Automatic Asymptotics and Generating Functions

Bruno Salvy
INRIA Rocquencourt

September 16, 1992

[summary by Bruno Salvy]

Abstract

Computer algebra systems can be of help in the asymptotic analysis of combinatorial sequences. Several algorithms are presented, most of which have been implemented in Maple.

Introduction

We assume a sequence is given, either by its first terms or by a combinatorial description of a class of objects it enumerates. The main tool we use is the *generating function* of the sequence. The idea is to consider this formal power series as an analytic function. When the series has a non-zero radius of convergence, Cauchy's theory makes it possible to find an asymptotic estimate of the sequence we started with.

1. From the sequence to the series

The preferred method naturally depends on the available information concerning the sequence.

Empirical method. When only the first few terms of the sequence are known, there are *a priori* an infinite number of possible sequences, and there seems to be little sense in looking for an asymptotic behaviour. However, there is quite often a "simple" sequence defined by these first terms. This approach was initiated by F. Bergeron and S. Plouffe [2], who looked for Padé approximants of the generating series. When the number of non-zero coefficients of the Padé approximant is "significantly" smaller than the number of given terms of the sequence, it is natural to conjecture that the generating series is rational and that a closed-form was found. This method can be extended by applying it to the logarithmic derivative or to the functional inverse of the given power series, which yields nice generating functions.

With P. Zimmermann, we applied this idea of looking for a "simple" generating function given its first coefficients to the quest of "holonomic" sequences, i.e. sequences satisfying a linear recurrence with polynomial coefficients. Rather than looking for a Padé approximant, this recurrence is sought by an undeterminate coefficients method. When the number of non-zero coefficients of the recurrence is "sufficiently" smaller than the number of given terms, the recurrence is conjectured as being satisfied by the whole sequence. This is implemented in the Gfun package [12].

Both these methods are very efficient in practice. Among the approximately 6000 sequences of the next edition of Sloane's book [14], roughly 25% of the sequences are thus conjectured rational, and an extra 5% are conjectured holonomic non-rational [9].

Combinatorial method. A large number of sequences f_n enumerate the number of objects of size n in some *decomposable* combinatorial data-structure. This means that the structure can be expressed in terms of a small combinatorial toolbox comprising cartesian product, disjoint union, list, set, cycle and basic atoms. Thus the structure "functional graph" (the graph of an application of a set of n elements into itself) is expressed as a set of connected components, these components being cycles of trees, these trees themselves being recursively defined as the cartesian product of a node (the root of the tree) by a set of trees.

The $\Lambda\Upsilon\Omega$ system, developed jointly with P. Zimmermann and Ph. Flajolet [3, 4] implements a translation of these combinatorial specifications into equations relating the corresponding generating functions. In the example of functional graphs, the first part of the system will produce the following equations:

$$\mathrm{FuncGraph}(z) = \exp(\mathrm{comp}(z)), \quad \mathrm{comp}(z) = \log[1/(1-\mathrm{tree}(z))], \quad \mathrm{tree}(z) = z\,\exp(\mathrm{tree}(z)).$$

A second part of the system then attempts to find an explicit form of the generating function from this system. For, in its current state, the asymptotic part of the $\Lambda\Upsilon\Omega$ system can only handle explicit generating functions. In this example, thanks to Maple's W function, the following "explicit" form is obtained:

$$\frac{1}{1+W(-z)}.$$

Conclusion. Two very different methods have been described to obtain the generating function of a sequence. The first one finds *holonomic* generating functions, i.e. solutions of linear differential equations with polynomial coefficients. The second one is more combinatorial and finds generating functions that obey functional equations expressed in terms of some "elementary" functions. In some cases, these equations can be solved.

Known algorithms to get "explicit" forms from these equations can be summarised as follows.

- Liouvillian solutions of linear differential equations can be obtained by Kovacic's algorithm for the case of order 2. This algorithm is (at least partially) implemented in most computer algebra systems. An algorithm due to M. Singer treats the general case, but is not practical. The third order has been made practical by F. Ulmer, but there is no generally available implementation;
- Hypergeometric solutions of linear differential equations can be found by an algorithm due principally to M. Petkovšek, without any limitation on the order of the equation [8];
- Elementary functional equations can only be solved in some special cases.

2. From generating functions to asymptotics

When the generating series defines an analytic function, Cauchy's formula yields the nth Taylor coefficient as

$$[z^n]f(z) = \frac{1}{2i\pi}\oint \frac{f(z)}{z^{n+1}}dz.$$

The path of integration is a closed contour containing the origin and no other singularity.

We are looking for an asymptotic estimate as n tends to infinity. First of all, Hadamard's rule implies that the coefficients grow roughly as $1/R^n$, where R is the radius of convergence. This relates the exponential growth of the Taylor coefficients of a generating function to the location of its singularities. Besides, simple functions whose coefficients are known, such as $1/(1-z)^\alpha$, give the intuition that sub-exponential growth of the coefficients is related to the local growth of the generating function in the neighbourhood of its singularity of smallest modulus. This can be made precise.

2.1. Singularity analysis. In 1878, G. Darboux treated the case of algebraic singularities. This result was extended by R. Jungen in 1934 to handle singularities in $(1-z)^\alpha \log^k(1-z)$, where k is a non-negative integer. Finally, Ph. Flajolet and A. Odlyzko [5] described the more general case where the exponents of $(1-z)$ and of the logarithm are complex numbers. These methods yield a full asymptotic expansion of the Taylor coefficients.

This leads to the following algorithm to find the asymptotic expansion of coefficients of a generating function.

(1) Locate the singularities of smallest modulus;
(2) Compute the expansion of the function in the neighbourhood of these singularities;
(3) Translate this expansion into the expansion of the coefficients.

The last step above is easy. We now insist on how the first two steps can be automated. This depends on the type of equation defining the generating function.

When the generating function is given as a solution to a linear differential equation, its singularities are found among the poles of the coefficients of the equation and the roots of its leading coefficient. Since the coefficients are polynomials, singularities in this case are therefore algebraic numbers. When the generating function is given explicitly in terms of elementary functions, it is easy to find a set of points containing the singularities by a recursive algorithm.

Then one has to compare the moduli of the singularities. Algebraic numbers can be compared by purely algebraic methods using resultants and Sturm sequences. It is also possible to make use of guaranteed numerical estimates, see [6]. In the more general case of elementary constants one is confined to heuristics, the problem being related to difficult questions of transcendency.

Once the dominant singularities have been located, one looks for the local behaviour of the generating function in the neighbourhood of these singularities. When the function is given explicitly as an exp-log function (functions built up from $\mathbb{Q}$ and x by field operation, exp and $x \mapsto \log|x|$), a recent algorithm due to J. Shackell [13] makes it possible to compute the local expansion. When the generating function is holonomic, the possible behaviours have been given by E. Fabry in 1885, and have the form

$$\exp[P(1/(1-(z/\rho)^{1/d}))](1-z/\rho)^{\alpha}\sum_{k=0}^{K}\phi_k(z)\log^k(1-z/\rho),$$

where ϕ_k are formal power series in $1-z/\rho$. Such local solutions can be determined automatically [15]. Once a basis of local solutions has been found, one has to find the right linear combination in terms of the first elements of the sequence. While these elements are given by the Taylor expansion of the function at the origin, we have a basis of local solutions at the singularity. Besides, the formal power series ϕ_k are generally divergent. One must then resort to the theory of resummation [1].

2.2. Saddle-point method. When the function is entire or has a singularity of a more "violent" type than a mere algebraico-logarithmic type, it is often possible to use a saddle-point method. Setting $h(z) = \log(f(z))-(n+1)\log z$, the contour of Cauchy's integral is deformed to pass through a point (*the saddle-point*) where $h'(z)=0$. With a few extra hypotheses, Cauchy's integral is then concentrated in the neighbourhood of the saddle-point and the integral can be approximated by a Gaussian. If we denote the saddle-point by R, the nth coefficient is then estimated as

$$[z^n]f(z) \approx \frac{f(R)}{R^{n+1}\sqrt{2\pi h''(R)}}.$$

To automate this method and the approximations it requires, one uses a theorem due to W. K. Hayman [7], which makes it possible to decide sufficient conditions under which the method applies. A last technical problem is that the saddle-point is often only available as an asymptotic expansion deduced from the equation $h'(R)=0$. An algorithm to compute this expansion under very general conditions has been developed in [11].

Bibliography

[1] Balser (W.), Braaksma (B. L. J.), Ramis (J.-P.), and Sibuya (Y.). – Multisummability of formal power series solutions of linear ordinary differential equations. *Asymptotic Analysis*, vol. 5, 1991, pp. 27–45.

[2] Bergeron (F.) and Plouffe (S.). – Computing the generating function of a series given its first terms. *Journal of experimental mathematics*, 1993.

[3] Flajolet (P.), Salvy (B.), and Zimmermann (P.). – *Lambda-Upsilon-Omega: The 1989 CookBook*. – Research Report n° 1073, Institut National de Recherche en Informatique et en Automatique, August 1989. 116 pages.

[4] Flajolet (P.), Salvy (B.), and Zimmermann (P.). – Automatic average-case analysis of algorithms. *Theoretical Computer Science, Series A*, vol. 79, n° 1, February 1991, pp. 37–109.

[5] Flajolet (Philippe) and Odlyzko (Andrew M.). – Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics*, vol. 3, n° 2, 1990, pp. 216–240.

[6] Gourdon (Xavier) and Salvy (Bruno). – Asymptotics of linear recurrences with rational coefficients. In Barlotti (A.), Delest (M.), and Pinzani (R.) (editors), *Formal Power Series and Algebraic Combinatorics*, pp. 253–266. – 1993. Proceedings of FPACS'5, Florence (Italy).

[7] Hayman (W. K.). – A generalization of Stirling's formula. *Journal für die reine und angewandte Mathematik*, vol. 196, 1956, pp. 67–95.

[8] Petkovšek (Marko) and Salvy (Bruno). – Finding all hypergeometric solutions of linear differential equations. In Bronstein (Manuel) (editor), *ISSAC'93*. pp. 27–33. – ACM Press, July 1993.

[9] Plouffe (S.). – *Approximations de séries génératrices et quelques conjectures.* – Master's thesis, Université du Québec à Montréal, September 1992. Also available as Research Report 92-61, Laboratoire Bordelais de Recherche en Informatique, Bordeaux, France.

[10] Salvy (Bruno). – *Asymptotique automatique et fonctions génératrices.* – PhD thesis, École Polytechnique, 1991.

[11] Salvy (Bruno) and Shackell (John). – Asymptotic expansions of functional inverses. In Wang (Paul S.) (editor), *Symbolic and Algebraic Computation*. pp. 130–137. – ACM Press, 1992. Proceedings of ISSAC'92, Berkeley.

[12] Salvy (Bruno) and Zimmermann (Paul). – *Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable.* – Technical Report n° 143, Institut National de Recherche en Informatique et en Automatique, 1992. To appear in *ACM Transactions on Mathematical Software*.

[13] Shackell (John). – Growth estimates for exp-log functions. *Journal of Symbolic Computation*, vol. 10, December 1990, pp. 611–632.

[14] Sloane (N. J. A.). – *A Handbook of Integer Sequences.* – Academic Press, 1973.

[15] Tournier (Évelyne). – *Solutions formelles d'équations différentielles.* – Doctorat d'État, Université scientifique, technologique et médicale de Grenoble, 1987.