Recursive Combinatorial Structures: Enumeration, Probabilistic Analysis and Random Generation

Bruno Salvy

Recursive Combinatorial Structures: Enumeration, Probabilistic Analysis and Random Generation

Bruno Salvy
Inria, LIP (U. Lyon, CNRS, ENS Lyon, UCBL), France
Bruno.Salvy@inria.fr

Abstract

In a probabilistic context, the main data structures of computer science are viewed as random combinatorial objects. Analytic Combinatorics, as described in the book by Flajolet & Sedgewick, provides a set of high-level tools for their probabilistic analysis. Recursive combinatorial definitions lead to generating function equations from which efficient algorithms can be designed for enumeration, random generation and, to some extent, asymptotic analysis. With a focus on random generation, this tutorial first covers the basics of Analytic Combinatorics and then describes the idea of Boltzmann sampling and its realisation.

The tutorial addresses a broad TCS audience and no particular pre-knowledge on analytic combinatorics is expected.

2012 ACM Subject Classification Mathematics of computing → Generating functions, Theory of computation → Generating random combinatorial structures

Keywords and phrases Analytic Combinatorics, Generating Functions, Random Generation

Digital Object Identifier 10.4230/LIPIcs.STACS.2018.1

Category Tutorial

Funding This work was partially supported by FastRelax ANR-14-CE25-0018-01.

1 Introduction

Combinatorial objects defined recursively by simple local rules tend to behave fairly regularly at large sizes. For instance, binary trees are defined by having nodes that are either leaves or binary internal nodes. From there, it turns out that all large random binary trees "look" the same. Also, many of their asymptotic characteristic persist for other classes of trees. The goal of analytic combinatorics is to understand and quantify those types of phenomena. Typical questions for binary trees could be: for a binary tree drawn uniformly at random among all binary trees with n nodes, what is the probability that the root has a leaf as one of its children? what is the expected distance from a random node to the root? what is the limiting distribution of this quantity?

The main applications are the probabilistic analysis of the average-case complexity of data structures and algorithms. Besides general "universality laws" of random discrete structures, the theory leads to very precise quantitative results. Analytic combinatorics is an active field of research, whose central core is described in detail, with numerous interesting examples, in the reference book *Analytic Combinatorics* by Flajolet and Sedgewick, published in 2009 and freely (and legally) available on-line [6]. Many methods of this theory can be made effective; this will be the focus of the tutorial.

35th Symposium on Theoretical Aspects of Computer Science (STACS 2018).
Editors: Rolf Niedermeier and Brigitte Vallée; Article No. 1; pp. 1:1–1:5

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


2 Constructible Classes

The classes of structures to which Analytic Combinatorics applies most directly are called *constructible*. They form an extension of context-free languages and can be defined recursively from a finite number of letters called atoms, of size 1 or 0, and combinatorial combinators: cartesian product, disjoint union, sequences, cycles and sets.

For example, the class of binary trees will be defined by the equation $\mathcal{B} = 1 + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$, expressing that a binary tree is either a '1' (a leaf encoded as an atom of size 0) or a triple formed by a root ($\mathcal{Z}$, an atom of size 1) and two binary trees attached to it. In terms of this class, the class of binary trees whose root has a leaf-child will be expressed by $\mathcal{Z} \times \mathcal{B} \times 1 + \mathcal{Z} \times 1 \times \mathcal{Z}$ (except that the binary tree with only one node is counted twice, which will not matter here). The class of non-planar rooted trees, also called Cayley trees, will be defined by $\mathcal{T} = \mathcal{Z} \times \mathrm{Set}(\mathcal{T})$, expressing that a tree is a node to which an arbitrary number of trees in an arbitrary order are attached. Series-parallel graphs will be defined by a system $\{\mathcal{G} = \mathcal{Z} + \mathcal{S} + \mathcal{P}, \mathcal{S} = \mathrm{Seq}_{>0}(\mathcal{Z} + \mathcal{P}), \mathcal{P} = \mathrm{Set}_{>0}(\mathcal{Z} + \mathcal{S})\}$, expressing that a graph ($\mathcal{G}$) is either a node ($\mathcal{Z}$) or a series graph ($\mathcal{S}$) or a parallel graph ($\mathcal{P}$), defined recursively in terms of each other. Such a system will be called a *specification* for the class. The motto of the theory is that "if a class can be specified, it can be analyzed."

3 Enumeration and Generating Functions

Given such a combinatorial specification for a class $\mathcal{F}$, the enumeration problem is to count the number f_n of distinct objects of a given size n in $\mathcal{F}$. For instance, knowing the number C_n (the Catalan number) of binary trees of size n and the number of those whose root has a leaf as one of its children lets one compute the probability that this event occurs.

The enumeration is simplified by the use of *generating functions*. The ordinary generating function of a sequence (f_n) is the formal power series $F(z) = \sum_{n\geq 0} f_n z^n$, while the exponential generating function is $\sum_{n\geq 0} f_n z^n/n!$. The use of one or the other depends on what exactly is counted as different. For instance there is only 1 sequence (list) of n atoms of size 1 if they are all identical, but $n!$ such sequences if they all carry a different label from 1 to n. Thus one distinguishes two 'universes': in the unlabelled universe, all atoms are alike and one uses ordinary generating functions, while in the labelled universe, an object of size n is formed of atoms labelled from 1 to n and one uses exponential generating functions. In both cases, there is an explicit dictionary translating the combinatorial specification into a system of equations for generating functions. Atoms of size 1 are translated into z and those of size 0 into $1 = z^0$; disjoint unions ('+') become additions and cartesian products become products of series. A sequence $\mathrm{Seq}(\mathcal{A})$ is translated into $1/(1 - A(z))$, where $A(z)$ is the generating function of $\mathcal{A}$. These first translation rules are identical in the labelled and unlabelled universes. Thus, in the example of binary trees, the equation is $B(z) = 1 + zB(z)^2$ in both cases and has for only power series solution the generating function of the Catalan numbers C_n. Similarly, in terms of this generating function, binary trees with one leaf-child at the root have generating function $2zB(z) - z$ (the term $-z$ discards the spurious tree of size 1), so that the probability mentioned in the introduction is $2C_{n-1}/C_n$ for $n > 1$. For sets and cycles, the rules differ, but remain simple in the labelled case, where Set becomes exp and $\mathrm{Cycle}(\mathcal{A})$ becomes $\log 1/(1 - A(z))$. For instance, in the case of Cayley trees, the exponential generating function is thus seen to satisfy $T(z) = z \exp(T(z))$.

4 Newton Iteration and Fast Enumeration

These equations give a first means to compute the enumeration sequences since they are fixed point equations over formal power series. A much more efficient algorithm is obtained by using Newton iteration for power series. The classical numerical Newton iteration solves an equation $f(y) = 0$ by approaching its root using the solution of successive linearized equations. Each iterate $y_{n+1} = y_n - f(y_n)/f'(y_n)$ is closer to the root when y_0 has been chosen appropriately. The extension of Newton iteration to formal power series is a standard algorithm in computer algebra [9]. Thus, for binary trees the Newton iteration reads $B_{n+1} = B_n + (1 + zB_n^2 - B_n)/(1 - 2zB_n)$. With the choice $B_0 = 0$, it produces a sequence of power series satisfying $B(z) - B_n = O(z^{2^n-1})$. Together with fast Fourier transform (FFT), Newton iteration lets one enumerate all constructible classes in quasi-optimal complexity [8][1]. It is even possible to obtain a combinatorial interpretation of the coefficients of the intermediate power series computed during the iteration: they enumerate combinatorial classes defined by a further lifting of the Newton iteration to combinatorial equations themselves, in the context of species theory [2, 1]. (Roughly speaking, species theory provides a sound theoretical basis grounded in category theory for what we have called "combinatorial class" so far.) For instance, in the case of binary trees, the combinatorial iteration then reads $\mathcal{B}_{n+1} = \mathcal{B}_n + \mathrm{Seq}(\mathcal{Z} \times \mathcal{B}_n \times \star + \mathcal{Z} \times \star \times \mathcal{B}_n) \times (1 + \mathcal{Z} \times \mathcal{B}_n \times \mathcal{B}_n \setminus \mathcal{B}_n)$, where $\star$ denotes an atom of size 0, which is interpreted as a 'bud' where trees can grow. The generating series of $\mathcal{B}_n$ is exactly the power series B_n produced by Newton iteration over power series.

5 Random Generation

Producing large random objects is the basis for simulation in a discrete world. Typical applications are the empirical evaluation of various parameters, software testing and the refinement of combinatorial models to suit an application.

A new family of efficient random generators called *Boltzmann samplers* was discovered at the beginning of the century [3, 4]. The principle is to draw each object of size n in a class $\mathcal{T}$ with a probability proportional to x^n for some prescribed positive real x and a factor of proportionality chosen so that the sum of probabilities is 1. Since the sum over all $t \in \mathcal{T}$ of $x^{\mathrm{size}(t)}$ is nothing but the evaluation of the generating function T of $\mathcal{T}$ at x, the probability will be $x^n/T(x)$, provided x is at most the radius of convergence of T. The algorithm of Boltzmann sampling itself is extremely simple and fits in about 10 lines. For atoms, the generator returns the atom; for cartesian products, it simply calls itself recursively on each of the components and assembles the results; for a disjoint union $\mathcal{A} + \mathcal{B}$, a random real number $t \in [0, 1]$ is compared to $u = A(x)/(A(x) + B(x))$ and the generator is called recursively on $\mathcal{A}$ if $t < u$ and on $\mathcal{B}$ otherwise. It is a simple exercise to check that the probabilities work out as expected. The values like $A(x)$ are provided by a numerical Newton iteration initialized at 0 [8]. The value of x can be adjusted to target a specific expected size $xT'(x)$ of the generated object.

[1] This is implemented in the `NewtonGF` Maple package, available at `http://perso.ens-lyon.fr/bruno.salvy/software/the-newtongf-package/`.

6 Asymptotic Analysis

Generating functions are also an entry point for the asymptotic analysis of their coefficients. The principle is to take these generating functions defined initially as power series given by a sequence of coefficients and view them as analytic functions in the complex plane. If $A(z) = \sum_{n\geq 0} a_n z^n$ has a positive radius of convergence R, then by Cauchy's theorem, its coefficient a_n can be recovered by the integral $a_n = \frac{1}{2\pi i} \oint (A(z)/z^{n+1})\, dz$, where for the contour of integration one can take a small circle around the origin of radius smaller than R. Classical complex analysis deforming the contour to locations where the integral concentrates asymptotically leads to very explicit and general results known as *transfer theorems*. The most useful case is when there is a unique singularity on the circle of convergence, at a point ρ (which will always be real), where the generating function behaves like $c(1 - z/\rho)^\alpha$, for $\alpha \notin \mathbb{N}$. Then the coefficients behave asymptotically like $c\rho^{-n} n^{-\alpha-1}/\Gamma(-\alpha)$. The outcome is a general 3-step method for the asymptotic analysis of generating functions: *(i).* locate the singularities of minimal modulus; *(ii).* compute the local behaviour of the generating function there (which can often be found directly from the defining equations); *(iii).* translate using a very simple dictionary. This process can be automated in a large part and full asymptotic expansions computed using computer algebra systems. In the case of binary trees, the quadratic equation satisfied by the generating function can be solved explicitly to yield $B(z) = (1 - \sqrt{1-4z})/(2z)$. From there, the explicit formula for the Catalan numbers $C_n = \binom{2n}{n}/(n+1)$ can be derived and the asymptotic behaviour could be computed by Stirling's formula. However, such closed forms are the exception rather than the rule, and the method applies in general. There, the dominant singularity is $1/4$, the local behaviour is $2 - 2\sqrt{1-4z} + O(1-4z)$. The coefficients of $B(z) - 2$ which are C_n for $n > 0$ thus behave asymptotically like $-2 \cdot 4^n n^{-3/2}/\Gamma(-1/2) = 4^n n^{-3/2}/\sqrt{\pi}$. One can also deal with the probability of a leaf-child at the root either using the formula $2C_{n-1}/C_n$ from §3 and simplification of binomials, or by applying the general method to the generating function $2zB(z) - z$. The singularity is the same as that of $B(z)$; the local behaviour at first order is $2\rho = 1/2$ times that of $B(z)$, so that $1/2$ is the limiting probability. A slighly more detailed computation by this method yields $1/2 + 3/(4n) + O(1/n^2)$.

These asymptotic techniques help analysis of parameters of combinatorial structures. One then introduces *multivariate* generating functions of the form $F(z,u) = \sum_{n,k} f_{n,k} u^k z^n$ (and the labelled counterpart), where $f_{n,k}$ denotes the number of objects of size n for which the parameter of interest takes the value k. As a concrete example, one can think of the internal path-length in a binary tree, that is, the sum of the distances from the nodes to the root. Equations for the generating series can often be derived by an extended dictionary [7]. In the case of path-length in binary trees, the equation reads $B(z,u) = z + B^2(zu,u)$. From such a bivariate generating function, the expected value of the parameter on objects of size n is obtained by dividing the coefficient of z^n in $\partial F/\partial u|_{u=1}$ by that of z^n in F. Both are univariate generating functions to which the previous method applies. In our example, one gets an average distance of the nodes to the roots growing like $\sqrt{\pi n}$.

The next step of Analytic Combinatorics is the study of limiting distributions of parameters, e.g., path-length in binary trees is asymptotically Gaussian after proper normalization. This goes beyond what can be covered in this tutorial and we refer the reader to the last part of the book by Flajolet & Sedgewick.

7 Conclusion

The message of this tutorial is that from a combinatorial specification, Analytic Combinatorics provides easy-to-use tools that provide counting, random generation and asymptotic analysis. Work is under way to automate this approach fully within computer algebra. Counting and the required parts of random generation are complete and asymptotic analysis is only partly done, but progress is being made. The ultimate goal would be a system taking as input a combinatorial specification and some sort of description of an algorithm and producing automatically the asymptotic average-case behaviour of the algorithm. The approach was tested a long time ago and works well for various grammars and parameters [5], but much remains to be done.

References

1 F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial species and tree-like structures*. Number 67 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1998. Translated from the 1994 French original by Margaret Readdy, With a foreword by Gian-Carlo Rota.

2 H. Décoste, G. Labelle, and P. Leroux. Une approche combinatoire pour l'itération de Newton-Raphson. *Advances in Applied Mathematics*, 3(4):407–416, 1982.

3 Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4–5):577–625, 2004. Special issue on Analysis of Algorithms.

4 Philippe Flajolet, Éric Fusy, and Carine Pivoteau. Boltzmann sampling of unlabelled structures. In David Applegate, Gerth Stølting Brodal, Daniel Panario, and Robert Sedgewick, editors, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics*, volume 126 of *SIAM Proceedings in Applied Mathematics*, pages 201–211. SIAM, 2007. Workshops held in New Orleans, LA, January 6, 2007.

5 Philippe Flajolet, Bruno Salvy, and Paul Zimmermann. Automatic average-case analysis of algorithm. *Theor. Comput. Sci.*, 79(1):37–109, 1991. doi:10.1016/0304-3975(91)90145-R.

6 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. 824 pages (ISBN-13: 9780521898065); also available electronically from the authors' home pages.

7 Marni Mishna. Attribute grammars and automatic complexity analysis. *Advances in Applied Mathematics*, 30(1):189–207, 2003.

8 Carine Pivoteau, Bruno Salvy, and Michèle Soria. Algorithms for combinatorial structures: Well-founded systems and newton iterations. *J. Comb. Theory, Ser. A*, 119(8):1711–1773, 2012. doi:10.1016/j.jcta.2012.05.007.

9 Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge University Press, New York, 2nd edition, 2003. URL: http://www.cambridge.org/fr/knowledge/isbn/item1170826.