be the flattening of $\pi(1)\pi(2)\cdots q_m p'_m d_{m+3}$ where $p'_m = d_{m+2} - 1$. Suppose to the contrary that there exists a value $z < r_m$ located to the right of q_m. If $\pi^{-1}(z) < \pi^{-1}(p'_m)$, then $p'_m \neq p_m$, and $r'_m p_m z p'_m d_{m+3}$ forms a 35142 pattern. So assume $\pi^{-1}(z) > \pi^{-1}(p'_m)$. Then we have a 53241, 52341 or 35142 pattern depending the position of r_m. So every value we removed to construct π_1 must be greater than r_m. Now, we prove π_1 is simple. Assume it is not simple, so we have a proper non-singleton block I. If $\pi_1(I) = [\hat{x}, \hat{y}]$ for some $\hat{x}, \hat{y}$ with $\hat{x} < \hat{y} < \hat{d}_{m+3}$, then I would be a block in π. Suppose $\pi_1(I) = [\hat{x}, \hat{d}_{m+3}]$. Since $\pi_1^{-1}(\hat{p}'_m) = \pi_1^{-1}(\hat{d}_{m+3}) - 1$, we must have $\hat{p}'_m \in \pi_1(I)$. Since $\hat{p}'_m > \hat{d}_{m+3}$, we have a contradiction. So assume $\pi_1(I) = [\hat{x}, \hat{r}'_m]$. Since $\hat{d}_{m+3} = \hat{r}'_m - 1$, we have $\hat{d}_{m+3} \in \pi_1(I)$. With $\pi_1^{-1}(\hat{r}'_m) < \pi_1^{-1}(\hat{p}'_m) < \pi_1^{-1}(\hat{d}_{m+3})$, we obtain $\hat{p}'_m \in \pi_1(I)$, but since $\hat{p}'_m > \hat{r}'_m$, we again have a contradiction. Suppose $\pi_1(I) = [\hat{x}, \hat{p}'_m]$. Because $\hat{r}'_m = \hat{p}'_m - 1$, $\hat{r}'_m \in \pi_1(I)$. With $\pi_1^{-1}(\hat{r}'_m) < \pi_1^{-1}(\hat{d}_{m+1}) < \pi_1^{-1}(\hat{p}'_m)$, we know $\pi_1^{-1}(\hat{d}_{m+1}) \in I$, and this implies $\hat{d}_m \in \pi_1(I)$. Because $\pi_1^{-1}(\hat{d}_m) < \pi_1^{-1}(\hat{d}_{m+2}) < \pi_1^{-1}(\hat{p}'_m)$, $\hat{d}_{m+2}$ must be in $\pi_1(I)$, but since $\hat{d}_{m+2} > \hat{p}'_m$, we achieve a contradiction. Finally, let $\pi_1(I) = [\hat{x}, \hat{d}_{m+2}]$. Since $\hat{p}'_m = \hat{d}_{m+2} - 1$, we must have $\hat{p}'_m \in \pi_1(I)$, but this implies $\hat{d}_{m+1} \in \pi_1(I)$ due to the fact $\pi_1^{-1}(\hat{d}_{m+2}) < \pi_1^{-1}(\hat{d}_{m+1}) < \pi_1^{-1}(\hat{p}'_m)$. The block I is now all of π_1, so this is a contradiction. Hence, π_1 must be simple.

With $\hat{d}_{m+2}$, $\hat{r}'_m$, $\hat{p}'_m$ and $\hat{d}_{m+3}$, π_1 has the structure described in Case 4 or 5 of Figure 5.21. On the other hand, with $\hat{d}_{m+2}$, $\hat{r}'_m$ and $\hat{d}_{m+1}$, π_2 has the structure described in Case 2 of Figure 5.22. Together, we can recover π by $\pi_1 \mathbin{\bigcirc\mkern-14mu\nwarrow}_4^0 \pi_2$.

Consequently, in every case, we can express π as $\pi_1 \mathbin{\bigcirc\mkern-14mu\nwarrow}_x^y \pi_2$ for appropriate π_1 and π_2 where $(x, y) \in \{(1,0), (1,1), (2,0), (2,1), (3,0), (4,0)\}$, if m is odd. So suppose m is even. Hence, $\pi \in H'$ with $m+3$ values denoted by d_i satisfies Equation 5.6(b). We need to show that for $\pi \in H'$ of length n with $m+4$ values denoted by d_i satisfies Equation 5.6(a). The proof can be done by the inverse argument of the proof for the case that m is odd. This completes the proof of Proposition 5.22. ■

5.2.3. Proof of Theorem 5.21 (Part 2)

We now prove the converse direction of Theorem 5.21 as Proposition 5.24, which states that an arbitrary permutation which has the form of either Equation 5.6(a) or 5.6(b) must be in H'. Before we start, we discuss a few differences between this proposition and Proposition 4.6, the analogous proposition in Chapter 4. Then we state and prove Lemma 5.23, which is necessary for the proof of Proposition 5.24.

In Proposition 4.6, we assume *a priori* that the permutation π with the structure in Figure 4.9 is simple. Here, we will not make that assumption. This is because in Chapter 6, we will not show that this assumption holds for a permutation decoded from a word in our language. Hence, we need to explicitly prove that π of the form in Equation 5.6(a) or 5.6(b) is simple in addition to showing that π avoids every permutation in $\{52341, 53241, 52431, 35142, 42513, 351624\}$.

Furthermore, we proved Proposition 4.6 using a graphical representation of permutations in H. In $\mathcal{A}'$, however, providing a graphical representation of π in H' is extremely complicated. Hence, we show Proposition 5.24 by induction on the number of glue sums.

We now establish the following lemma, which says that if π has the form of Equation 5.6(a) or 5.6(b), then π has exactly $m+3$ values denoted by d_i where each d_i is defined in the same way as in the proof of 5.22.

Lemma 5.23 *Let π be an arbitrary permutation of length n. If there exist simple permutations σ_i (i odd) in $\mathcal{A}'$ of extreme pattern 2413 and simple permutations τ_i (i even) in $\mathcal{A}'$ of extreme pattern 3142 such that*

$$\pi = \sigma_1 \mathbin{\bigcirc\!\!\!\!\!\nwarrow}^{y_1}_{x_1} \tau_2 \mathbin{\bigcirc\!\!\!\!\!\searrow}^{y_2}_{x_2} \sigma_3 \mathbin{\bigcirc\!\!\!\!\!\nwarrow}^{y_3}_{x_3} \tau_4 \mathbin{\bigcirc\!\!\!\!\!\searrow}^{y_4}_{x_4} \cdots \mathbin{\bigcirc\!\!\!\!\!\searrow}^{y_{m-1}}_{x_{m-1}} \sigma_m \qquad \textit{for some odd integer } m$$

or

$$\pi = \sigma_1 \mathbin{\bigcirc\!\!\!\!\!\nwarrow}^{y_1}_{x_1} \tau_2 \mathbin{\bigcirc\!\!\!\!\!\searrow}^{y_2}_{x_2} \sigma_3 \mathbin{\bigcirc\!\!\!\!\!\nwarrow}^{y_3}_{x_3} \tau_4 \mathbin{\bigcirc\!\!\!\!\!\searrow}^{y_4}_{x_4} \cdots \mathbin{\bigcirc\!\!\!\!\!\nwarrow}^{y_{m-1}}_{x_{m-1}} \tau_m \qquad \textit{for some even integer } m$$

where $(x_\ell, y_\ell) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$ ($1 \leq \ell \leq m-1$), then π has $m+3$ distinct values denoted by d_i where $d_1 = \pi(1)$ and

$$d_i = \begin{cases} \pi(\max\{s : \pi(s) < \pi(d_{i-1})\}) & \textit{if } i \textit{ is even} \\ \max\{t : \pi^{-1}(t) < \pi^{-1}(d_{i-1})\} & \textit{if } i \textit{ is odd} \end{cases}$$

for i with $2 \leq i \leq m+3$.

Proof. We use mathematical induction to prove the statement. As the base case of $m = 1$, suppose $\pi = \sigma$ where σ is a simple permutation in $\mathcal{A}'$ of extreme pattern 2413 whose length is n. Then $d_1 = \pi(1)$, $d_2 = 1$, $d_3 = n$ and $d_4 = \pi(n)$. By definition, we obtain $d_i = d_3$ if $i \geq 5$ is odd, and $d_i = d_4$ if $i \geq 6$ is even. Thus, we have $4 = m+3$ distinct values denoted by d_i, so the statement holds for the base case.

Suppose the statement holds for some odd integer m. Consider

$$\pi = \sigma_1 \mathbin{\bigcirc\!\!\!\!\!\nwarrow}^{y_1}_{x_1} \tau_2 \mathbin{\bigcirc\!\!\!\!\!\searrow}^{y_2}_{x_2} \cdots \mathbin{\bigcirc\!\!\!\!\!\searrow}^{y_{m-1}}_{x_{m-1}} \sigma_m$$

of length n for some σ_i ($1 \leq i \leq m$, i odd) and τ_i ($2 \leq i \leq m-1$, i even) where $(x_\ell, y_\ell) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$ $(1 \leq \ell \leq m-1)$. Assume σ_m has the structure of Case 1 of Figure 5.21. Let τ_{m+1} be a simple permutation ($|\tau_{m+1}| = k$) in $\mathcal{A}'$ of extreme pattern 3142 which has the structure described in Case 1 of Figure 5.22. We let d_i $(1 \leq i \leq m+3)$ for π and d'_i for $\pi \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_1^0 \tau_{m+1}$ as defined by d_i in the statement of Lemma 5.23. Note that $d_{m+2} = n$ and $d_{m+3} = \pi(n)$ by definition. We need to show there are $m+4$ distinct values denoted by d'_i. The only values of π that are modified to construct $\pi \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_1^0 \tau_{m+1}$ are $d_{m+2} = \pi(n)$ and $d_{m+3} = \pi(n)$. The value in $\pi \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_1^0 \tau_{m+1}$ corresponding to d_{m+2} is adjusted upward by $(\tau_{m+1}(1) - 3)$, and the value d_{m+3} is eliminated. Moreover, every value in $\pi \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_1^0 \tau_{m+1}$ corresponding to τ_{m+1} is greater than or equal to d_{m+3}. Hence, $d'_i = d_i$ for $1 \leq i \leq m+1$. Also, the value in $\pi \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_1^0 \tau_{m+1}$ corresponding to d_{m+2} is still the greatest value whose position is to the left of d'_{m+1}, so it is denoted by d'_{m+2}, though it is not equal to d_{m+2}. Since $\tau_{m+1}(k) < \tau_{m+1}(1)$, $\pi \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_1^0 \tau_{m+1}(n+k-3)$, which is the value corresponding to $\tau_{m+1}(k)$, is denoted by d'_{m+3}, and the one corresponding to k is denoted by d'_{m+4}. We then have $d'_i = d_{m+4}$ if $i \geq m+6$ is odd, and $d'_i = d_{m+3}$ if $i \geq m+5$ is even. Consequently, we have $m+4$ distinct values denoted by d'_i.

Proofs for for the cases that σ_m and τ_{m+1} having different structures, and thus, different types of NW glue sums, are very similar. We can also show that the statement is true for $m+1$ when we assume m is even by applying the inverse argument. Altogether, we have the desired result. ∎

We are now ready to state and prove Proposition 5.24.

Proposition 5.24 *Let π be an arbitrary permutation of length n. If there exist simple permutations σ_i (i odd) in $\mathcal{A}'$ of extreme pattern 2413 and simple permutations τ_i (i even) in $\mathcal{A}'$ of extreme pattern 3142 such that*

$$\pi = \sigma_1 \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_{x_1}^{y_1} \tau_2 \mathbin{\searrow\!\!\!\!\!\bigcirc}_{x_2}^{y_2} \sigma_3 \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_{x_3}^{y_3} \tau_4 \mathbin{\searrow\!\!\!\!\!\bigcirc}_{x_4}^{y_4} \cdots \mathbin{\searrow\!\!\!\!\!\bigcirc}_{x_{m-1}}^{y_{m-1}} \sigma_m \qquad \textit{for some odd integer } m$$

or

$$\pi = \sigma_1 \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_{x_1}^{y_1} \tau_2 \mathbin{\searrow\!\!\!\!\!\bigcirc}_{x_2}^{y_2} \sigma_3 \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_{x_3}^{y_3} \tau_4 \mathbin{\searrow\!\!\!\!\!\bigcirc}_{x_4}^{y_4} \cdots \mathbin{\nwarrow\!\!\!\!\!\bigcirc}_{x_{m-1}}^{y_{m-1}} \tau_m \qquad \textit{for some even integer } m$$

where $(x_\ell, y_\ell) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$ $(1 \leq \ell \leq m-1)$, then π is in H'.

Proof. If a permutation is expressed as in Equation 5.6(a) or 5.6(b), then it satisfies the condition of $\pi^{-1}(1) \geq 5$ or $\pi(1) = 2$ because the first summand is σ_1 which is a simple permutation of extreme pattern 2413. So we have to show that a permutation expressed as in Equation 5.6(a) or 5.6(b) is simple and avoids every permutation in $\{52341, 53241, 52431, 35142, 42513, 351624\}$. We use mathematical induction on m to complete the proof. As the base case, suppose $\pi = \sigma$

where σ is a simple permutation in $\mathcal{A}'$ of extreme pattern 2413. Then we have the desired result.

Now, assume that the statement holds for some odd integer m. Let

$$\pi = \sigma_1 \textcircled{$\nwarrow$}^{y_1}_{x_1} \tau_2 \textcircled{$\searrow$}^{y_2}_{x_2} \cdots \textcircled{$\searrow$}^{y_{m-1}}_{x_{m-1}} \sigma_m$$

of length n for some σ_i ($1 \leq i \leq m$, i odd) and τ_i ($2 \leq i \leq m-1$, i even) where $(x_\ell, y_\ell) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$ ($1 \leq \ell \leq m-1$). We need to show that, for every appropriate τ_{m+1} of length k and σ_m with $\textcircled{$\nwarrow$}^{y_m}_{x_m}$ ($(x_m, y_m) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$), $\pi \textcircled{$\nwarrow$}^{y_m}_{x_m} \tau_{m+1}$ is simple, and avoids permutations in $\{52341, 53241, 52431, 35142, 42513, 351624\}$. Let $\rho = \pi \textcircled{$\nwarrow$}^{y_m}_{x_m} \tau_{m+1}$. We first show that ρ is simple.

Define values d_j of ρ for positive integers i as before. By Lemma 5.23, there are $m+4$ values denoted by d_i of ρ. Suppose to the contrary that ρ is not simple, so we have a proper non-singleton block I. Assume I contains at least two positions corresponding to d_j and d_k for some j, k with $1 \leq j, k \leq m+4$. Note that I cannot contain both positions of d_1 and d_{m+3} because this results in $I = [1, n]$. So suppose I does not contain $1 = \rho^{-1}(d_1)$. Let s be the least position in I such that $\rho(s) = d_j$ for some j with $2 \leq j \leq m+4$. Suppose j is even. If $j = m+3$, then there is no more value to the right of d_{m+3}, so $j \leq m+1$. Hence, assume $j \leq m+1$. Then the position of d_{j+3} must be in I, but with $d_j < d_{j-1} < d_{j+3}$, we must have $\rho^{-1}(d_{j-1}) \in I$ while $\rho^{-1}(d_{j-1}) < s$, so we have a contradiction. Now, assume j is odd. Then I contains the position of d_{j-1}. Because $d_{j-1} < d_{j-2} < d_j$, we have $\rho^{-1}(d_{j-2}) \in I$, but $\rho^{-1}(d_{j-2}) < s$, so we again achieve a contradiction. We can apply a similar argument for the case of I not containing $n = \rho^{-1}(d_{m+3})$, so I cannot contain two or more positions corresponding to values denoted by d_i.

This time, suppose I does not contain a position corresponding to any of d_j of ρ. For every $(x_m, y_m) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$, positions of all values of π that are modified by $\textcircled{$\nwarrow$}^{y_m}_{x_m}$ to construct ρ are greater than or equal to $\rho^{-1}(d_{m+2})$. Thus, if I is a subset of $[1, \rho^{-1}(d_{m+2}) - 1]$, then I is also a proper non-singleton block of π. Since π is a simple permutation by induction hypothesis, this is a contradiction. Similarly, if I is a subset of $[\rho^{-1}(d_{m+4}) + 1, \rho^{-1}(d_{m+3}) - 1]$, then I is a proper non-singleton block of τ_{m+1}. Hence, we have either $I \subseteq [\rho^{-1}(d_{m+2}) + 1, \rho^{-1}(d_{m+1}) - 1]$ or $I \subseteq [\rho^{-1}(d_{m+1}) + 1, \rho^{-1}(d_{m+4}) - 1]$. Suppose $I \subseteq [\rho^{-1}(d_{m+2}) + 1, \rho^{-1}(d_{m+1}) - 1]$. Then I must contain at least one value in $[\rho^{-1}(d_{m+2}) + 1, \rho^{-1}(d_{m+1}) - 1]$ corresponding to the value of π that was modified by $\textcircled{$\nwarrow$}^{y_m}_{x_m}$, because, otherwise, I would also a proper non-singleton block of π. In cases of $\textcircled{$\nwarrow$}^0_1$, $\textcircled{$\nwarrow$}^1_1$ and $\textcircled{$\nwarrow$}^0_3$, there is no such value. If $\textcircled{$\nwarrow$}^{y_m}_{x_m} = \textcircled{$\nwarrow$}^0_2$, $\textcircled{$\nwarrow$}^1_2$ or $\textcircled{$\nwarrow$}^0_4$, either $d_{m+2} - 1$ or $d_{m+2} - 2$ is the only such value, but in either case, d_{m+3} is less than this value, implying that I contains $\rho^{-1}(d_{m+3})$, so we achieve a contradiction for every type of NW glue sum.

So assume $I \subseteq [\rho^{-1}(d_{m+1})+1, \rho^{-1}(d_{m+4})-1]$. We first consider the case where $(x_m, y_m) \in \{(1,0),(1,1),(2,0),(2,1)\}$. If I only contains values corresponding to τ_{m+1}, then I would be a proper non-singleton block of τ_{m+1} as well, because for every $(x_m, y_m) \in \{(1,0),(1,1),(2,0),(2,1)\}$, $\bigcirc\kern-0.8em\nwarrow_{x_m}^{y_m}$ only shifts values of τ_{m+1} upward by the same amount to construct ρ. For a similar reason, if I only contains values corresponding to π, then I would be a proper non-singleton block of π. Thus, I must contain values corresponding to both π and τ_{m+1}. Because I contains values from both π and τ_{m+1}, the position of $\tau'_{m+1}(3)$, the left-most point of τ_{m+1} in $[\rho^{-1}(d_{m+1})+1, \rho^{-1}(d_{m+4})-1]$, must be in I since every value to its right comes from τ. If $\tau'_{m+1}(3) > d_{m+3}$, then $\rho^{-1}(d_{m+3})$ would be in I since every value of ρ coming from π in $[\rho^{-1}(d_{m+1})+1, \rho^{-1}(d_{m+4})-1]$ is less than d_{m+3}, so we must have $\tau'_{m+1}(3) < d_{m+3}$. By Proposition 5.16, this implies that $\tau_{m+1}(3)$ is a part of a 312-position chain. Due to the structure of 312-position chain, every value of ρ corresponding to some value in this 312-position chain must be in I. Denote by M the position of the right-most value involved in the 312-position chain containing $\tau_{m+1}(3)$. Then $\tau_{m+1}(M-1) > d_{m+3}$, since $\tau_{m+1}(M-1)$ is the scissor of this 312-position chain. Hence, $\rho^{-1}(d_{m+3}) \in I$, which is a contradiction.

For the case of $\bigcirc\kern-0.8em\nwarrow_4^0$, I must also contain values from both π and τ_{m+1}. The position of $\tau'_{m+1}(4)$, the left-most point of τ_{m+1} in $[\rho^{-1}(d_{m+1})+1, \rho^{-1}(d_{m+4})-1]$, must be in I since every value to its right comes from τ. However, by proposition 5.16, $\tau'_{m+1}(4) > d_{m+3}$. Hence, $\rho^{-1}(d_{m+3})$ would be in I since every value of ρ coming from π in $[\rho^{-1}(d_{m+1})+1, \rho^{-1}(d_{m+4})-1]$ is less than d_{m+3}.

Finally, consider the case of $\bigcirc\kern-0.8em\nwarrow_3^0$. The block I cannot contain only values corresponding to τ_{m+1} due to the same reason as other NW glue sums. Now, if I only contains values corresponding to π, then it must contain $\pi'(n-2)$, because it is the only value in $[\rho^{-1}(d_{m+1})+1, \rho^{-1}(d_{m+4})-1]$ that is shifted upward by $\bigcirc\kern-0.8em\nwarrow_3^0$ to construct ρ. However, since $\pi'(n-2) > d_{m+3}$, we end up with $\rho^{-1}(d_{m+3}) \in I$ as usual, so I cannot contain only values corresponding to π. Thus, I must contain values corresponding to both π and τ_{m+1}. Furthermore, $\rho^{-1}(\pi'(n-2)) \notin I$, because, otherwise, we again obtain $\rho^{-1}(d_{m+3}) \in I$. So the only possibility is $I = [\rho^{-1}(\pi'(n-1)), z]$ for some position z whose value corresponds to a value of τ_{m+1}. Note that, however, π has the structure shown in Case 3 of Figure 5.21. Thus, $\pi(n-2) = n-1$, $\pi(n) = n-2$ and $\pi(n-1) = n-4$ are all parts of the same 312-value chain, which implies that $\pi(n-1)+1$ is located to the left of $\pi(n-1)$. Consequently, $\pi'(n-1)+1 \in \rho(I)$ has a position less than $\rho^{-1}(\pi'(n-1))$, so we achieve a contradiction. Altogether, we have a contradiction for the case of I not containing a position corresponding to any of d_j of ρ.

Lastly, we assume I contains exactly one position corresponding to d_j for some j with $1 \le j \le m+3$. For every $(x_m, y_m) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$, values of π that are modified by $\obslash_{x_m}^{y_m}$ to construct ρ are all located to the right of d_{m+2}. Thus, if $j \le m$, then I would be a proper non-singleton block in π. Similarly, if $j = m+3$ or $m+4$, then I would be a proper non-singleton block in τ_{m+1}. So j must be either $m+1$ or $m+2$. For the case $j = m+1$, the arguments of previous paragraphs apply. So suppose $j = m+2$. If I contains $\rho^{-1}(d_{m+2}) - 1$, then I also has to contain $\rho^{-1}(d_{m+3})$ because $\rho(\rho^{-1}(d_{m+2}) - 1) < d_{m+3}$. Thus, $I = [\rho^{-1}(d_{m+2}), z]$ for some position z. The value immediately to the right of d_{m+2}, however, is less than d_{m+3}, except when it is $d_{m+2} - 2$ and $\obslash_{x_m}^{y_m} = \obslash_4^0$. In this case, $\tau'_{m+1}(4)$, the left-most value from τ_{m+1} is $d_{m+2} - 1$, so $\rho^{-1}(d_{m+2} - 1) \in I$. Since $\rho^{-1}(d_{m+2} - 1) > \rho^{-1}(d_{m+1})$, $\rho^{-1}(d_{m+1}) \in I$, which is a contradiction.

Consequently, ρ cannot have a proper non-singleton block I, so ρ is simple. Next, we prove that ρ avoids every permutation in $\{52341, 53241, 52431, 35142, 42513, 351624\}$.

Suppose ρ contains some permutation β in $\{52341, 53241, 52431, 35142, 42513, 351624\}$. Note that $\pi, \tau_{m+1} \in \mathcal{A}'$, and modifications of every value of π and τ_{m+1} by each $\obslash_{x_m}^{y_m}$ to construct ρ are either upward shift of certain values by the same amount or removal. Thus, whichever permutation β in the basis that ρ contains, the containment must involve both values corresponding to π and values corresponding to τ_{m+1}, because, otherwise, $\beta \preceq \pi$ or $\beta \preceq \tau_{m+1}$. Hence, involving both values from π and values from τ_{m+1} for a containment of β is a necessary condition of $\beta \preceq \rho$, but it is not a sufficient condition.

We first discuss the case $\beta \in \{52341, 53241, 52431\}$. Note that, every value corresponding to τ_{m+1} is greater than any value corresponding to π, except for the shifted ones. Thus, for every $\beta \in \{52341, 53241, 52431\}$, we must assign one of the shifted values to $\beta(1) = 5$ in order to assign some value $\tau'_{m+1}(i)$ to $\beta(5) = 1$. Hence, the only values which can play the role of 5 are d_{m+2} for every $(x_m, y_m) \in \{(1,0),(1,1),(2,0),(2,1),(3,0),(4,0)\}$, $d_{m+2} - 1$ for $(x_m, y_m) = (2,0)$ and $(2,1)$, $\rho(n-2) = \pi'(n-2)$ for $(x_m, y_m) = (3,0)$, and $d_{m+2} - 2$ for $(x_m, y_m) = (4,0)$.

Consider the cases of $(x_m, y_m) \in \{(1,0),(1,1)\}$. Then d_{m+2} is playing the role of $\beta(1) = 5$. In addition, the values playing the roles of $\beta(2)$, $\beta(3)$ and $\beta(4)$ are all from τ_{m+1}, since we have to assign a value corresponding to τ_{m+1} to $\beta(5) = 1$. However, note that the flattening of $d_{m+2}\tau'_{m+1}(3)\tau'_{m+1}(4) \cdots \tau'_{m+1}(k)$, call it γ, is contained in τ_{m+1}. Hence, no matter which values of ρ corresponding to τ_{m+1} plays the role of $\beta(2)$, $\beta(3)$ and $\beta(4)$, $\beta \preceq \gamma$ implies $\beta \preceq \tau_{m+1}$. Since $\tau_{m+1} \in \mathcal{A}'$, this is a contradiction.

Next, consider $(x_m, y_m) \in \{(2,0),(2,1)\}$. As explained, at least one shifted value must be playing the role of 5, so in this case, it is either d_{m+2} or $d_{m+2}-1$. Suppose only one of them is involved for a containment of β, say d_{m+2}. Then again, γ, the flattening of $d_{m+2}\tau'_{m+1}(3)\tau'_{m+1}(4)\cdots\tau'_{m+1}(k)$ is contained in τ_{m+1}, and all four values playing the roles of $\beta(2)$, $\beta(3)$, $\beta(4)$ and $\beta(5) = 1$ respectively must correspond to τ_{m+1}, so $\beta \preceq \tau_{m+1}$, a contradiction. We achieve the exact same result if $d_{m+2} - 1$ plays the role of 5. Thus, suppose both d_{m+2} and $d_{m+2} - 1$ are involved for a containment of β. Since d_{m+2} is located to the left of $d_{m+2} - 1$, d_{m+2} must play the role of $\beta(1) = 5$. In addition, we cannot assign $d_{m+2} - 1$ to either 2 or 3, because there is no integer value between $d_{m+2} - 1$ and d_{m+2}, so we don't have a value for 4. Since every β in $\{52341, 53241, 52431\}$ has either $\beta(2) = 2$ or $\beta(2) = 3$, we cannot involve both d_{m+2} and $d_{m+2}-1$ to have $\beta \preceq \rho$. Consequently, $\rho = \pi \obslash_2^0 \tau_{m+1}$ cannot contain $\beta \in \{52341, 53241, 52431\}$.

We move onto the case $(x_m, y_m) = (3,0)$. Let s be the value corresponding to the scissor of the last 312-value chain of π which is merged with the first 312-position chain of τ by the type 3-0 NW glue sum. In addition, let t be the position of the value corresponding to the left-most value of the last 312-value chain of π. Note that $t = \rho^{-1}(s-1) - 1$. Due to the construction of ρ by $\obslash_3^0$, the flattening of $d_{m+2}sd_{m+1}\rho(t)\rho(t+1)\cdots\rho(n+k-(\ell+3))$ is exactly τ_{m+1}. Let $\overline{\tau}_{m+1}$ be this sequence of values. A containment of β must involve some value other than the values in $\overline{\tau}_{m+1}$, because, otherwise, $\beta \preceq \tau_{m+1}$. Whether d_{m+2} or $\rho(n-2)$ plays the role of 5, some value z corresponding to τ_{m+1} must be assigned to 1. Since every value greater than z is in the sequence $\overline{\tau}_{m+1}$, we cannot assign any value to $\beta(2)$, $\beta(3)$ and $\beta(4)$ that are not in $\overline{\tau}_{m+1}$. Hence, $\beta \not\preceq \rho$.

Lastly, the case $(x_m, y_m) = (4,0)$ is similar to the case of $(x_m, y_m) \in \{(2,0),(2,1)\}$. If d_{m+2} is involved in a containment of β as 5, but not $d_{m+2}-1$, then again, we are forced to assign some values of τ_{m+1} to $\beta(2)$, $\beta(3)$ and $\beta(4)$. Because the flattening of $d_{m+2}\tau'_{m+1}(4)\tau'_{m+1}(5)\cdots\tau'_{m+1}(k)$ is contained in τ_{m+1}, we obtain $\beta \preceq \tau_{m+1}$, which is a contradiction. We achieve the same result for the case of $d_{m+2}-2$ playing the role of 5. So suppose both d_{m+2} and $d_{m+2}-2$ are involved in a containment of β. Hence, d_{m+2} and $d_{m+2} - 2$ play the role of $\beta(1) = 5$ and $\beta(2)$ respectively. Note that we only have one value, $d_{m+2} - 1$, that is in between $d_{m+2} - 2$ and d_{m+2}. Thus, if $\beta = 52341$ or 52431, we do not have a value for either $\beta(2)$ or $\beta(3)$. Therefore, suppose $\beta = 53241$. Since the value assigned to $\beta(3) = 2$ is less than $d_{m+2} - 2$ and its is in between $\rho^{-1}(d_{m+2} - 2)$ and $\rho^{-1}(d_{m+2} - 1)$, this value must correspond to π. However, this value is less than the value corresponding to τ_{m+1} that is assigned to 1, so we cannot form 53241 pattern. Hence, $53241 \not\preceq \rho$.

Altogether, we conclude that for every $\beta \in \{52341, 53241, 52431\}$, ρ avoids β. We now look at the case of $\beta = 35142$.

First, consider $(x_m, y_m) \in \{(1,0),(1,1)\}$. If we assign a value of ρ from π to $\beta(4) = 4$, then we are forced to choose another value from π to play the role of $\beta(5) = 2$. Since we don't have a value from τ_{m+1} corresponding to some value of β, we have $\beta \preceq \pi$, which is a contradiction. Hence, suppose some value corresponding to τ_{m+1} plays the role of 4. If a value corresponding to τ_{m+1} plays the role of $\beta(2) = 5$, then $\beta(3) = 1$ is automatically played by a value corresponding to τ_{m+1} as well. We need to assign a value from π to $b(1) = 3$, but the only value available is d_{m+2}. Since γ, the flattening of $d_{m+2}\tau'_{m+1}(3)\tau'_{m+1}(4)\cdots\tau'_{m+1}(k)$ is contained in τ_{m+1}, we now have $\beta \preceq \tau_{m+1}$, a contradiction. Thus, suppose some value corresponding to π plays the role of 5. Then again, d_{m+2} is the only value which can play the role of 5. We are forced to assign a value z corresponding to π and located to the left of d_{m+2} to 3. However, there is no value from τ_{m+1} that is less than z, so we do not have a value for 2. Hence, 35142 cannot be contained in ρ in the case of $(x_m, y_m) \in \{(1,0),(1,1)\}$.

For $(x_m, y_m) \in \{(2,0),(2,1),(4,0)\}$, it is possible to assign $d_{m+2} - 1$ (when $x_m = 2$) or $d_{m+2} - 2$ (when $x_m = 4$) to 4, and assign some value z from τ_{m+1} to 2. Since the only value located to the left of and greater than the one assigned to 4 is d_{m+2}, it has to play the role of 5, forcing the position of the value playing the role of 3 to be to the left of d_{m+2}. However, there is no value located there that is greater than z, so β cannot be contained in ρ this way. The rest of the proof and the case $(x_m, y_m) = (3,0)$ is very similar to the case of $(x_m, y_m) \in \{(1,0),(1,1)\}$. Essentially, 5 must be played by either d_{m+2} or the other shifted value. Then the value playing the role of 3 is located to the left of the value for 5, but this value cannot be greater than any value corresponding to τ_{m+1}, so we don't have a value to assign to 2. Thus, in every case, $35142 \npreceq \rho$.

We now let $\beta = 42513$. Again, since both values corresponding to π and values corresponding to τ_{m+1} must be involved in a containment of β, we know the value for $\beta(1) = 4$ is from π and the value for $\beta(5) = 3$ is from τ_{m+1}. For each NW glue sum, there are at most two values corresponding to π that are greater than some value corresponding to τ, but they are in decreasing order from left to right. Hence, the value playing the role of 4 must be one of the two shifted values, and the value playing the role of $\beta(3) = 5$ must be from τ_{m+1} for each case of NW glue sum. This implies the value for $\beta(4) = 1$ is also corresponding to τ_{m+1}. If the value for $\beta(2) = 2$ is from τ_{m+1}, then again, the flattening of values assigned to a value of β is contained in τ_{m+1}. Thus, the value assigned to 2 must be from π. Furthermore, this value has to be the other shifted value to assign, and particularly $d_{m+2} - 2$. Otherwise, we do not have values from τ_{m+1} that we can assign to each of 1 and 3. Therefore, $(x_m, y_m) \in \{(3,0),(4,0)\}$, and we assign

d_{m+2} to 4, $d_{m+2} - 2$ to 2 and three values from τ_{m+1} to each of 5, 1 and 3. However, because the flattening of values d_{m+2}, $d_{m+2} - 2$ and all the values corresponding to τ_{m+1} is contained in τ_{m+1}, we must have $\beta \preceq \tau_{m+1}$. This is a contradiction, so we conclude that $42513 \not\preceq \rho$.

The final case is $\beta = 351624$. For the same reason as the case of 42513, the value $\beta(2) = 5$ must be played by one of the shifted values, and the value $\beta(4) = 6$ must be from τ_{m+1} for each NW glue sum. Thus, the value playing the role of $\beta(1) = 3$ is a value corresponding to π, and this value is not shifted by $\text{\textcircled{$\nwarrow$}}_{x_m}^{y_m}$. Because the value playing the role of 6 is from τ_{m+1}, so is the value playing the role of $\beta(5) = 2$, but we do not have a value corresponding to τ_{m+1} that is less than the value playing the role of 3. Consequently, $351624 \not\preceq \rho$.

Therefore, when the statement of the proposition holds for odd m, it is also true for $m+1$. Showing the case of even m is the same argument applied to the inverses of all permutations involved. Altogether, we have the desired result. ■

Chapter 6.

Enumeration of the class $\mathcal{A}'$

6.1. Enumeration of simple permutations in $\mathcal{A}'$

With the structure we discussed in Chapter 5, we are finally ready to enumerate the class $\mathcal{A}'$. How we are going to accomplish this is somewhat similar to the method we used in Chapter 4 for $\mathcal{A}$, but we need to break down the whole procedure into small steps. We first give an alphabet Σ' and define an encoding function ϕ' from H' to Σ'^*. Then we define a language $L' \subseteq \Sigma'^*$ and show ϕ' is a bijection between H' and L'. Afterwards, we construct ten languages $\overline{L}_i'$ $(1 \leq i \leq 10)$ associated with L'. Frankly, there are ten distinct kinds of prefixes a word w in $\overline{L}$ can have. Thus, each $\overline{L}_i'$ is prepared to generate the number of words having different prefixes. We then define ten deterministic finite-state automatons M_i' $(1 \leq i \leq 10)$ and show $\overline{L}_i' = \mathcal{L}(M_i)$ for each i with $1 \leq i \leq 10$. Once we obtain ten distinct generating functions, we combine them to obtain the following result.

Theorem 6.1 *Let $f_{Si(\mathcal{A}')\backslash\mathcal{S}_2}$ be the generating function for the set of simple permutations in $\mathcal{A}'$ excluding $\mathcal{S}_2 = \{12, 21\}$. Then*

$$f_{\mathrm{Si}(\mathcal{A}')\backslash\mathcal{S}_2} = \frac{2x^4\left(x^{10}+7x^9+18x^8+23x^7+16x^6+10x^5+12x^4+9x^3+2x^2+1\right)}{(x+1)\left(2x^9+12x^8+16x^7+3x^6-11x^5-5x^4-3x^2-3x+1\right)}.$$

In this section, let N and S be the set of simple permutations of extreme pattern 2413 and 3142 respectively. Let m be the total number of simple permutations in N and S together to construct $\pi \in H'$ with glue sums. Furthermore, let d_i, p_i, q_i, q_i' and r_i be defined as in the proof of Proposition 5.22. Note that each of p_i, q_i, q_i' and r_i is defined differently based on whether i is odd or even.

6.1.1. Defining the encoding function ϕ' and the language L'

Let

$$\Sigma' = \{a, a', a'', b, b_s, b', b'', \underline{\mathrm{b}}, \overline{\mathrm{b}}, c, c', c'', d, d_a', d_a'', d_c', d_c'', \underline{d}, \overline{\mathrm{d}}, \underline{\overline{\mathrm{d}}}, d_\ell, x, x', x'', \underline{x}, y, y', y'', \overline{y}, z\}.$$

Now, we define an encoding function ϕ' from H' to Σ'^*. Comparing to the encoding function ϕ for H, the description of ϕ' is far more complicated, so we divide into several algorithms to define ϕ'.

First, by Theorem 5.21, given $\pi \in H'$, π can be uniquely expressed as Equation 5.6 where σ_i (i odd) is in N and τ_i (i even) is in S.

Next, we define two encoding algorithms. One is to encode a simple permutation in N, called N-ENCODE, and the other one is for a simple permutation in S, called S-ENCODE. Both algorithms encode simple permutations in $\mathcal{A}'$ into a word in $\{a, a', a'', b, b_s, b', b'', c, c', c''\}^*$. So let $\Sigma_1 = \{a, a', a'', b, b_s, b', b'', c, c', c''\}$ which is a subset of Σ'.

We first define N-ENCODE. The algorithm N-ENCODE assigns each value of σ a letter in Σ_1, then reads the letters assigned to each value from 1 to n in increasing order. Letters are assigned as follows.

First, suppose the position of the value t is in $[1, \pi^{-1}(n))$. Then, if t plays the role of 1 of 21 or 1 of 231 of a 231-value chain, it is encoded as a', if t plays the role of 2 of 21 or 3 of 231 of a 231-value chain, it is encoded as a'', and if t plays the role of 2 of 231 of a 231-value chain or 1 of 1 in Equation 5.2, then it is encoded as a. Next, assume the position of the value t is in $[\pi^{-1}(n), \pi^{-1}(1)]$. Then, if t plays the role of 1 of 12 in Equation 5.1, it is encoded as b', if t plays the role of 2 of 12 in Equation 5.1, it is encoded as b'', if t is a scissor of a 231-value chain or a 312-value chain (and it is not already b' or b''), it is encoded as b_s, and otherwise, it is encoded as b. Finally, suppose the position of the value t is in $(\pi^{-1}(1), n]$. Then, if t plays the role of 1 of 21 or 1 of 312 of a 312-value chain, it is encoded as c', if t plays the role of 2 of 21 or 3 of 312 of a 312-value chain, it is encoded as c'', and if t plays the role of 2 of 312 of a 312-value chain or 1 of 1 in Equation 5.2, then it is encoded as c.

Now, let us define the language $K_1 \subseteq \Sigma_1^*$ to have the following conditions, which will turn out to be the image of N-ENCODE.

1. Prefix condition.
   A word w must begin with ba, $bb'a$, $ba'a$, $ba'b_sa''$, $ba'b'a''$ or $ba'a'a''$.

2. Suffix condition.
   A word w must end with cb, $cb''b$, $cc''b$, $c'b_sc''b$, $c'b''c''b$ or $c'c''c''b$.

3. Conditions on a' and a''.
   Every a' and a'' in w is a part of a subword in $\{a', a'a\}\{a'a'', a'a''a\}^*\{b_sa'', b'a''\}$. Note the number of a' and the number of a'' are equal in each sequence, and thus, in w.

4. Conditions on c' and c''.
   Every c' and c'' in w is a part of a subword in $\{c'b_s, c'b''\}\{c'c'', cc'c''\}^*\{c'', cc''\}$. Note the

number of c' and the number of c'' are equal in each sequence, and thus, in w.

5. Conditions on b_s.
   The letter b_s is only allowed in the third condition and the fourth condition, *i.e.* as a part of a subword of a' and a'' or a subword of c' and c''.

6. Conditions on b' and b''.
   Every b' and b'' in w is a part of a subword in $\{b'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{b''\}$ with at least one of a'', a, c or c' being present. Note the number of b' and the number of b'' are equal in each sequence, and thus, in w.

7. Repetition restrictions.
   w must not contain aa, bb or cc.

We define N-DECODE on K_1, the inverse algorithm of N-ENCODE, and show that N and K_1 are bijective due to these two algorithms. The algorithm N-DECODE takes $w \in K_1$ as an input and outputs a permutation as the following.

Algorithm N-DECODE
INPUT: A word w in K_1.
OUTPUT: A permutation π.

Initialize: Draw a point for the first b and locate coordinates for P_a, P_b and P_c.
Draw a point at $(1, 1)$. Let $P_a = (0, 1)$ and $P_b = P_c = (1, 1)$. Let $t = 1$. Let α be the second letter in w.

Identify: Determine the case due to what letter α is.

a. If $\alpha = a'$, then GOTO Case 1.

b. If $\alpha = b'$, then GOTO Case 2.

c. If $\alpha = c'$, then GOTO Case 3.

d. If α is none of the above, then GOTO Case 4.

Case 1: Construct and draw a 231-value chain in the following manner.
Take the subword of w starting from α up to a'' which equalizes the numbers of a' and a''. Hence, this subword is in $\{a', a'a\}\{a'a'', a'a''a\}^*\{b_sa'', b'a''\}$. Remove the first a' and the first a''. Additionally, if there exists a in between these a' and a'', remove it as well. Iteratively remove a', a'' and a until b_s or b' is the only letter left. Every time we remove a' and a'' together, sum 21 with $\oplus_1$, and every time we remove a',

a'' and a together, sum 231 with $\oplus_1$. Draw the 231-value chain that we obtain from this procedure so that the chain is horizontally in between P_a and P_b, and vertically greater than t. Set t to be the greatest y coordinate of all points in the chain and P_a to be the right-most point in the chain.

For b_s or b', draw a point located horizontally in between P_a and P_b, and vertically in between the point with the second greatest y coordinate and the point with the greatest y coordinate in the chain. Set α to be the letter immediately after the subword.

a. If b_s is a part of the subword, then set P_b to be the point drawn for b_s. Then GOTO Identify.

b. If b' is a part of the subword, then set P_ℓ to be the point drawn for b' and identify what letter α is. If $\alpha = a$, then GOTO a of Case 2. If $\alpha = c$, then GOTO b of Case 2. If $\alpha = b''$, then GOTO c of Case 2. If $\alpha = c'$, then GOTO Case 3.

Case 2: Draw points for b' and letters up to b''.

Draw a point at (x, y) where $P_a^{(x)} < x < P_b^{(x)}$ and $t < y$. Set P_ℓ to be this new point and $t = P_\ell^{(y)}$. Set α to be the next letter. If $\alpha = a$, then GOTO a. If $\alpha = c$, then GOTO b. If $\alpha = b''$, then GOTO c. If $\alpha = c'$, then GOTO Case 3.

a. Draw a point at (x, y) where $P_a^{(x)} < x < P_\ell^{(x)}$ and $t < y$. Set P_a to be this new point and $t = P_a^{(y)}$. Set α to be the next letter. If $\alpha = c$, then GOTO b. If $\alpha = b''$, then GOTO c. If $\alpha = c'$, then GOTO Case 3.

b. Draw a point at (x, y) where $P_c^{(x)} < x$ and $t < y$. Set P_c to be this new point and $t = P_c^{(y)}$. Set α to be the next letter. If $\alpha = b''$, then GOTO c. If $\alpha = c'$, then GOTO Case 3.

c. Draw a point at (x, y) where $P_\ell^{(x)} < x < P_b^{(x)}$ and $t < y$. Set $P_b = P_\ell$ and t to be the y coordinate of the point just drawn. Set α to be the next letter. Then GOTO Identify.

Case 3: Construct and draw a 312-value chain in the following manner.

Take the subword of w starting from α up to c'' which equalizes the numbers of c' and c''. Hence, this subword is in $\{c'b_s, c'b''\}\{c'c'', cc'c''\}^*\{c'', cc''\}$. Remove the first c' and the first c''. Additionally, if there exists c in between these c' and c'', remove it as well. Iteratively remove c', c'' and c until b_s or b'' is the only letter left. Every time we remove c' and c'' together, sum 21 with $\oplus_1$, and every time we remove c', c'' and c together, sum 312 with $\oplus_1$. Draw the 312-value chain that we obtain from this procedure so that the chain is horizontally located to the right of P_c, and vertically greater than t. Set t to be the greatest y coordinate of all points in the chain and P_c to be the right-most point in the chain.

```
        a. If $b_s$ is a part of the subword, then draw a point located horizontally in between $P_a$ and $P_b$, and vertically in between the point with the least $y$ coordinate and the point with the second least $y$ coordinate in the chain. Set $P_b$ to be the point drawn for $b_s$. Set $\alpha$ to be the letter immediately after the subword. Then GOTO Identify.
        b. If $b''$ is a part of the subword, then draw a point located horizontally in between $P_\ell$ and $P_b$, and vertically in between the point with the least $y$ coordinate and the point with the second least $y$ coordinate in the chain. Set $P_b = P_\ell$. Set $\alpha$ to be the letter immediately after the subword. Then GOTO Identify.
Case 4: Draw a point due to which letter $\alpha$ is.
        a. If $\alpha = a$ or $\alpha = b$, then draw a point at $(x, y)$ where $P_a^{(x)} < x < P_b^{(x)}$ and $t < y$. Set $P_a$ to be this new point and $t = P_a^{(y)}$.
            i. If $\alpha = b$ is the last letter in $w$, then GOTO Flatten.
            ii. Otherwise, set $\alpha$ to be the next letter. Then GOTO Identify.
        b. If $\alpha = c$, then draw a point at $(x, y)$ where $P_c^{(x)} < x$ and $t < y$. Set $P_c$ to be this new point and $t = P_c^{(y)}$. Set $\alpha$ to be the next letter. Then GOTO Identify.
Flatten: Finalize the permutation by flattening.
        Let $\pi$ be a permutation obtained by flattening the constructed graph. OUTPUT $\pi$.
```

We define the function ϕ_N on N that $\phi_N(\sigma)$ is a word obtained by applying N-ENCODE to σ, and the function ψ_N on K_1 that $\psi_N(w)$ is a permutation obtained by applying N-DECODE to w. We then have the following lemma.

Lemma 6.2 *The function ϕ_N is a bijection between N and K_1.*

Proof. We first show that the image of ϕ_N is in K_1. Let σ be in N and $w = \phi_N(\sigma)$. It is clear that w is in Σ_1^*, so we need to show that w satisfies all conditions of K_1. We can easily verify that each case of Case 1 through Case 6 in Figure 5.20 corresponds to each prefix of ba, $bb'a$, $ba'a$, $ba'b_sa''$, $ba'b'a''$ and $ba'a'a''$ respectively. Similarly, each case of Case 1 through Case 6 in Figure 5.21 respectively corresponds to each suffix of cb, $cb''b$, $cc''b$, $c'b_sc''b$, $c'b''c''b$ and $c'c''c''b$.

For the condition on a' and a'', notice that a 231-value chain of the form $21 \oplus_1 21 \oplus_1 \cdots \oplus_1 21$ together with the scissor of the chain is encoded as $a'a'a''a'a'' \cdots a'a''b_sa''$ (or b' instead of b_s). Suppose there exists a summand 231 instead of 21, so the value playing the role of 2 is encoded as a. If the first summand is 231, then a appears immediately after a', corresponding to 1 of the same 231. Otherwise, due to the structure of a 231-value chain, a appears immediately after a'', corresponding to 2 of 21 or 3 of 231 of the previous summand. Since 231-value chains are the only ways for w to have a' and a'', the third condition is satisfied. Proving the fourth condition is satisfied is almost identical. The fifth condition is also immediate, since N-ENCODE is defined so that b_s is only used for scissors of 231-value chains and 312-value chains.

The condition on b' and b'' is tied with the sixth condition of Proposition 5.10, which states there are at most four values in between two values playing the roles of 1 and 2 of 12 in Equation 5.1. The positions of two values x_1 and x_2 ($x_1 < x_2$) are in $[1, \sigma^{-1}(n))$ and the positions of the other two values y_1 and y_2 ($y_1 < y_2$) are in $(\sigma^{-1}(1), n]$. Moreover, in the proof of Lemma 5.14, we showed that x_1 and y_2 respectively must be the maximum value of a 231-value chain and the minimum value of a 312-chain. Altogether, these values are encoded into a subword in $\{b'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{\beta''\}$. Also, since there must be at least one value in between two values playing the roles of 1 and 2 of 12 in Equation 5.1, one of a'', a, c and c' must be in between b' and b''. Thus, w satisfies the sixth condition of K_1.

Finally, for the seventh condition, suppose w contains aa. This implies two values for $1 \oplus 1$ in Equation 5.2 are consecutive, so it violates the first condition of Proposition 5.8. We can apply the same arguments for bb and cc with the second and third conditions of Proposition 5.8 respectively, so w does not contain aa, bb and cc. Consequently, the image of ϕ_N is in K_1.

To show that the image of ψ_N is in N, notice that together with how N-DECODE is defined and conditions of K_1, $\psi_N(w)$ is a permutation of extreme pattern 2413 and obeys all conditions of Proposition 5.8 and 5.10. Therefore, by Proposition 5.19, $\psi_N(w)$ is in N, so the image of ψ_N is in N.

Due to the constructions of ϕ_N and ψ_N, they are inverse to each other, so this completes the proof. ■

Next, we define S-ENCODE and S-DECODE, the algorithm to encode simple permutations in S, and vice versa. To make things less complicated, we simply define these algorithms as N-ENCODE and N-DECODE applied to the inverse permutations. To be precise, To S-ENCODE τ in S, we N-ENCODE τ^{-1}, and to S-DECODE a word w in K_1, we N-DECODE to obtain a permutation σ in N, then output σ^{-1}.

We define ϕ_S and ψ_S that $\phi_S(\tau)$ is a word obtained by applying S-ENCODE to a permutation τ in S, and $\psi_S(w)$ is a permutation obtained by applying S-DECODE to a word w in K_1. Thus, $\phi_N(\sigma)$ and $\phi_S(\tau)$ are equal if and only if σ and τ are inverse to each other. It is straightforward to prove the following lemma by definition of ϕ_S and Lemma 6.2.

Lemma 6.3 *The function ϕ_S is a bijection between S and K_1.*

Later with ϕ', we first decompose a permutation π in H' into the sum of m simple permutations in N and S, and apply N-ENCODE and S-ENCODE to each of these simple permutations alternatively. We abuse the notations of glue sums, and obtain $\overline{w} = w_1 \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_1}^{y_1} w_2 \mathbin{\bigcirc\mkern-14mu\searrow}_{x_2}^{y_2} w_3 \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_3}^{y_3} \cdots w_m$ to retain which glue sums were used in the expression of π. Thus, we obtain total m words in K_1 connected by glue sums. In particular, w_i for odd i is obtained by using N-ENCODE to each σ_i and w_i for even i is obtained by using S-ENCODE to each τ_i. To make it simpler, we define ϕ_1 on H' such that

$$\phi_1(\pi) = \begin{cases} \phi_N(\sigma_1) \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_1}^{y_1} \phi_S(\tau_2) \mathbin{\bigcirc\mkern-14mu\searrow}_{x_2}^{y_2} \cdots \mathbin{\bigcirc\mkern-14mu\searrow}_{x_{m-1}}^{y_{m-1}} \phi_N(\sigma_m) & \text{if } m \text{ is odd} \\ \phi_N(\sigma_1) \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_1}^{y_1} \phi_S(\tau_2) \mathbin{\bigcirc\mkern-14mu\searrow}_{x_2}^{y_2} \cdots \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_{m-1}}^{y_{m-1}} \phi_S(\tau_m) & \text{if } m \text{ is even} \end{cases}$$

Let $I_j = \{w_1 \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_1}^{y_1} w_2 \mathbin{\bigcirc\mkern-14mu\searrow}_{x_2}^{y_2} \cdots w_j : w_j \in K_1$ for all i with $1 \leq i \leq j\}$ and K_2 be the image of ϕ_1. We discuss that K_2 is a subset of $\bigcup_{j=1}^{m} I_j$, but not equal.

Recall that each glue sum requires summands to satisfy specific conditions. For example, type 1-0 and 1-1 require the left summand and the right summand to have Case 1 in Figure 5.21 and Case 1 in Figure 5.22 respectively. These restrictions coming from each glue sum force w_i and w_{i+1} to have specific suffix and prefix for all i $(1 \leq i \leq m-1)$. In particular, for every w in K_2, if the i-th glue sum is type 1-0 or 1-1, then w_i ends with cb and w_{i+1} begins with ba. If it is type 2-0 or 2-1, then w_i ends with $cb''b$ and w_{i+1} begins with ba. If it is type 3-0, then w_i ends with $cc''b$ and w_{i+1} begins with $ba'a$, and finally, if it is type 4-0, then w_i ends with $c'b_sc''b$ or $c'b''c''b$ and w_{i+1} begins with $bb'a$.

Now, we define $\Sigma_2 = \Sigma' \setminus \{d'_a, d''_a, d'_c, d''_c, d_\ell\}$. The next algorithm, called W-COMBINE iteratively connects these words to construct one word w in Σ_2^*. So let us define W-COMBINE as the following.

Algorithm W-COMBINE
INPUT: A sequence of words connected by glue sums as $\overline{w} = w_1 \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_1}^{y_1} w_2 \mathbin{\bigcirc\mkern-14mu\searrow}_{x_2}^{y_2} \cdots w_m$ in K_2.
OUTPUT: A word w in Σ_2^*.

```
Initialize: Let $w = w_1$ and write $\overline{w}$ as $w \mathbin{\bigcirc\mkern-14mu\nwarrow}_{x_1}^{y_1} w_2 \mathbin{\bigcirc\mkern-14mu\searrow}_{x_2}^{y_2} \cdots w_m$.

 Identify: If $\overline{w} = w$, then OUTPUT $w$. Otherwise, let $i+1$ be the least index of $w_k$ in $\overline{w}$.
     a. If $(x_i, y_i) = (1, 0)$ or $(x_i, y_i) = (1, 1)$, then GOTO Case 1.
     b. If $(x_i, y_i) = (2, 0)$ or $(x_i, y_i) = (2, 1)$, then GOTO Case 2.
```

c. If $(x_i, y_i) = (3, 0)$, then GOTO Case 3.

d. If $(x_i, y_i) = (4, 0)$, then GOTO Case 4.

Case 1: A suffix of w is cb and a prefix of w_{i+1} is ba.

a. If $(x_i, y_i) = (1, 0)$, replace the suffix cb of w with d and erase the prefix ba of w_{i+1}. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

b. If $(x_i, y_i) = (1, 1)$, replace the suffix b of w with d and erase the prefix ba of w_{i+1}. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

Case 2: A suffix of w is $cb''b$, $c\underline{b}b$ or $cy''b$, and a prefix of w_{i+1} is ba.

a. Suppose a suffix of w is $cb''b$.

i. If $(x_i, y_i) = (2, 0)$, replace the suffix $cb''b$ of w with $\underline{d}d$ and erase the prefix ba of w_{i+1}. Replace the last b' in w with $\overline{b}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

ii. If $(x_i, y_i) = (2, 1)$, replace the suffix $b''b$ of w with $\underline{d}d$ and erase the prefix ba of w_{i+1}. Replace the last b' in w with $\overline{b}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

b. Suppose a suffix of w is $c\underline{b}b$.

i. If $(x_i, y_i) = (2, 0)$, replace the suffix $c\underline{b}b$ of w with $\underline{d}d$ and erase the prefix ba of w_{i+1}. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

ii. If $(x_i, y_i) = (2, 1)$, replace the suffix $\underline{b}b$ of w with $\underline{d}d$ and erase the prefix ba of w_{i+1}. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

c. Suppose a suffix of w is $cy''b$.

i. If $(x_i, y_i) = (2, 0)$, replace the suffix $cy''b$ of w with $\underline{d}d$ and erase the prefix ba of w_{i+1}. Replace the last y' in w with $\overline{y}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

ii. If $(x_i, y_i) = (2, 1)$, replace the suffix $y''b$ of w with $\underline{d}d$ and erase the prefix ba of w_{i+1}. Replace the last y' in w with $\overline{y}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

Case 3: A suffix of w is $c'b_scc''b$, $c'b''cc''b$, $c'\underline{b}cc''b$, $c'c''cc''b$ or $c'y''cc''b$, and a prefix of w_{i+1} is $ba'avb_sa''$ or $ba'avb'a''$ where v is a word in $\{a'a'', a'a''a\}^*$.

For a suffix of w,

a. if it is $c'b_scc''b$, then replace it with zxd.

b. if it is $c'b''cc''b$, then replace it with $zx''d$ and the last b' in w with x'.

c. if it is $c'\underline{b}cc''b$, then replace it with $z\underline{x}d$.

d. if it is $c'c''cc''b$, then replace it with $zc''d$.

e. if it is $c'y''cc''b$, then replace it with $zy''d$.

For a prefix of w_{i+1},

a. if it is $ba'avb_sa''$ ($v \in \{a'a'', a'a''a\}^*$), then replace the whole prefix $ba'avb_sa''$ with y.

b. if it is $ba'avb'a''$, then replace the whole prefix $ba'avb'a''$ with y'. Additionally, replace the first b'' in w_{i+1} with y''.

Concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

Case 4: A suffix of w is $c'b_sc''b$, $c'b''c''b$, $c'\underline{b}c''b$ or $c'y''c''b$, and a prefix of w_{i+1} is $bb'a$.

a. If the suffix of w is $c'b_sc''b$, then replace the suffix $c'b_sc''b$ of w with $\overline{d}d$ and erase the prefix $bb'a$ of w_{i+1}. Replace the first b'' in w_{i+1} with $\underline{b}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

b. If the suffix of w is $c'b''c''b$, then replace the suffix $c'b''c''b$ of w with $\overline{\underline{d}}d$ and erase the prefix $bb'a$ of w_{i+1}. Replace the first b'' in w_{i+1} with $\underline{b}$. Replace the last b' in w with $\overline{b}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

c. If the suffix of w is $c'\underline{b}c''b$, then replace the suffix $c'\underline{b}c''b$ of w with $\overline{\underline{d}}d$ and erase the prefix $bb'a$ of w_{i+1}. Replace the first b'' in w_{i+1} with $\underline{b}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

d. If the suffix of w is $c'y''c''b$, then replace the suffix $c'y''c''b$ of w with $\overline{\underline{d}}d$ and erase the prefix $bb'a$ of w_{i+1}. Replace the last y' in w with $\overline{y}$ and the first b'' in w_{i+1} with $\underline{b}$. Then concatenate w with w_{i+1}. Redefine w to be the resultant word. Then GOTO Identify.

Let K_3 be the language over Σ_2 with the following conditions. We claim that K_3 is in bijection with K_2.

1. Prefix condition.

   A word w must begin with ba, $bb'a$, $ba'a$, $ba'b_sa''$, $ba'b'a''$, $ba'a'a''$, $b\overline{b}a$, $ba'\overline{b}a''$, $bx'a$ or $ba'x'a''$.

2. Suffix condition.

   A word w must end with cb, $cb''b$, $cc''b$, $c'b_sc''b$, $c'b''c''b$, $c'c''c''b$, $c\underline{b}b$, $c'\underline{b}c''b$, $cy''b$ or $c'y''c''b$.

3. Conditions on a' and a''.

   Every a' and a'' in w is a part of a subword in $\{a', a'a\}\{a'a'', a'a''a\}^*\{b_sa'', b'a'', \overline{b}a'', x'a''\}$. Note the number of a' and the number of a'' are equal in each sequence, and thus, in w.

4. Conditions on c' and c''.

   Every c' and c'' in w is a part of a subword in $\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*\{c'', cc'', zc'', czc''\}$. Note the number of c' and the number of c'' are equal in each sequence, and thus, in w.

5. Conditions on b_s.

   The letter b_s is only allowed in the third condition and the fourth condition, *i.e.* as a part of a subword of a' and a'' or a subword of c' and c''.

6. Conditions on b' and b''.

   Every b' and b'' in w is a part of a subword in $\{b'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{b''\}$ with at least one of a'', a, c or c' being present. Note the number of b' and the number of b'' are equal in each sequence, and thus, in w.

7. Conditions on letters with overlines and underlines.

   For every letter with an overline, there is a corresponding letter with an underline.

   - Every $\overline{b}$ in w is a part of a subword in $\{\overline{b}\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \overline{\underline{d}}d\}$.
   - Every $\overline{y}$ in w is a part of a subword in $\{\overline{y}\}\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \overline{\underline{d}}d\}$.
   - Every $\overline{d}$ and $\underline{d}$ in w is a part of a subword in one of the following.
     - $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{c', \lambda\}\{\underline{b}\}$.
     - $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{z\underline{x}\}$.
     - $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{\underline{d}d, \overline{\underline{d}}d\}$.

8. Conditions on x, y, z and other related letters.

   Every x, x', x'', $\underline{x}$, y, y', y'', $\overline{y}$ and z is a part of a subword v_1dv_2 where v_1 is in

   - $\{zx, z\underline{x}, zc'', zy''\}$ or
   - $\{x'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{zx''\}$, and

v_2 is in

- $\{y, \overline{y}\}$, or
- $\{y'\}\{a, \lambda\}\{c, \lambda\}\{y'', c'y'', zy''\}$.

9. Repetition restrictions.
   w must not contain aa, bb, cc or da.

Before we move onto the next algorithm W-DECOMPOSE, let us take a closer look at which letter is replaced by each of letters in $\{\underline{b}, \overline{b}, d, \underline{d}, \overline{d}, \overline{\underline{d}}, x, x', x'', \underline{x}, y, y', y'', \overline{y}, z\}$ when words are connected by W-COMBINE. For this observation, we examine the simplest case, which is applying W-COMBINE to $w_\sigma \obslash_x^y w_\tau$ where σ and τ be simple permutations of extreme pattern 2413 and 3142 respectively with $|\sigma| = m$ and $|\tau| = n$, and $w_\sigma = \phi_N(\sigma)$, $w_\tau = \phi_S(\tau)$.

When σ and τ are operated with $\obslash_1^0$ or $\obslash_1^1$, w_σ ends with cb which corresponds to values $\sigma(m)$ and m, and w_τ starts with ba which corresponds to $\tau(1)$ and 1. Recall that, for the case of $\obslash_1^0$, it eliminates the last value of σ and 1 of τ, and combines m and $\tau(1)$ as it shifts up accordingly. Thus, c of cb and a of ba are these eliminated values, and d takes the role of combined m and $\tau(1)$. If $\obslash_1^1$ is used instead, then c of cb stays there, as $\obslash_1^1$ does not erase the last value of σ. Therefore, the letter d is the only replacement happened in both cases. Technically, d represents m of σ and $\tau(1)$ of τ, but to refer back what the previous letter was, we set the convention to look at w_σ, not w_τ. Thus, d was originally encoded as b in w_σ.

If σ and τ are operated with $\obslash_2^0$ or $\obslash_2^1$, w_σ ends with $cb''b$, corresponding to values $\sigma(m) = m-2$, $m-1$ and m respectively, and w_τ starts with ba corresponding to $\tau(1)$ and 1. For $\obslash_2^0$, there are two things W-COMBINE does in the process of concatenation of w_σ and w_τ. First, where w_σ and w_τ are concatenated, we would have $cb''bba$, but W-COMBINE replace this with $\underline{d}d$. We may think that c and a are simply deleted, since the last value of σ and the first value of τ are eliminated. The first b is combined with the second b which results in d, and b'' is replaced with $\underline{d}$. Second, W-COMBINE replaces the letter b' that is matched up with b'' of $cb''b$ with $\overline{b}$. The only difference for $\obslash_2^1$ is that it does not erase c of $cb''b$ as $\sigma \obslash_2^1 \tau$ keeps the last value of σ. Hence, in w_σ, d was originally encoded as b, $\underline{d}$ was encoded as b'', and $\overline{b}$ was encoded as b'.

Next, we look at the case of $\obslash_3^0$. Recall that σ and τ must have a 312-value chain involving $\sigma(m)$ and a 312-position chain involving 1 respectively that are similar to each other. Thus, the last three greatest values of σ are placed as shown in Case 3 of 5.21 where $\sigma(m) = m-2$, $m-1$ and m are encoded by N-ENCODE as c, c'' and b respectively. For $m-3$, there are three cases.

If the 312-value chain is simply $\alpha = 312$, then $m-3$ would be the scissor of this chain. As the first case, if this scissor is 1 of 1 in Equation 5.1, then it is encoded as b_s. Alternatively, if it is 2 of 12 in Equation 5.1, then it is b''. The last case is $\alpha \neq 312$. Then $m-3$ is a part of the chain, and it is encoded as c''. In either case, $m-4$ must be 1 of the last 312 in the 312-value chain, so it is c'. Altogether, the five greatest values of σ are encoded as $c'b_scc''b$, $c'b''cc''b$ or $c'c''cc''b$. On the other hand, τ having the similar 312-position chain to start with, encoding of values of τ by S-ENCODE up to the end of the chain is either $ba'avb_sa''$ or $ba'avb'a''$ where $v \in \{a'a'', a'a''a\}^*$.

Now, for each case of w_σ having $c'b_scc''b$, $c'b''cc''b$ and $c'c''cc''b$, W-COMBINE replaces it with zxd, $zx''d$ and $zc''d$ respectively. In particular, we consider as follows.

- The first letter c' is replaced with z.
- The second letter is replaced with x, x'' and c'' accordingly (thus, in the case of c'', it remains unchanged). In case of replacing b'' with x'', W-COMBINE additionally replaces the paired b' with x'.
- The third letter c is erased as $\sigma \oslash^0_3 \tau$ eliminates $\sigma(m)$.
- The fourth letter c'' is merged into the scissor b_s of $ba'avb_sa''$ or b' of $ba'avb'a''$ in w_τ, which will be encoded latter.
- The last letter b is replaced with d.

Since the whole 312-position chain of τ is merged into the 312-value chain of σ by computing $\sigma \oslash^0_3 \tau$, every letter in $ba'avb_sa''$ or $ba'avb'a''$ of w_τ are erased, except the letter b_s or b' which corresponds to the scissor value. As this value gets combined with the value $m-1$ of σ, we replace it with the knew letter y or y' respectively. In case of replacing b' with y', we also replaces the paired b'' with y''. Hence, to summarize this case, previously in w_σ, z, x, x'', d, y and y' were all encoded as c', b_s, b'', b, b_s and b' respectively.

Finally, for $\oslash^0_4$, for the four greatest values of σ, w_σ can either have $c'b_sc''b$ or $c'b''c''b$, whereas w_τ must start with $bb'a$. Since $\oslash^0_4$ eliminates $\sigma(m-1) = m-1$ and $\sigma(m) = m-3$ of σ, c' and c'' of both suffixes of w_σ are erased. Also, $\oslash^0_4$ erases $\tau(1)$, so a of the prefix $bb'a$ of w_τ is deleted. The combination of m and $\tau(1)$, corresponding to b of w_σ and b of w_τ respectively, is replaced with d, just like in every other glue sum. If $m-2$ of σ is b_s, then it is replaced with $\overline{d}$ together with b' of w_τ. The value $\tau(2)$ corresponding to b' of $bb'a$ is 1 of 12 in Equation 5.1, and its paired 2 of 12 encoded as b'' is also replaced with $\underline{b}$. If $m-2$ of σ is b'', then it is replaced with $\underline{\overline{d}}$ instead. The letter b'' of w_τ that is paired with b' in the prefix is again replaced with $\underline{b}$, and in addition, b' that is paired with b'' in the suffix of w_σ is also replaced with $\overline{b}$. Therefore, in w_σ, $\overline{b}$, $\overline{d}$, $\overline{\underline{d}}$ and d were encoded as b', b_s, b'' and b respectively, and $\underline{b}$ in w_τ was encoded as b''.

The above observations are just for the case of $w_\sigma \text{\textcircled{$\scriptstyle\nwarrow$}}_x^y\, w_\tau$. As W-COMBINE recursively concatenates words, two consecutive glue sums sometimes result in the necessities of $\underline{x}$ and $\overline{y}$. For each, if we examine W-COMBINE closely, we find that they were originally b'' and b' respectively.

Next, let us define the inverse algorithm called W-DECOMPOSE on K_3, and we will prove K_2 and K_3 are in bijection due to these algorithms.

Algorithm W-DECOMPOSE
INPUT: A word w in K_3.
OUTPUT: A sequence of words connected by glue sums as $\overline{w} = w_1 \text{\textcircled{$\scriptstyle\nwarrow$}}_{x_1}^{y_1}\, w_2 \text{\textcircled{$\scriptstyle\searrow$}}_{x_2}^{y_2} \cdots w_m$ in $\bigcup_{j=1}^m I_j$.

```
Initialize: Let $m$ be the number of $d$ in $w$. Let $w'_m = w$ Also, write $\overline{w} = w'_m$.
 Identify: Let $i$ be the index such that
               $$\overline{w} = \begin{cases} w'_i \text{\textcircled{$\scriptstyle\nwarrow$}}_{x_i}^{y_i}\, w_{i+1} \text{\textcircled{$\scriptstyle\searrow$}}_{x_{i+1}}^{y_{i+1}} \cdots w_m & \text{if } i \text{ is odd} \\ w'_i \text{\textcircled{$\scriptstyle\searrow$}}_{x_i}^{y_i}\, w_{i+1} \text{\textcircled{$\scriptstyle\nwarrow$}}_{x_{i+1}}^{y_{i+1}} \cdots w_m & \text{if } i \text{ is even} \end{cases}$$
           If $i = 1$, then OUTPUT $\overline{w}$. Otherwise, locate the last $d$ in $w'_i$.
           a. If it is not a part of a subword $\underline{d}d$, $zsd$ ($s \in \{x, x'', \underline{x}, c'', y'\}$), $\overline{d}d$ or $\overline{\underline{d}}d$, then GOTO
              Case 1.
           b. If it is a part of a subword $\underline{d}d$, then GOTO Case 2.
           c. If it is a part of a subword $zsd$ where $s \in \{x, x'', \underline{x}, c'', y''\}$, then GOTO Case 3.
           d. If it is a part of a subword $\overline{d}d$ or $\overline{\underline{d}}d$, then GOTO Case 4.
   Case 1: Split $w'_i$ into $w'_{i-1}$ and $w_i$ by type 1-0 or 1-1 glue sum.
           Decatenate the suffix of $w'_i$ starting from the letter immediately after the last $d$. Concatenate $ba$ with this suffix of $w'_i$ and call it $w_i$. Let $w'_{i-1}$ be the remaining subword of $w'_i$.
           a. If the letter immediately before the last $d$ in $w'_{i-1}$ is not $c$, then replace the last $d$
              in $w'_{i-1}$ with $cb$.
               i. If $i$ is odd, then replace $w'_i$ in $\overline{w}$ with $w'_{i-1} \text{\textcircled{$\scriptstyle\searrow$}}_1^0\, w_i$. GOTO Identify.
              ii. If $i$ is even, then replace $w'_i$ in $\overline{w}$ with $w'_{i-1} \text{\textcircled{$\scriptstyle\nwarrow$}}_1^0\, w_i$. GOTO Identify.
           b. If the letter immediately before the last $d$ in $w'_{i-1}$ is $c$, then replace the last $d$ in
              $w'_{i-1}$ with $b$.
```

  - i. If i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\searrow}^1_1 w_i$. GOTO Identify.
  - ii. If i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}^1_1 w_i$. GOTO Identify.

Case 2: Split w'_i into w'_{i-1} and w_i by type 2-0 or 2-1 glue sum.

Decatenate the suffix of w'_i starting from the letter immediately after the last d. Concatenate ba with this suffix of w'_i and call it w_i. Let w'_{i-1} be the remaining subword of w'_i.

- a. Suppose the last letter with an overline in w'_{i-1} is $\overline{b}$. Replace this $\overline{b}$ with b'.
  - i. If the letter immediately before the last $\underline{d}$ in w'_{i-1} is not c, then replace the suffix $\underline{d}d$ in w'_{i-1} with $cb''b$.
    - A. If i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\searrow}^0_2 w_i$. GOTO Identify.
    - B. If i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}^0_2 w_i$. GOTO Identify.
  - ii. If the letter immediately before the last $\underline{d}$ in w'_{i-1} is c, then replace the suffix $\underline{d}d$ in w'_{i-1} with $b''b$.
    - A. If i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\searrow}^1_2 w_i$. GOTO Identify.
    - B. If i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}^1_2 w_i$. GOTO Identify.
- b. Suppose the last letter with an overline in w'_{i-1} is $\overline{d}$ or $\overline{\underline{d}}$.
  - i. If the letter immediately before the last $\underline{d}$ in w'_{i-1} is not c, then replace the suffix $\underline{d}d$ in w'_{i-1} with $c\underline{b}b$.
    - A. If i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\searrow}^0_2 w_i$. GOTO Identify.
    - B. If i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}^0_2 w_i$. GOTO Identify.
  - ii. If the letter immediately before the last $\underline{d}$ in w'_{i-1} is c, then replace the suffix $\underline{d}d$ in w'_{i-1} with $\underline{b}b$.
    - A. If i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\searrow}^1_2 w_i$. GOTO Identify.
    - B. If i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}^1_2 w_i$. GOTO Identify.
- c. Suppose the last letter with an overline in w'_{i-1} is $\overline{y}$. Replace this $\overline{y}$ with y'.
  - i. If the letter immediately before the last $\underline{d}$ in w'_{i-1} is not c, then replace the suffix $\underline{d}d$ in w'_{i-1} with $cy''b$.
    - A. If i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\searrow}^0_2 w_i$. GOTO Identify.
    - B. If i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\hspace{-0.8em}\nwarrow}^0_2 w_i$. GOTO Identify.
  - ii. If the letter immediately before the last $\underline{d}$ in w'_{i-1} is c, then replace the suffix $\underline{d}d$ in w'_{i-1} with $y''b$.

A. If i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\mathrlap{\bigcirc}{\searrow}}^{1}_{2} w_i$. GOTO Identify.

B. If i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\mathrlap{\bigcirc}{\nwarrow}}^{1}_{2} w_i$. GOTO Identify.

Case 3: Split w'_i into w'_{i-1} and w_i by type 3-0 glue sum.

Decatenate the suffix of w'_i starting from the letter immediately after the last d and call it v. Let w'_{i-1} be the remaining subword of w'_i.

- If v starts with y, replace this y with $b_s a''$.
- If v starts with y', replace this y' with $b'a''$. Additionally, replace y'' in v with b''.

To construct w_i,

a. if w'_{i-1} has a suffix zxd, $zx''d$, $z\underline{x}d$ or $zy''d$, concatenate $ba'a$ with v, and call it w_i.

b. if w'_{i-1} has a suffix $zc''d$, take the whole sequence of c' and c'' which zc'' of $zc''d$ is a part of, and call it t. Thus, t is a subword of w'_{i-1} in $\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*$ $\{zc'', czc''\}$. Let $u = ba'a$. After the prefix $c'b_s$, $c'b''$, $c'\underline{b}$ or $c'y''$ in t, for every $c'c''$ and zc'', concatenate u with $a'a''$, and for every $cc'c''$ and czc'', concatenate u with $a'a''a$. Once this procedure is completed, concatenate u with v, and call it w_i.

To complete w'_{i-1},

a. if w'_{i-1} has a suffix zxd, then replace it with $c'b_scc''b$.

b. if w'_{i-1} has a suffix $zx''d$, then replace it with $c'b''cc''b$ and the last x' in w'_{i-1} with b'.

c. if w'_{i-1} has a suffix $z\underline{x}d$, then replace it with $c'\underline{b}cc''b$.

d. if w'_{i-1} has a suffix $zc''d$, then replace it with $c'c''cc''b$.

e. if w'_{i-1} has a suffix $zy''d$, then replace it with $c'y''cc''b$.

Then, finally,

- if i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\mathrlap{\bigcirc}{\searrow}}^{0}_{3} w_i$. GOTO Identify.
- if i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\mathrlap{\bigcirc}{\nwarrow}}^{0}_{3} w_i$. GOTO Identify.

Case 4: Split w'_i into w'_{i-1} and w_i by type 4-0 glue sum.

Decatenate the suffix of w'_i starting from the letter immediately after the last d. Concatenate $bb'a$ with this suffix of w'_i and call it w_i. Replace $\underline{b}$ with b''. Let w'_{i-1} be the remaining subword of w'_i. For w'_{i-1},

a. if its suffix is $\overline{d}d$, then replace it with $c'b_sc''b$.

b. if its suffix is $\underline{\overline{d}}d$, observe the last letter with an overline in w'_{i-1}.

i. If the last letter with an overline in w'_{i-1} is $\overline{b}$, then replace it with b' and the suffix $\underline{\overline{d}}d$ of w'_{i-1} with $c'b''c''b$.

ii. If the last letter with an overline in w'_{i-1} is $\overline{d}$ or $\overline{\underline{d}}$, then replace the suffix $\overline{\underline{d}}d$ of w'_{i-1} with $c'\underline{d}c''b$.

iii. If the last letter with an overline in w'_{i-1} is $\overline{y}$, then replace it with y' and the suffix $\overline{\underline{d}}d$ of w'_{i-1} with $c'y''c''b$.

Then,

- if i is odd, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\kern-0.8em\searrow}^{0}_{4} w_i$. GOTO Identify.
- if i is even, then replace w'_i in $\overline{w}$ with $w'_{i-1} \mathbin{\bigcirc\kern-0.8em\nwarrow}^{0}_{4} w_i$. GOTO Identify.

We define ϕ_2 on K_2 that $\phi_2(\overline{w}) = w$ is obtained by applying N-COMBINE to an arbitrary $\overline{w} \in K_2$, and ψ_2 on K_3 that $\psi_2(w) = \overline{w}$ is obtained by applying W-DECOMPOSE to an arbitrary $w \in K_3$. By using these two functions, we prove the following.

Lemma 6.4 *The function ϕ_2 is a bijection between K_2 and K_3.*

Proof. We first claim that the image of ϕ_2 is in K_3. In each w_i of $\overline{w} = w_1 \mathbin{\bigcirc\kern-0.8em\nwarrow}^{y_1}_{x_1} w_2 \mathbin{\bigcirc\kern-0.8em\searrow}^{y_2}_{x_2} \cdots w_m$, there are no aa, bb and cc by definition of K_1. Thus, if w contains any of aa, bb, cc or da, then this is caused by the process of W-COMBINE.

For w_i and w_{i+1} in $\overline{w}$ to be combined, a certain suffix of w_i is modified to a sequence of letters ending with d first in each case of W-COMBINE. After that, a certain prefix of w_{i+1} is either erased (Case 1, Case 2 and Case 4) or replaced (Case 3), and then connected to w_i. Erased prefixes are either ba or $bb'a$, so the letter after these prefixes cannot be a. Also, in the case of replacement, it is always with y or y'. Thus, in all cases, da is not contained in w.

In some cases, such as Case 2,a.i, b', b'' or y' is replaced with another letter. However, none of these is replaced with a, b, c or d, so it is impossible for this arrangement to cause w containing aa, bb, cc or da. Hence, $\phi_2(\overline{w})$ satisfies the ninth condition of K_3.

The other conditions of K_3 can be verified by observing each case of W-COMBINE. Thus, the image of ϕ_2 is in K_3.

Next, we show that the image of ψ_2 is in K_2. Frankly, the algorithm W-DECOMPOSE reads a word w from right to left and splits w every time d occurs. Based on the conditions of K_3, all possible cases of subwords that d can be a part of are $\underline{d}d$, zsd where $s \in \{x, x'', \underline{x}, c'', y''\}$, $\overline{d}d$, $\overline{\underline{d}}d$ and sd where $s \in \{a, a'', b, b'', c, c''\}$. In each case, W-DECOMPOSE decatenates w_i as all letters in $\Sigma_2 \setminus \Sigma_1$ are replaced with letters in Σ_1. We can observe that the word decatenated by W-DECOMPOSE is in K_1 for every case, implying the whole $\overline{w}$ is in K_2. Thus, the image of ψ_2

is in K_2.

Due to the constructions of ϕ_2 and ψ_2, they are inverse to each other, so ϕ_2 is a bijection from K_2 to K_1. ■

We are only one step away to the regular language L' which is bijective to H'. Our final algorithm is to modify a prefix and a suffix of w in K_3, which is the following.

Algorithm AFFIX-CONVERT

```
INPUT: A word $w$ in $K_3$.
OUTPUT: A word in $\Sigma'^*$.

GOTO Prefix.

Prefix: Convert the prefix of $w$.
    • If it is $ba$, then replace it with $dd$. GOTO Suffix.
    • If it is $bb'a$, then replace it with $db'd$. GOTO Suffix.
    • If it is $ba'a$, then replace it with $da'd$. GOTO Suffix.
    • If it is $ba'b_sa''$, then replace it with $dd'_ab_sd''_a$. GOTO Suffix.
    • If it is $ba'b'a''$, then replace it with $dd'_ab'd''_a$. GOTO Suffix.
    • If it is $ba'a'a''$, then replace it with $dd'_aa'd''_a$. GOTO Suffix.
    • If it is $b\bar{b}a$, then replace it with $d\bar{b}d$. GOTO Suffix.
    • If it is $ba'\bar{b}a''$, then replace it with $dd'_a\bar{b}d''_a$. GOTO Suffix.
    • If it is $bx'a$, then replace it with $dx'd$. GOTO Suffix.
    • If it is $ba'x'a''$, then replace it with $dd'_ax'd''_a$. GOTO Suffix.

Suffix: Convert the suffix of $w$.
    • If it is $cb$, then replace it with $dd_\ell$. OUTPUT $w$.
    • If it is $cb''b$, then replace it with $db''d_\ell$. OUTPUT $w$.
    • If it is $cc''b$, then replace it with $dc''d_\ell$. OUTPUT $w$.
    • If it is $c'b_sc''b$, then replace it with $d'_cb_sd''_cd_\ell$. OUTPUT $w$.
    • If it is $c'b''c''b$, then replace it with $d'_cb''d''_cd_\ell$. OUTPUT $w$.
    • If it is $c'c''c''b$, then replace it with $d'_cc''d''_cd_\ell$. OUTPUT $w$.
    • If it is $c\underline{b}b$, then replace it with $d\underline{b}d_\ell$. OUTPUT $w$.
```

- If it is $c'\underline{b}c''b$, then replace it with $d'_c\underline{b}d''_cd_\ell$. OUTPUT w.
- If it is $cy''b$, then replace it with $dy''d_\ell$. OUTPUT w.
- If it is $c'y''c''b$, then replace it with $d'_cy''d''_cd_\ell$. OUTPUT w.

Define ϕ_3 be a function on K_3 induced by AFFIX-CONVERT. By how AFFIX-CONVERT is defined, it is not difficult to verify that the image of ϕ_3 is equal to the language L' over Σ' with following conditions.

1. Prefix condition.

   A word w must begin with dd, $db'd$, $da'd$, $dd'_ab_sd''_a$, $dd'_ab'd''_a$, $dd'_aa'd''_a$, $d\overline{b}d$, $dd'_a\overline{b}d''_a$, $dx'd$ or $dd'_ax'd''_a$. Therefore,

   (a) $da'd$ is followed by a subword in $\{a'a'', a'a''a\}^*\{b_sa'', b'a'', \overline{b}a'', x'a''\}$.

   (b) $dd'_aa'd''_a$ is followed by a subword in $\{a, \lambda\}\{a'a'', a'a''a\}^*\{b_sa'', b'a'', \overline{b}a'', x'a''\}$.

   (c) $db'd$ is followed by a subword in $\{c, \lambda\}\{c', \lambda\}\{b''\}$ or the suffix $db''d_\ell$, $d'_cb''d''_cd_\ell$.

   (d) $dd'_ab'd''_a$ is followed by a subword in $\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{b''\}$ or the suffix $db''d_\ell$, $d'_cb''d''_cd_\ell$.

   (e) $d\overline{b}d$ is followed by a subword in $\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$.

   (f) $dd'_a\overline{b}d''_a$ is followed by a subword in $\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$.

   (g) $dx'd$ is followed by a subword v_1dv_2 where v_1 is in $\{c, \lambda\}\{zx''\}$ and v_2 is in $\{y, \overline{y}\}$ or $\{y'\}\{a, \lambda\}\{c, \lambda\}\{y'', c'y'', zy''\}$.

   (h) $dd'_ax'd''_a$ is followed by a subword v_1dv_2 where v_1 is in $\{a, \lambda\}\{c, \lambda\}\{zx''\}$ and v_2 is in $\{y, \overline{y}\}$ or $\{y'\}\{a, \lambda\}\{c, \lambda\}\{y'', c'y'', zy''\}$.

   Letters d'_a and d''_a are only permitted in the above listed prefixes.

2. Suffix condition.

   A word w must end with dd_ℓ, $db''d_\ell$, $dc''d_\ell$, $d'_cb_sd''_cd_\ell$, $d'_cb''d''_cd_\ell$, $d'_cc''d''_cd_\ell$, $d\underline{b}d_\ell$, $d'_c\underline{b}d''_cd_\ell$, $dy''d_\ell$ or $d'_cy''d''_cd_\ell$. Therefore,

   (a) $dc''d_\ell$ is preceded by a subword in $\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*$.

   (b) $d'_cc''d''_cd_\ell$ is preceded by a subword in $\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*\{c, \lambda\}$.

   (c) $db''d_\ell$ is preceded by a subword in $\{b'\}\{a'', \lambda\}\{a, \lambda\}$ or the prefix $db'd$, $dd'_ab'd''_a$.

   (d) $d'_cb''d''_cd_\ell$ is preceded by a subword in $\{b'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}$ or the prefix $db'd$, $dd'_ab'd''_a$.

   (e) $d\underline{b}d_\ell$ is preceded by $\overline{d}d$ or $\underline{\overline{d}}d$.

   (f) $d'_c\underline{b}d''_cd_\ell$ is preceded by a subword in $\{\overline{d}d, \underline{\overline{d}}d\}\{c, \lambda\}$.

   (g) $dy''d_\ell$ is preceded by a subword v_1dv_2 where v_1 is in $\{zx, z\underline{x}, zc'', zy''\}$ or $\{x'\}\{a'', \lambda\}$ $\{a, \lambda\}\{c, \lambda\}\{zx''\}$ and v_2 is y' or $y'a$.

(*h*) $d'_c y'' d''_c d_\ell$ is preceded by a subword $v_1 d v_2$ where v_1 is in $\{zx, z\underline{x}, zc'', zy''\}$ or $\{x'\}\{a'', \lambda\}$ $\{a, \lambda\}\{c, \lambda\}\{zx''\}$ and v_2 is in $\{y'\}\{a, \lambda\}\{c, \lambda\}$.

Letters d'_c, d''_c and d_ℓ are only permitted in the above listed suffixes. In particular, every w must end with d_ℓ.

3. Conditions on a' and a''.

   Every a' and a'' in w is a part of a prefix described in (*a*) or (*b*) in the prefix condition, or a part of a subword in $\{a', a'a\}\{a'a'', a'a''a\}^*\{b_s a'', b'a'', \overline{b}a'', x'a''\}$. Note the numbers of a' and a'' are the same in each sequence, and thus, in the entire w.

4. Conditions on c' and c''.

   Every c' and c'' in w is a part of a suffix described in (*a*) or (*b*) in the suffix condition, or a part of a subword in $\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*\{c'', cc'', zc'', czc''\}$. Note the numbers of c' and c'' are the same in each sequence, and thus, in the entire w.

5. Conditions on b_s.

   The letter b_s is only allowed in the prefix $dd'_a b_s d''_a$, in the suffix $d'_c b_s d''_c d_\ell$, as a part of a subword of a' and a'' listed in the third condition, or as a part of a subword of c' and c'', listed in the fourth condition.

6. Conditions on b' and b''.

   Every b' and b'' in w is a part of a prefix described in (*c*) or (*d*) in the prefix condition, a part of a suffix described in (*c*) or (*d*) in the suffix condition, or a part of a subword in $\{b'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{b''\}$ with at least one of a'', a, c or c' being present. Note the numbers of b' and b'' are the same in each sequence, and thus, in the entire w.

7. Conditions on letters with overlines and underlines.

   For every letter with an overline, there is a corresponding letter with an underline.

   - Every $\overline{b}$ in w is a part of a prefix described in (*e*) or (*f*) in the prefix condition, or a part of a subword in $\{\overline{b}\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$.
   - Every $\overline{y}$ in w is a part of a subword in $\{\overline{y}\}\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$.
   - Every $\overline{d}$ and $\underline{\overline{d}}$ in w is a part of a subword in one of the following.
     - $\{\overline{d}d, \underline{\overline{d}}d\}\{d\underline{b}d_\ell\}$.
     - $\{\overline{d}d, \underline{\overline{d}}d\}\{c, \lambda\}\{d'_c\underline{b}d''_c d_\ell\}$.
     - $\{\overline{d}d, \underline{\overline{d}}d\}\{c, \lambda\}\{c', \lambda\}\{\underline{b}\}$.
     - $\{\overline{d}d, \underline{\overline{d}}d\}\{c, \lambda\}\{z\underline{x}\}$.
     - $\{\overline{d}d, \underline{\overline{d}}d\}\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$.

8. Conditions on x, y, z and other related letters.

   In addition to (g), (h) in the prefix condition and (g), (h) in the suffix condition, every x, x', x'', $\underline{x}$, y, y', y'', $\overline{y}$ and z is a part of a subword v_1dv_2 where v_1 is in

   - $\{zx, z\underline{x}, zc'', zy''\}$ or
   - $\{x'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{zx''\}$, and

   v_2 is in

   - $\{y, \overline{y}\}$, or
   - $\{y'\}\{a, \lambda\}\{c, \lambda\}\{y'', c'y'', zy''\}$.

9. Other restrictions.

   w must not contain aa, bb, cc, da and cdd_ℓ.

Finally, with every bijection we defined, let us define ϕ' on H' as $\phi' = \phi_3 \circ \phi_2 \circ \phi_1$. Then, we have the following proposition. The proof is immediate with the lemmas we established.

Proposition 6.5 *The encoding function ϕ' is a bijection between H' and L'.*

To visualize the encoding of a permutation in H' to L', we refer to the Python code provided in Appendix B.

6.1.2. Defining the automaton M'

Recall that we defined the language $\overline{L}$ associated with L in Chapter 4. While every word in L must begin with dd, we defined $\overline{L}$ to be the set of words in L without the prefix dd, and showed $\overline{L}$ is regular.

For L', words have 10 distinct prefixes as described. We prepare associated 10 languages $\overline{L}_i$ $(1 \leq i \leq 10)$, one for each prefix, and construct an automaton for each of these languages to show that they are all regular languages. For each i with $1 \leq i \leq 10$, we define $\overline{L}_i$ as the following.

$$\begin{aligned}
&\overline{L}_1 = \{w \in \Sigma'^* : ddw \in L'\}, && \overline{L}_2 = \{w \in \Sigma'^* : dd'_a b_s d''_a w \in L'\}, \\
&\overline{L}_3 = \{w \in \Sigma'^* : da'dw \in L'\}, && \overline{L}_4 = \{w \in \Sigma'^* : dd'_a a' d''_a w \in L'\}, \\
&\overline{L}_5 = \{w \in \Sigma'^* : db'dw \in L'\}, && \overline{L}_6 = \{w \in \Sigma'^* : dd'_a b' d''_a w \in L'\}, \\
&\overline{L}_7 = \{w \in \Sigma'^* : dx'dw \in L'\}, && \overline{L}_8 = \{w \in \Sigma'^* : dd'_a x' d''_a w \in L'\}, \\
&\overline{L}_9 = \{w \in \Sigma'^* : d\overline{b}dw \in L'\}, && \overline{L}_{10} = \{w \in \Sigma'^* : dd'_a \overline{b} d''_a w \in L'\}.
\end{aligned}$$

Each $\overline{L}_i$ is a language over $\overline{\Sigma} = \Sigma' \setminus \{d'_a, d''_a\}$ that shares all conditions of L', except for the prefix condition. Since an arbitrary w in each $\overline{L}_i$ is a word in L' without the certain prefix, the prefix condition for each $\overline{L}_i$ shall be stated with what the new prefix has to be. Thus, the prefix condition for $\overline{L}_i$ $(1 \leq i \leq 10)$ is as the following.

L_1: The first letter must be in $\{a', b, b', \overline{b}, c, c', d, d'_c, \overline{d}, x', z\}$.

L_2: The first letter must be in $\{a, a', b, b', \overline{b}, c, c', d, d'_c, \overline{d}, x', z\}$.

L_3: A word w must begin with a subword in $\{a'a'', a'a''a\}^*\{b_s a'', b'a'', \overline{b}a'', x'a''\}$.

L_4: A word w must begin with a subword in $\{a, \lambda\}\{a'a'', a'a''a\}^*\{b_s a'', b'a'', \overline{b}a'', x'a''\}$.

L_5: A word w must begin with a subword in $\{c, \lambda\}\{c', \lambda\}\{b''\}$ or $w = db''d_\ell$, $d'_c b'' d''_c d_\ell$.

L_6: A word w must begin with a subword in $\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{b''\}$ or $w = db''d_\ell$, $d'_c b'' d''_c d_\ell$.

L_7: A word w must begin with a subword $v_1 d v_2$ where v_1 is in $\{c, \lambda\}\{zx''\}$ and v_2 is in $\{y, \overline{y}\}$ or $\{y'\}\{a, \lambda\}\{c, \lambda\}\{y'', c'y'', zy''\}$.

L_8: A word w must begin with a subword $v_1 d v_2$ where v_1 is in $\{a, \lambda\}\{c, \lambda\}\{zx''\}$ and v_2 is in $\{y, \overline{y}\}$ or $\{y'\}\{a, \lambda\}\{c, \lambda\}\{y'', c'y'', zy''\}$.

L_9: A word w must begin with a subword in $\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$.

L_{10} A word w must begin with a subword in $\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$.

Next, we define 10 deterministic finite-state automatons M'_i $(1 \leq i \leq 10)$, and show that $\mathcal{L}(M'_i) = \overline{L}_i$ for each i. Before we do so, we note that the only difference among each M'_i is the initial state. Each automaton runs over Σ' and shares the same set of states, transition function and accept state. The set of states Q' for each M'_i contains 83 distinct states. For this reason, it is unarguably not reasonable to present their transition function as a diagram. The description of the transition function is given in Table A.1 in Appendix A. We ask readers to refer to the same table to see each of 83 states in as well.

Let us now introduce M'_i for each i with $1 \leq i \leq 10$. For every i $(1 \leq i \leq 10)$, we define an automaton $M'_i = (Q', \Sigma', \delta', q_i, \{D_\ell\})$ where Q contains all states presented in Table A.1, δ' is the transition function described in Table A.1 in Appendix A, and q_i is the initial state such that

$$\begin{array}{lllll} q_1 = A & q_2 = A'' & q_3 = A[A] & q_4 = A''[A] & q_5 = A[B] \\ q_6 = A''[B] & q_7 = A[X] & q_8 = A''[X] & q_9 = \overline{B}A & q_{10} = \overline{B}A'' \end{array}$$

Before we show $\mathcal{L}(M'_i) = \overline{L}_i$ for each i, we give a general description of each state in Q. Frankly, every state is denoted based on which letter in Σ' is previously used to obtain the state.

A state in

$$\{A, A'', B, B'', \underline{B}, \overline{B}, C, C'', D, D''_c, \underline{D}, \overline{D}, \underline{\overline{D}}, D_\ell, X, X'', \underline{X}, Y, Y'', \overline{Y}, Z\}$$

is denoted by a single capital letter, which indicates the transition that was used to arrive at the state. For instance, $\underline{D} = \delta(q, \underline{d})$ for some state $q \in Q$.

Many states come with square brackets with one or two capital letters inside, such as $A'[A]$ and $A'[A, A]$. The letter outside of the brackets indicates that the associated lower case letter in Σ' that was just used for transition to arrive at the state. Each letter in the brackets, on the other hand, states that there must eventually be a transition using the associated lower case letter with a double prime. The appearance of a letter in square brackets is initially caused by a transition with a prime sign in Σ' such as a' and b'. For example, from a state with no brackets, say A, we have $\delta'(A, b') = B'[B]$ as described in Table A.1. Until its paired b'' is used for transition, the letter B is shown in brackets of the states following $B'[B]$. Indeed, from $B'[B]$, we eventually arrive at B'', $B''[C]$, DB'' or $B''[D]$ by b'' from a previous state as shown in Figure 6.1. Note that labels of transitions are omitted in Figure 6.1.

Notice that, as we see in $\delta'(B'[B], c') = C'[B, C]$, it is possible to have another transition with a prime sign until we reach a state with B'' involved. In this case, we also must have the transition c'' eventually, so we include C in the square brackets as well.

Next, we give a description for states denoted by two consecutive upper case letters. Namely, the list of these states are given as follows.

$$\{\overline{B}A, \overline{B}A'', \overline{B}C, CD, DB'', D\underline{B}, DC'', \underline{D}D, \overline{D}D, DY'', XD, ZC'', ZY''\}$$

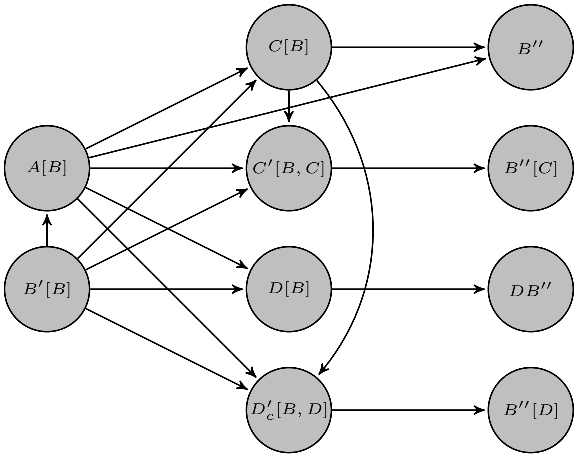


Figure 6.1.: Partial state diagram of M'.

The transition to a state q in the above set is the lower case of the second letter in the expression of q. For instance, to arrive at ZC'', we use c'' from some appropriate state. On the other hand, the first letter does not indicate anything relevant about a transition. One may think that the first letter informs which letter was used to obtain the state immediately before the current state, but this is only true for some cases. For example, the only way to arrive at $\overline{B}A''$ is from $\overline{B}[A]$ with the transition a''. From here, it is possible to go to $\overline{B}A$ with the transition a. As we can see, the previous transition a'' is not recorded in the notation $\overline{B}A$.

Finally, we explain the states with arrows involved, that is, the states in the set

$$\{C \to \{\underline{B}, \underline{D}, \underline{\overline{D}}, \underline{X}\}, C'[C] \to \underline{B}[C], D \to \underline{B}, D'_c[D] \to \underline{B}[D], Z \to \underline{X}\}.$$

Each state in this set is denoted based on which transition with an underline is used in the future. In order to arrive at a state in the above set, we must first obtain $\overline{D}D$. Once the transitions $\overline{d}$ or $\underline{d}$ occur, we obtain $\overline{D}$ or $\underline{D}$ respectively, and as a result, we arrive at $\overline{D}D$. From here, we are required to have either $\underline{b}$, $\underline{d}$, $\underline{\overline{d}}$ or $\underline{x}$ in the near future. Hence, we arrive at $\underline{B}$, $\underline{B}[C]$, $\underline{B}[D]$, $\underline{D}$, $\underline{\overline{D}}$ or $\underline{X}$. These states with arrows are necessary to distinguish them from other states, which do not require future transitions to underlined sates.

In each expression of a state with an arrow, the letter at the tail of the arrow indicates the associated transition to reach there. To reach $C \to \{\underline{B}, \underline{D}, \underline{\overline{D}}, \underline{X}\}$, for example, we must have c from $\overline{D}D$. The state, or the set of states at the head of the arrow shows which transition with an underline will be possibly used. This transition with an underline, however, does not have to occur immediately from the state with an arrow, as we can observe with $\delta'(C \to \{\underline{B}, \underline{D}, \underline{\overline{D}}, \underline{X}\}, c') = C'[C] \to \underline{B}[C]$. Square brackets are used for the same purpose as before.

We are now ready to prove the following proposition.

Proposition 6.6 *For every i with $1 \leq i \leq 10$, $\mathcal{L}(M'_i) = \overline{L}_i$.*

Proof. We first show that $\mathcal{L}(M'_i) \subseteq \overline{L}_i$ for each i with $1 \leq i \leq 10$. We define the set of letters Γ as

$$\Gamma = \{a, a', b, b', \overline{b}, c, c', d, d'_c, \overline{d}, x', z\}.$$

We also let P' be a subset of Q' defined as $P' = \{A, A'', B, B'', \underline{B}, C, C'', CD, D, \underline{D}D, Y, Y''\}$.

Let i be arbitrary with $1 \leq i \leq 10$. In M'_i, we verify that every sequence of transitions from a state to a non-jail state constructs a subword obeying the conditions of $\overline{L}_i$. As we go on, we also make sure that the initial state for each M_i' is the appropriate one.

Let w be in $\mathcal{L}(M_i')$. Due to certain conditions of $\overline{L}_i$, for the letters in

$$\overline{\Sigma} \setminus \Gamma = \{a'', b'', b_s, \underline{b}, c'', d_c'', \underline{d}, \overline{d}, d_\ell, x, \underline{x}, y, y', y'', \overline{y}\}$$

to appear in w, it requires some specific letters to be previously appearing in w. For instance, b'' is not allowed to appear in w unless there is an unmatched b' previously showing up in w. Hence, the letters in Γ are the ones which do not require any particular preceding letters in w.

On the other hand, the letters in the set

$$\Delta = \{a, a', b, b', b_s, \overline{b}, c, c', d, d_c', d_c'', \underline{d}, \overline{d}, \overline{\underline{d}}, x, x', x'', \underline{x}, y', \overline{y}, z\}$$

restricts what the next letter or the next few letters can be. For instance, a cannot occur after a, a sequence of letters following b' must be in $\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{b''\}$, and so forth. Therefore, when there is no restriction coming from other letters, we are allowed to have any letter in Γ from a letter in $\Delta^c \setminus \{d_\ell\} = \{a'', b'', \underline{b}, c'', y, y''\}$. Achieving each letter in $\Delta^c \setminus \{d_\ell\}$ when there is no restriction corresponds to the states A'', B'', $\underline{B}$, C'', Y and Y'' respectively. In Table A.1, we can see that from each one of these states, we have every transition of letters in Γ transiting to a non-jail states. For $\mathcal{L}(M_2')$, since w in $\overline{L}_2$ must begin with a letter in Γ, it is appropriate to have A'' as its initial state.

In addition to letters in $\Delta^c \setminus \{d_\ell\}$, the only restriction what a letter after a can be is that it cannot be a due to the ninth condition of L'. Indeed, every letter in $\Gamma\{a\}$ is a transition from A to a non-jail state. This also verifies that the initial state of $\mathcal{L}(M_1')$ being A is valid. Similarly, we can confirm transitions from B, C, D and $\underline{D}D$ are all valid as well. From D, we can transit to D_ℓ with d_ℓ, ending w with dd_ℓ which satisfies the suffix condition of $\overline{L}_i$. The state $CD = \delta'(C, d)$ is to avoid having d_ℓ after having cd, as cdd_ℓ is forbidden in L'. Except that we cannot have the transition d_ℓ from CD, other transitions are identical to the ones from D, verifying the row of CD.

From any of the states we have discussed so far, having d_c' will take us to $D_c'[D]$. From here, the only path is to $B_s[D]$, D_c'', and then D_ℓ, which gives us the suffix $d_c'b_sd_c''d_\ell$. This is one of the acceptable suffixes of $\overline{L}_i$, so the transitions from $D_c'[D]$ and $B_s[D]$ are confirmed.

Next, we examine transitions involving states having $[A]$ in their expressions. Except for the initial states of $\mathcal{L}(M_3')$ and $\mathcal{L}(M_4')$, in order to enter states with $[A]$, we always must go to $A'[A]$ first, since $[A]$ indicates that a' has been mentioned and currently unmatched. On the other hand, to exit out of states with $[A]$, we must have a'' from either $B_s[A]$, $\overline{B}[A]$, $B'[A,B]$ or $X'[A,X]$ to transit to B'', $\overline{B}A''$, $A''[B]$ or $A''[X]$ respectively. Before we arrive at any of $B_s[A]$, $\overline{B}[A]$, $B'[A,B]$ and $X'[A,X]$, we can bypass through states $A[A]$, $A'[A,A]$, $A''[A]$ in this order finite number of times with $A[A]$ being optional. In this case, we can transit out to $B_s[A]$, $\overline{B}[A]$, $B'[A,B]$ or $X'[A,X]$ from $A[A]$ or $A''[A]$, but not $A'[A,A]$. Observing these cases, the set of the sequence of letters allowed by these transitions is

$$\{a'\}\{aa'a'', a'a''\}^*\{a, \lambda\}\{b_s a'', b'a'', \overline{b}a'', x'a''\} = \{a', a'a\}\{a'a'', a'a''a\}^*\{b_s a'', b'a'', \overline{b}a'', x'a''\},$$

which is the set of subwords of a word in L' containing a' and a'' as it is listed in Condition 3 in the definition of L'. We can also verify that the initial states of $\mathcal{L}(M_3')$ and $\mathcal{L}(M_4')$ are appropriate with the prefix conditions of $\overline{L}_3$ and $\overline{L}_4$.

We now look at transitions from states with $[B]$. Besides the initial states of $\mathcal{L}(M_5')$ and $\mathcal{L}(M_6')$, two ways to enter states with $[B]$ are either by transiting to $B'[A,B]$ from $A[A]$, $A'[A]$ or $A''[A]$, or by transiting to $B'[B]$ from any other states that allows the transition b'. As previously observed, from $B'[A,B]$, we only have one transition available, which is a'' to move to $A''[B]$. Now, whether from $A''[B]$ or $B'[B]$, we must arrive at B'', $B''[C]$, $B''[D]$ or DB'' to exit out of states with $[B]$. Until we arrive at B'', we can bypass through $A[B]$ and $C[B]$ in that order. From $A'[B]$, going through $A[B]$ and $C[B]$ are unnecessary. On the other hand, if we are coming from $B'[B]$, stopping at one of them is necessary, since we do not have the transition b'' from $B'[B]$. Similarly, we can observe the possibilities to arrive at each one of $B''[C]$, $B''[D]$ and DB'' from $A''[B]$ and $B'[B]$ to see all possible subwords induced by the transitions after b' up to b'' are in one of the following sets.

$$\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{c', \lambda\}\{b''\}, \quad \{a'', \lambda\}\{a, \lambda\}\{db''\}, \quad \{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{d_c' b''\}.$$

Having the initial states of $\mathcal{L}(M_5')$ and $\mathcal{L}(M_6')$ as $A[B]$ and $A''[B]$ respectively, we can confirm that w in $\mathcal{L}(M_5')$ satisfies the prefix condition of $\overline{L}_5$ and w in $\mathcal{L}(M_6')$ satisfies the prefix condition of $\overline{L}_6$. Also, obtaining db'' takes us from $D[B]$ to DB'', and the only transition from there is d_ℓ to go to D_ℓ, which meets one of the suffix condition of $\overline{L}_i$. Similarly, $d_c' b''$ corresponds to the transition from $D_c'[B,D]$ to $B''[D]$, and this continues with D_c'' and then D_ℓ. Hence, we have $d_c' b'' d_c'' d_\ell$, an acceptable suffix of w in $\overline{L}_i$.

We move onto the transitions with overlines and underlines. From any state in P', we arrive at $\overline{B}$ with $\overline{b}$ from which, we transit to either $\underline{D}$ or $\underline{\overline{D}}$ with optional paths through $\overline{B}A$ and $\overline{B}C$ in that order. If we come from $A[A]$, $A'[A]$ or $A''[A]$, we transit to $\overline{B}[A]$, and then $\overline{B}A''$ with a''. Again, having optional states $\overline{B}A$ and $\overline{B}C$, we arrive at $\underline{D}$ or $\underline{\overline{D}}$. The only transition from $\underline{D}$ and $\underline{\overline{D}}$ is d to $\underline{D}D$ and $\overline{D}D$ respectively. Thus, a sequence of letters constructed by these paths is in

$$\{\overline{b}\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \overline{\underline{d}}d\}$$

which satisfies the seventh condition of $\overline{L}_i$. For $\mathcal{L}(M'_9)$ and $\mathcal{L}(M'_{10})$, the initial states are $\overline{B}A$ and $\overline{B}A''$ respectively. Thus, a word in $\mathcal{L}(M'_9)$ has a prefix in $\{c, \lambda\}\{\underline{d}d, \overline{\underline{d}}d\}$, and a word in $\mathcal{L}(M'_{10})$ has a prefix in $\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \overline{\underline{d}}d\}$.

Instead of $\overline{b}$, we can also use $\overline{d}$, $\overline{\underline{d}}$ or $\overline{y}$ to enter states with overlines, and arrive at states with underlines later. For now, we examine the cases of $\overline{d}$ and $\overline{\underline{d}}$ to have $\overline{D}$ and $\overline{\underline{D}}$ respectively, and we will discuss the case of $\overline{y}$ later. When we have a transition $\overline{d}$ or $\overline{\underline{d}}$ from a certain state, we arrive at $\overline{D}$ or $\overline{\underline{D}}$ respectively, which are both followed by $\overline{D}D$. From $\overline{D}D$, there are several ways to achieve a state with an underline. First, going to $D \rightarrow \underline{B}$, $D\underline{B}$ and then D_ℓ gives us either $\overline{d}dd\underline{b}d_\ell$ or $\overline{\underline{d}}dd\underline{b}d_\ell$, which are appropriate suffixes of a word in $\overline{L}_i$. Similarly, after the optional $C \rightarrow \{\underline{B}, \underline{D}, \overline{\underline{D}}, \underline{X}\}$, we can transit through $D'_c[D] \rightarrow \underline{B}[D]$, $\underline{B}[D]$, D''_c and then D_ℓ to have a suffix in $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{d'_c\underline{b}d''_c d_\ell\}$, which is also a valid suffix in $\overline{L}_i$. Additionally, there are five more cases. In each case, $C \rightarrow \{\underline{B}, \underline{D}, \overline{\underline{D}}, \underline{X}\}$ is available after $\overline{D}D$ as an optional state. Thus, these five cases are:

- $\underline{B}$ to obtain $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{\underline{b}\}$.
- $C'[C] \rightarrow \underline{B}[C]$ and $\underline{B}[C]$ to obtain $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{c'\underline{b}\}$.
- $\underline{D}$ and $\underline{D}D$ to obtain $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{\underline{d}d\}$.
- $\overline{\underline{D}}$ and $\overline{D}D$ to obtain $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{\overline{\underline{d}}d\}$.
- $Z \rightarrow \underline{X}$, $\underline{X}$ and XD to obtain $\{\overline{d}d, \overline{\underline{d}}d\}\{c, \lambda\}\{z\underline{x}d\}$.

All of them are acceptable under the seventh condition of $\overline{L}_i$.

Let us now look at transitions from states with $[C]$. Entering these states can be done in multiple ways. Namely, from $A[B]$, $A''[B]$, $B'[B]$ and $C[B]$ to $C'[B, C]$ while b' is unmatched, from $A[Y]$, $C[Y]$ and $Y'[Y]$ to $C'[Y, C]$ while y' is unmatched, from $\overline{D}D$ and $C \rightarrow \{\underline{B}, \underline{D}, \overline{\underline{D}}, \underline{X}\}$ to $C'[C] \rightarrow \underline{B}[C]$ while $\overline{d}$ is unmatched, and simply from a state in P' to $C''[C]$. The states immediately after each case are $B''[C]$, $Y''[C]$, $\underline{B}[C]$ and $B_s[C]$ respectively, and they share the same transitions to the same states afterwards. Similar to the case of $[A]$, we can transit through $C[C]$, $C'[C, C]$ and $C''[C]$ in this order finite number of times with $C[C]$ being optional. We can exit out of these states from $C[C]$ by having c'' to C'', d'_c to $D'_c[C, D]$ or z to $Z[C]$, or from $B''[C]$, $Y''[C]$, $\underline{B}[C]$, $B_s[C]$ and $C''[C]$ by having d to $D[C]$, d'_c to $D'_c[C, D]$ or z to $Z[C]$. Since $Z[C]$ is followed by ZC'', if we arrive at C'' or $Z[C]$, the sequence of letters constructed by these paths are in

$$\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*\{c'', cc'', zc'', czc''\},$$

which obeys the fifth condition of $\overline{L}_i$. Similarly, if we arrive at $D[C]$ or $D'_c[C, D]$, the rest of the run of M'_i are DC'' then D_ℓ and $C''[D]$, D''_c then D_ℓ respectively. Suffixes constructed by paths are respectively in $\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*\{dc''d_\ell\}$ and $\{c'b_s, c'b'', c'\underline{b}, c'y''\}\{c'c'', cc'c''\}^*$ $\{c, \lambda\}\{d'_c c'' d''_c d_\ell\}$, which satisfy the suffix condition of $\overline{L}_i$.

Finally, we observe transitions from the rest of the states, which are related to X, Y and Z. We first describe four ways to arrive at XD first, and any sequence of letters up to XD obeys the eighth condition of $\overline{L}_i$. Frankly, these four cases are recognized by which state of Z, $Z[X]$, $Z \to \underline{X}$ and $Z[C]$ we obtain by the transition z. There is one more state that we can arrive at with z, which is $Z[Y]$, but we will look at this case later.

The transition x' can occur if and only if b' can happen. From $A[A]$, $A'[A]$ or $A''[A]$, we arrive at $X'[A, X]$, and from any other states that allow the transition x', we transit to $X'[X]$. From $X'[A, X]$, a'' is the only transition available to $A''[X]$. Whether from $A''[X]$ or $X'[X]$, we have the optional states $A[X]$ and $C[X]$ in this order. Afterwards, we must arrive at $Z[X]$ which is followed by X'', and then XD. Therefore, words constructed by these sequences of transitions starting from x' are in $\{x'\}\{a'', \lambda\}\{a, \lambda\}\{c, \lambda\}\{zx''\}$, which is listed under the eighth condition of $\overline{L}_i$. For $\mathcal{L}(M'_7)$ and $\mathcal{L}(M'_8)$, a word begins with a prefix in $\{c, \lambda\}\{zx''\}$ and $\{a, \lambda\}\{c, \lambda\}\{zx''\}$ respectively, and it is followed by d afterwards.

From any state in P', we have the transition z to Z followed by X and XD, so this path produces zxd. Also, as explained before, having $\overline{d}$ or $\underline{\overline{d}}$ previously allows us to transit to $Z \to \underline{X}$, $\underline{X}$ and XD to obtain $z\underline{x}d$. The last case, $Z[C]$ can be obtained whenever $C'[C, C]$ is available. Once we arrive at $Z[C]$, we transit to ZC'' with c'' instead of x, then XD, giving us $zc''d$. Hence, in all four cases of passing through Z, $Z[X]$, $Z \to \underline{X}$ or $Z[C]$, a sequence of letters up to XD obeys the eighth condition of $\overline{L}_i$.

We note that primarily, x after z is a necessary transition, but this x is replaceable with x'', $\underline{x}$ or c''. In the case x' or c' is unmatched, we have x'' or c'' respectively instead of x. Similarly, if $\overline{d}$ or $\underline{\overline{d}}$ is unmatched with a transition with an underline, then we have $\underline{x}$.

From XD, there are three transitions available. Having y takes us to Y, which completes the sequence of transitions related to x, y and z. On the other hand, if we transit to $\overline{Y}$ with $\overline{y}$, we have optional $\overline{B}A$ and $\overline{B}C$ before we get to $\underline{D}$ or $\underline{\overline{D}}$, constructing a word in

$\{\overline{y}\}\{a, \lambda\}\{c, \lambda\}\{\underline{d}d, \underline{\overline{d}}d\}$. Finally, if we go to $Y'[Y]$ with y', there are a few different ways to achieve y''. With optional $A[Y]$, $C[Y]$ and $C'[Y, C]$ in this order, we can arrive at either Y'' or $Y''[C]$, giving us $\{y'\}\{a, \lambda\}\{c, \lambda\}\{y'', c'y''\}$. Now, after optional $A[Y]$ and $C[Y]$, it is also possible to have $Z[Y]$ followed by ZY'' to have $\{y'\}\{a, \lambda\}\{c, \lambda\}\{zy''\}$. In this case, y'' is taking the primary role of x, so it goes to XD afterwards, creating another sequence of letters starting with z. The other two states after Y' are $D[Y]$, DY'' and then D_ℓ, or $D'_c[Y, D]$, $Y''[D]$, D''_c and then D_ℓ, resulting in suffixes $dy''d_\ell$ or $d'_cy''d''_cd_\ell$ respectively.

With any sequence of transitions up to XD combined with a sequence of transitions from XD, the constructed subword satisfies the conditions of $\overline{L}_i$.

We have examined every sequence of transitions recursively occurring to form a word obeys all conditions of $\overline{L}_i$. Consequently, a word in $\mathcal{L}_(M'_i)$ is in $\overline{L}_i$, completing the proof of $\mathcal{L}(M'_i) \subseteq \overline{L}_i$.

To show that $\overline{L}_i \subseteq \mathcal{L}(M'_i)$, we take the same approach as we did in Chapter 4 for $\overline{L}$ and $\mathcal{L}(M)$. That is, suppose w is not in $\mathcal{L}(M'_i)$. The only ways w cannot be accepted by M_i are either the run of M_i on w contains the jail state or the last state is not D_ℓ. The latter violates the suffix condition of $\overline{L}_i$, so w is not in $\overline{L}_i$. For the case the run of M on w contains the jail state, we need to show that every transition to the jail state is due to a failure of w to meet one of the conditions of $\overline{L}_i$.

For some cases such as w containing the jail state due to (A, a) or (B, b), w must contain a subword that is prohibited in the ninth condition of $\overline{L}_i$. In other cases, which are the majority of the transitions to the jail state, w must disobey at least one of the second to the eighth conditions of $\overline{L}_i$. For instance, having a'' with any state q without the notation of $[A]$ implies that there is no a' previously. Thus, this violates the third condition of $\overline{L}_i$. As we can see, it is a straightforward exercise (but long and tedious one) to verify that every transition to the jail state violates some conditions of $\overline{L}_i$.

Consequently, this completes the proof of $\mathcal{L}(M'_i) = \overline{L}_i$. ■

With Proposition 6.5 and 6.6, we are finally at the place to derive the generating function for all simple permutations of length 4 or more in $\mathcal{A}'$.

Proof of Theorem 6.1. Let us apply the transfer matrix method to each $\mathcal{L}(M'_i)$ $(1 \leq i \leq 10)$. Since each M'_i shares the same transitions with different initial states, we only need to provide one adjacency matrix. The adjacency matrix for each M'_i is extremely large, so it is given in

Appendix A.

In this matrix, there are three weights, x, $\bar{F}$ and $\bar{G}$. These weights are used for inflation in the next section. In order to we replace all weights $\bar{F}$ and $\bar{G}$ with x. Let P' be such adjacency matrix. We denote by $(I-P)^{-1}_{q_1,q_2}$ the (q_1,q_2)-entry of $(I-P)^{-1}$. For each $\mathcal{L}(M'_i)$, we examine $(I-P)^{-1}_{q_i,D_\ell}$ where q_i is the initial state for M'_i. Each one gives the generating function for n-letter words in $\mathcal{L}(M'_i)$, and hence, for n-letter words in $\overline{L}_i$. A word in each $\overline{L}_i$, is missing a certain prefix, so we need to multiply either x^2, x^3 or x^4 accordingly to count all associated words in L'. We then add all of these generating functions to obtain the generating function for L', which enumerates all permutations in H'. Thus,

$$\begin{aligned} f_{H'} &= x^2\cdot(I-P)^{-1}_{A,D_\ell} + x^4\cdot(I-P)^{-1}_{A'',D_\ell} + x^3\cdot(I-P)^{-1}_{A[A],D_\ell} + x^4\cdot(I-P)^{-1}_{A''[A],D_\ell} \\ &+ x^3\cdot(I-P)^{-1}_{A[B],D_\ell} + x^4\cdot(I-P)^{-1}_{A''[B],D_\ell} + x^3\cdot(I-P)^{-1}_{A[X],D_\ell} + x^4\cdot(I-P)^{-1}_{A''[X],D_\ell} \\ &+ x^3\cdot(I-P)^{-1}_{\overline{B}A,D_\ell} + x^4\cdot(I-P)^{-1}_{\overline{B}A'',D_\ell} \\ &= \frac{x^4\left(x^{10}+7x^9+18x^8+23x^7+16x^6+10x^5+12x^4+9x^3+2x^2+1\right)}{(x+1)\left(2x^9+12x^8+16x^7+3x^6-11x^5-5x^4-3x^2-3x+1\right)}. \end{aligned}$$

By doubling this result, we obtain the desired generation function $f_{\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2}$. ■

6.2. Enumeration of the whole class $\mathcal{A}'$

In order to finish the enumeration of the whole class, we show that every simple permutation in $\mathcal{A}'$ satisfies the hypothesis of Proposition 2.5, just as we did in Chapter 4.

Before we state and prove the statement, we make a few important notes. For $\pi \in N$, values in LRmax(π) have positions in $[1, \pi^{-1}(n)]$. In particular, except the value n, a value of π is in LRmax(π) if and only if it is 1 of 1, 2 of 21 in 231-value chains, or 2 or 3 of 231 in 231-value chains in Equation 5.2. This implies that LRmax$(\pi)\setminus\{n\}$ is the set of values that are encoded as a or a'' by N-ENCODE. By the reverse complement symmetry, we also know that RLmin$(\pi)\setminus\{1\}$ is the set of values that are encoded as c or c' by N-ENCODE. Hence, values $\pi(i)$ that are covered in the fourth condition of the latter four must be playing the roles of 1 of 21 or 231 in 231-value chains in Equation 5.2, 1 of 21 or 312 in 312-value chains in Equation 5.3, 1 of 1 in Equation 5.1 which is a scissor of a value chain, or 1 or 2 of 12 in Equation 5.1. Respectively, these are the values encoded as a', c'', b_s, b' and b'' by N-ENCODE.

We are now ready to prove the following lemma.

Lemma 6.7 *Let π be a simple permutation of extreme pattern 2413 or 3142 whose length is n. The condition $\alpha = \sigma[\sigma_1, \ldots, \sigma_n] \in \mathcal{A}'$ is equivalent to the condition stating that for all i with $1 \leq i \leq n$,*

- *if $\pi(i) \in \mathrm{LRmax}(\pi)$, then $\sigma_i \in Av(4123, 4213, 4132)$,*
- *if $\pi(i) \in \mathrm{RLmin}(\pi)$, then $\sigma_i \in Av(2341, 3241, 2431)$,*
- *if $\pi(i)$ is 1 of 1 in Equation 5.1 and it is not a scissor of either value chain, then $\sigma_i \in Av(123, 213, 132)$, and*
- *otherwise, $\sigma_i \in Av(12, 21)$.*

Proof. We only consider the case of π is of extreme pattern 2413, since we can apply the inverse symmetry to prove the case of extreme pattern 3142.

First, we show that the first condition implies the latter four conditions. Suppose the latter condition is false. That is, at least one of the above four conditions is not met. Assume it is the first one. Then there exists i ($1 \leq i \leq n$) such that $\pi(i) \in \mathrm{LRmax}(\pi)$ and σ_i contains at least one of 4123, 4213 and 4132. Now, whichever point $(i, \pi(i))$ is, the point with the value 1 is located to the right. Thus, the value 1 of α together with the subsequence corresponding to 4123, 4213 or 4132 of α within the inflation σ_i, α contains 52341, 53241 or 52431 respectively, so $\alpha \notin \mathcal{A}'$. If the second condition is not met, we can apply the reverse complement argument of the previous one to show α contains one of 52341, 53241 and 52431. If the third condition is false, say at least one of 123, 213 and 132 is contained in σ_i for some i, then with the values $|\alpha|$ and 1, α contains 52341, 53241 or 52431, $\alpha \notin \mathcal{A}'$. Finally, for the last condition, each of these values is either 2 or 3 of 4231 pattern in π. Hence, inflating with σ_i which contains 12 or 21 will cause α to contain 52341, 53241 or 52431, so again, $\alpha \notin \mathcal{A}'$. Consequently, $\alpha = \pi[\sigma_1, \ldots, \sigma_n] \in \mathcal{A}'$ implies the latter condition.

Next, assume a permutation $\alpha = \pi[\sigma_1, \ldots, \sigma_n]$ where $\pi \in H'$ is not in $\mathcal{A}'$. Thus, α contains at least one permutation β in the basis. Since π avoids every permutation in the basis, it means there exist $\sigma_{i_1}, \ldots, \sigma_{i_k}$ (for all $j \in \{1, \ldots, k\}$, $1 \leq i_j \leq n$) such that α contains β within the union of subintervals corresponding to $\sigma_{i_1}, \ldots, \sigma_{i_k}$.

Because every permutation in $\{35142, 42513, 351624\}$ is simple, so we only need to consider $\beta \in \{52341, 53241, 52431\}$. For now, suppose $\beta = 52341$. If there exist $\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}, \sigma_{i_4}, \sigma_{i_5}$ such that each point of β is contained in intervals corresponding to $\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}, \sigma_{i_4}, \sigma_{i_5}$ respectively, then $\beta \preceq \pi$ which cannot be true. On the other hand, if there exists a single σ_i such that $\beta \preceq \sigma_i$, then one of the latter conditions is false, since β contains whatever σ_i cannot contain. Hence, there exist two, three or four subintervals of α such that the containment of β is involved in.

We first assume that a containment of β occurs among four subintervals of α. Let $\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}, \sigma_{i_4}$ with $i_1 < i_2 < i_3 < i_4$ be the permutations from the expression $\alpha = \pi[\sigma_1, \ldots, \sigma_n]$ which correspond to each subinterval. By the pigeonhole principle, there is one subinterval containing two values of β which have consecutive positions. Due to inflation, these two values cannot be 52 and 41 of β. Hence, it is either 23 or 34, and each case implies $12 \preceq \sigma_{i_2}$ or $12 \preceq \sigma_{i_3}$ respectively. Suppose the case of 23 contained in one subinterval, so we have $12 \preceq \sigma_{i_2}$. Note that $\pi(i_2)$ cannot be in $\text{LRmax}(\pi)$ because then we do not have any value for $\pi(i_1)$ to the left of $\pi(i_2)$. Similarly, $\pi(i_2) \notin \text{RLmin}(\pi)$. Also, if $\pi(i_2)$ plays the role of 1 of 21 or 231 in 231-value chains in Equation 5.2 or 1 of 21 or 312 in 312-value chains in Equation 5.3, then the fourth condition is not met. Therefore, suppose $i_2 \in (\pi^{-1}(n), \pi^{-1}(1))$. If $\pi(i_2)$ plays the role of 1 or 2 of 12 in Equation 5.1 or 1 of 1 in Equation 5.1 that is a scissor of some value chain, then again, the fourth condition is not satisfied. Thus, $\pi(i_2)$ plays the roles of 1 of 1 in Equation 5.1 that is not a scissor of a value chain. Then $i_3, i_4 \in [\pi^{-1}(1), n]$ with $\pi(i_4) < \pi(i_2) < \pi(i_3)$, but this means $\pi(i_2)$ is a scissor of a 312-value chain, so we have a contradiction. We achieve same results by assuming 34 is contained in one subinterval. Consequently, if a containment of β occurs among four subintervals of α, then the latter conditions are false.

Next, suppose a containment of β involves three subintervals of α. Let $\sigma_{i_1}, \sigma_{i_2}, \sigma_{i_3}$ with $i_1 < i_2 < i_3$ be the permutations from the expression $\alpha = \pi[\sigma_1, \ldots, \sigma_n]$ which correspond to each subinterval. In this case, the only possibility is that the middle interval corresponding to σ_{i_2} contains the positions of 234, and each of the other two interval contains the positions of 5 and 1. Hence, we have $123 \preceq \sigma_{i_2}$. As before, $\pi(i_2)$ cannot be in $\text{LRmax}(\pi)$ and $\text{RLmin}(\pi)$. Now, if $\pi(i_2)$ corresponds to 1 of 1 in Equation 5.1 that is not a scissor of a value chain, then the third condition is disobeyed. Otherwise, the fourth condition is false, since $12 \preceq 123 \preceq \sigma_{i_2}$.

Finally, suppose the containment of β is shared by two subintervals of α, say the ones of σ_{i_1} and σ_{i_2} with $i_1 < i_2$. Then this must imply $4123 \in \sigma_{i_1}$ or $2341 \in \sigma_{i_2}$. Suppose $4123 \in \sigma_{i_1}$. Then $\pi(i_1) \notin \text{RLmin}(\pi)$ since there is no value for $\pi(i_2)$ to the right of $\pi(i_1)$. However, because 12 and 123 are contained in 4123, this implies σ_1 disobeys one of the latter conditions. We obtain the same result for $2341 \in \sigma_{i_2}$. So once again, the latter conditions are false.

Proving for the cases of $\beta = 53241$ and $\beta = 52431$ are very similar. With every observation we made, the latter conditions imply $\alpha = \pi[\sigma_1, \ldots, \sigma_n] \in \mathcal{A}'$. Consequently, those two statements are equivalent. ∎

By using Lemma 6.7, we claim that every simple permutation in H' also satisfies the hypothesis of Proposition 2.5. Let σ, τ be simple permutations of extreme pattern 2413 and 3142 respectively where $|\sigma| = m$ and $|\tau| = n$. We examine $\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^1$ as an example. Let i be the position of m in σ. In $\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau$, for every position k except for i, $\sigma'(k) = \sigma(k)$. Because all values coming from τ are greater than $\sigma'(k)$, subclasses of inflation for each $\sigma'(k)$ are exactly the same for $\sigma(k)$ of σ. Similarly, for $\tau'(k)$ with $3 \leq k \leq n$, as $\sigma'(i)$ taking the place of $\tau(1)$, subclasses of inflation for each $\tau'(k)$ are exactly the same for $\tau(k)$ of τ. For $\sigma'(i)$, since it is now in $\mathrm{LRmax}(\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau)$, its subclass of inflation is $\mathrm{Av}(4123, 4213, 4132)$, which is the same as the one for $\sigma(i)$ in σ. Hence, for every value of $\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau$, its subclass of inflation remains the same. We obtain the same result for $\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_1^0 \tau$.

For $(x, y) \in \{(2, 0), (2, 1), (3, 0), (4, 0)\}$, there is another value of σ that is shifted upward in $\sigma \mathbin{\bigcirc\mkern-15mu\nwarrow}_x^y \tau$. In particular, for $(x, y) = (2, 0)$ and $(x, y) = (2, 1)$, it is $\sigma(i + 2)$, for $(x, y) = (3, 0)$, it is $\sigma(m - 2)$, and for $(x, y) = (4, 0)$, it is $\sigma(s)$ where s is the position of the value $m - 2$. All of these values correspond to either 2 or 3 of 4231 pattern in σ, so their subclass of inflation is $\mathrm{Av}(12, 21)$. After they are shifted by $\mathbin{\bigcirc\mkern-15mu\nwarrow}_x^y$, they are still playing the roles of 2 or 3, so their subclass of inflation do not change. In addition, just like the case of $\mathbin{\bigcirc\mkern-15mu\nwarrow}_1^1$, other values of σ and values of τ keep their subclasses of inflation when σ and τ are operated by $\mathbin{\bigcirc\mkern-15mu\nwarrow}_x^y$.

We can establish the same result for SE glue sums. Thus looking at how W-COMBINE and AFFIX-CONVERT replaces letters, we have the following proposition as a consequence of Lemma 6.7.

Proposition 6.8 *Let π be a simple permutation in H'. The condition $\alpha = \pi[\sigma_1, \ldots, \sigma_n] \in \mathcal{A}'$ is equivalent to the condition stating that for all i with $1 \leq i \leq n$,*

- *if $\pi(i)$ is encoded by ϕ' as a letter in $\{a, a'', c, c', d, d''_a, d'_c, d_\ell, z\}$, then $\sigma_i \in Av(4123, 4213, 4132)$ or $\sigma_i \in Av(2341, 3241, 2431)$ (depending on the specific letter and whether $\pi(i)$ originally belonged to N or S),*
- *if $\pi(i)$ is encoded as b, then $\sigma_i \in Av(123, 213, 132)$, and*
- *if $\pi(i)$ is encoded as a letter in $\{a', b_s, b', b'', \underline{b}, \overline{b}, c'', \underline{d}, \overline{d}, \overline{\underline{d}}, x, x', x'', \underline{x}, y, y', y'', \overline{y}\}$, then $\sigma_i \in Av(12, 21)$.*

The generating functions for $\mathrm{Av}(4123, 4213, 4132)$ and $\sigma_i \in \mathrm{Av}(2341, 3241, 2431)$ are exactly the same by reverse complement, as stated in Lemma 3.2. For their inflation, we need the generating function $\bar{G}$. The generating function for $\mathrm{Av}(123, 213, 132)$ was also previously explained in Chapter 3 as Lemma 3.1. We denote by F the generating function for this class, and use $\bar{F}$ for inflation. Finally, $\mathrm{Av}(12,21) = \{\varepsilon, 1\}$, so $f_{\mathrm{Av}(12,21)} = 1+x$, implying $\bar{f}_{\mathrm{Av}(12,21)} = x$.

We now revisit the adjacency matrix P'. By replacing weights of transitions in $\{a, a'', c, c', d, d''_a, d'_c, d_\ell, z\}$ with $\bar{G}$ and the transition b with $\bar{F}$, we obtain the adjacency matrix $\hat{P}'$ shown in Table A.2. The generating function for permutations in $\mathcal{A}'$ that are obtained by inflation of simple permutations in H' is

$$\begin{aligned} f_{\mathrm{ifl}(H')} &= \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A,D_\ell} + x^2 \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A'',D_\ell} + x \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A[A],D_\ell} \\ &+ x^2 \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A''[A],D_\ell} + x \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A[B],D_\ell} + x^2 \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A''[B],D_\ell} \\ &+ x \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A[X],D_\ell} + x^2 \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{A''[X],D_\ell} + x \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{\overline{B}A,D_\ell} \\ &+ x^2 \cdot \bar{G}^2 \cdot (I-\hat{P})^{-1}_{\overline{B}A'',D_\ell}. \end{aligned} \tag{6.1}$$

As usual, we multiply 2 to include the inverse case, so we have $f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2)} = 2f_{\mathrm{ifl(H')}}$.

For the inflation of $\pi = 21$, we need to inflate $\pi(1) = 2$ with a skew-indecomposable permutation σ_1. We claim that $\alpha = 21[\sigma_1, \sigma_2] \in \mathcal{A}'$ with skew-indecomposable σ_1 is equivalent to the condition that $\sigma_1 \in \mathrm{Av}(4123, 4213, 4132)$ and σ_1 is skew-indecomposable, and $\sigma_2 \in \mathrm{Av}(2341, 3241, 2431)$. The condition of σ_1 being skew-indecomposable cannot be dropped to enforce the uniqueness of inflation. It is clear that either of $\beta_1 \preceq \sigma_1$ for any $\beta_1\{4123, 4213, 4132\}$ or $\beta_2 \preceq \sigma_2$ for any $\beta_2\{2341, 3241, 2431\}$ implies $\alpha \notin \mathcal{A}'$. Suppose that $\alpha \notin \mathcal{A}'$. Then $\beta \in \alpha$ for some $\beta \in \{52341, 53241, 52431, 35142, 42513, 351624\}$. Since 35142, 42513 and 351624 are simple, if $\beta \in \{35142, 42513, 351624\}$, then $\beta \preceq \sigma_1$ or $\beta \preceq \sigma_2$. In either case, either one of 4123, 4213, 4132 is contained in σ_1 or either one of 2341, 3241, 2431 is contained in σ_2, so the second condition is not met. If $\beta = 52341$, then it is immediate that $4123 \preceq \sigma_1$ or $2341 \preceq \sigma_2$. We obtain the similar results for $\beta = 53241$ and $\beta = 52431$. Thus, the condition $\alpha = 21[\sigma_1, \sigma_2] \in \mathcal{A}'$ with skew decomposable σ_1 and the condition $\sigma_1 \in \mathrm{Av}(4123, 4213, 4132)$ where σ_1 is skew-indecomposable are equivalent.

By Proposition 2.5,

$$\bar{f}_{\mathrm{ifl}(21)} = \bar{f}^{\ominus}_{\mathrm{Av}(4123,4213,4132)} \cdot \bar{f}_{\mathrm{Av}(2341,3241,2431)}.$$

By Lemma 3.3, we know $\bar{f}^{\ominus}_{\mathrm{Av}(4123,4213,4132)} = (1-x-x^2)\bar{G}$. With $\bar{f}_{\mathrm{Av}(2341,3241,2431)} = \bar{G}$, we obtain

$$f_{\mathrm{ifl}(21)} = (1-x-x^2)\bar{G}^2.$$

The last case we need to consider is $\pi = 12$. We can inflate both 1 and 2 by any permutations σ_1 and σ_2 of $\mathcal{A}'$ itself, provided that σ_1 is a sum-indecomposable permutation in $\mathcal{A}'$.

Three cases for σ_1 being sum-indecomposable are $\sigma_1 = 1$, σ_1 is skew-decomposable, or σ_1 is an inflated permutation of $\pi \in \mathrm{Si}(\mathcal{A}') \setminus \mathcal{S}_2$. Generating functions for each case are x, $f_{\mathrm{ifl}(21)}$ and $f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2)}$ respectively. Thus,

$$f_{\mathrm{ifl}(12)} = (x + f_{\mathrm{ifl}(21)} + f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A})\setminus\mathcal{S}_2)}) \cdot \bar{f}_{\mathcal{A}'}.$$

Consequently, the generating function for $\mathcal{A}'$ satisfies the functional equation

$$\begin{aligned} f_{\mathcal{A}'} &= 1 + x + f_{\mathrm{ifl}(12)} + f_{\mathrm{ifl}(21)} + f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2)} \\ &= 1 + x + (x + f_{\mathrm{ifl}(21)} + f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2)}) \cdot \bar{f}'_{\mathcal{A}} + f_{\mathrm{ifl}(21)} + f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2)}. \end{aligned}$$

Finally, with $\bar{f}_{\mathcal{A}'} = f_{\mathcal{A}'} - 1$, we solve for $f_{\mathcal{A}'}$. Then, we obtain

$$f_{\mathcal{A}'} \;=\; \frac{1}{1 - x - f_{\mathrm{ifl}(21)} - f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2)}}.$$

We substitute $f_{\mathrm{ifl}(21)}$ with $(1-x-x^2)\bar{G}^2$ and $f_{\mathrm{ifl}(\mathrm{Si}(\mathcal{A}')\setminus\mathcal{S}_2)} = 2f_{\mathrm{inf}(H')}$ with Equation 6.1. After we simplify the whole expression of $f_{\mathcal{A}'}$ as well as Equation 6.1, we obtain the final desired result as the following.

Theorem 6.9 *The generating function for the class $\mathcal{A}'$ is defined by*

$$f_{\mathcal{A}'} = \frac{\sum_{i=0}^{5} a_i \bar{G}^i}{\sum_{i=0}^{6} b_i \bar{G}^i}$$

where $\bar{G} = G - 1$ and G is the generating function satisfying the equation

$$G = 1 + \frac{xG}{1 - xG^2},$$

and

$$\begin{aligned} a_0 &= -1 + 14x - 39x^2 + 28x^3 + 9x^4 - 11x^5 + x^6, \\ a_1 &= -12 + 81x - 100x^2 + 15x^3 + 46x^4 - 19x^5, \\ a_2 &= -8 + 35x - 20x^2 - 25x^3 + 31x^4 - 6x^5 - x^6, \\ a_3 &= 7, \qquad a_4 = 1, \qquad a_5 = -2. \end{aligned}$$

$$\begin{aligned} b_0 &= -1 + 57x - 125x^2 + 143x^3 - 48x^4 - 64x^5 + 51x^6 - 2x^8, \\ b_1 &= -54 + 260x - 386x^2 + 250x^3 + 81x^4 - 226x^5 + 74x^6 + 15x^7 - 3x^8, \\ b_2 &= -18 + 114x - 104x^2 - 22x^3 + 148x^4 - 123x^5 + 11x^6 + 14x^7 - x^8, \\ b_3 &= 24, \qquad b_4 = -2, \qquad b_5 = -5, \qquad b_6 = 1. \end{aligned}$$

The first several terms in power series expression are

$$f_{\mathcal{A}'} = 1 + x + 2x^2 + 6x^3 + 24x^4 + 115x^5 + 607x^6 + 3370x^7 + 19235x^8 + 111571x^9 + \cdots$$

6.3. Conclusions

As authors noted in [7], the technique they used to enumerate $\mathcal{A}$ was indeed applicable to enumerate the permutations indexing local complete intersection Schubert varieties. Although the whole process turned out to be extremely complicated, as of now, examining the geometric structures and constructing an encoding to apply the transfer matrix method is the only known way to enumerate this class. There may be a much more efficient method by closely learning the class $\mathcal{A}'$ and observing Rothe diagrams of permutations described in [38].

Lastly, this dissertation owes huge thanks to PermLab [1] and its author, Michael Albert. Without this extremely efficient program, viewing structures of permutations in $\mathcal{A}'$ would have been impossible.

Appendices

A. Transitions of M'_i $(1 \leq i \leq 10)$ and adjacency matrix associated with M'_i

Two large tables which are referred in Chapter 6 are placed in this section. Table A.1 shows the description of δ', the transition function for the automaton M'_i for each i with $1 \leq i \leq 10$. Note that Jail states and transitions to them are omitted.

Table A.2 shows the associated adjacency matrix $\hat{P}'$ with weights x, $\bar{F}$ and $\bar{G}$. Since the matrix is extremely large, it is presented by dividing into eight sub-matrices A through H in alphabetical order, where

$$\hat{P}' = \left[\begin{array}{c|c|c|c} A & B & C & D \\ \hline E & F & G & H \end{array}\right].$$

As it was described in Chapter 6, in order to obtain P', the adjacency matrix for each M'_i, we need replace all weights $\bar{F}$ and $\bar{G}$ with x. Let P' be such adjacency matrix.

	a	a'	a''	b	b'	b''	b_s
A		$A'[A]$		B	$B'[B]$		
A''	A	$A'[A]$		B	$B'[B]$		
$A[A]$		$A'[A, A]$			$B'[A, B]$		$B_s[A]$
$A'[A]$	$A[A]$	$A'[A, A]$			$B'[A, B]$		$B_s[A]$
$A''[A]$	$A[A]$	$A'[A, A]$			$B'[A, B]$		$B_s[A]$

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
A		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
A''		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
$A[A]$		$\overline{B}[A]$					
$A'[A]$		$\overline{B}[A]$					
$A''[A]$		$\overline{B}[A]$					

	d''_c	$\underline{d}$	$\overline{d}$	$\overline{\underline{d}}$	d_ℓ	x	x'
A			$\overline{D}$				$X'[X]$
A''			$\overline{D}$				$X'[X]$
$A[A]$							$X'[A, X]$
$A'[A]$							$X'[A, X]$
$A''[A]$							$X'[A, X]$

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
A							Z
A''							Z
$A[A]$							
$A'[A]$							
$A''[A]$							

Table A.1.: Partial state diagram of M'.

	a	a'	a''	b	b'	b''	b_s
$A[B]$						B''	
$A''[B]$	$A[B]$					B''	
$A[X]$							
$A''[X]$							
$A[Y]$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$A[B]$			$C[B]$	$C'[B, C]$		$D[B]$	$D'_c[B, D]$
$A''[B]$			$C[B]$	$C'[B, C]$		$D[B]$	$D'_c[B, D]$
$A[X]$			$C[X]$				
$A''[X]$			$C[X]$				
$A[Y]$			$C[Y]$	$C'[Y, C]$		$D[Y]$	$D'_c[Y, D]$

	d''_c	$\underline{d}$	$\overline{d}$	$\overline{\underline{d}}$	d_ℓ	x	x'
$A[B]$							
$A''[B]$							
$A[X]$							
$A''[X]$							
$A[Y]$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$A[B]$							
$A''[B]$							
$A[X]$							$Z[X]$
$A''[X]$							$Z[X]$
$A[Y]$					Y''		$Z[Y]$

	a	a'	a''	b	b'	b''	b_s
$A'[A,A]$			$A''[A]$				
B	A	$A'[A]$			$B'[B]$		
B''	A	$A'[A]$		B	$B'[B]$		
$\underline{B}$	A	$A'[A]$		B	$B'[B]$		
$\overline{B}$	$\overline{B}A$						

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$A'[A,A]$							
B		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
B''		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
$\underline{B}$		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
$\overline{B}$			$\overline{B}C$				

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$A'[A,A]$							
B			$\overline{D}$				$X'[X]$
B''			$\overline{D}$				$X'[X]$
$\underline{B}$			$\overline{D}$				$X'[X]$
$\overline{B}$		$\underline{D}$		$\underline{\overline{D}}$			

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$A'[A,A]$							
B							Z
B''							Z
$\underline{B}$							Z
$\overline{B}$							

	a	a'	a''	b	b'	b''	b_s
$\overline{B}A$							
$\overline{B}A''$	$\overline{B}A$						
$\overline{B}C$							
$B_s[A]$			A''				
$\overline{B}[A]$			$\overline{B}A''$				

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$\overline{B}A$			$\overline{B}C$				
$\overline{B}A''$			$\overline{B}C$				
$\overline{B}C$							
$B_s[A]$							
$\overline{B}[A]$							

	d''_c	$\underline{d}$	$\overline{d}$	$\overline{\underline{d}}$	d_ℓ	x	x'
$\overline{B}A$		$\underline{D}$		$\overline{\underline{D}}$			
$\overline{B}A''$		$\underline{D}$		$\overline{\underline{D}}$			
$\overline{B}C$		$\underline{D}$		$\overline{\underline{D}}$			
$B_s[A]$							
$\overline{B}[A]$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$\overline{B}A$							
$\overline{B}A''$							
$\overline{B}C$							
$B_s[A]$							
$\overline{B}[A]$							

	a	a'	a''	b	b'	b''	b_s
$B'[B]$	$A[B]$						
$B_s[C]$							
$B''[C]$							
$\underline{B}[C]$							
$B_s[D]$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$B'[B]$			$C[B]$	$C'[B,C]$		$D[B]$	$D'_c[B,D]$
$B_s[C]$			$C[C]$	$C'[C,C]$	C''	$D[C]$	$D'_c[C,D]$
$B''[C]$			$C[C]$	$C'[C,C]$	C''	$D[C]$	$D'_c[C,D]$
$\underline{B}[C]$			$C[C]$	$C'[C,C]$	C''	$D[C]$	$D'_c[C,D]$
$B_s[D]$							

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$B'[B]$							
$B_s[C]$							
$B''[C]$							
$\underline{B}[C]$							
$B_s[D]$	D''_c						

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$B'[B]$							
$B_s[C]$							$Z[C]$
$B''[C]$							$Z[C]$
$\underline{B}[C]$							$Z[C]$
$B_s[D]$							

	a	a'	a''	b	b'	b''	b_s
$B''[D]$							
$\underline{B}[D]$							
$B'[A,B]$			$A''[B]$				
C	A	$A'[A]$		B	$B'[B]$		
C''	A	$A'[A]$		B	$B'[B]$		

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$B''[D]$							
$\underline{B}[D]$							
$B'[A,B]$							
C		$\overline{B}$		$C'[C]$		CD	$D'_c[D]$
C''		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$B''[D]$	D''_c						
$\underline{B}[D]$	D''_c						
$B'[A,B]$							
C			$\overline{D}$				$X'[X]$
C''			$\overline{D}$				$X'[X]$

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$B''[D]$							
$\underline{B}[D]$							
$B'[A,B]$							
C							Z
C''							Z

	a	a'	a''	b	b'	b''	b_s
$C \to \{\underline{B}, \underline{D}, \underline{\overline{D}}, \underline{X}\}$							
CD		$A'[A]$		B	$B'[B]$		
$C[B]$						B''	
$C[C]$							
$C'[C]$							$B_s[C]$

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$C \to \{\underline{B}, \underline{D}, \underline{\overline{D}}, \underline{X}\}$	$\underline{B}$			$C'[C] \to \underline{B}[C]$			$D'_c[D] \to \underline{B}[D]$
CD		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
$C[B]$				$C'[B, C]$			$D'_c[C, D]$
$C[C]$				$C'[C, C]$	C''		$D'_c[C, D]$
$C'[C]$							

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$C \to \{\underline{B}, \underline{D}, \underline{\overline{D}}, \underline{X}\}$		$\underline{D}$		$\underline{\overline{D}}$			
CD			$\overline{D}$				$X'[X]$
$C[B]$							
$C[C]$							
$C'[C]$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$C \to \{\underline{B}, \underline{D}, \underline{\overline{D}}, \underline{X}\}$							$Z \to \underline{X}$
CD							Z
$C[B]$							
$C[C]$							$Z[C]$
$C'[C]$							

	a	a'	a''	b	b'	b''	b_s
$C'[C] \to \underline{B}[C]$							
$C''[C]$							
$C''[D]$							
$C[X]$							
$C[Y]$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$C'[C] \to \underline{B}[C]$	$\underline{B}[C]$						
$C''[C]$			$C[C]$	$C'[C,C]$	C''	$D[C]$	$D'_c[C,D]$
$C''[D]$							
$C[X]$							
$C[Y]$				$C'[Y,C]$			$D'_c[Y,D]$

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$C'[C] \to \underline{B}[C]$							
$C''[C]$							
$C''[D]$	D''_c						
$C[X]$							
$C[Y]$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$C'[C] \to \underline{B}[C]$							
$C''[C]$							$Z[C]$
$C''[D]$							
$C[X]$							$Z[X]$
$C[Y]$					Y''		$Z[Y]$

	a	a'	a''	b	b'	b''	b_s
$C'[B,C]$						$B''[C]$	
$C'[C,C]$							
$C'[Y,C]$							
D		$A'[A]$		B	$B'[B]$		
D''_c							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$C'[B,C]$							
$C'[C,C]$					$C''[C]$		
$C'[Y,C]$							
D		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
D''_c							

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$C'[B,C]$							
$C'[C,C]$							
$C'[Y,C]$							
D			$\overline{D}$		D_ℓ		$X'[X]$
D''_c					D_ℓ		

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$C'[B,C]$							
$C'[C,C]$							
$C'[Y,C]$					$Y''[C]$		
D							Z
D''_c							

	a	a'	a''	b	b'	b''	b_s
$\underline{D}$							
$\overline{D}$							
$\overline{\underline{D}}$							
D_ℓ							
DB''							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$\underline{D}$						$\underline{D}D$	
$\overline{D}$						$\overline{D}D$	
$\overline{\underline{D}}$						$\underline{D}D$	
D_ℓ							
DB''							

	d''_c	$\underline{d}$	$\overline{d}$	$\overline{\underline{d}}$	d_ℓ	x	x'
$\underline{D}$							
$\overline{D}$							
$\overline{\underline{D}}$							
D_ℓ							
DB''					D_ℓ		

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$\underline{D}$							
$\overline{D}$							
$\overline{\underline{D}}$							
D_ℓ							
DB''							

	a	a'	a''	b	b'	b''	b_s
$D\underline{B}$							
$D \to \underline{B}$							
DC''							
$\underline{D}D$		$A'[A]$		B	$B'[B]$		
$\overline{D}D$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$D\underline{B}$							
$D \to \underline{B}$	$D\underline{B}$						
DC''							
$\underline{D}D$		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
$\overline{D}D$	$\underline{B}$		$C \to \{\underline{B}, \underline{D}, \overline{\underline{D}}, \underline{X}\}$	$C'[C] \to \underline{B}[C]$		$D \to \underline{B}$	$D'_c[D] \to \underline{B}[D]$

	d''_c	$\underline{d}$	$\overline{d}$	$\overline{\underline{d}}$	d_ℓ	x	x'
$D\underline{B}$					D_ℓ		
$D \to \underline{B}$							
DC''					D_ℓ		
$\underline{D}D$			$\overline{D}$				$X'[X]$
$\overline{D}D$		$\underline{D}$		$\overline{\underline{D}}$			

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$D\underline{B}$							
$D \to \underline{B}$							
DC''							
$\underline{D}D$							Z
$\overline{D}D$							$Z \to \underline{X}$

	a	a'	a''	b	b'	b''	b_s
DY''							
$D[B]$						DB''	
$D[C]$							
$D'_c[D]$							$B_s[D]$
$D'_c[D] \to \underline{B}[D]$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
DY''							
$D[B]$							
$D[C]$					DC''		
$D'_c[D]$							
$D'_c[D] \to \underline{B}[D]$	$\underline{B}[D]$						

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
DY''					D_ℓ		
$D[B]$							
$D[C]$							
$D'_c[D]$							
$D'_c[D] \to \underline{B}[D]$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
DY''							
$D[B]$							
$D[C]$							
$D'_c[D]$							
$D'_c[D] \to \underline{B}[D]$							

	a	a'	a''	b	b'	b''	b_s
$D[Y]$							
$D'_c[B,D]$						$B''[D]$	
$D'_c[C,D]$							
$D'_c[Y,D]$							
X							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$D[Y]$							
$D'_c[B,D]$							
$D'_c[C,D]$					$C''[D]$		
$D'_c[Y,D]$							
X						XD	

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$D[Y]$							
$D'_c[B,D]$							
$D'_c[C,D]$							
$D'_c[Y,D]$							
X							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$D[Y]$					DY''		
$D'_c[B,D]$							
$D'_c[C,D]$							
$D'_c[Y,D]$					$Y''[D]$		
X							

	a	a'	a''	b	b'	b''	b_s
X''							
$\underline{X}$							
XD							
$X'[X]$	$A[X]$						
$X'[A,X]$			$A''[X]$				

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
X''						XD	
$\underline{X}$						XD	
XD							
$X'[X]$			$C[X]$				
$X'[A,X]$							

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
X''							
$\underline{X}$							
XD							
$X'[X]$							
$X'[A,X]$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
X''							
$\underline{X}$							
XD			Y	$Y'[Y]$		$\overline{Y}$	
$X'[X]$							$Z[X]$
$X'[A,X]$							

	a	a'	a''	b	b'	b''	b_s
Y	A	$A'[A]$		B	$B'[B]$		
Y''	A	$A'[A]$		B	$B'[B]$		
$\overline{Y}$	$\overline{B}A$						
$Y''[C]$							
$Y''[D]$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
Y		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
Y''		$\overline{B}$	C	$C'[C]$		D	$D'_c[D]$
$\overline{Y}$			$\overline{B}C$				
$Y''[C]$			$C[C]$	$C'[C,C]$	C''	$D[C]$	$D'_c[C,D]$
$Y''[D]$							

	d''_c	$\underline{d}$	$\overline{d}$	$\overline{\underline{d}}$	d_ℓ	x	x'
Y			$\overline{D}$				$X'[X]$
Y''			$\overline{D}$				$X'[X]$
$\overline{Y}$		$\underline{D}$		$\overline{\underline{D}}$			
$Y''[C]$							
$Y''[D]$	D''_c						

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
Y							Z
Y''							Z
$\overline{Y}$							
$Y''[C]$							$Z[C]$
$Y''[D]$							

	a	a'	a''	b	b'	b''	b_s
$Y'[Y]$	$A[Y]$						
Z							
ZC''							
$Z \to \underline{X}$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
$Y'[Y]$			$C[Y]$	$C'[Y, C]$		$D[Y]$	$D'_c[Y, D]$
Z							
ZC''						XD	
$Z \to \underline{X}$							

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
$Y'[Y]$							
Z						X	
ZC''							
$Z \to \underline{X}$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
$Y'[Y]$					Y''		$Z[Y]$
Z							
ZC''							
$Z \to \underline{X}$		$\underline{X}$					

	a	a'	a''	b	b'	b''	b_s
ZY''							
$Z[C]$							
$Z[X]$							
$Z[Y]$							

	$\underline{b}$	$\overline{b}$	c	c'	c''	d	d'_c
ZY''						XD	
$Z[C]$					ZC''		
$Z[X]$							
$Z[Y]$							

	d''_c	$\underline{d}$	$\overline{d}$	$\underline{\overline{d}}$	d_ℓ	x	x'
ZY''							
$Z[C]$							
$Z[X]$							
$Z[Y]$							

	x''	$\underline{x}$	y	y'	y''	$\overline{y}$	z
ZY''							
$Z[C]$							
$Z[X]$	X''						
$Z[Y]$					ZY''		

$$
\begin{array}{c|ccccccccccccccccccccc|}
 & A & A" & A[A] & A'[A] & A"[A] & A[B] & A"[B] & A[X] & A"[X] & A[Y] & A'[A,A] & B & B" & \underline{B} & \overline{B} & \overline{B}A & \overline{B}A" & \overline{B}C & B_s[A] & \overline{B}[A] & B'[B] \\
A & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
A" & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
A[A] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 \\
A'[A] & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 \\
A"[A] & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & x & 0 \\
A[B] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A"[B] & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A[X] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A"[X] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A[Y] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A'[A,A] & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
B" & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
\underline{B} & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
\overline{B} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & \bar{G} & 0 & 0 & 0 \\
\overline{B}A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 \\
\overline{B}A" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & \bar{G} & 0 & 0 & 0 \\
\overline{B}C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B_s[A] & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\overline{B}[A] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 \\
B'[B] & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B_s[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B"[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{B}[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B_s[D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B"[D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{B}[D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B'[A,B] & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
C" & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
C \rightarrow \{\underline{B},\underline{D},\underline{\overline{D}},\underline{X}\} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
CD & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
C[B] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C'[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C'[C] \rightarrow \underline{B}[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C"[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C"[D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C[X] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C[Y] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C'[B,C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C'[C,C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Table A.2.: Adjacency matrix associated with M_i'.

$$\left[\begin{array}{ccccccccccccccccccc}
B_s[C] & B"[C] & \underline{B}[C] & B_s[D] & B"[D] & \underline{B}[D] & B'[A,B] & C & C" & C \rightarrow \{\underline{B},\underline{D},\underline{\overline{D}},\underline{X}\} & CD & C[B] & C[C] & C'[C] & C'[C] \rightarrow \underline{B}[C] & C"[C] & C"[D] & C[X] & C[Y] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0
\end{array}\right]$$

$$
\begin{array}{ccccccccccccccccccccc}
C'[B,C] & C'[C,C] & C'[Y,C] & D & D" & \underline{D} & \overline{D} & \underline{\overline{D}} & D_\ell & DB" & D\underline{B} & D \to \underline{B} & DC" & \underline{D}D & \overline{D}D & DY" & D[B] & D[C] & D'_c[D] & D'_c[D] \to \underline{B}[D] & D[Y] \\
0 & 0 & 0 & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 \\
\bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & x & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & x & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & x & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & x & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 \\
0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 \\
0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 \\
0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & x & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 \\
0 & 0 & 0 & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
\bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$\begin{array}{cccccccccccccccccccccc}
D'_c[B,D] & D'_c[C,D] & D'_c[Y,D] & X & X" & \underline{X} & XD & X'[X] & X'[A,X] & Y & Y" & \overline{Y} & Y"[C] & Y"[D] & Y'[Y] & Z & ZC" & Z \rightarrow \underline{X} & ZY" & Z[C] & Z[X] & Z[Y] \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0 \\
0&0&\bar{G}&0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0&0&0&\bar{G} \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0 \\
0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0 \\
0&0&\bar{G}&0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0&0&0&\bar{G} \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}$$

$$
\begin{array}{c|ccccccccccccccccccccc}
 & A & A" & A[A] & A'[A] & A"[A] & A[B] & A"[B] & A[X] & A"[X] & A[Y] & A'[A,A] & B & B" & \underline{B} & \overline{B} & \overline{B}A & \overline{B}A" & \overline{B}C & B_s[A] & \overline{B}[A] & B'[B] \\
\hline
C'[Y,C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
D" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{D} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\overline{D} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{\overline{D}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D_\ell & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
DB" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D\underline{B} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D \to \underline{B} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
DC" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{D}D & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
\overline{D}D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
DY" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D[B] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D'_c[D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D'_c[D] \to \underline{B}[D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D[Y] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D'_c[B,D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D'_c[C,D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D'_c[Y,D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\underline{X} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
XD & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X'[X] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X'[A,X] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Y & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
Y" & \bar{G} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{F} & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & x \\
\overline{Y} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & \bar{G} & 0 & 0 & 0 \\
Y"[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Y"[D] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Y'[Y] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
ZC" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Z \to \underline{X} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
ZY" & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Z[C] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Z[X] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Z[Y] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$\left[\begin{array}{ccccccccccccccccccc}
B_s[C] & B"[C] & \underline{B}[C] & B_s[D] & B"[D] & \underline{B}[D] & B'[A,B] & C & C" & C \to \{\underline{B},\underline{D},\overline{\underline{D}},\underline{X}\} & CD & C[B] & C[C] & C'[C] & C'[C] \to \underline{B}[C] & C"[C] & C"[D] & C[X] & C[Y] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]$$

$$\begin{array}{ccccccccccccccccccccc}
C'[B,C] & C'[C,C] & C'[Y,C] & D & D" & \underline{D} & \overline{D} & \overline{\underline{D}} & D_\ell & DB" & D\underline{B} & D \rightarrow \underline{B} & DC" & \underline{D}D & \overline{D}D & DY" & D[B] & D[C] & D'_c[D] & D'_c[D] \rightarrow \underline{B}[D] & D[Y] \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&\bar{G}&0&0&x&0&\bar{G}&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&\bar{G}&0&0&x&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&0&0&0&0&x&0&x&0&0&0&\bar{G}&0&0&0&0&0&0&0&\bar{G}&0 \\
0&0&0&0&0&0&0&0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&x&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&x&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&\bar{G}&0&0&x&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&0&0&\bar{G}&0&0&x&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0 \\
0&0&0&0&0&x&0&x&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G}&0&0&0 \\
0&0&0&0&x&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&\bar{G}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\bar{G} \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}$$

$$\begin{array}{cccccccccccccccccccccc}
D'_c[B,D] & D'_c[C,D] & D'_c[Y,D] & X & X" & \underline{X} & XD & X'[X] & X'[A,X] & Y & Y" & \overline{Y} & Y"[C] & Y"[D] & Y'[Y] & Z & ZC" & Z \rightarrow \underline{X} & ZY" & Z[C] & Z[X] & Z[Y] \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & x & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} \\
0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{G} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0
\end{array}$$

B. Python code

In this section, we provide Python programming code which demonstrate the encoding of a simple permutation in H' of length n into a n-letter word word in L'. Together with PermLab [1] and all Python files provided in this section, we are able to construct a list of all n-letter words in L' as a text file where n is a reasonable length.

Note that the code was written first, and was used greatly to discover the encoding rules stated in Chapter 6. What we provide here is kept as original, and was not modified even after proofs in Chapter 6 were established. For this reason, there may be a slight difference in the process of encoding between Python code and the sequence of encoding defined in Chapter 6, but the obtained result is the same. For instance, `prefixConverter.py` and `suffixConverter.py` take the places of W-COMBINE and AFFIX-CONVERT.

`makeList.py`

```
def readFile(fname):
    """
    reads the list of permutations from txt file line by line.
    """
    with open (fname) as f:
        permList = f.read().splitlines()
    return permList

def convList1(permList):
    """
    converts the list of string type permutations in permList
    to int type, specifically length <10.
    """
    listofperm=[]
    for i in range(len(permList)):
        temp=[]
        for k in range(len(permList[i])):
            temp.append(int(permList[i][k]))
        listofperm.append(temp)
    return listofperm

def convList2(permList):
    """
    converts the list of string type permutations in permList
    to int type, specifically length >=10.
    """
    listofperm =[]
    for i in range(len(permList)):
        temp = permList[i].split()
        for k in range(len(temp)):
            temp[k]=int(temp[k])
        listofperm.append(temp)
    return listofperm
```

list2str.py

```
"""
The following function converts a list to a string.
"""

def list2Str(word):
    string = '␣'.join(word)
    return string

def test(blah):
    word = []
    while len(blah) > 0:
        i = 0
        while blah[i] != '␣':
            i += 1
        temp = ''.join(blah[0:i])
        word.append(temp)
        temp2 = temp2[i+1:len(temp2)]
    return word
```

invert.py

```
"""
The following function simply inverts a permutation.
"""
```

```
def invPerm(perm):
    inverse = []
    for i in range(1,max(perm)+1):
        if i in perm:
            inverse.append(perm.index(i)+1)
        else:
            pass
    return inverse
```

ep.py

```
def ep2413(perm):
    """
    keeps permutations of extreme pattern 2413, and discard others.
    """
    if perm[0]<perm[len(perm)-1] and invPerm(perm)[0]>
    invPerm(perm)[len(invPerm(perm))-1]:
        return perm
    else:
        pass

def ep3142(perm):
    """
    keeps permutations of extreme pattern 3142, and discard others.
    """
    if perm[0]>perm[len(perm)-1] and invPerm(perm)[0]<
    invPerm(perm)[len(invPerm(perm))-1]:
        return perm
    else:
        pass

def Nstart(perm):
    """
    This function eliminates a simple adbi permutation which starts
    with S set.
    """
    if (perm[0] == 2 or perm[0] == 3 or perm[0] == 4) and
    perm[1] != 1:
        return perm
    else:
        pass
```

createFile.py

```
”””
The following function creates a text file listing all the desired
words.
”””

from os import path

def createFile(dest,words):
    if not (path.isfile(dest)):
        f = open(dest, ’w’)
        for i in words:
            f.write(i+’\n’)
        f.close
```

counter.py

```
”””
The following function will count how many N’s and S’s are embedded
in a permutation.
”””

def counter(perm):
    d1 = perm[0] #The first number. It is either 2, 3 or 4.#
    d2 = 1 #This is always 1.#
    temp1 = perm[perm.index(d1):perm.index(d2)+1]
    #Get the interval from d1 to d2”
    d3 = max(temp1) #The maximum number of temp 1 is d3.#
    temp2 = perm[0:len(perm)] #Get a copy of a permutation.#
    for x in range(d3+1,max(temp2)+1):
        temp2.remove(x) #Erase everything > d3.#
    d4 = temp2[-1] #The last number of temp 2 is d4.#
    count = 1
    while d2 != d4:
        d1 = d3
        d2 = d4 #Move up d1 and d2.#
        temp1 = perm[perm.index(d1):perm.index(d2)+1]
        d3 = max(temp1) #Determine the new d3.#
        count += 1 #Count goes up.#
        if d1 != d3: #Checking if d1 and d3 are the same.#
        #If not, determine the new d4.#
            temp2 = perm[0:len(perm)]
            for x in range(d3+1,max(temp2)+1):
                temp2.remove(x)
            d4 = temp2[-1]
            count += 1
        else: #If d1 and d3 become the same, we end it here.#
            break
    return count

”””
The number n = 'count' tells us how many N's and S's we have.
If n is odd, then there are (n-1)/2 N's and (n-1)/2 S's in a
permutation. If n is even, then there are n/2 N's and (n/2)-1
S's in a permutation.
”””
```

Nsplit.py

```
”””
We define two functions. The first one determines where the first
N shape ends. It returns the split N segment and some extras. The
second one takes the N set away from the whole permutation. It
returns the rest.
”””

def Nsplit(perm):
    ”””
```

```
First, we detect d1, the greatest element between the first
and the least numbers.
"""
temp1 = perm[0:perm.index(1)+1]
d1 = max(temp1)
"""
For convenience, we determine d2 as well. Remove everything
greater than d1. Let the last number be d2.
"""
temp2 = perm[0:len(perm)]
for x in range(d1+1,max(temp2)+1):
    temp2.remove(x)
d2 = temp2[-1]
"""
Finally, we identify the splitter number. This is necessary
only when d1 is scissored vertically by a number that is not
equal to d2.
"""
splitter = d1-1
"""
We determine if d1 is the max of the whole permutation. If so,
d2 is the last number in N-C. If not, we see which one of d1-2,
d1-1 and d1+1 appears first after d1.
"""
if d1 == max(perm):
    CLast = d2
    """
    If d1-2 appears first, one of the following three is
    happening.
    Case 1: The following S set is either empty, or just one
    number in C segment.
    Case 2: d1 is scissored vertically by d1-1.
    Case 3: d1-1 and d1-2 are a scission pair, or d1-1 and d2
    are a scission pair with d1-2 being a scissor.
    """
elif perm.index(d1-2) < perm.index(d1+1) and perm.index(d1-2)
< perm.index(d1-1):
```

```
temp3 = temp1[0:len(perm)]
temp3.remove(d1)
#Case 1a: d1-2 comes from A or B segment.#
if max(temp3) == d1-2:
#Detecting if d1-2 belongs to either A or B segment.#
#In order to distinguish from Case 2, we need to find#
#what separates d1 and d1-2.#
#If d1-1 is not d2, then it is Case 2. Otherwise, it is#
#Case 1a.#
    if d2 != d1-1: #Case 2#
        if perm[perm.index(d1-1)+1] == d2 or
        perm[perm.index(d1-1)+2] == d2:
        #One of scission pair of d2 is being a scissor.#
            temp4 = perm[perm.index(1):perm.index(d2)+1]
            temp5 = []
            for x in temp4:
                if x < d2:
                    temp5.append(x)
                else:
                    pass
            CLast = temp5[-1]
        elif perm[perm.index(d1-1)-2]>d1-1:
        #Case of d d^ c c' b_ ...#
            CLast = perm[perm.index(d1-1)-3]
        elif perm[perm.index(d1-1)-1]>d1-1:
        #Case of d d^ c' b_ ... or d d^ c b_ ...#
            CLast = perm[perm.index(d1-1)-2]
        else:
            CLast = perm[perm.index(d1-1)-1]
    else:
        temp4 = [] #REVISED (3/4/15)#
        for i in perm:
            if i < d1-2:
                temp4.append(i)
            else:
                pass
        CLast = temp4[-1]
```

```
#Case 1b: d1-2 comes from C segment.#
elif d1-2 < d2:
    if perm[perm.index(d1-2)+1]<d1-2:
    #Detecting if d1-2 is a part of scission.#
        if perm.index(d1-2)<perm.index(d1-3):
        #Detecting if d1-2 scission pairs are not the last#
        #numbers in C.#
            CLast = d1-3
        else:
        #d1-2 scission pairs are the last numbers in C.#
            CLast = perm[perm.index(d1-2)+1]
    else: #d1-2 is not a part of scission.#
        CLast = d1-2
#The last one is Case 3: d1-1 and d1-2 are a scission pair#
#in B segment of S shape, or d1-1 and d2 are scission pairs#
#with d1-2 being a scissor.#
#Note that for the former case, d1-1 and d1-2 can be y' and#
#y'', and for the latter case, d1-2 can be y^.#
else:
    if perm[perm.index(d1-2)+1] > d1-2:
    #No scissor in A segment.#
        CLast = perm[perm.index(d1-2)-1]
    else:
        s1 = perm[perm.index(d1-2)+1]
        s2 = s1+1
        temp6 = perm[perm.index(1):perm.index(s1)+1]
        while s2 in temp6 and perm[perm.index(s2)+1] < s2:
            s1 = perm[perm.index(s2)+1]
            s2 = s1+1
        if perm.index(s2) < perm.index(s1) and s2 in temp6:
        #Case of S–A segment self-scission chain. 2nd one#
        #is a scission.#
            CLast = perm[perm.index(s2)-1]
        elif perm.index(s2) > perm.index(s1):
        #Case of S–A segment self-scission chain. 2nd one#
        #is not a scission.#
            CLast = perm[perm.index(s1)-2]
            elif perm.index(s2) < perm.index(s1) and s2
            not in temp6:
            #Case of N-C segment self-scission chain with the#
            #last one being z.#
                CLast = perm[perm.index(d1-2)+1]
    """
    In the case d1+1 appears first, the number immediately
    left of d1+1 is the last number in C segment.
    """
elif perm.index(d1+1) < perm.index(d1-1) and perm.index(d1+1)
< perm.index(d1-2):
    CLast = perm[perm.index(d1+1)-1]
    """
    Finally, in the case d1-1 shows up first, there are
    following six cases.
    Case 1: d1 and d1-1 are a scission pair.
    Case 2: d1-1 is encoded as y.
    Case 3: d1-1 scissors a chain of S-A self-scissions or
    one of N-C self-scissions as d1-1 is being y.
    Case 4: d1-1 and d2 are a scission pair.
    Case 5: d1-1 scissors c' and c'' in S set.
    Case 6: Nothing special. Encoding goes d b, or d b'.
    """
elif perm.index(d1-1) < perm.index(d1+1) and perm.index(d1-1)
< perm.index(d1-2):
    if d1-1 == perm[perm.index(d1)+2] and perm[perm.index(d1)+1]
    != 1: #Case 1, d1 and d1-1 are a scission pair.#
        temp7 = perm[0:len(perm)]
        for x in range(1,temp7[temp7.index(d1)+1]):
            temp7.remove(x)
        if temp7[temp7.index(d1-1)+1]
        == temp7[temp7.index(d1)+1]+1:
            CLast = temp7[temp7.index(d1)+1]+1
        elif temp7[temp7.index(d1-1)+1]
        == temp7[temp7.index(d1)+1]+2 and
        temp7.index(temp7[temp7.index(d1)+1]+1)
        < temp7.index(temp7[temp7.index(d1)+1]+2):
          CLast = temp7[temp7.index(d1)+1]+2
      else:
          temp7 = perm[0:len(perm)]
          for x in
          range(temp7[temp7.index(d1)+1],max(temp7)+1):
              temp7.remove(x)
          CLast = temp7[-1]

  elif perm[perm.index(d1-1)+1] < d1-1 and
  perm[perm.index(d1-1)+1]+1 in
  perm[0:perm.index(perm[perm.index(d1-1)+1])+1]:
  #Case 2 and 3.#
      s1 = perm[perm.index(d1-1)+1]
      s2 = s1+1
      temp8 = perm[perm.index(1):perm.index(s1)+1]
      while s2 in temp8 and perm[perm.index(s2)+1] < s2:
          s1 = perm[perm.index(s2)+1]
          s2 = s1+1
      if perm.index(s2) < perm.index(s1) and s2 in temp8:
      #Case 3, S-A segment self-scission chain. 2nd one is#
      #a scission.#
          CLast = perm[perm.index(s2)-1]
      elif perm.index(s2) > perm.index(s1):
      #Case 3, S-A segment self-scission chain. 2nd one is#
      #not a scission.#
          CLast = perm[perm.index(s1)-2]
      elif perm.index(s2) < perm.index(s1) and s2 not in
      temp8: #Case 2, d1-1 is encoded as y. There could be#
      #an N-C segment self-scission chain.#
          CLast = perm[perm.index(d1-1)+1]
  elif d1-2 == d2 and perm[perm.index(d1-1)-1] > d1 and
  perm.index(d2)-2 == perm.index(d1-1): #Case 4, d1-1 and d2#
  #are a scission pair.# #REVISED (3/5/15)#
      temp9 = perm[0:len(perm)]
      for x in range(d2,max(temp9)+1):
          temp9.remove(x)
      CLast = temp9[-1]
```

```
            elif perm[perm.index(d1-1)-1] == d1+2: #Case 5, d1-1#
            #scissors c' and c'' in S set.#
                CLast = perm[perm.index(d1-1)-2]
            else: #Case 6, Nothing special is present.#
                CLast = perm[perm.index(d1-1)-1]
        subperm = [] #This is the split subpermutation.#
        for i in range(0,perm.index(CLast)+1):
                subperm.append(perm[i]) #subperm now is the#
                #subpermutation up to the last number in C segment that#
                #we identified.#
                #*** Refer this list as (1) in the latter function.#
        if perm[perm.index(d1)+1] == d1-2 or perm[perm.index(d1)+2]
        == d1-2: #Determine if d1 is scissored.#
            if perm.index(d1-2) > perm.index(1): #REVISED (3/4/15)#
                pass
            else:
                if splitter not in subperm and splitter != d2:
                    subperm.append(splitter) #We add the splitter number#
                    #in case it belongs to S-B or the scission of d2.#
                else:
                    pass
        else:
            pass
        if CLast != d2:
            subperm.append(d2)
            #Finally, add d2 to the subperm list, and we are done.#
        return subperm, CLast

    def setMinus(perm,CLast,subword):
        """
        perm is the whole permutation, CLast is the last number in N-C
         obtained by Nsplit function. subword is the encoded
        word after the modification of both prefix/suffix converters.
        What we do with this function are the following:
        1. We first take away everything from perm up to CLast number
        (including itself).
        2. By looking at the suffix of subword, we add back the necessary
```

```
numbers.
"""
subperm = [] #This is the split subpermutation.#
for i in perm[0:perm.index(CLast)+1]:
    subperm.append(i) #*** Same list as (1)#
    #from the above function.#
for i in perm[0:perm.index(CLast)+1]:
    perm.remove(i)
orig = subperm + perm #We need to refer back in some cases.#
d0 = min(subperm) #i.e., 1. We always need to add this one back.#
d1 = max(subperm) #Similarly, we need to add this back.#
if subword[-2:len(subword)] == ['d^', 'd']: #In this case,#
#d = d1. We need to add back d^ as well. d^ is simply d1-2.#
    recover = [d1, d1-2, d0]
elif subword[-2:len(subword)] == ['d_^', 'd']: #Similar to the#
#above case. d_^ is d1-2.#
    recover = [d1, d1-2, d0]
elif subword[-3:len(subword)] == ['z', 'x', 'd'] or
subword[-3:len(subword)] == ['z', 'x"', 'd'] or
subword[-3:len(subword)] == ['z', 'x_', 'd'] or
subword[-3:len(subword)] == ['z', 'c"', 'd'] or
subword[-3:len(subword)] == ['z', 'y"', 'd']:
    for i in subperm[0:subperm.index(d1)+1]:
        subperm.remove(i) #Get rid of A segment, including d1.#
    for i in
    subperm[subperm.index(d0)+1:subperm.index(subperm[-1])+1]:
        subperm.remove(i) #Get rid of C segment,#
        #NOT including d0.#
    x = max(subperm) #This is the max of B segment, i.e., x, x",#
    #x_, y" or b (b") which is#
    #associated with z _ d y encoding. We need to add this#
    #back to the next perm.#
    if x-1 == CLast: #All the cases except x being b or b",#
    #causing the C-chain scissions.(Need revision here???)#
    #In these cases, simply CLast is z. All we need to add back#
    #is d1, x, d0, y and z.#
        y = orig[orig.index(CLast)-1]
```

```
            recover = [d1, x, d0, y, CLast]
        else: #In this case, we must add back the whole C chain#
        #including x which initiates the chain.#
            firstcPrime = x-1
            firstcDPrime = orig[orig.index(firstcPrime)-1]
            CChain =
            orig[orig.index(firstcDPrime):orig.index(CLast)+1]
            recover = [d1, x, d0] + CChain
    else:
        recover = [d1, d0]
    perm = recover + perm
    return perm
```

encodeNS.py

```
import counter
import ep
from os import path
import invert
import Nsplit

"""
Looking at a permutation, or subsequence embedded in a permutation
which has the extreme pattern 2413 (N shape), the following function
encodes it into a word under certain rules. The Alphabet is the set
containing following 30 letters.

# a, a', a", b, b', b", bs, b_, b^, c, c', c", d, da', da",  #
# dc', dc", d_, d^, d_^, dl, x, x', x", x_, y, y', y", y^, z #

'perm' is an arbitrary permutation of extreme pattern 2413.
'locator' tells us where the subsequence is in a main permutation.
'transit1' is the information about the previous S shape.
'transit2' is the information about the next S shape.
"""

def encodeS(perm):
```

```
”””
encodes a permutation of extreme pattern 3142 into a word.
”””
if perm[0]>perm[len(perm)−1] and perm.index(len(perm)) >
perm.index(1):
    word = []
    for i in range(len(perm)):
        if i == 0 or i == len(perm)−1:
            word.append(’d’) #First and last letters must be d.#
        ##=====Prefix coding=====##
        elif i == 1 and (perm[i] == 1 or perm[i] == 2):
            if perm[i] == 1:
                word.append(’d’)
            else:
                word.append(’da\’’)
        elif i == 2 and perm[i] == 1:
            word.append(’d’)
        elif i == 3 and perm[i] == 1:
            word.append(’da\”’)
        ##=====Surffix coding=====##
        elif i == len(perm)−2 and (perm[i] == len(perm) or
        perm[i] == len(perm)−1):
            if perm[i] == len(perm):
                word.append(’d’)
            else:
                word.append(’dc\”’)
        elif i == len(perm)−3 and perm[i] == len(perm):
            word.append(’d’)
        elif i == len(perm)−4 and perm[i] == len(perm):
            word.append(’dc\’’)
        ##=====Infix coding=====##
        elif perm[i] > perm[0]:
            if perm.index(perm[i]−1) > i:
                word.append(’c\’’)
            elif perm.index(perm[i]+1) < i:
                word.append(’c\”’)
            else:
                    word.append('c')
                elif perm[i] < perm[0] and perm[i] > perm[len(perm)-1]:
                    if perm.index(perm[i]+1) > i:
                        word.append('b\'')
                    elif perm.index(perm[i]-1) < i:
                        word.append('b\"')
                    else:
                        word.append('b')
                else:
                    if perm.index(perm[i]-1) > i:
                        word.append('a\'')
                    elif perm.index(perm[i]+1) < i:
                        word.append('a\"')
                    else:
                        word.append('a')
            for i in range(len(word)):
                if word[i] == 'b':
                    if word[i-1] == 'c\'' or word[i+1] == 'a\"' or
                    word[i-1] == 'dc\'' or word[i+1] == 'da\"':
                        word[i] = 'bs'
                    else:
                        pass
                else:
                    pass
            return word
        else:
            print("Error.␣The␣entered␣permutation␣does␣not␣have␣extreme
␣␣␣␣␣␣␣␣pattern␣3142.")


def encodeN(perm):
    """
    encodes a permutation of extreme pattern 2413 into a word.
    """
    inverse = invert.invPerm(perm)
    word = encodeS(inverse)
    return word
```

prefixConverter.py

```
"""
Taking an arbitrary N or S set out of a simple adbi permutation of
extreme pattern 2143, the prefix & suffix of it must be modified
according to:
1. where it comes from, and
2. how it connects to the previous/next N & S sets.
The following functions will modify the prefix of the N or S set that
we are currently encoding.
"""

import Nsplit
import encodeNS

def initialPrefix(word):
    """
    This function is particularly applied if the N set we are
    encoding is the initial set of a whole permutation, but not the
    final set. In addition to the existing 6 ways to start the
    encoded word, there are 4 more possibilities associated with
    x' and b^.
    """
    codedWord = word[0:len(word)]
    if codedWord.count("b'") == 1:
    #This line ensures that the first b' showing up needs to be#
    #modified to x' or b^.#
        if codedWord[0:3] == ['d', "b'", 'd'] and
        codedWord[-5:len(codedWord)] == ["c'", 'b"', 'd', 'c"', 'd']:
            codedWord[1] = "x'"
            #This is the modification for Case 9: d x' d...#
        elif codedWord[0:4] == ['d', "da'", "b'", 'da"'] and
        codedWord[-5:len(codedWord)] == ["c'", 'b"', 'd', 'c"', 'd']:
            codedWord[2] = "x'"
            #This is the modification for Case 10: d da' x' da"...#
        elif codedWord[0:3] == ['d', "b'", 'd'] and
        (codedWord[-3:len(codedWord)] == ['d', 'b"', 'd'] or
        codedWord[-4:len(codedWord)] == ["dc'", 'b"', 'dc"', 'd']):
            codedWord[1] = 'b^'
            #This is the modification for Case 18: d b^ d...#
        elif codedWord[0:4] == ['d', "da'", "b'", 'da"'] and
        (codedWord[-3:len(codedWord)] == ['d', 'b"', 'd'] or
        codedWord[-4:len(codedWord)] == ["dc'", 'b"', 'dc"', 'd']):
            codedWord[2] = 'b^'
            #This is the modification for Case 19: d da' b^ da"...#
        else:
            pass
    else:
        pass
    return codedWord

def noninitialPrefix(word,m,n):
    """
    This function is applied if the N set we are encoding is not the
    initial set of a whole permutation. word is the encoded word by
    encodeN(word) function, m is the current set number
    (3 leq m leq n), and n is the final set (found by counter(perm)
    function).
    """
    codedWord = word[0:len(word)]
    if codedWord[0:2] == ['d', 'd']:
    #This may happen if the previous set ended with:#
    #1. Simply with d d.#
    #2. d_ d. We do not carry d_ to the following set.#
    #We simply erase d d, since they are previously encoded.#
        codedWord = codedWord[2:]
    elif codedWord[0:3] == ['d', "a'", 'd']:
    #This is associated with z x d y interaction.#
    #a' could be various things. The associated a" is z, or . Those#
    #are previously encoded,#
    #so we make sure to erase d a' d and a".#
        codedWord = codedWord[3:]
    #There are three possiblilities immediately after d a' d, namely#
    #bs, b' or a'.#
```

```
#If it is a’, then the suffix of the previous set was encoded by#
#z c” d.#
#We first get rid of this case.#
    while codedWord[0] != ’bs’ and codedWord[0] != ”b’”:
        codedWord = codedWord[1:]
    codedWord.remove(’a”’)
#Now, it’s down to two possibilities; bs or b’.#
    if codedWord[0] == ’bs’:
    #If it is followed by bs, then all we have to do is to#
    #replace this with y.#
        codedWord[0] = ’y’
    elif codedWord[0] == ”b’”:
    #If it is followed by b’, this should be y’ or y^.#
    #Now, if the number of b’ & b” pair is 1, AND b” happens#
    #in the suffix,#
    #then this case needs a special attention.#
        if codedWord.count(”b’”) == 1:
            if codedWord[-3:len(codedWord)] == [’d’, ’b”’, ’d’]:
            #We do not modify ’b”’ because it is#
            #included in the suffixConverter.#
                if m != n:
                #The current set we are encoding is not the
                #final set. Then b’ -> y^.#
                    codedWord[0] = ’y^’
                else:
                #I.e., m == n. In this case, b’ -> y’.#
                #Then d y” d at the end.#
                    codedWord[0] = ”y’”
            elif codedWord[-4:len(codedWord)] ==
            [”dc’”, ’b”’, ’dc”’, ’d’]:
            #Similar to the above case.#
            #The difference is whether the following b”#
            #becomes d_ (previous case) or d_^ (this case).#
            #Again, We do not modify ’b”’ because it is
            #included in the suffixConverter.#
                if m != n:
                #The current set we are encoding is not the#
                #final set. Then b' -> y^.#
                    codedWord[0] = 'y^'
                else: #I.e., m == n. In this case, b' -> y'.#
                #Then dc' y" dc" d at the end.#
                    codedWord[0] = "y'"
            elif codedWord[-5:len(codedWord)] ==
            ["c'", 'b"', 'd', 'c"', 'd']:
            #Very special case.#
            #This is the structure of#
            #d a' d b' a" (a) (c) c' b" d c" d, resulting#
            #y' z y" d y encoding. We do not convert b"#
            #since it is a part of the suffix.#
                codedWord[0] = "y'"
            else:
            #The case b'& b" pair happens once, but b" is not#
            #in the suffix.#
            #In this case, b" -> y" as well as b' -> y'.#
            #Since b" isn't included in#
            #the suffix, we must convert it here.#
                codedWord[0] = "y'"
                codedWord[codedWord.index('b"')] = 'y"'
        else:
        #The case b' & b" pair happens more than once.#
        #Same as before, b" -> y" as well as#
        #b' -> y' because b" isn't included in the suffix.#
            codedWord[0] = "y'"
            codedWord[codedWord.index('b"')] = 'y"'
elif codedWord[0:3] == ['d', "b'", 'd']:
#This is associated with d^ b_ interaction.#
#b' is either d^ or d_^. This is previously encoded, so we#
#simply erase d b' d.#
#The associated b" can be many different things, such as#
#b_, d_, d_^, and x_.#
#If b" is a part of the suffix, then we postpone the#
#modification till suffixConverter.#
#If this isn't the case, then we modify b" here.#
    codedWord = codedWord[3:]
```

```
        if codedWord.count('b"') == 1:
        #Since we have erased b', we check if b" only shows up once.#
            if codedWord[-3:len(codedWord)] == ['d', 'b"', 'd']:
            #We do not modify b" because it is#
            #included in the suffixConverter.#
                pass
            elif codedWord[-4:len(codedWord)] ==
            ["dc'", 'b"', 'dc"', 'd']:
            #We do not modify b" because it is#
            #included in the suffixConverter.#
                pass
            elif codedWord[-5:len(codedWord)] ==
            ["c'", 'b"', 'd', 'c"', 'd']:
                if m != n:
                #If the set isn't final, we leave b" for the#
                #suffixConverter.#
                    pass
                else: #Otherwise, b" needs to be modified to b_,#
                #since finalSuffix function don't#
                #deal with the case of ending with d c" d.#
                    codedWord[codedWord.index('b"')] = 'b_'
            else: #The case b'& b" pair happens once, but b" is not#
            #in the suffix.#
            #In this case, we must convert b" -> b_.#
                codedWord[codedWord.index('b"')] = 'b_'
        else: #The case b' & b" pair happens more than once. Same as#
        #before, b" -> b_#
        #because b" isn't included in the suffix.#
            codedWord[codedWord.index('b"')] = 'b_'
    else: #Any other prefix should never appear due to the way we#
    #split each set.#
        codedWord = print("Error.␣Something␣went␣wrong...")
    return codedWord
```

suffixConverter.py

```
"""
```

```
Taking an arbitrary N or S set out of a simple adbi permutation of
extreme pattern 2143, the prefix & suffix of it must be modified
according to:
1. where it comes from, and
2. how it connects to the previous/next N & S sets.
The following functions will modify the suffix of the N or S set
that we are currently encoding.
There is a slight difference from prefixConverter functions. word is
the original, and codedWord is the encoded word after prefixConverter
is applied.
”””

import Nsplit
import encodeNS
import prefixConverter

def finalSuffix(word,codedWord):
    ”””
    This function is particularly applied if the N set we are
    encoding is the final set of a whole permutation, but not the
    initial set. In addition to the existing 6 ways to end the
    encoded word, there are 4 more possibilities associated with
    y” and b_.
    ”””
    if word.count(”b’”) == 1: # This line ensures that the first b’#
    #showing up needs to be modified to x’ or b^.#
        if word[-3:len(word)] == [’d’, ’b”’, ’d’] and word[0:5] ==
        [’d’, ”a’”, ’d’, ”b’”, ’a”’]:
            codedWord[-2] = ’y”’ #This is the modification for the#
            #reverse of Case 9: ...d y” d ending case.#
        elif word[-4:len(word)] == [”dc’”, ’b”’, ’dc”’, ’d’] and
        word[0:5] == [’d’, ”a’”, ’d’, ”b’”, ’a”’]:
            codedWord[-3] = ’y”’ #This is the modification for the#
            #reverse of Case 10: ...dc’ y” dc” d#
        elif word[-3:len(word)] == [’d’, ’b”’, ’d’] and (word[0:3] ==
        [’d’, ”b’”, ’d’] or word[0:4] == [’d’, ”da’”, ”b’”, ’da”’]):
            codedWord[-2] = ’b_’ #This is the modification for the#
            #reverse of Case 18: ...d b_ d#
        elif word[-4:len(word)] == [”dc’”, ’b”’, ’dc”’, ’d’] and
        (word[0:3] == [’d’, ”b’”, ’d’] or word[0:4] ==
        [’d’, ”da’”, ”b’”, ’da”’]):
            codedWord[-3] = ’b_’ #This is the modification for the#
            #reverse of Case 19: ...dc’ b_ dc” d#
        else:
            pass
    else:
        pass
    return codedWord

def nonfinalSuffix(word,codedWord,m):
    ”””
    This function is applied if the N set we are encoding is not the
    final set of a whole permutation. word is the original encoded
    word which is the result of encodeN(word) function, codedWord is
    the encoded word by either initialPrefix(word) or
    noninitialPrefix(word) function, m is the current set number
    (2 leq m leq (# of final set)-1).
    ”””
    last1 = len(word)
    last2 = len(codedWord) #Just to make things a little easier...#
    if word[-2:last1] == [’d’, ’d’]: #We erase the second d, since#
    #this will be encoded in the next set.#
        codedWord = codedWord[0:last2 -1]
    elif word[-4:last1] == [”dc’”, ’bs’, ’dc”’, ’d’]:
    #The case which didn’t rise in the prefixConverter.#
    #We need to erase dc’ and dc”, and convert b -> d^. dc” will be#
    #converted in the next set.#
        codedWord = codedWord[0:last2 -4]
        #Simply erase all four of them first.#
        codedWord.append(’d^’) #Then add d^, and#
        codedWord.append(’d’) #add back d.#
    elif word[-3:last1] == [’d’, ’c”’, ’d’]:
    #This is associated with z x d y interaction.#
    #c’ is z. The associated c” is y. The scissoring element for#
```

```
#these c’ & c” could be various things.#
#We first erase the suffix d c” d, modify whatever needs to be#
#modified, then add back necessary stuff.#
#c” and the first d are encoded in the future, so for now, we#
#will not add these back.#
    word = word[0:last1 -3]
    codedWord = codedWord[0:last2 -3]
    codedWord[-2] = ’z’ #Fix c’ to be z here.#
#There are three possiblilities immediately before d c” d#
#namely bs, b” or c”.#
#If it is c”, then we keep it as c”, so we have z c” d.#
    if word[-1] == ’c”’:
        codedWord.append(’d’)
    elif word[-1] == ’bs’: #If it is bs, then all we have to do#
    #is to replace this with x.#
        codedWord[-1] = ’x’
        codedWord.append(’d’)
    elif word[-1] == ’b”’:
    #If it is b”, this should be x”, x_ or y”.#
    #Now, if the number of b’ & b” pair is 1, AND b’ happens in#
    #the prefix,#
    #then this case needs a special attention.#
        if word.count(”b’”) == 1:
            if word[0:3] == [’d’, ”b’”, ’d’]:
            #We do not modify ”b’” because it is#
            #included in the prefixConverter.#
                if m != 2:
                #The current set we are encoding is not the#
                #initial set. Then b” -> x_.#
                    codedWord[-1] = ’x_’
                    codedWord.append(’d’)
                else: #I.e., m == 2. In this case, b” -> x”.#
                #d x” d should be the actual suffix.#
                    codedWord[-1] = ’x”’
                    codedWord.append(’d’)
            elif word[0:4] == [’d’, ”da’”, ”b’”, ’da”’]:
            #We do not modify b’ because it is#
    #included in the prefixConverter.#
        codedWord[-1] = 'x"'
        #Now, Since this particular prefix only#
        #happens when the set is#
        #initial, we may assume m == 2 here.#
        #Thus, this can never be x_.#
        codedWord.append('d')
    elif word[0:5] == ['d', "a'", 'd', "b'", 'a"']:
    #Very special case.#
    #This is the structure of#
    #d a' d b' a" (a) (c) c' b" d c" d, resulting#
    #y' z y" d y encoding. We do not convert b'#
    #since it is a part of the prefix.#
        if m != 2: #The current set we are encoding is#
        #not the initial set. Then b" -> y".#
            codedWord[-1] = 'y"'
            codedWord.append('d')
        else: #If it is the initial set, then b" becomes#
        #x" instead. In addition, we must change#
        #the associated b' to x'.#
            codedWord[-1] = 'x"'
            codedWord[codedWord.index("b'")] = "x'"
            codedWord.append('d')
    else: #The case b'& b" pair happens once, but b' is#
    #not in the prefix.#
    #In this case, b' -> x' as well as b" -> x".#
    #Since b' isn't included in#
    #the prefix, we must convert it here.#
        codedWord[-1] = 'x"'
        codedWord[codedWord.index("b'")] = "x'"
        codedWord.append('d')
else: #The case b' & b" pair happens more than once.#
#Same as before, b' -> x' as well as#
#b" -> x" because b' isn't included in the prefix.#
    codedWord[-1] = 'x"'
    bPrimeIndexList =
    [i for i, j in enumerate(codedWord) if j == "b'"]
```

```
            #Make a list with indices of all b’ in#
            #codedWord list.#
            codedWord[bPrimeIndexList[−1]] = ”x’”
            codedWord.append(’d’)
elif word[−3:len(word)] == [’d’, ’b”’, ’d’]:
#This is associated with bˆ d_ interaction.#
#b” is d_. As usual, after erasing d b” d, we add back#
#whatever we need. The associated b’ can be many different#
#things, such as bˆ, dˆ, d_ˆ, and yˆ. If the associated b’ is a#
part of the prefix, then we must have modified it previously.#
#If this isn’t the case, then we modify b’ here.#
    word = word[0:last1 −3]
    codedWord = codedWord[0:last2 −3]
    if word.count(”b’”) == 1:
    #As usual, count the number of b’ appearing.#
        if word[0:3] == [’d’, ”b’”, ’d’]:
        #We do not modify b’ because it is#
        #included in the prefixConverter.#
            pass
        elif word[0:4] == [’d’, ”da’”, ”b’”, ’da”’]:
        #We do not modify b’ because it is#
        #included in the prefixConverter.#
            pass
        elif word[0:5] == [’d’, ”a’”, ’d’, ”b’”, ’a”’]:
            if m != 2:
            #If the set isn’t initial, we modify b’ in the#
            #prefixConverter.#
                pass
            else: #Otherwise, b’ needs to be modified to bˆ,#
            #since initialPrefix function don’t#
            #deal with the case of beginning with d a’ d.#
                codedWord[codedWord.index(”b’”)] = ’bˆ’
        else: #The case b’& b” pair happens once, but b’ is#
        #not in the prefix. In this case, we must convert#
        #b’ −> bˆ.#
            if codedWord.count(”b’”) == 1:
            #This double checks if it really isn’t fixed in the#
            #prefixConverter. In the case of N-C chain, the#
            #above three cannot cover it.#
                codedWord[codedWord.index("b'")] = 'b^'
            else:
                pass
    else: #The case b' & b" pair happens more than once.#
    #Same as before, b' -> b^#
    #because b' isn't included in the prefix.#
        bPrimeIndexList =
        [i for i, j in enumerate(codedWord) if j == "b'"]
        #Make a list with indices of all b' in codedWord list.#
        codedWord[bPrimeIndexList[-1]] = 'b^'
    codedWord.append('d_')
    codedWord.append('d')
elif word[-4:len(word)] == ["dc'", 'b"', 'dc"', 'd']:
#This is associated with b^ d_^ interaction. The case which#
#didn't rise in the prefixConverter.
#The modification is identical to the case of b^ d_.#
    word = word[0:last1 -4]
    codedWord = codedWord[0:last2 -4]
    if word.count("b'") == 1:
    #As usual, count the number of b' appearing.#
        if word[0:3] == ['d', "b'", 'd']:
        #We do not modify b' because it is#
        #included in the prefixConverter.#
            pass
        elif word[0:4] == ['d', "da'", "b'", 'da"']:
        #We do not modify b' because it is#
        #included in the prefixConverter.#
            pass
        elif word[0:5] == ['d', "a'", 'd', "b'", 'a"']:
            if m != 2: #If the set isn't initial, we modify b'#
            #in the prefixConverter.#
                pass
            else: #Otherwise, b' needs to be modified to b^,#
            #since initialPrefix function don't#
            #deal with the case of ending with d a' d.#
```

```
                    codedWord[codedWord.index("b'")] = 'b^'
            else: #The case b'& b" pair happens once, but b' is not#
            #in the prefix. In this case, we must convert b' -> b^.#
                if codedWord.count("b'") == 1:
                #This double checks if it really isn't fixed in the#
                #prefixConverter. In the case of N-C chain, the#
                #above three cannot cover it.#
                    codedWord[codedWord.index("b'")] = 'b^'
                else:
                    pass
        else: #The case b' & b" pair happens more than once.#
        #Same as before, b' -> b^#
        #because b' isn't included in the prefix.#
            bPrimeIndexList =
            #[i for i, j in enumerate(codedWord) if j == "b'"]
            #Make a list with indices of all b' in codedWord list.#
            codedWord[bPrimeIndexList[-1]] = 'b^'
        codedWord.append('d_^')
        codedWord.append('d')
    else: #Any other prefix should never appear due to the way we#
    #split each set.#
        print("Error.")
        codedWord += ['<-␣Something␣went␣wrong␣here.']
    return codedWord
```

encode.py

```
"""
The following function takes a permutation having extreme pattern of
either 2413, or 2143 starting with N set.
"""

import counter
import encodeNS
import invert
import Nsplit
import prefixConverter
import suffixConverter

def encode(perm):
    word = [] #We will return this list as the final word.#
    m = 2 #Initial number.#
    n = counter.counter(perm) #Final number.#
    if n == 2: #The permutation is of extreme pattern 2413.#
        word = encodeNS.encodeN(perm)
    else:
        while m != n+1:
            subperm = Nsplit.Nsplit(perm)[0] #Split it.#
            CLast = Nsplit.Nsplit(perm)[1]
            subword1 = encodeNS.encodeN(subperm)
            #Encode the split subpermutation.#
            #Perform the modifications of prefix/suffix Converters.#
            if m == 2: #Initial set.#
                subword2 = prefixConverter.initialPrefix(subword1)
                #Result of the prefixConverter.#
                subword2 =
                suffixConverter.nonfinalSuffix(subword1,subword2,m)
                #Result of the suffixConverter.#
            elif m == n: #Final set.#
                subword2 =
                prefixConverter.noninitialPrefix(subword1,m,n)
                #Result of the prefixConverter.#
                subword2 =
                suffixConverter.finalSuffix(subword1,subword2)
                #Result of the suffixConverter.#
            else: #Neither initial nor final.#
                subword2 =
                prefixConverter.noninitialPrefix(subword1,m,n)
                #Result of the prefixConverter.#
                subword2 =
                suffixConverter.nonfinalSuffix(subword1,subword2,m)
                #Result of the suffixConverter.#
            word += subword2
            perm = Nsplit.setMinus(perm,CLast,subword2)
            perm = invert.invPerm(perm)
            m += 1
    word[−1] = ’dl’
    return word
```

main.py

```
import makeList
import createFile
import ep
import encode
import list2Str
import Nsplit

input(”We␣first␣read␣a␣text␣file␣of␣permutations,␣and␣convert␣the
string␣type␣to␣the␣integer␣type.”)
print(”Enter␣a␣text␣file␣directory.”)

while(True):
    try:
        permList = input()
        permList = makeList.readFile(permList)
        break
    except:
        print(”There␣is␣no␣such␣a␣file␣directory.”)
if len(permList[0])<10:
    permList = makeList.convList1(permList)
else:
    permList = makeList.convList2(permList)

print(”Done.␣Permutations␣in␣the␣txt␣file␣are␣stored␣in␣permList.\n”)
operation1 = input(”Now,␣we␣delete␣all␣permutations␣which␣satisfy␣at
least␣one␣of␣the␣following␣conditions.\n␣1.The␣first␣number␣is␣not␣2,
3,␣or␣4.␣\n␣2.The␣second␣number␣is␣1.”)
halfList=[]
for i in permList:
        halfList.append(ep.Nstart(i))
```

```
while(True):
    try:
        halfList.remove(None)
    except:
        break
print(”Done.␣Permutations␣starting␣with␣2,␣3,␣or␣4␣AND␣the␣second
number␣is␣not␣1␣are␣stored␣in␣halfList.”)
operation2 = input(”Type␣encode␣to␣create␣a␣text␣file␣of␣coded␣words.
Press␣enter␣to␣exist.\n”)
words=[]
while operation2 != ’encode’ and operation2 != ’’:
    print(”Type␣encode␣or␣press␣enter␣to␣exit.”)
    operation2 = input()
if operation2 == ’encode’:
    destination =
    ’C:\\Users\\Ikeda\\Desktop\\encode(%d).txt’%len(permList[0])
    for i in halfList:
        words.append(list2Str.list2Str(encode.encode(i)))
    createFile.createFile(destination,words)
    print(”Done.␣The␣text␣file␣is␣now␣created.”)
else:
    pass

def uniqueCheck(words):
    for i in range(len(words)-1):
        for j in words[i+1:]:
            if words[i] != j:
                pass
            else:
                print(j)

def createWordList(halfList):
    wordList = []
    for i in halfList:
        wordList.append(encode.encode(i))
    return wordList
```

```
def lengthCheck(wordList):
    n = len(wordList[0])
    for i in wordList:
        if len(i) != n:
            print(i)
        else:
            pass
```

References

[1] M. H. Albert, *Permlab: Software for permutation patterns*, 2012.

[2] M. H. Albert and M. D. Atkinson, *Simple permutations and pattern restricted permutations*, Discrete Math. **300** (2005), no. 1-3, 1–15. MR2170110 (2006d:05007)

[3] M. H. Albert, M. D. Atkinson, and Robert Brignall, *The enumeration of permutations avoiding* 2143 *and* 4231, Pure Math. Appl. (PU.M.A.) **22** (2011), no. 2, 87–98. MR2924740

[4] M. H. Albert, M. D. Atkinson, and M. Klazar, *The enumeration of simple permutations*, J. Integer Seq. **6** (2003), no. 4, Article 03.4.4, 18. MR2051958

[5] Michael H. Albert, M. D. Atkinson, Mathilde Bouvel, Nik Ruškuc, and Vincent Vatter, *Geometric grid classes of permutations*, Trans. Amer. Math. Soc. **365** (2013), no. 11, 5859–5881. MR3091268

[6] Michael H. Albert, M. D. Atkinson, and Vincent Vatter, *Counting 1324, 4231-avoiding permutations*, Electron. J. Combin. **16** (2009), no. 1, Research Paper 136, 9. MR2577304 (2011e:05003)

[7] Michael H. Albert and Robert Brignall, *Enumerating indices of Schubert varieties defined by inclusions*, J. Combin. Theory Ser. A **123** (2014), 154–168. MR3157805

[8] Michael H. Albert, Cheyne Homberger, Jay Pantone, Nathaniel Shar, and Vincent Vatter, *Generating permutations with restricted containers* (2015), available at `arXiv:math/1510.00269v3`.

[9] Noga Alon and Ehud Friedgut, *On the number of permutations avoiding a given pattern*, J. Combin. Theory Ser. A **89** (2000), no. 1, 133–140. MR1736130 (2000i:05007)

[10] David Bevan, *Permutations avoiding 1324 and patterns in Łukasiewicz paths* (2015), available at `arXiv:math/1406.2890v2`.

[11] Miklós Bóna, *Exact enumeration of* 1342*-avoiding permutations: a close link with labeled trees and planar maps*, J. Combin. Theory Ser. A **80** (1997), no. 2, 257–272. MR1485138 (98j:05003)

[12] ———, *The solution of a conjecture of Stanley and Wilf for all layered patterns*, J. Combin. Theory Ser. A **85** (1999), no. 1, 96–104. MR1659444 (99i:05005)

[13] ———, *A new record for 1324-avoiding permutations*, European Journal of Mathematics **1** (2015), no. 1, 198–206 (English).

[14] Mireille Bousquet-Mélou and Steve Butler, *Forest-like permutations*, Ann. Comb. **11** (2007), no. 3-4, 335–354. MR2376109 (2009b:05003)

[15] Robert Brignall, *A survey of simple permutations*, Permutation patterns, 2010, pp. 41–65. MR2732823 (2012d:05008)

[16] Murray Elder and Vincent Vatter, *Problems and conjectures presented at the third international conference on permutation patterns, university of florida, march 7-11, 2005* (2005), available at `arXiv:math/0505504`.

[17] P. Erdös and G. Szekeres, *A combinatorial problem in geometry*, Compositio Math. **2** (1935), 463–470. MR1556929

[18] Jacob Fox, *Stanley-wilf limits are typically exponential* (2013), available at `arXiv:math/1310.8378v1`.

[19] S. Garrabrant and I Pak, *Pattern avoidance is not P-recursive* (2015), available at `arXiv:math/1505.06508v1`.

[20] V. Gasharov and V. Reiner, *Cohomology of smooth Schubert varieties in partial flag manifolds*, J. London Math. Soc. (2) **66** (2002), no. 3, 550–562. MR1934291 (2003i:14064)

[21] Ira M. Gessel, *Symmetric functions and P-recursiveness*, J. Combin. Theory Ser. A **53** (1990), no. 2, 257–285. MR1041448 (91c:05190)

[22] M. Haiman, *Smooth schubert varieties*, Preprint (1992).

[23] Tomáš Kaiser and Martin Klazar, *On growth rates of closed permutation classes*, Electron. J. Combin. **9** (2002/03), no. 2, Research paper 10, 20. Permutation patterns (Otago, 2003). MR2028280 (2004m:05026)

[24] Martin Klazar, *The Füredi-Hajnal conjecture implies the Stanley-Wilf conjecture*, Formal power series and algebraic combinatorics (Moscow, 2000), 2000, pp. 250–255. MR1798218 (2001k:05005)

[25] D. Knuth, *The art of computer programming*, Vol. 1, Addison-Wesley, Reading, MA, 1968.

[26] ———, *The art of computer programming*, Vol. 3, Addison-Wesley, Reading, MA, 1973.

[27] Darla Kremer, *Permutations with forbidden subsequences and a generalized Schröder number*, Discrete Math. **218** (2000), no. 1-3, 121–130. MR1754331 (2001a:05005)

[28] ______, *Postscript: "Permutations with forbidden subsequences and a generalized Schröder number" [Discrete Math.* **218** *(2000), no. 1-3, 121–130; MR1754331 (2001a:05005)]*, Discrete Math. **270** (2003), no. 1-3, 333–334. MR1997910 (2004d:05010)

[29] V. Lakshmibai and B. Sandhya, *Criterion for smoothness of Schubert varieties in* $\mathrm{Sl}(n)/B$, Proc. Indian Acad. Sci. Math. Sci. **100** (1990), no. 1, 45–52. MR1051089 (91c:14061)

[30] P. MacMahon, *Combinatory analysis*, Vol. 1, Cambridge University Press, London, 1915.

[31] Adam Marcus and Gábor Tardos, *Excluded permutation matrices and the Stanley-Wilf conjecture*, J. Combin. Theory Ser. A **107** (2004), no. 1, 153–160. MR2063960 (2005b:05009)

[32] published electronically at http://oeis.org. N. J. A. Sloane, *Online encyclopedia of integer sequences.* http://oeis.org/.

[33] John Noonan and Doron Zeilberger, *The enumeration of permutations with a prescribed number of "forbidden" patterns*, Adv. in Appl. Math. **17** (1996), no. 4, 381–407. MR1422065 (97j:05003)

[34] T. Kyle Petersen and Bridget Eileen Tenner, *The depth of a permutation* (2012), available at `arXiv:math/1202.4765v3`.

[35] Vaughan R. Pratt, *Computing permutations with double-ended queues. Parallel stacks and parallel queues*, Fifth Annual ACM Symposium on Theory of Computing (Austin, Tex., 1973), 1973, pp. 268–277. MR0489115 (58 #8588)

[36] Rodica Simion and Frank W. Schmidt, *Restricted permutations*, European J. Combin. **6** (1985), no. 4, 383–406. MR829358 (88a:05006)

[37] Daniel A. Spielman and Miklós Bóna, *An infinite antichain of permutations*, Electron. J. Combin. **7** (2000), Note 2, 4 pp. (electronic). MR1741337 (2000k:05014)

[38] Henning Ulfarsson and Alexander Woo, *Which Schubert varieties are local complete intersections?*, Proc. Lond. Math. Soc. (3) **107** (2013), no. 5, 1004–1052. MR3126390

[39] Vincent Vatter, *Small permutation classes*, Proc. Lond. Math. Soc. (3) **103** (2011), no. 5, 879–921. MR2852292 (2012j:05022)

[40] ______, *Permutation classes* (2015), available at `arXiv:math/1409.5159v3`.