INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Algorithms Seminar, 1999–2000

Frédéric CHYZAK, éditeur scientifique

N ° ????
Novembre 2000

THÈME 2

Rapport de recherche

ISSN 0249-6399

Algorithms Seminar, 1999–2000

Frédéric CHYZAK, éditeur scientifique

Thème 2 — Génie logiciel
et calcul symbolique
Projet Algo

Rapport de recherche n ° ???? — Novembre 2000 — 150 pages

Abstract: These seminar notes constitute the proceedings of a seminar devoted to the analysis of algorithms and related topics. The subjects covered include combinatorics, symbolic computation, asymptotic analysis, computational biology, and average-case analysis of algorithms and data structures.

Key-words: combinatorics, symbolic computation, analysis of algorithms, probabilistic methods, computational biology

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33) 01 39 63 55 11 – Télécopie : (33) 01 39 63 53 30

Séminaire algorithmes, 1999–2000

Résumé : Ces notes de séminaires constituent les actes, le plus souvent en anglais, d'un séminaire consacré à l'analyse d'algorithmes et aux domaines connexes. Les thèmes abordés comprennent : la combinatoire, le calcul symbolique, l'analyse asymptotique, la biologie computationnelle et l'analyse en moyenne d'algorithmes et de structures de données.

Mots-clé : combinatoire, calcul symbolique, analyse d'algorithmes, méthodes probabilistes, biologie computationnelle

ALGORITHMS SEMINAR
1999–2000

Frédéric Chyzak[1]
(Editor)

Abstract

These seminar notes constitute the proceedings of a seminar devoted to the analysis of algorithms and related topics. The subjects covered include combinatorics, symbolic computation, probabilistic methods, and average-case analysis of algorithms and data structures.

This is the ninth in our series of seminar proceedings. The previous ones have appeared as INRIA Research Reports numbers 1779, 2130, 2381, 2669, 2992, 3267, 3504, and 3830. The content of these proceedings consists of summaries of the talks, usually written by a reporter from the audience.[2]

The primary goal of the seminar is to cover the major methods for the average-case analysis of algorithms and data structures. Neighbouring topics of study are combinatorics, symbolic computation, asymptotic analysis, probabilistic methods, and computational biology.

The study of combinatorial objects—their description, their enumeration according to various parameters—arises naturally in the process of analysing algorithms that often involve classical combinatorial structures like strings, trees, graphs, and permutations.

Beside the traditional topics of combinatorics of words and algorithmics on words, over the years an increasing interest has been given in the seminar to biological applications of combinatorics.

Symbolic computation, and in particular computer algebra, plays an increasingly important role in these areas. It provides a collection of tools that allows one to attack complex models of combinatorics and the analysis of algorithms via *generating functions*; at the same time, it inspires the quest for developing ever more systematic solutions and decision procedures for the analysis of well-characterized classes of problems.

The 31 articles included in this book represent snapshots of current research in these areas. A tentative organization of their contents is given below.

PART I. COMBINATORICS

In addition to its own traditions rooted in mathematics, the study of *combinatorial models* arises naturally in the process of analysing algorithms that often involve classical combinatorial structures like permutations, strings, trees, random walks, and graphs. Maps are a special class of graphs that are drawn in the plane. This is an active field of research to which our seminar already dedicated several sessions last year. Further progress has been made recently; this is reported in [**1**], [**2**], [**3**], and [**4**]. The talks [**5**] and [**6**] are concerned with other types of graph enumerations: those of non-crossing configurations in the plane and of constrained subgraphs of rectangular grids. Models of random automata have been developed recently. A simple class of automata is introduced in [**7**], and a random generation algorithm is presented. A combinatorially meaningful question is to

[1]Partially supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

[2]The summaries for the past nine years are available on the web at the URL `http://algo.inria.fr/seminars/`.

classify combinatorial models according to the nature—rational, algebraic, D-finite, non-D-finite—of the corresponding generating functions. Two different viewpoints are given on this problem: the talk **[8]** classifies the solutions to a general class of multivariate recurrence systems, while several models of random walks are compared in **[9]**. The "n! Conjecture" in algebraic combinatorics associates some vector space of polynomials to each partition of the integer n and states that each of these spaces has dimension $n!$. A refinement of the conjecture relates to Macdonalds polynomials and has been proved only recently. A vector space that includes all the spaces above is the object of study in **[10]**.

- **[1]** Enumeration of Planar Rooted Triangulations. *J. Z. Gao.*
- **[2]** Some Sharp Concentration Results about Random Planar Triangulations. *J. Z. Gao.*
- **[3]** Planar Maps and Composition Schemes. *G. Schaeffer.*
- **[4]** Coalescence: Emergence of the Map–Airy Law. *C. Banderier.*
- **[5]** Enumeration of Geometric Configurations on a Convex Polygon. *M. Noy.*
- **[6]** Tutte Polynomials in Square Grids. *M. Noy.*
- **[7]** Random Group Automata. *C. Nicaud.*
- **[8]** Solving Discrete Initial- and Boundary-Value Problems. *M. Petkovšek.*
- **[9]** Classifying ECO-Systems and Random Walks. *C. Banderier.*
- **[10]** Combinatorics of Harmonic Polynomials. *F. Bergeron.*

PART II. COMPUTER ALGEBRA AND SYMBOLIC METHODS

For a computer algebra system, it is crucial to optimize the arithmetical operations on basic objects. In this spirit, clever algorithmic optimizations of existing algorithms are discussed in **[11]**, and novel methods of lazy evaluation are presented in **[12]**. These works are part of the few works that provide computer algebra procedures with accurate complexity analysis. Also fundamental is the LLL algorithm; for example, it has recently been used as a crucial ingredient in an efficient factoring algorithm. Summary **[13]** analyses three variants of LLL which output bases of a similar quality, but in a much faster way on average. The Galois theory for differential equations is now classical and over the years has been the topic of several talks in our seminar. Recently, a Galois theory has been developed for the case of difference equations. The results are conceptually similar but have required a non-trivial adaptation. This is the topic of **[14]**. A new algorithm to solve a certain class of linear difference equations is presented in **[15]**. As a transition to the next part, a general symbolic methodology to perform automatic average-case analysis of algorithms is presented in **[16]**.

- **[11]** Efficient Algorithms on Numbers, Polynomials, and Series. *P. Zimmermann.*
- **[12]** Relax But Don't Be Too Lazy. *J. van der Hoeven.*
- **[13]** Threshold Phenomena in Random Lattices and Reduction Algorithms. *A. Akhavi.*
- **[14]** Eigenring and Reducibility of Difference Equations. *R. Bomboy.*
- **[15]** Difference Equations with Hypergeometric Coefficients. *M. Bronstein.*
- **[16]** Attribute Grammars and Automatic Complexity Analysis. *M. Mishna.*

PART III. ANALYSIS OF ALGORITHMS AND DATA STRUCTURES

Continued fractions have made several appearances in this year's session of the seminar. Expansion into continued fraction is closely related to the Euclidean algorithm; following previous works on the arithmetical complexity of these algorithms, **[17]** considers the corresponding bit complexity. The distribution of digits in continued fractions and other number representation systems is studied in **[18]**, where sorting algorithms based on such expansions are also analysed. The next two talks also deal with continued fractions, but are of a more number-theoretic nature. As a follow-up to last year's series of talks by the same author, **[19]** provides the limiting distribution of the alternating sum of the coefficients of a continued fraction; **[20]** detects the transcendence of numbers from the digit structure of their expansions into continued fractions or in some base b. Summary **[21]** deals with the allocation of resources to connection requests in a network, a problem of graph colouring in disguise. A general basis for the analysis and synthesis of digital circuits is provided in **[22]**, together with unexpected connections between hardware design and the classical notion of automatic sequences in number theory.

[17] Average Bit-Complexity of Euclidean Algorithms. *B. Vallée.*
[18] Continued Fractions, Comparison Algorithms and Fine Structure Constants. *Ph. Flajolet.*
[19] Continued Fractions and Modular Forms. *I. Vardi.*
[20] Transcendence of Numbers whose Expansion in Base b or into Continued Fractions is "Too Regular." *J.-P. Allouche.*
[21] Routing Permutations on Trees. *S. Corteel.*
[22] Synchronous Decision Diagrams: a Data Structure for Representing Finite Sequential Digital Functions. *J. Vuillemin.*

PART IV. COMPUTATIONAL BIOLOGY AND COMBINATORICS OF WORDS

The first three talks are of a biological flavour. Summary **[23]** is concerned with determining the local statistical distribution of nucleotides along a chromosome. Searching genomic databases has motivated the work **[24]** in which the key tool is the classical description of the possible periods in strings. Trees are another combinatorial structure central to computational biology. Indeed, phylogenetic trees exhibit the evolution of a species, a gene, and so on. In this vein, **[25]** analyses several methods of construction of classification trees. More classically about combinatorics of words, **[26]** presents a new data structure used to design efficient string matching algorithms: a minimal automaton that stores the factors of a word.

[23] Bayesian Approach to DNA Segmentation into Regions with Different Average Nucleotide Composition. *V. Makeev.*
[24] Enumeration of Autocorrelations and Computation of Their Populations. *É. Rivals.*
[25] Classification by Trees: the Shape of the Inferred Tree Depends on the Algorithmic Scheme Selected. *O. Gascuel.*
[26] Factor Oracle, Suffix Oracle. *M. Raffinot.*

PART V. MISCELLANY

Two talks are concerned with the analysis of algorithms or data structures, but are of a more probabilistic flavour. Random walks on graphs are studied in **[27]**. Measures related to internal path length in various models of possibly randomized search trees and to the Quickfind algorithm are analysed in **[28]**. The information-theoretic problem of source coding is considered in great generality in **[29]**. The key question is to analyse the redundancy of a source. This relates to data compression by Huffman codes, Shannon–Fano codes, and Lempel–Ziv algorithms. A dynamical system exhibiting chaos is studied in **[30]**; the iteration process is described in terms of a language whose complexity is sought. Finally, **[31]** discusses classical models of statistical mechanics. This reflects the recent increase of interest in such problems in our seminar.

[27] On Random Graph Homomorphisms into $\mathbb{Z}$. *E. Mossel.*
[28] Distributional Analysis of Recursive Algorithms by the Contraction Method. *R. Neininger.*
[29] Analytic Information Theory and the Redundancy Rate Problem. *W. Szpankowski.*

[**30**] Queues, Stacks, and Transcendentality at the Transition to Chaos. *C. Moore.*
[**31**] Colorings, Potts Models, Height Representations, and Entropic Forces. *C. Moore.*

Acknowledgements. The lectures summarized here emanate from a seminar attended by a community of researchers in the analysis of algorithms, from the Algorithms Project at INRIA (the organizers are Philippe Flajolet and Bruno Salvy) and the greater Paris area—especially the University of Paris Sud at Orsay (Dominique Gouyou-Beauchamps). The editor expresses his gratitude to the various persons who have actively supported this joint enterprise and offered to write summaries, most notably Cyril Banderier and Philippe Flajolet for writing more than their share. Thanks are also due to the speakers and to the authors of summaries. Many of them have come from far away to attend one seminar and kindly accepted to write the summary. We are also greatly indebted to Virginie Collette for making all the organization work smoothly.

The editor,
F. Chyzak

Part I

Combinatorics

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 3–6.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Enumeration of Planar Rooted Triangulations

Jason Zhicheng Gao
School of Mathematics and Statistics, Carleton University

June 8, 2000

Summary by Gilles Schaeffer

This talk presents a joint work with I. M. Wanless and N. C. Wormald [5].

1. Introduction

A *planar map* is a connected graph embedded in the plane. In this talk, loops and multiple edges are forbidden. A map is *rooted* if one edge is oriented. The start point of this root is called the *root vertex*, the face on its right the *root face* and the face on its left the *near face*. If the root and near face of a planar map are the same, the root is a bridge. By convention the root is always taken so that the root face is the infinite face. The other faces are then called *interior faces*. The *degree* of a face is its number of incidences of edges (i.e., bridges count for two). A *triangulation* is a map whose faces all have degree three. A map is *p-connected* if at least p vertices must be removed to separate it into two connected components.

Euler already enumerated triangulations of polygons in the 19th century (they are Catalan), but the enumeration of triangulations as defined here started with Tutte's work in the sixties. Several families of planar rooted triangulations were in fact enumerated:

- 4-connected triangulations (see [8]: algebraic generating function and asymptotic are given),
- 3-connected triangulations (see [2, 8]: with n vertices they are $(4n+1)!/(n+1)!\,(3n+2)!$),
- 2-connected allowing multiple edges (see [7]: with n vertices they are $3\cdot 2^n(3n)!/n!\,(2n+2)!$),
- all triangulations allowing loops and multiple edges (see [6]: algebraic generating function and asymptotic are given).

All these family of planar rooted triangulations have algebraic generating functions and asymptotic behaviors of the same form,

$$c_i n^{-5/2}(1/\rho_i)^n,$$

where n denotes the number of vertices. For connectivity i from 1 to 4, the values of ρ_i are

$$\sqrt{3}/36, \quad 2/27, \quad 27/256, \quad 4/27,$$

respectively. For no planar map is 6-connected, the only missing connectivity for triangulations was 5, which is the subject of the present study: it turns out that for 5-connected triangulations, the generating function is algebraic of degree 6, and the asymptotic behavior is similar, with ρ_5 given as a root of a certain polynomial P of degree 6 such that

$$\rho_5 \approx 0.2477.$$

It is amusing to remark that ρ_5 is not the smallest positive root the polynomial P.

The proof is based on skillful refinements of the three original ingredients of Tutte's method: root edge deletion, the quadratic method, and composition schemes.

2. Root Edge Deletion

The deletion of the root edge is maybe the simplest possible idea to decompose a map. It turns out to be very efficient in providing functional equation for generating functions of "not-too-connected" maps.

In general there are two cases in the root edge deletion process applied to a planar map M of a family $\mathcal{F}$:

- either the root edge deletion separates M into two pieces that more or less belong to $\mathcal{F}$,
- or it yields directly a map M' that belongs more or less to the family $\mathcal{F}$. In this case, M' usually has a larger root face degree than M: the removal of the root has merged the root and near faces of M.

This decomposition can be made one-to-one, at the expense of taking the root face degree into account. It then results into functional equations for the generating function

$$F(x,y) = \sum_{n,k} f_{n,k} x^n y^k,$$

where $f_{n,k}$ denote the number of maps with n inner vertices and a root face of degree k.

For instance let $F(x,y)$ be the generating function of *near-triangulations*, i.e., maps with all faces of degree three, except maybe the root face. Then the root edge deletion yields

$$F(x,y) = y^2 + y^{-1}F(x,y)^2 + xy^{-1}\left(F(x,u) - y^2 - yF_3(x)F(x,y)\right),$$

where $F_3(x)$ is the generating function of triangulations, i.e., $F(x,y) = y^2 + F_3(x)y^3 + O(y^4)$. Indeed in the right hand side, the three summands correspond to three cases in the decomposition of a near-triangulation M:

- M is the degenerate triangulation with one edge and two vertices,
- M is made of a couple of triangulations separated by a rooted triangle,
- or removing the root of M directly yields a triangulation M'. In this case, M' must not be the degenerate triangulation, nor have a short diagonal cutting it into a triangulation and a near-triangulation (otherwise, replacing the root of M would create a double edge).

In order to enumerate 5-connected triangulations, it turns out to be necessary to enumerate *M-type maps*, i.e., maps whose interior faces have degree three or four. Their generating function

$$M(x,y,z) = \sum_{n,l,k} m_{n,l,k} x^n y^l z^k,$$

where $m_{n,l,k}$ denotes the number of rooted M-type maps with n triangular interior faces, l interior quadrangular faces and a root face of degree k, satisfies

$$M(x,y,z) = 1 + z^2 + M_3(x,y)z^3 + M_4(x,y)z^4 + O(z^5)$$

where M_3 and M_4 denote the generating functions of M-type maps with root face of degree three and four respectively.

The root edge deletion applied to M-type maps yields, with $M' = M - 1$,

$$M' = z^2M^2 + xz^{-1}(M' - z^2M - zM_3M') + yz^{-2}(M' - z^2 - z^3M_3M - z^2M_4M').$$

3. The Quadratic Method

The equations provided by root edge deletion have always the same flavor: they involve a principal generating function ($F(y)$ for near-triangulations) in which the equation is quadratic, and a secondary generating function not depending on y (F_3 for near-triangulations).

The quadratic method, as used by Tutte, proceeds as follows

- Write the equation in the form $A^2 = B$, where A and B are polynomials in all variables and generating functions *and where B does not contain the principal generating function.* E.g., for near-triangulations this gives
$$A = F + \frac{1}{2}(x - xyF_3 - y), \quad \text{and} \quad B = \frac{1}{4}(x - xyF_3 - y)^2 + xy^2 - y^3.$$
In general this is possible because the equation is quadratic in F.
- Show that there exists a power series $Y(x)$ such that
$$A\Big(x, Y(x), F_3(x), F\big(x, Y(x)\big)\Big) = 0.$$
- Then
$$B\big(x, Y(x), F_3(x)\big) = \frac{\partial B}{\partial y}\big(x, Y(x), F_3(x)\big) = 0$$
and, provided this system is not degenerate, this proves that F_3 and Y are algebraic.

In the case of M-type maps, the situation is somewhat more involved, because of the presence of two secondary generating functions. However using a theorem of Brown on power series that are square roots [3], Bender and Canfield have dealt with a similar situation in [1]. Upon finding appropriate parametrizations to make the computation tractable with `Maple`, this approach yields
$$M_3 = u^3 - 2uv + u, \quad \text{and} \quad M_4 = 3u^4 - 5u^2v + u^2 - v^2 + v + 2,$$
where $u = u(x, y)$ and $v = v(x, y)$ are the power series uniquely determined by
$$x = \frac{3u^3 - 2uv + u}{(1+v)^3}, \quad \text{and} \quad y = \frac{v - u^2}{(1+v)^3}.$$

4. Composition Schemes and Non-Uniqueness

Root edge deletion does not work well on 4-connected triangulations or triangulations with higher connectivity, because the deletion of the root can produce maps with smaller connectivity that are hard to decompose back into maps with high connectivity. To enumerate 4-connected triangulations, Tutte introduced compositions schemes.

First remark that a triangulation is 3-connected as soon as it contains no loop and multiple edges, 4-connected if all its cycles of length three bound faces, and 5-connected if moreover it contains no 4-cycles with a vertex inside.

In the last two sections we were able to determine the generating function $F_3(x)$ of (3-connected) triangulations. Now take a 3-connected triangulation M. Its cycles of length three are ordered by inclusion. In particular they are all inside the outer cycle of the root face; call a cycle of length three *maximal* if it is not inside any other one. A maximal cycle either bounds a face or contains at least one vertex in its inside. In the latter case, the maximal cycle and its inside form a triangulation.

Removing the triangulation inside each maximal cycle yields the decomposition of M into a 4-connected triangulation M' plus one triangulation per face of M' (possibly reduced to a triangle). In terms of generating functions, this yields
$$F_3(x) - 1 = \sum_{k \geq 1} G_k x^k F_3(x)^{2k+1}$$
where G_k is the number of 4-connected triangulations with k vertices (and $2k+1$ inner faces). This yields a fonctional equation of the composition type
$$F_3(x) = 1 + F_3(x) G\big(xF_3(x)^2\big),$$

which properly determines the generating function $G(a)$ in terms of $f(x) = F_3(x)^2$. Indeed, consider the equation $a = xf(x)$. As $f(x) = 1 + O(x)$ this equation properly defines a power series $x(a)$ and from the composition equation,

$$\big(1 - G(a)\big)^2 f\big(x(a)\big) = \big(1 - G(a)\big)^2 a/x(a) = 1.$$

Now as $F_3(x)$ is algebraic, so is $f(x)$ and there is a polynomial $P(x, f)$ such that $P(x, f(x)) = 0$. Take $x = x(a)$ so that

$$P\Big(x(a), f\big(x(a)\big)\Big) = P\big(x(a), a\big) = 0,$$

and we conclude that $x(a)$, and thus $G(a)$, are algebraic.

The next step is to go from 4-connected triangulation to 5-connected ones. The idea is again to start with a triangulation and remove the inside of any non empty cycle of length three or four. However in general this yields an M-type map and not a triangulation.

The composition scheme has thus to be defined between 4- and 5-connected M-type maps. The same technique immediately applies to remove cycles of length three in M-type maps, but for cycle of length four, a new difficulty appear: two four cycles can overlap, making the definition of maximal four-cycles no so easy.

Finally, it turns out that a careful case study allows to classify overlapping four-cycles and work out the desired composition schemes.

5. **Conclusion**

Using the latter composition scheme and the results for $M_3(x, y)$ and $M_4(x, y)$, algebraic equations for the generating function $T(x)$ of 5-connected triangulations can finally be derived. These equations take the form of a parametrization $T(x) = \Phi(x, s)$ where s has a relatively compact algebraic equation (unlike $T(x)$).

The asymptotic is then obtained from a careful analysis of the possible sources of singularity in the parametrization. This indirect approach seems more easily tractable than dealing with the explicit polynomial equation giving $T(x)$.

This concludes the story for planar triangulations. As far as exact expression for generating functions are concerned, for general planar maps, there is no more than Tutte's result giving 3-connected ones. On higher genus surfaces, 2-connectivity was the limit until the very recent result of [4] for 3-connected triangulations of the projective plane.

Bibliography

[1] Bender (Edward A.) and Canfield (E. Rodney). – The number of degree-restricted rooted maps on the sphere. *SIAM Journal on Discrete Mathematics*, vol. 7, n° 1, 1994, pp. 9–15.

[2] Brown (William G.). – Enumeration of triangulations of the disk. *Proceedings of the London Mathematical Society. Third Series*, vol. 14, 1964, pp. 746–768.

[3] Brown (William G.). – On kth roots in power series rings. *Mathematische Annalen*, vol. 170, 1967, pp. 327–333.

[4] Gao (Z. C.) and Wang (J. Y.). – Exact enumeration of rooted 3-connected triangular maps on the projective plane. – Preprint.

[5] Gao (Z. C.), Wanless (I. M.), and Wormald (N. C.). – Counting 5-connected planar triangulations. – Preprint.

[6] Gao (Zhi-Cheng). – The number of rooted triangular maps on a surface. *Journal of Combinatorial Theory. Series B*, vol. 52, n° 2, 1991, pp. 236–249.

[7] Mullin (R. C.). – On counting rooted triangular maps. *Canadian Journal of Mathematics*, vol. 17, 1965, pp. 373–382.

[8] Tutte (W. T.). – A census of planar triangulations. *Canadian Journal of Mathematics*, vol. 14, 1962, pp. 21–38.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 7–10.

Available online at the URL
http://algo.inria.fr/seminars/.

Some Sharp Concentration Results about Random Planar Triangulations

Jason Zhicheng Gao
School of Mathematics and Statistics, Carleton University

June 8, 2000

Summary by Cyril Banderier

Abstract

The theory of random maps has a relatively short history when compared to the theory of random graphs. In this talk, we mention some recent results concerning sharp concentration properties of parameters in random planar triangulations. Examples include the maximum vertex degree, the largest component, the number of copies of a given submap, and the number of flippable edges. This is joint work by Jason Zhicheng Gao and Nicholas Wormald (Melbourne, Australia).

1. Introduction

Draw a graph on a sphere and then mark (or "root") a face, an edge of this face and a vertex of this edge. Then project the graph on the plane (e.g., by a stereographic projection): you get a *planar map*. Without loss of generality, you can rotate the sphere so that the marked face contains the north pole; then the projection transforms the marked face into an unbounded face, which is called the *external face*.

For background, we refer to the summary of Gao's other talk (pages 3–6 in these proceedings) for several definitions and examples on maps and triangulations.

For ten years, Jason Zhicheng Gao (often collaborating with other specialists of maps, namely Bender, Canfield, Richmond, McKay, Wormald, ...) has studied several parameters of maps (strong connexity, pattern occurrences, vertex degree, symmetries, cycles, Eulerian properties), finding new functional equations, solving them, and also obtaining precise asymptotic estimations. We specialize here the discussion to two parameters that appear to be intimately related: vertex degree and submap occurrence. Concentration results are obtained by the second moment method.

2. Submap Density Result

Many combinatorial structures satisfy *Borges's Theorem*,[1] meaning that any pattern will appear with high probability in a large enough structure. Any word of length $\ln n$ appears with high probability in a random word of length n (for more details, see the study by Guibas and Odlyzko [6], and then by Nicodème et al. [7, 8]). The occurrence of patterns in random graphs has also been studied [2]. However for maps, the situation is different as these objects live in quite a different probability space. Let's make a bet: choose a map of size 6, while I generate a random map of size 1000; would you bet that your map is a submap of mine? This talk makes explicit the conditions under which you can make good (or bad) bets.

[1]Philippe Flajolet coined this naming in reference to Borges's novel *The Library of Babel*.

Let $\mathcal{T}$ be any fixed triangulation and $\eta_n(\mathcal{T})$ be the random variable counting the *number of copies* of $\mathcal{T}$ in a random triangulation with n vertices. Richmond and Wormald [9] showed that

$$\mathbf{P}\big(\eta_n(\mathcal{T}) > cn\big) > 1 - \exp^{-\delta n}$$

for some positive constants c and δ depending on $\mathcal{T}$. Bender, Gao and Richmond [1] showed that the above result holds for many families of maps, and recently Gao and Wormald [4] proved that $\eta_n(\mathcal{T})$ is sharply concentrated around cn for some constant c. More precisely:

Theorem 1. *Let $\mathcal{T}$ be a 3-connected triangulation with $j+3$ vertices such that there are r distinct ways to root $\mathcal{T}$. Let $c = 2r(27/256)^j$. Then, provided that $cn \to \infty$,* $\mathbf{P}\big(\big|\eta_n(\mathcal{T}) - cn\big| = o(cn)\big) \to 1$.

A *near-triangulation* is composed of triangulations except that it can have more than 3 vertices on its external face. Define

$$\mu_k = \frac{8(k-2)}{4k^2-1}\left(-\frac{3}{4}\right)^k \binom{-3/2}{k}.$$

Theorem 2. *Let $\mathcal{M}$ be a 3-connected near-triangulation with external face of degree k and with j internal vertices such that there are r distinct ways to root the external face. Then, for fixed j and k with $k \geq 4$, one has*

$$\mathbf{P}\left(\big|\eta_n(\mathcal{M}) - r\mu_k(27/256)^{j-1}n\big| = o(n)\right) \to 1.$$

Proof. The method used here relies on Chebyshev-like inequalities: $\mathbf{P}(X > t\mu) \leq 1/t$ and $\mathbf{P}\big(|X - \mu| \geq t\sigma\big) \leq 1/t^2$ for a nonnegative random variable X with average μ and variance σ^2. A consequence is that if $\sigma = o(\mu)$, then one gets a concentration result.

Let us first study the number $\zeta_n(k)$ of vertices of degree k in a random triangulation with $n+2$ vertices. The quantity T_n denotes the number of rooted triangulation with $n+2$ vertices; $T_{n,k}$ denotes the number of rooted triangulation with $n+2$ vertices and root vertex of degree k; $T_{n,k,l}$ denotes the number of rooted triangulation with $n+2$ vertices, root vertex of degree k and another distinguished vertex of degree l. The scheme of the proof is as follows:

Step 1: Use combinatorial arguments to show the relations

$$\mathbf{E}\big[\zeta_n(k)\big] = \frac{6n}{k}\frac{T_{n,k}}{T_n} \qquad \text{and} \qquad \mathbf{E}\big[\zeta_n(k)(\zeta_n(k)-1)\big] = \frac{6n}{k}\frac{T_{n,k,k}}{T_n}.$$

Step 2: Obtain functional equations for the generating functions for $T_{n,k,l}$, $T_{n,k}$, and T_n, and perform singularity analysis.

Step 3: Derive a suitable multivariate version of Flajolet and Odlyzko's transfer theorem (see Lemmas 2 and 3 below), and obtain the following asymptotics, uniformly for $k = O(\ln n)$:

$$T_n = \sqrt{6}/(32\sqrt{\pi})n^{-5/2}(256/27)^n\big(1 + O(1/n)\big),$$

$$T_{n,k} = \frac{k\sqrt{6}}{192\sqrt{\pi}}\mu_k n^{-5/2}(256/27)^n\big(1 + O(k^{20}/n)\big),$$

$$T_{n,k,k} = \frac{k\sqrt{6}}{192\sqrt{\pi}}\mu_k^2 n^{-3/2}(256/27)^n\big(1 + O(k^{20}/n)\big).$$

Step 4: Derive asymptotics for the first two moments of $\zeta_n(k)$;

$$\mathbf{E}\big[\zeta_n(k)\big] = n\mu_k\big(1 + O(k^{20}/n)\big) \qquad \text{and} \qquad \mathbf{Var}\big[\zeta_n(k)\big] = n\mu_k + (n\mu_k)^2 O(k^{20}/n),$$

uniformly for $k = O(\ln n)$. It follows from Chebyshev's inequality that

$$\mathbf{P}\big(\big|\zeta_n(k) - \mu_k n\big| = o(\mu_k n)\big) \to 1$$

uniformly for $k < \big(\ln n - (\ln\ln n)/2\big)/\ln(4/3) - \Omega(n)$. $\square$

The proof is based on three lemmas.

Lemma 1. *Let $\mathcal{T}$ be a 3-connected near-triangulation with $j+3$ vertices such that j is $o(n)$ and that there are r distinct ways to root $\mathcal{T}$. Let $\eta_n(\mathcal{T})$ be the number of copies of $\mathcal{T}$ in a random rooted triangulation with $n+2$ vertices. Then*

$$\mathbf{E}\big[\eta_n(\mathcal{T})\big] = r\left(\frac{27}{256}\right)^{j-1}\mathbf{E}\big[\zeta_{n+1-j}(3)\big]\big(1+o(1)\big),$$

$$\mathbf{E}\big[\eta_n(\mathcal{T})(\eta_n(\mathcal{T})-1)\big] = r^2\left(\frac{27}{256}\right)^{2j-2}\mathbf{E}\big[\zeta_{n+2-2j}(3)(\zeta_{n+2-2j}(3)-1)\big]\big(1+o(1)\big).$$

In order to state precise results, one needs the following notation: let ϵ be a small positive constant, ϕ be a constant satisfying $0<\phi<\pi/2$, and $\bar{y}$ be $(y_1, y_2, \ldots, y_d)$. Define:

$$\Delta(\epsilon,\phi) = \big\{\, x \text{ such that } |x|\le 1+\epsilon,\ x\neq 1,\ |\mathrm{Arg}(x-1)|\ge\phi \,\big\},$$

$$\mathcal{R}(\epsilon,\phi) = \big\{\, (x,\bar{y}) \text{ such that } |y_j|<1,\ 1\le j\le d,\ x\in\Delta(\epsilon,\phi) \,\big\}.$$

Let $\beta_j>0$ for $1\le j\le d$, and α be any real number. The following two notations are also useful.

Definition 1 ($\widetilde{O}$ notation)**.** We write $f(x,\bar{y}) = \widetilde{O}\big((1-x)^{-\alpha}\prod_{j=1}^d(1-y_j)^{-\beta_j}\big)$ if there exist $\epsilon>0$ and $0<\phi<\pi/2$ such that, in $\mathcal{R}(\epsilon,\phi)$, $f(x,\bar{y})$ is analytic and $f(x,\bar{y}) = O\big(|1-x|^{-\alpha}\prod_{j=1}^d(1-|y_j|)^{-\beta_j}\big)$ as $(1-x)(1-y_j)^{-p}\to 0$, for $1\le j\le d$, and some $p\ge 0$; for some $q\ge 0$ and some real number α', one has $f(x,\bar{y}) = O\big(|1-x|^{-\alpha'}\prod_{j=1}^d(1-|y_j|)^{-q}\big)$.

Definition 2 ($\approx$ notation)**.** We write $f(x,\bar{y}) \approx c(1-x)^{-\alpha}\prod_{j=1}^d(1-y_j)^{-\beta_j}$ if $f(x,\bar{y})$ can be expressed as $f(x,\bar{y}) = c(\bar{y})(1-x)^{-\alpha}\prod_{j=1}^d(1-y_j)^{-\beta_j} + \sum_{j=0}^d C_j(x,\bar{y}) + E(x,\bar{y})$ where $C_0(x,\bar{y})$ is a polynomial in x, and for $1\le j\le d$, $C_j(x,\bar{y})$ is a polynomial in y_j and where $E(x,\bar{y}) = \widetilde{O}\big(|1-x|^{-\alpha'}\prod_{j=1}^d(1-y_j)^{-\beta'_j}\big)$ for some $\alpha'<\alpha$ and $\beta'_j\ge 0$, $1\le j\le n$ and finally where $c(\bar{y}) = c + O(\sum_{j=1}^d|1-y_j|)$ and is analytic when $\bar{y}\in\Delta(\epsilon,\phi)^d$, with $c(\bar{1}) = c\neq 0$.

Lemma 2. *Suppose that*

$$f(x,\bar{y}) = \widetilde{O}\left((1-x)^{-\alpha}\prod_{j=1}^d(1-y_j)^{-\beta_j}\right);$$

then as $n\to\infty$ and $1\ge k_j = O(\ln n)$,

$$[x^n\bar{y}^k]f(x,\bar{y}) = O\left(n^{\alpha-1}\prod_{j=1}^d k_j^{\beta_j}\right);$$

and for any $0<\epsilon'<1$ and for all n and k_j,

$$[x^n\bar{y}^k]f(x,\bar{y}) = O\left(n^{\alpha-1}\prod_{j=1}^d(1-\epsilon')^{\beta_j}\right).$$

Lemma 3. *Let $d\ge 1$ and $f(x,\bar{y})\approx c(1-x)^{-\alpha}\prod_{j=1}^d(1-y_j)^{-\beta_j}$, where α is neither a negative integer nor 0, and $c\neq 0$. Then as $n\to\infty$ and $k_j = O(\ln n)$,*

$$[x^n\bar{y}^k]f(x,\bar{y}) = \frac{c}{\Gamma(\alpha)}\prod_{j=1}^d\left(k_j^{\beta_j-1}/\Gamma(\beta_j)\right)\left(1+O\left(\sum_{j=1}^d 1/k_j\right)\right).$$

3. Maximum Vertex Degree

Let d_n be the maximum vertex degree of a random map in a family of maps of size n. Devroye, Flajolet, Hurtado, Noy, and Steiger [3] showed that, for triangulations of an n-gon,

$$\mathbf{P}\Big(\big|d_n - \ln(n)/\ln 2\big| \le (1+\epsilon)\ln\ln n/\ln 2\Big) \to 1.$$

Gao and Wormald [5] improved this last result and extended it to general families of maps. They showed that, for any function Ω going to infinity arbitrarily slowly, one has

- for triangulations of an n-gon: $\mathbf{P}\left(\left|d_n - \frac{\ln n + \ln\ln n}{\ln 2}\right| \le \Omega(n)\right) \to 1$,
- for 3-connected triangulations of n vertices: $\mathbf{P}\left(\left|d_n - \frac{\ln n + (\ln\ln n)/2}{\ln(4/3)}\right| \le \Omega(n)\right) \to 1$,
- for all maps of n edges: $\mathbf{P}\left(\left|d_n - \frac{\ln n + (\ln\ln n)/2}{\ln(6/5)}\right| \le \Omega(n)\right) \to 1$.

4. A Few Open Problems

At the end of the talk, a few questions were raised, and the following conjectures appear plausible but might involve hard work.

Conjecture 1. *The generating function of maps without a given submap is algebraic.*

There is no doubt that a functional equation could be obtained for each pattern (however, as the overlaps can be very intricate, the functional equation would be horrendous, and it is not clear that a generalization of the quadratic method would allow us to solve it, and prove algebraicity).

Conjecture 2. *A Gaussian limit law should hold.*

Once more, we expect the behavior to be qualitatively the same for words, trees and maps. Another interesting study (which is as of now out of reach) is the occurrence of a given pattern, not in a local sense but in a global one (such patterns are called "minors").

Other concentration results about random planar triangulations such as the largest component and the number of flippable edges were finally not presented in the talk but can be found in Gao's articles at his homepage `http://mathstat.math.carleton.ca/~zgao/`.

Bibliography

[1] Bender (Edward A.), Gao (Zhi-Cheng), and Richmond (L. Bruce). – Submaps of maps. I. General 0–1 laws. *Journal of Combinatorial Theory. Series B*, vol. 55, n° 1, 1992, pp. 104–117.

[2] Bollobás (Béla). – *Random graphs.* – Academic Press Inc., London, 1985, xvi+447p.

[3] Devroye (L.), Flajolet (P.), Hurtado (F.), Noy (M.), and Steiger (W.). – Properties of random triangulations and trees. *Discrete & Computational Geometry*, vol. 22, n° 1, 1999, pp. 105–117.

[4] Gao (Zhicheng) and Wormald (Nicholas C.). – Sharp concentration of the number of submaps in random planar triangulations. – Preprint.

[5] Gao (Zhicheng) and Wormald (Nicholas C.). – The distribution of the maximum vertex degree in random planar maps. *Journal of Combinatorial Theory. Series A*, vol. 89, n° 2, 2000, pp. 201–230.

[6] Guibas (L. J.) and Odlyzko (A. M.). – String overlaps, pattern matching, and nontransitive games. *Journal of Combinatorial Theory. Series A*, vol. 30, n° 2, 1981, pp. 183–208.

[7] Nicodème (Pierre), Salvy (Bruno), and Flajolet (Philippe). – Motif statistics. *Theoretical Computer Science.* – To appear.

[8] Nicodème (Pierre), Salvy (Bruno), and Flajolet (Philippe). – Motif statistics. In Nešetřil (Jaroslav) (editor), *Algorithms, ESA'99. Lecture Notes in Computer Science*, vol. 1643, pp. 194–211. – Springer, Berlin, 1999. Proceedings of the 7th Annual European Symposium, Prague, Czech Republic, July 1999.

[9] Richmond (L. Bruce) and Wormald (Nicholas C.). – Random triangulations of the plane. *European Journal of Combinatorics*, vol. 9, n° 1, 1988, pp. 61–71.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), p. 11.

Available online at the URL
http://algo.inria.fr/seminars/.

Planar Maps and Composition Schemes

Gilles Schaeffer
Polka Project, INRIA Lorraine

March 20, 2000

Abstract

This talk is concerned with presenting the enumerative theory of planar maps, following Tutte's original approach. Combinatorial proofs of many beautiful formulæ discovered by Tutte have been given recently (cf. last year's talk). However, in "less beautiful" cases, one is invariably back to decompositions and generating series. In other words, the best tools still are those introduced by Tutte and Brown in the 1960's. The decompositions by "deletion/contraction" of edges translate into quadratic bivariate discrete differential equations, that transform into algebraic equations by the (rather miraculous) "quadratic method." Decompositions by "composition" of maps translate into composition schemes. From the viewpoint of singularity analysis, these schemes are all of the same type: the critical composition of two singularities in $(\rho - x)^{3/2}$. The goal of this presentation is especially to show where the composition schemes and the random generation algorithm reported on in Cyril Banderier's companion talk stem from.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 13–16.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Coalescence: Emergence of the Map–Airy Law

Cyril Banderier
Algorithms Project, INRIA Rocquencourt

March 20, 2000

Summary by Michel Nguyễn-Thế

Abstract

Maps are planar graphs presented together with an embedding in the plane, and as such, they model the topology of many geometric arrangements. This talk is concerned with the statistical properties of random maps, and focuses on connectivity issues. The analysis that we introduce is largely based on a method of "coalescing saddle points." We exhibit here a new class of "universal" phenomena that are of the exponential-cubic type $\exp ix^3$, corresponding to nonstandard distributions that involve the Airy function. Consequences include the analysis and fine optimization of random generation algorithms for multiply connected planar graphs.
(Joint work of C. Banderier with P. Flajolet, G. Schaeffer and M. Soria.)

1. Statistical Properties of Random Maps

Generically, $\mathcal{M}$ and $\mathcal{C}$ will be two classes of maps, respectively the "basic maps" and the "core-maps," with $\mathcal{M}_n$ and $\mathcal{C}_n$ the subsets of elements of size n. Here, the class $\mathcal{C}$ is always a subset of $\mathcal{M}$ satisfying additional properties, such as higher connectivity.

1.1. **Combinatorics of maps.** Let M_n and C_k be the cardinalities of $\mathcal{M}_n$ and $\mathcal{C}_k$. The *generating functions* of $\mathcal{M}$ and $\mathcal{C}$ are respectively defined by $M(z) = \sum_{n\geqslant 1} M_n z^n$ and $C(z) = \sum_{k\geqslant 1} C_k z^k$.

(i) Root-face decomposition. From the quadratic method [6, Sec. 2.9] and from root-face decomposition [9], one can find two power series ψ and ϕ, such that $M(z) = \psi\big(L(z)\big)$, where L is implicitly determined by $L(z) = z\phi\big(L(z)\big)$. For nonseparable maps, one has $\phi(y) = (1+y)^3$ and $\psi(y) = y(1-y)$. Lagrange inversion theorem [6] hence yields:

$$M_n = [z^n]M(z) = \frac{1}{n}[y^n]\psi'(y)\phi(y)^n, \text{ that is, for nonseparable maps: } M_n = \frac{4(3n)!}{n!\,(2n+2)!}.$$

(ii) Substitution decomposition. Noticing that the generating functions $z + \frac{2M(z)^2}{1+M(z)}$ and $C\big(M(z)\big)$ enumerate respectively the maps without core (i.e., no submap that is element of $\mathcal{C}_n$) and the maps formed of a nondegenerate core in which maps are substituted, we deduce that $M(z)$ satisfies [9]:

$$M(z) = \left(z + \frac{2M(z)^2}{1+M(z)}\right) + C\big(M(z)\big).$$

Define the *bivariate generating function* $M(z,u) = \sum_{n,k} M_{n,k} u^k z^n$, with $M_{n,k} = \operatorname{Card}\mathcal{M}_{n,k}$, where $\mathcal{M}_{n,k}$ is the set of maps of size n having a core of $k+1$ edges. Tutte proved the refinement $M(z,u) = C\big(uM(z)\big)$.

1.2. **Connectivity issues.** We are interested in the probability $\mathbf{P}[X_n = k]$ that a map of $\mathcal{M}_n$ has a core with $k+1$ edges. This probability is given by

$$\mathbf{P}[X_n = k] = \frac{C_k\,[z^n]M(z)^k}{M_n}, \quad \text{with} \quad [z^n]M(z)^k = \frac{k}{n}\,[y^{n-1}]y\psi'(y)\psi(y)^{k-1}\phi(y)^n,$$

where the second equality results from Lagrange inversion.

1.3. **The asymptotics of maps.** Thanks to transfer methods [4], one can derive asymptotics for M_n and C_k [1]. One obtains positive numbers b and b', as well as ρ and τ, such that:

$$M_n \underset{n\to\infty}{\sim} \frac{3b}{4\sqrt{\pi}}\frac{\rho^{-n}}{n^{5/2}}, \qquad \text{in particular} \qquad M_n^{(\text{non sep.})} \underset{n\to\infty}{\sim} \frac{\sqrt{3}}{2\sqrt{\pi}}\left(\frac{27}{4}\right)^n n^{-5/2};$$

$$C_k \underset{n\to\infty}{\sim} \frac{3b'}{4\sqrt{\pi}}\,\psi(\tau)^{-k}k^{-5/2}, \qquad \text{in particular} \qquad C_k^{(3\text{-conn.})} \underset{n\to\infty}{\sim} \frac{8}{243\sqrt{\pi}}4^k k^{-5/2}.$$

Hence, studying $\mathbf{P}[X_n = k]$ essentially consists in estimating $[z^n]M(z)^k$.

2. Two Saddle Points

Let us start the estimation of $[z^n]M(z)^k$ by Cauchy's formula,

$$[z^n]M^k(z) = \frac{k}{n}\frac{1}{2i\pi}\int_\Gamma z\left(\psi(z)^k\right)'\phi(z)^n\frac{dz}{z^{n+1}} = \frac{k}{n}\frac{1}{2i\pi}\int_\Gamma G(z)\,\psi(z)^k\,(\phi(z)/z)^n\,dz$$

where Γ is a contour encircling the origin and $G(z) = \psi'(z)/\psi(z) = (1-2z)/(z(1-z))$.

We make use of the saddle-point method. The idea consists in deforming the contour Γ in the complex plane in order to have it cross a saddle point of the integrand f (i.e., a zero of the derivative) and to take advantage of concentration of the integral near the saddle point.

The problem at hand furnishes with two saddle points, $z_+ = 1/2$ and $z_- = (n-k)/(n+k)$, solutions of the equation $\frac{\partial}{\partial z}\big(k\ln\psi + n\ln(\phi/z)\big) = 0$. We distinguish four cases.

2.1. **Distinct saddles.** When $k < n/3$, the saddle point $z_+ = 1/2$ is dominant, and when $k > n/3$, $z_- = (n-k)/(n+k)$ dominates. If k is far enough from $n/3$, the basic saddle-point method applies and we use for contour a circle Γ_0 centered around the origin and passing through the dominant saddle point τ. Local expansions are of the "exponential quadratic" type and, the contour being orthogonal to the real axis in τ, the real-variable Laplace method permits one to estimate the integral asymptotically [3]. Then we have:

Theorem 1 (Tails and distinct saddles [5]). *Let $\lambda(n)$ be an arbitrary function with $\lambda(n) \to +\infty$ and $\lambda(n) = o(n^{1/3})$. Then, the probability distribution of the core of random element of $\mathcal{M}_n$ satisfies*

$$\mathbf{P}[X_n = k] \sim \frac{32}{243\sqrt{\pi}}\cdot\frac{n^{5/2}}{k^{3/2}(n-3k)^{5/2}}, \qquad \textit{uniformly for } \lambda(n) < k < \frac{n}{3} - n^{2/3}\lambda(n);$$

$$\mathbf{P}[X_n = k] = O\left(\exp(-n(k/n-1/3)^3)\right), \qquad \textit{uniformly for } k > \frac{n}{3} + n^{2/3}\lambda(n).$$

2.2. **A double saddle.** Here we directly attack the analysis of the "center" of the distribution, that is, the case where $n = 3k$ exactly. Then, the saddle points become equal: $z_- = z_+ = \tau$. The function f can be written $f(z) = f(\tau) + f^{(3)}(\tau)(z-\tau)^3/6 + O\big((z-\tau)^4\big)$, with $f^{(3)}(\tau)$ real and negative. Hence the curves of steepest descent, corresponding to real and nonpositive $f^{(3)}(\tau)(z-\tau)^3/6$ when z is close to τ, either follow the positive real axis or form an angle of $\pm 2\pi/3$ with it. We approximate

Maps ($\mathcal{M}$), $\mathcal{M}_n$	Cores ($\mathcal{C}$), Scheme	α_0	c
general, n edges	1, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}\mathcal{M}^2]$	1/3	$3/4^{2/3}$
general, n edges	bridgeless, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}(\mathcal{X}\mathcal{M})^*]$	4/5	$(5/3)^{2/3}/4$
general, n edges	loopless, $\mathcal{M} \simeq \mathcal{L} + \mathcal{C}[\mathcal{X}((\mathcal{X}\mathcal{M})^*)^2]$	2/3	3/2
loopless , n edges	simple, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}\mathcal{M}]$	2/3	$3^{4/3}/4$
bipartite, n edges	bipartite simple, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}\mathcal{M}]$	5/9	$3^{8/3}/20$
bipartite, n edges	bipartite nonseparable, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}\mathcal{M}^2]$	5/13	$(13/6)^{5/3} \cdot 3/10$
bipartite, n edges	bipartite bridgeless, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}(\mathcal{X}\mathcal{M})^*]$	3/5	$(15/2)^{5/3}/18$
nonseparable, n edges	simple nonseparable, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}\mathcal{M}]$	4/5	$15^{5/3}/36$
nonseparable, $n+1$ edges	3-connected, $\mathcal{M} \simeq \mathcal{D} + \mathcal{C}[\mathcal{M}]$	1/3	$3^{4/3}/4$
cubic nonseparable, $n+2$ faces	cubic 3-connected, $\mathcal{M} \simeq \mathcal{C}[\mathcal{X}(1+\mathcal{M})^3]$	1/2	$(3/2)^{1/3}$
cubic 3-connected, $n+2$ faces	cubic 4-connected, $\mathcal{M} \simeq \mathcal{M} \cdot \mathcal{C}[\mathcal{X}\mathcal{M}^2]$	1/2	$6^{2/3}/3$

TABLE 1. A selection of composition schemes ($\mathcal{X}$ an edge, $\mathcal{L}$, $\mathcal{D}$ auxiliary families).

those last two curves by replacing a small arc of Γ_0 by two small segments Δ_1 and Δ_2 intersecting τ at an angle of $\pm 2\pi/3$. A few computations then deliver:

$$\mathbf{P}[X_{3k} = k] = \frac{4}{27}\frac{\Gamma(2/3)}{3^{1/6}\pi} k^{-2/3}\left(1 + O\left((\ln k)^4 k^{-1/3}\right)\right), \quad \text{with} \quad \frac{4}{27}\frac{\Gamma(2/3)}{3^{1/6}\pi} \approx 0.0531.$$

The estimation remains valid for $n = 3k+1$ and $n = 3k+2$. A similar result holds for $n = 3k + O(1)$.

2.3. **Nearby saddles.** When k is close to $n/3$, we choose a contour Γ with the same shape as previously but going through the mid-point $\zeta := (z_- + z_+)/2$, so that it *simultaneously* catches the contributions of the two saddle points z_- and z_+. Local estimates of the integrand lead to an expression involving Airy functions. With the "map–Airy" distribution $\mathcal{A}$ defined by

$$\mathcal{A}(x) = 2\exp\left(-\frac{2}{3}x^3\right)\left(x\mathrm{Ai}(x^2) - \mathrm{Ai}'(x^2)\right), \text{ where } \mathrm{Ai}(z) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{i(zt+t^3/3)}\,dt,$$

we have indeed: $\displaystyle \sup_{a \leqslant \frac{k-n/3}{n^{2/3}} \leqslant b} \left| n^{2/3}\mathbf{P}[X_n = k] - \frac{16}{81}\frac{3^{4/3}}{4}\mathcal{A}\left(\frac{3^{4/3}}{4}\frac{k-n/3}{n^{2/3}}\right)\right| \underset{n\to\infty}{\longrightarrow} 0.$

2.4. **Coalescing saddles.** This case is an improvement of the former one, in so far as we provide a uniform description of the transition regions around $n/3$, allowing k to range anywhere between $\lambda(n)$ and $n - \lambda(n)$, for any $\lambda(n) = o(n)$ with $\lambda(n) \to \infty$. We set $k = (1/3 + \beta)n$, and make β vary in any compact subinterval of $\big(-1/3, 2/3\big)$. By a change of variable, one reduces the computation to the case of (the exponential of) a cubic integrand [1, 2, 10]—the simplest case enabling a double saddle point— to get:

$$\mathbf{P}[X_n = n/3 + \beta n] = \frac{16}{81(1+3\beta)^{3/2}n^{2/3}}\left(\frac{a_1}{2}\mathcal{A}(\chi) + \frac{a_4}{n^{2/3}}\exp\left(-\frac{2}{3}\chi^3\right)\mathrm{Ai}(\chi^2)\right)\left(1 + O(1/n)\right),$$

where: *(i)* $\chi = n^{1/3}\gamma$; *(ii)* the error term is uniform for β in any compact subinterval of $\big(-1/3, 2/3\big)$ and is also uniform for any $k > \lambda(n)$, up to replacing $O(1/n)$ with $O\big(\lambda(n)^{-1}\big)$; *(iii)* γ, a_1, a_4 are functions of β made explicit in [1].

3. Applications to Maps and Random Sampling

The former framework was applied to the families of random maps presented in Table 1, whose generating functions are all of Lagrangian type. Each family is characterized by two parameters α_0 and c, displayed in Table 1.

Theorem 2. *Consider any scheme of Table 1 with parameters* α_0 *and* c. *The probability* $\mathbf{P}[X_n = k]$ *that a map of size* n *has a core of size* k *has a local limit law of the map–Airy type with centering constant* α_0 *and scale parameter* c.

Suppose now that one wants to generate a random map from a given family in Table 1. For general, nonseparable, bipartite, and cubic nonseparable maps, an algorithm `Map` is already given in [8] that takes an integer n and outputs in linear time a map of size n uniformly at random. For the other families of Table 1, one can use the following probabilistic algorithm `Core(k)` with parameter $f(k)$:

1. use `Map(n)` to generate a random map $M \in \mathcal{M}$ of size $n = f(k)$;
2. extract the largest component C of M with respect to the scheme;
3. if C does not have size k, then go back to step 1; otherwise output C.

Except for an exponentially small number of failures, this algorithm produces an element of $\mathcal{C}_k$ with uniform probability. Among other results, we have:

Theorem 3. *In all extraction/rejection algorithms of* [8], *the choice* $f(k) = k/\alpha_0$ *yields an algorithm whose average number of iterations satisfies*

$$\ell_n \sim n^{2/3}/\big(\mathcal{A}(0)c\big).$$

Let $x_0 \approx 0.44322$ *be the position of the peak of the map–Airy density function given by the equation*

$$(1 - 4x_0^3)\mathrm{Ai}(x_0^2) + 4x_0^2\mathrm{Ai}'(x_0^2) = 0.$$

The optimal choice $f(k) = k/\alpha_0 - (x_0/\alpha_0 c)\,(k/\alpha_0)^{2/3}$ *reduces the expected number of loops by* $1 - \mathcal{A}(0)/\mathcal{A}(x_0) \approx 30\%$.

This proves that the extraction/rejection algorithms have overall complexity $O(k^{5/3})$, as do variant algorithms of [7, 8] that are uniform over all $\mathcal{C}_k$. This complexity drops to $O(k)$ if one allows some small tolerance on the size of the generated map.

Bibliography

[1] Banderier (Cyril), Flajolet (Philippe), Schaeffer (Gilles), and Soria (Michèle). – Planar maps and Airy phenomena. In Montanari (Ugo), Rolim (José D. P.), and Welzl (Emo) (editors), *Automata, languages and programming. Lecture Notes in Computer Science*, vol. 1853, pp. 388–402. – Springer, New York, 2000. Proceedings of the 27th ICALP Conference, Geneva, Switzerland, July 2000.

[2] Bleistein (Norman) and Handelsman (Richard A.). – *Asymptotic expansions of integrals.* – Dover Publications Inc., New York, 1986, xvi+425p. A reprint of the second Holt, Rinehart and Winston edition, 1975.

[3] de Bruijn (N. G.). – *Asymptotic methods in analysis.* – Dover Publications Inc., New York, 1981, third edition, xii+200p. A reprint of the third North Holland edition, 1970 (first edition, 1958).

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

[5] Gao (Zhicheng) and Wormald (Nicholas C.). – The size of the largest components in random planar maps. *SIAM Journal on Discrete Mathematics*, vol. 12, n° 2, 1999, pp. 217–228.

[6] Goulden (I. P.) and Jackson (D. M.). – *Combinatorial enumeration.* – John Wiley & Sons Inc., New York, 1983, xxiv+569p. With a foreword by Gian-Carlo Rota, Wiley-Interscience Series in Discrete Mathematics.

[7] Schaeffer (Gilles). – *Conjugaison d'arbres et cartes combinatoires aléatoires.* – PhD thesis, Université Bordeaux I, 1998.

[8] Schaeffer (Gilles). – Random sampling of large planar maps and convex polyhedra. In *Proceedings of the thirty-first annual ACM symposium on theory of computing (STOC'99).* pp. 760–769. – ACM press, Atlanta, Georgia, may 1999.

[9] Tutte (W. T.). – Planar enumeration. In *Graph theory and combinatorics (Cambridge, 1983)*, pp. 315–319. – Academic Press, London, 1984.

[10] Wong (R.). – *Asymptotic approximations of integrals.* – Academic Press Inc., Boston, MA, 1989, xiv+546p.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 17–22.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Enumeration of Geometric Configurations on a Convex Polygon

Marc Noy

Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya

December 16, 1999

Summary by Michel Nguyễn-Thế

Abstract

We survey recent work on the enumeration of non-crossing configurations on the set of vertices of a convex polygon, such as triangulations, trees, and forests. Exact formulæ and limit laws are determined for several parameters of interest. In the second part of the talk we present results on the enumeration of chord diagrams (pairings of $2n$ vertices of a convex polygon by means of n disjoint pairs). We present limit laws for the number of components, the size of the largest component and the number of crossings. The use of generating functions and of a variation of Levy's continuity theorem for characteristic functions enable us to establish that most of the limit laws presented here are Gaussian. (Joint work by Marc Noy with Philippe Flajolet and others.)

1. Analytic Combinatorics of Non-crossing Configurations [3]

1.1. **Connected graphs and general graphs.** Let $\Pi_n = \{v_1, \ldots, v_n\}$ be a fixed set of points in the plane, conventionally ordered counter-clockwise, that are vertices of a regular n-gon K. Define a *non-crossing graph* as a graph with vertex set Π_n whose edges are straight line segments that do not cross. A graph is *connected* if any two vertices can be joined by a path. Parameters of interest are the number of edges of connected graphs and general graphs, and the number of components of general graphs.

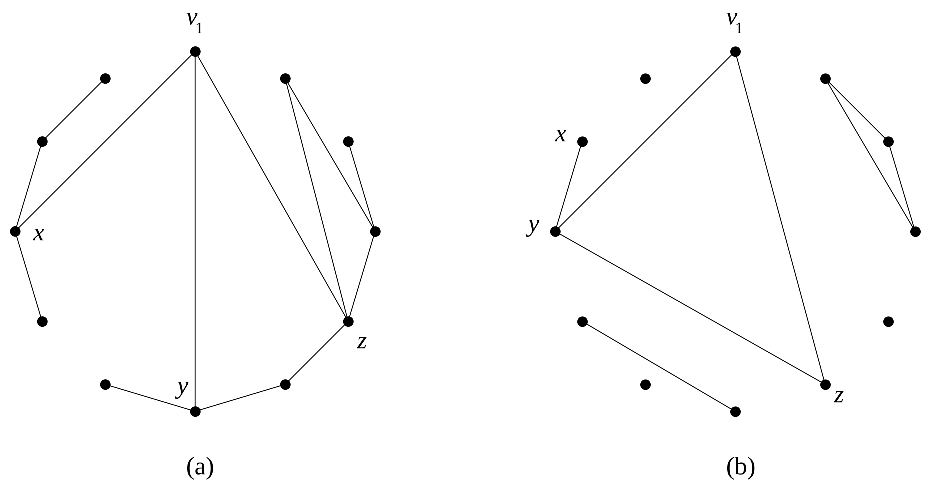


FIGURE 1. (a) A connected non-crossing graph; (b) an arbitrary non-crossing graph.

1.2. **Trees and forests.** A *(general) tree* is a connected acyclic graph and the number of edges in a tree is one less than the number of vertices. The study of trees becomes easier with the introduction of *butterflies* [3], defined to be ordered pairs of trees with a common vertex; a tree appears to be a sequence of butterflies attached to a root. A *forest* is an acyclic graph, in other words a graph whose components are trees.

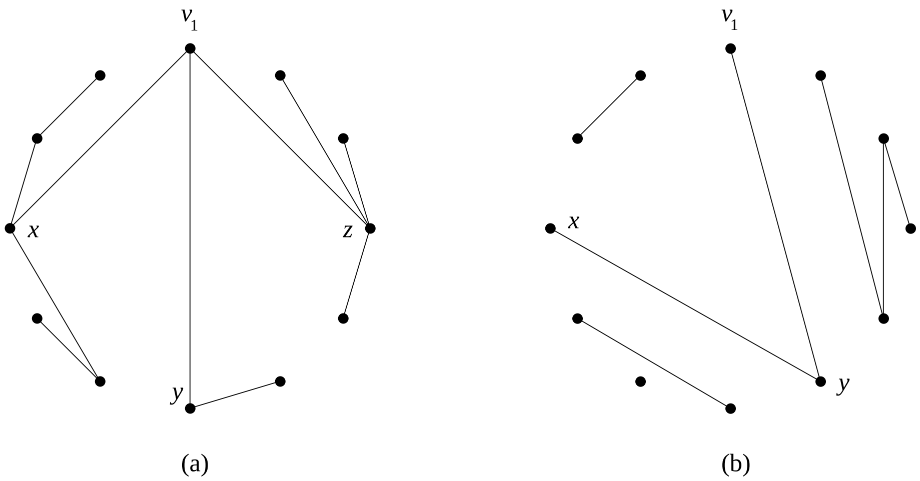


FIGURE 2. (a) A tree; (b) a forest.

1.3. **Triangulations.** A *triangulation* [7] is a set $\mathcal{T}_n$ of $n-3$ non-crossing diagonals v_iv_j which partitions K into $n-2$ triangles. As each triangle corresponds to an internal node of a binary tree (see the generating function of exercise 7.22 of [6]) via a classical bijection due to Euler [11], the number $\widehat{T_n}$ of triangulations is given by the $(n-2)$-th Catalan number $\widehat{T_n} = C_{n-2} = \binom{2n-4}{n-2}/(n-1)$. Let d_i denote the *degree* of the vertex v_i (i.e., the number of diagonals incident with v_i) and $\|v_iv_j\| = \min\big(|i-j|, n-|i-j|\big)$ the length of a diagonal v_iv_j. Define [2]:

$$\Delta_n(\tau) = \max\left\{ d_i \;\middle|\; i = 0, \dots, n-1 \right\},$$

the *maximal degree* of the vertices, and

$$\lambda_n(\tau) = \max\left\{ \|v_iv_j\| \;\middle|\; v_iv_j \in \mathcal{T}_n \right\},$$

the *length of the longest diagonal* in the triangulation.

Those features are of interest for a triangulation τ because they convey information about the corresponding tree $b(\tau)$: $\Delta_n(\tau)$ measures the *external-node separation* of $b(\tau)$, i.e., the maximal distance between successive external nodes; $\lambda_n(\tau)$ measures its *nearly half* measure, i.e., the size of the largest subtree with not more than half the external nodes.

Using combinatorial bijections and probability lemmas [2], we find:

$$\mathbf{E}[\Delta_n] \sim \log_2 n, \text{ and } \mathrm{E}[\lambda_n] \sim \alpha n, \quad \text{where} \quad \alpha = \frac{\sqrt{3}}{\pi} + \frac{1}{3} - \frac{\log\left(2+\sqrt{3}\right)}{\pi} \simeq 0.4654.$$

Let an *ear* of a triangulation τ be a triangle sharing two sides with the polygon, and e_n the number of ears of a triangulation. Let us view triangulations as binary trees and ears as leaves (internal node whose children are external nodes [11]) or roots with at least one child that is an external node, and let $\widehat{B}$ enumerate binary trees by size and number of leaves and $\widehat{T}$ enumerate triangulations by size and number of ears.[1] These generating series satisfy [5]

$$z^2\widehat{T}(z,w) = \big(1+2z(w-1)\big)\widehat{B}(z,w), \quad \text{where} \quad \widehat{B}(z,w) = z\big(w + 2\widehat{B}(z,w) + \widehat{B}(z,w)^2\big),$$

leading to $\mathbf{Var}[e_n] \sim \sqrt{n}/4$ and a Gaussian limit law (see §1.5 below). The expectation

$$\mathbf{E}[e_n] = \frac{n(n-1)}{2(2n-5)} \sim \frac{n}{4}$$

was already known from a combinatorial manipulation of Catalan numbers described in [7].

1.4. **Generating functions.** The combinatorial objects and parameters above, except for extremal ones, lead to univariate and bivariate generating functions, given in Table 1 below.

Configuration	Generating function equation
Connected graphs	$C^3 + C^2 - 3zC + 2z^2 = 0$
—, edges	$wC^3 + wC^2 - (1+2w)zC + (1+w)z^2 = 0$
Graphs	$G^2 + (2z^2 - 3z - 2)G + 3z + 1 = 0$
—, edges	$wG^2 + \big((1+w)z^2 - (1+2w)z - 2w\big)G + w + (1+2w)z = 0$
—, components	$G^3 + (2w^3z^2 - 3w^2z + w - 3)G^2 + (3w^2z - 2w + 3)G + w - 1 = 0$
Trees	$T^3 - zT + z^2 = 0$
—, leaves	$T^3 + (z^2w - z^2 - z)T + z^2 = 0$
Forests	$F^3 + (z^2 - z - 3)F^2 + (z+3)F - 1 = 0$
—, components	$F^3 + (w^3z^2 - w^2z - 3)F^2 + (w^3z + 3)F - 1 = 0$
Triangulations	$z^4\widehat{T}^2 + (2z^2 - z)\widehat{T} + 1 = 0$
—, ears	$z^4\widehat{T}^2 + \big(1+2z(w-1)\big)(2z^2 - z)\widehat{T} + w\big(1+2z(w-1)\big)^2 = 0$

TABLE 1. Generating function equations (z and w mark vertices and the secondary parameter).

A few tricks enable one to make Lagrange inversion applicable and to derive exact formulæ—sometimes involving summations—for all coefficients. For example, the change of variable $T = z + zy$ followed by Lagrange's formula yields:

$$T_n = \frac{1}{2n-1}\binom{3n-3}{n-1} \quad \text{and} \quad T_{n,k} = \frac{1}{n-1}\binom{n-1}{k}\sum_{j=0}^{k-1}\binom{n-1}{j}\binom{n-k-1}{k-1-j}2^{n-2k+j}.$$

Finding $C_{n,k}$ goes through a parameterization of the functional equation of C. To get the coefficients $\widehat{T}$, we use the equality $\widehat{T}_{n,k} = \widehat{B}_{n+2,k-1} + 2\widehat{B}_{n+1,k} - 2\widehat{B}_{n+1,k}$ deduced from $z^2\widehat{T}(z,w) = \big(1+2z(w-1)\big)\widehat{B}(z,w)$.

1.5. **Asymptotics.** All of the univariate generating functions above, and a few others (dissections and partitions of convex polygons) not presented in the talk but available in [3], have a unique dominant singularity ρ in $(0,1)$, and can be written

$$f(z) = c_0 + c_1\left(1 - \frac{z}{\rho}\right)^{1/2} + O\left(1 - \frac{z}{\rho}\right), \quad \text{entailing} \quad [z^n]f(z) = \frac{c_1}{\Gamma(-1/2)}\left(1 + O\left(\frac{1}{n}\right)\right).$$

For example the numbers T_n and F_n of respectively general trees and forests satisfy

$$T_n \asymp (27/4)^n = 6.75^n \quad \text{and} \quad F_n \asymp 8.2246^n, \quad \text{whence} \quad T_n = o(F_n).$$

[1]The expression of $\widehat{T}$, entailing the Gaussian limit of the distribution of ears of triangulations, was established by the author of this summary.

The numbers C_n and G_n of respectively connected and general graphs satisfy

$$C_n \sim \left(\frac{\sqrt{6}}{9} - \frac{\sqrt{2}}{6}\right) \asymp 10.39^n \quad \text{and} \quad G_n \sim \frac{1}{4}\sqrt{99\sqrt{2} - 140} \times \frac{2^n(3+2+\sqrt{2})^n}{\sqrt{\pi}n^{3/2}} \asymp 11.65^n,$$

entailing $C_n/G_n \to 0$ when $n \to \infty$.

The bivariate generating function seen before admits the form

$$f(z,w) = c_0(w) + c_1(w)\left(1 - \frac{z}{\rho(w)}\right)^{1/2} + O\left(1 - \frac{z}{\rho(w)}\right);$$

this leads to

$$f_n(w) = \gamma(w)\left(\frac{1}{\rho(w)}\right)^n \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right), \text{ or } \frac{f_n(w)}{f_n(1)} = \frac{\gamma(w)}{\gamma(1)}\left(\frac{\rho(1)}{\rho(w)}\right)^n \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right).$$

From the Quasi-Powers theorem [5, 8], which is a consequence of Levy's continuity theorem for characteristic functions, one deduces that f_n is asymptotically normal. The mean μ_n and variance σ_n satisfy $\mu_n \sim \kappa n$ and $\sigma_n^2 \sim \lambda n$ for algebraic numbers κ and λ.

For instance, for the distribution of the number of edges in the space of connected graphs of given size, we have $\kappa = \left(1+\sqrt{3}\right)/2 \simeq 1.366$.

2. **Analytics Combinatorics of Chord Diagrams** [4]

2.1. **Definitions.** Take $2n$ points on a circle, labelled 1, 2, ..., $2n$, and join them in disjoint pairs by n chords. The resulting configuration is called a *chord diagram.* A diagram is *connected* if no set of chords can be separated from the remaining chords by a line. A *component* is a maximal connected subdiagram.

2.2. **Components.**

2.2.1. *Number of components.* Let $C(z) = \sum_{n\geqslant 0} C_n z^n$ be the generating function of connected diagrams of size n. The bivariate generating function $I(z,w) = \sum_{n,k\geqslant 0} I_{n,k} w^k z^n$ of diagrams of size n and k components satisfies $I(z,w) = 1 + wC\big(zI(z,w)^2\big)$.

We have the following result:

Theorem 1. *Let X_n be the number of components in a random diagram of size n.*

1. *For $k \geqslant 1$, one has* $\mathbf{P}[X_n = k] \underset{n\to\infty}{=} \dfrac{e^{-1}}{(k-1)!}\big(1+o(1)\big)$.
2. *The mean μ_n and the variance σ_n of the distribution satisfy $\mu_n \underset{n\to\infty}{\sim} 2$ and $\sigma_n^2 \underset{n\to\infty}{\sim} 1$.*

Sketch of proof. The proof of the first point makes use of "monoliths," or "monolithic diagrams," where a diagram is said to be monolithic if: *(i)* it consists solely of the connected component that contains 1 (called the root component) and of isolated edges; *(ii)* for any two such isolated edges (a,b) and (c,d), one never has $a < c < d < b$ or $c < a < b < d$ (in other words, two isolated chords are never in a dominance relation).

The ordinary generating function of monoliths reads $M(z) = C\big(z/(1-z)^2\big)$, and according to Stein and Everett [12] $C_n/I_n = e^{-1} + o(1)$, so one can deduce the relation $M_n \sim I_n$, i.e., that almost every diagram is a monolith. The number $M_{n,k}$ of monoliths of size n with k components is given by

$$M_{n,k} = \binom{2n-k}{k-1} C_{n-k+1} \sim \frac{e^{-1}}{(k-1)!} I_n.$$

As to the second point, using $2zC(z)C'(z) = C(z)^2 + C(z) - z$, which is deduced from

$$C_n = (n-1)\sum_{j=1}^{n-1} C_j C_{n-j}$$

and $C_1 = 1$ [9, 13], one finds

$$\mu_n = \frac{\partial I}{\partial w}(z,w)\Big|_{w=1} = \frac{1}{z}\big(I(z) + h(z) - 2\big), \quad \text{where} \quad h(z) = I(z)^{-1}.$$

Hence, letting $g_n = h_n/I_n$, one obtains

$$g_n = 1 - \sum_{k=1}^{n-1} g_k \binom{n}{k}\binom{2n}{2k}^{-1} = 1 - \frac{1}{n} + \frac{3}{4n^2} + O(n^{-3}),$$

and $\mu_n = \dfrac{I_{n+1} + h_{n+1}}{I_n} = \dfrac{2n+1}{n+1} + O(n^{-1}) \sim 2$. Similar computations yield the variance. □

2.2.2. *Largest connected component.*

Theorem 2. *Let L_n be the size of the largest connected component in a random diagram of size n. Then, as $n \to \infty$, the mean μ_n and the variance σ_n of the distribution of L_n are*

$$\mathbf{E}[L_n] = n - 1 + o(1), \qquad \mathbf{Var}[L_n] = 1 + o(1),$$

and for any fixed $k \geqslant 1$, one has $\mathbf{P}[n - L_n = k] = \dfrac{e^{-1}}{k!}\big(1 + o(1)\big)$. In other words, thre random variable $n - L_n$ follows a Poisson law of parameter 1.

The proof relies on the analysis of the largest component in a monolith, namely, the root component with probability $1 - o(1)$, the other components being only edges. The number $M_{n,k}$ of monoliths of size n with root component of size $n - k$ is given by:

$$M_{n,k} = \binom{2n-k-1}{k} C_{n-k} \underset{n\to\infty}{\sim} \frac{e^{-1}}{(k-1)!} I_n.$$

2.3. **Crossings.** Let κ denote the number of chord crossings in a chord diagram, and let $\mathcal{I}_n$ be the set of all diagrams of size n. Flajolet and Noy proved the following result:

Theorem 3. *Let X_n be the random variable equal to the value of κ taken over the set of chord diagrams $\mathcal{I}_n$ of size n endowed with the uniform probability distribution.*

1. *The mean μ_n and the variance σ_n of the distribution of X_n are given by*

$$\mu_n = \mathbf{E}[X_n] = \frac{n(n-1)}{6} \quad \text{and} \quad \sigma_n^2 = \mathbf{Var}[X_n] = \frac{n(n-1)(n+3)}{45}, \quad \text{respectively.}$$

2. *The distribution of X_n is Gaussian in the asymptotic limit: for all real x, one has*

$$\lim_{n\to\infty} \mathbf{P}\left[\frac{X_n - \mu_n}{\sigma_n} \leqslant x\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2}\, dy.$$

Sketch of proof. Flajolet and Noy prove a stronger result by computing the moments of any order. They use the *exact formula* discovered by Touchard [14] and Riordan [10], namely that the series

$$\phi_n(q) = \sum_{w\in\mathcal{I}_n} q^{\kappa(w)} \quad \text{equals} \quad \frac{1}{(1-q)^n} \sum_{k=-n}^{n} (-1)^k q^{k(k-1)/2} \binom{2n}{n+k}.$$

Using the equality $e^{a^2/2} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} e^{ax}\, dx$ for $a = k\sqrt{t}$, one obtains:

$$\phi_n(e^t) = \frac{1}{(1-e^t)^n} \sum_{k=-n}^{n} (-1)^k e^{-kt/2} \binom{2n}{n+k} e^{k^2 t/2} = \frac{1}{2\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-x^2/2} x^{2n} H(x,t)^n\, dx,$$

where $H(x,t) = \dfrac{2\sinh^2(x\sqrt{t}/2 - t/4)}{x^2 \exp(t/2)\sinh(t/2)}$.

Taking derivatives with respect to t and taking the limit when $t \to 0$ yields the moments of any order; this proves the first point of the claim.

The Laplace method delivers the asymptotic relation

$$e^{-u\mu_n/\sigma_n} \frac{\phi_n(u/\sigma_n)}{\phi_n(1)} = e^{u^2/2}\big(1 + O(n^{-1/5})\big).$$

From Levy's continuity theorem for Laplace transforms [1], one concludes that $(X_n - \mu_n)/\sigma_n$ converges in distribution towards $\mathcal{N}(0,1)$. $\square$

Bibliography

[1] Billingsley (Patrick). – *Probability and measure.* – John Wiley & Sons Inc., New York, 1995, third edition, xiv+593p. A Wiley-Interscience Publication.

[2] Devroye (L.), Flajolet (P.), Hurtado (F.), Noy (M.), and Steiger (W.). – Properties of random triangulations and trees. *Discrete & Computational Geometry*, vol. 22, n° 1, 1999, pp. 105–117.

[3] Flajolet (Philippe) and Noy (Marc). – Analytic combinatorics of non-crossing configurations. *Discrete Mathematics*, vol. 204, n° 1-3, 1999, pp. 203–229.

[4] Flajolet (Philippe) and Noy (Marc). – Analytic combinatorics of chord diagrams. In Krob (D.), Mikhalev (A. A.), and Mikhalev (A. V.) (editors), *Formal Power Series and Algebraic Combinatorics*. pp. 191–201. – Springer Verlag, 2000. Proceedings of FPSAC'2000, June 2000, Moscow.

[5] Flajolet (Philippe) and Sedgewick (Robert). – *The average case analysis of algorithms: multivariate asymptotics and limit distributions.* – Research Report n° 3162, Institut National de Recherche en Informatique et en Automatique, 1997. 123 pages.

[6] Graham (Ronald L.), Knuth (Donald E.), and Patashnik (Oren). – *Concrete mathematics.* – Addison-Wesley Publishing Co., Reading, MA, 1989, xiv+625p. A foundation for computer science.

[7] Hurtado (F.) and Noy (M.). – Ears of triangulations and Catalan numbers. *Discrete Mathematics*, vol. 149, n° 1-3, 1996, pp. 319–324.

[8] Hwang (Hsien-Kuei). – *Théorèmes limites pour les structures combinatoires et les fonctions arithmétiques.* – Thèse de doctorat, École polytechnique, December 1994.

[9] Nijenhuis (Albert) and Wilf (Herbert S.). – The enumeration of connected graphs and linked diagrams. *Journal of Combinatorial Theory. Series A*, vol. 27, n° 3, 1979, pp. 356–359.

[10] Riordan (John). – The distribution of crossings of chords joining pairs of $2n$ points on a circle. *Mathematics of Computation*, vol. 29, 1975, pp. 215–222. – Collection of articles dedicated to Derrick Henry Lehmer on the occasion of his seventieth birthday.

[11] Sedgewick (Robert) and Flajolet (Philippe). – *An introduction to the analysis of algorithms.* – Addison-Wesley Publishing Co., Reading, MA, 1996.

[12] Stein (P. R.) and Everett (C. J.). – On a class of linked diagrams. II. Asymptotics. *Discrete Mathematics*, vol. 21, n° 3, 1978, pp. 309–318.

[13] Stein (Paul R.). – On a class of linked diagrams. I. Enumeration. *Journal of Combinatorial Theory. Series A*, vol. 24, n° 3, 1978, pp. 357–366.

[14] Touchard (Jacques). – Sur un problème de configurations et sur les fractions continues. *Canadian Journal of Mathematics*, vol. 4, 1952, pp. 2–25.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 23–26.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Tutte Polynomials in Square Grids

Marc Noy
Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya

December 16, 1999

Summary by Frédéric Chyzak

Abstract

The Tutte polynomial of a graph G is a two-variable polynomial that records much information on G. In particular, different evaluations at integers provide the number of spanning trees, forests (acyclic spanning subgraphs), and acyclic orientations of G. We estimate these values when G is an $n \times n$ square grid so as to deduce refined upper and lower bounds for the numbers of forests and acyclic orientations on such grids.

1. Polynomial Invariants of Graphs

1.1. **Chromatic polynomials.** A general graph $G = (V, E)$ is a undirected graph with loops and multiple edges allowed; it is described by its set V of vertices and its set E of edges. The *chromatic polynomial* $p(G;\lambda)$, introduced by Birkhoff in 1912 is a very important invariant of G: it counts the number of its λ-*colourings*, i.e., the number of ways to assign colours to the vertices of G in such a way that no two adjacent vertices share the same colour, and that the number of colours used is at most λ. This polynomial records many statistics of the graph: indeed, for a graph on n vertices, we have the expansion $p(G;\lambda) = \lambda^n - |E|\lambda^{n-1} + a\lambda^{n-2} - \cdots \pm \lambda^{\kappa(G)}$ where $a = |E|\big(|E| - 1\big)/2 - t(G)$ relates to the number $t(G)$ of triangles in G, and where $\kappa(G)$ is the number of connected components of G. Also, the coefficients of $p(G;\lambda)$ alternate in signs. Table 1 provides other interesting graph statistics as evaluations of the chromatic polynomial.

Unfortunately, the computation of a chromatic polynomial is hard: already the problem of computing the chromatic number of a graph G, i.e., the smallest integer λ such that there exists a λ-colouring, is NP-complete; evaluating the chromatic polynomial itself is #P-hard, as is even computing the chromatic polynomial at any algebraic number different from 0, 1, and 2. A simple exponential algorithm to compute $p(G;\lambda)$ is based on *contraction and deletion of edges*: the graph $G \,/\, e$ resulting from the contraction of an edge e in a graph G is obtained by removing the edge and identifying both incident vertices; the mere deletion of an edge e in a graph G results in the graph $G \setminus e$ with same vertex set V and new edge set $E \setminus \{e\}$. The algorithm consists in following the recurrence $p(G;\lambda) = p(G \setminus e;\lambda) - p(G \,/\, e;\lambda)$ provided that G is connected and that e is neither a loop nor a *bridge* (also called *isthmus* or *co-loop*, i.e., an edge whose deletion does not disconnect the graph). Finally, the chromatic polynomial of a (possibly disconnected) graph is the product of the chromatic polynomials of its connected components.

1.2. **Tutte polynomials.** A generalization of the chromatic polynomial is the *Tutte polynomial* $T(G;x,y)$ of a graph G [5, 6], most easily defined as the variant $T(G;x,y) = R(G;x-1,y-1)$ of Whitney's *rank generating function* $R(G;x,y)$ [9]. The *rank* of a graph G is defined as the size of any of its spanning forests, which is $|V| - \kappa(G)$. This notion stems from the *matroid* interpretation

$p(G;0)$	0	$T(G;1,1)$	# of spanning trees
$p(G;1)$	1 if G is empty	$T(G;2,1)$	# of forests
$p(G;1)$	0 if G contains an edge	$T(G;1,2)$	# of connected subgraphs
$p(G;2)$	$2^{\kappa(G)}$ if G is bipartite	$T(G;2,0)$	# of acyclic orientations [4]
$p(G;2)$	0 if G is not bipartite	$T(G;1,0)$	# of ac. or. with a single source
$\lvert p(G;-1)\rvert$	# of acyclic orientations [4]	$T(G;0,2)$	# of totally cyclic orientations

TABLE 1. Special evaluations of the chromatic (left) and Tutte (right) polynomials.

of graphs [7, 8], which, informally, views circuits (i.e., cycles) in a graph as dependency relations and forests as sets of independent edges. Now, by definition

$$R(G;x,y) = \sum_{A\subseteq E} x^{r(E)-r(A)} y^{|A|-r(A)} = x^{r(E)} \sum_{A\subseteq E} y^{|A|}/(xy)^{r(A)}, \tag{1}$$

where $r(A)$ denotes the rank of the subgraph $G_A = (V, A)$ of the graph $G = (V, E)$ obtained by retaining the subset $A \subseteq E$ of its edges only. Note that $r(A) = r(E)$ means that G_A has the same number of connected components as G, while $r(A) = |A|$ means that G_A is acyclic. The chromatic polynomial is recovered through the relation $p(G;\lambda) = (-1)^{r(G)}\lambda^{\kappa(G)}T(G;1-\lambda,0)$; on the other hand, the relation $f(G;\lambda) = (-1)^{|G|}T(G;0,1-\lambda)$ defines the *flow polynomial* of G, which counts the number of flows on G with edges weighted by elements of $\mathbb{Z}/\lambda\mathbb{Z}$, once any orientation has been chosen on G. (A *flow* is an assignment of weights to edges in such a way that the weights corresponding to all edges incident to the same vertex add up to zero.) Table 1 provides other interesting graph statistics as evaluations of the Tutte polynomial.

An algorithm similar to the one in the case of the chromatic polynomial above computes the Tutte polynomial, and is based on the relations: $T(G;x,y) = 1$ if G is empty; $T(G;x,y) = T(G\,/\,e;x,y)$ if e is a bridge; $T(G;x,y) = T(G\setminus e;x,y)$ if e is a loop; and $T(G;x,y) = T(G\,/\,e;x,y) + T(G\setminus e;x,y)$ otherwise. Finally, the Tutte polynomial of a (possibly disconnected) graph is the product of the Tutte polynomials of its connected components.

1.3. **Tutte–Grothendieck invariants.** A restatement of this is that the Tutte polynomial is an example of *Tutte–Grothendieck invariant* [2], i.e., a function v from the set of graphs to a fixed commutative ring—$\mathbb{Z}[x,y]$ in the case of the Tutte polynomial—with the relations:

1. $v(G) = v(G\,/\,e) + v(G\setminus e)$ provided G is connected and e is neither a loop nor a bridge;
2. the invariant of a graph is the product of the invariants of its connected components;
3. the invariants of two isomorphic graphs are the same.

A result by Brylawski [2] is that any Tutte–Grothendieck invariant is uniquely determined by its values on the loop and bridge graphs, consisting of a single loop around a single vertex and of a single edge between two vertices, respectively, and the invariant $v(G)$ is the evaluation of the Tutte polynomial at $x = v(\text{loop graph})$ and $y = v(\text{bridge graph})$.

The Tutte polynomial satisfies the following more general universality theorem (cf. [1, Chap. X]). Let v be any function from the set of graphs to the commutative ring $\mathbb{Z}[x,y,\alpha,\sigma,\tau]$ which satisfies conditions 2. and 3. in the description of Tutte–Grothendieck invariants and the relations $u(G) = \alpha^{|G|}$ if G is empty; $u(G) = xu(G\,/\,e)$ if e is a bridge; $u(G) = yu(G\setminus e)$ if e is a loop; $u(G) = \sigma u(G\setminus e) + \tau u(G\,/\,e)$ otherwise. Then v is given in terms of the Tutte polynomial of G by the relation $v(G) = \alpha^{\kappa(G)}\sigma^{|G|}\tau^{r(G)}T(G;\alpha x/\tau, y/\sigma)$. Special cases are the chromatic and Tutte polynomials, respectively obtained when (x,y,α,σ,τ) is set to $(1-x,0,x,1,-1)$ and $(x,y,1,1,1)$.

1.4. **Matroidal interpretation of graphs.** Matroids [7, 8] are a general concept used to represent the combinatorics of dependency between objects of many different types, like linear dependency, affine dependency, algebraic dependency, the structure of cycles (or circuits) in a graph, and so on. Chromatic and Tutte polynomials extend to this setting with the same type of properties. Applications include lattice theory, graph theory, knot theory, coding theory, geometry, networks, percolation theory, and statistical mechanics.

2. Counting Problems on the $n \times n$ Grid

Although the following combinatorial objects are well-defined on any graph, we consider their enumeration on the square $n \times n$ grid L_n (with simple edges only) where we proceed to derive new asymptotic estimates:

1. A *matching* is a pairing of neighbouring vertices by edges of the graph, possibly leaving some of its vertices unpaired. Enumerating matchings relates to the study of a lattice gas model of statistical physics for a gas consisting of monomers and dimers.
2. A *perfect matching* is a matching that leaves no vertex on its own. This corresponds to a gas with dimers only.
3. A set of vertices is *independent* if no two of them can be joined by an edge. This corresponds to *Fibonacci arrays*, i.e., arrays consisting of 0's and 1's only, with no two consecutive 1's, either vertically or horizontally.
4. A *spanning tree* is a tree made of edges of the graph and that exhausts its vertices.
5. An *acyclic orientations* is an orientation of the edges of the graph that induces no cycle.

Upon substitution of each vertex of L_n by a square centred at this vertex, and after gluing squares that correspond to adjacent vertices, a matching becomes a *tiling with dominoes and squares* while a perfect matching becomes a *domino tiling*. Obviously, the above-mentioned transformation is a one-to-one correspondence. The following combinatorial algorithm by Temperley provides another bijection, between spanning trees on L_n and perfect matchings on L_{2n+1} deprived of one vertex: *(i)* spanning trees are rooted at some fixed vertex; *(ii)* dominoes are then placed on the branches of trees, from leaves to the root, and the same process is applied to the dual graph of the tree; *(iii)* domino tilings are changed into perfect matchings. The common counting number $t(n)$ on the grid L_n is given as $T(L_n; 1, 1)$ (see Table 1) and is known to satisfy $\lim_{n\to\infty} t(n)^{1/n^2} = t$ where $t = 3.2099125\ldots$

Upper and lower bounds for forests and acyclic orientations. The numbers of forests and acyclic orientations on the graph L_n are expressable in terms of its Tutte polynomial, and are $T(L_n; 2, 1)$ and $T(L_n; 2, 0)$, respectively (see Table 1). Since a spanning tree is a forest and a forest is merely an unconstrained choice of edges, the bounds $t_n < f_n < 2^{2n(n-1)} < 4^{n^2}$ hold for the number of forests. On the other hand, orienting all vertical edges towards the top endows L_n with an acyclic orientation, and acyclic orientations are orientations. This yields the bounds $2^{n(n-1)} < a_n < 2^{2n(n-1)} < 4^{n^2}$. Again, the limits $f = \lim_{n\to\infty} f(n)^{1/n^2}$ and $a = \lim_{n\to\infty} a(n)^{1/n^2}$ exist; the relations above yield the trivial bounds $t = 3.2099125\ldots < f < 4$ and $2 < a < 4$. Merino, Noy, and Welsh have obtained the improved bounds

$$t = 3.64497 \le f \le 3.74698 \quad \text{and} \quad 3.41358 \le a \le 3.56322.$$

The method used to derive the new, better upper bounds is to view the square grid L_n as a composite of m/n rectangular $m \times n$ grids $L_{m,n}$, relying on the computation of $T(L_{m,n}; 2, 1)$ as the cardinal of a rational language. The idea is to extend a forest, respectively an acyclic orientation, on $L_{m,n}$ to one on $L_{m,n+1}$. To this end, the m vertices on the nth column of the original graph are tagged in order to keep track of vertices that are members of the same tree. The number of such configurations is finite (in particular, the m vertices can be in at most m different trees). Among the 2^{2m-1} choices of edges that may be used to extend the original graph, only part of them do not produce a cycle. This provides a finite-state automaton that recognizes the relevant configurations on $L_{m,n}$. The generating series that enumerates this configurations is thus rational, and the counting numbers grow as the exponential α_m^n of an algebraic number α_m. Gluing n/m configurations on $L_{m,n}$ in any way yields the upper bounds $f_n \leq (\alpha_m^n)^{n/m} 2^{n(n/m-1)} \leq (2\alpha_m/m)^{n^2}$ (since blind gluing may produce cycles), as well as similar bounds for a_n (with a different α_m).

The case of the new lower bounds is very similar. Again, the forests, resp. acyclic orientations, on L_n are obtained by gluing relevant configurations on $L_{m,n}$. However, an additional constraint is that the selected configurations on $L_{m,n}$ induce forests, resp. acyclic orientations, on the graph $L^*_{m,n}$ obtained by contracting the mth row to a single vertex. This ensures that no cycle is created while gluing the rectangular grids. Again, the configurations on $L^*_{m,n}$ are counted by a rational language, yielding lower bounds of the same form as the upper bounds above. The numerical values indicated were obtained for $m = 8$. An article is in preparation [3].

3. Computing the Tutte Polynomial of $L_{m,n}$ by a Recurrence in n

The interpretation in terms of rational languages also applies to the computation of Tutte polynomials for $L_{m,n}$, based on the right-most representation (1) of Whitney's rank generating function. This form makes explicit the way to extend the rational automaton recognizing the forests of $L_{m,n}$, which has been described in the previous section. This extension only needs to keep track of the number of vertices ($+m$ at each column), the number of connected components (whose variation is between $-m$ and $+m$ at each column), and the number of edges (which by difference yields the rank). To each state s corresponding to a structure of connected components on the nth column of $L_{m,n}$, we associate a generating function $F^{(s)}(x,y,z) = \sum_n R_n^{(s)}(x,y)z^n$ where $R_n^{(s)}(x,y)$ is the contribution to the sum (1) restricted to configurations A of edges whose last column corresponds to state s. This induces a linear system of recurrences between the $F^{(s)}(x,y,z)$, with Laurent polynomial entries in x and y.

For fixed m, the rational generating function of the rank generating functions of the family of graphs $L_{m,n}$ is thus obtained as one of the $F^{(s)}(x,y,z)$ for a suitable state s. The rational generating function of the Tutte polynomials is then obtained by shifting x and y.

Bibliography

[1] Bollobás (Béla). – *Modern graph theory.* – Springer-Verlag, New York, 1998, xiv+394p.

[2] Brylawski (Thomas H.). – A decomposition for combinatorial geometries. *Transactions of the AMS*, vol. 171, 1972, pp. 235–282.

[3] Calkin (N.), Merino (C.), Noble (S.), and Noy (M.). – Improved bounds for the number of forests and acyclic orientations in the square lattice. – In preparation.

[4] Stanley (Richard P.). – A chromatic-like polynomial for ordered sets. In *Proc. Second Chapel Hill Conf. on Combinatorial Mathematics and its Applications (Univ. North Carolina, Chapel Hill, N.C., 1970)*, pp. 421–427. – Univ. North Carolina, Chapel Hill, N.C., 1970.

[5] Tutte (W. T.). – A ring in graph theory. *Proceedings of the AMS*, vol. 43, 1947, pp. 26–40.

[6] Tutte (W. T.). – A contribution to the theory of chromatic polynomials. *Canadian Journal of Mathematics*, vol. 6, 1954, pp. 80–91.

[7] Welsh (D. J. A.). – Matroids: fundamental concepts. In *Handbook of combinatorics, Vol. 1*, pp. 481–526. – Elsevier, Amsterdam, 1995.

[8] Welsh (Dominic). – Colouring problems and matroids. In *Surveys in combinatorics (Proc. Seventh British Combinatorial Conf., Cambridge, 1979)*, pp. 229–257. – Cambridge University Press, Cambridge, 1979.

[9] Whitney (Hassler). – The coloring of graphs. *Annals of Mathematics*, vol. 33, 1932, pp. 688–718.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 27–30.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Random Group Automata

Cyril Nicaud
LIAFA, Université Paris 7

February 21, 2000

Summary by Marianne Durand

Abstract

A group automaton is a complete deterministic automaton such that each letter of the alphabet acts on the set of states as a permutation [1, 5]. The aim is to describe an algorithm for the random generation of a minimal group automaton with n states. The treatment is largely based on properties of random permutations and random automata.

1. Properties

A group automaton is a complete deterministic automaton such that each letter of the alphabet acts on the set of states as a permutation [1, 5]. We consider a group automaton $\mathcal{A}$, with states $1, 2, \dots, n$. The state 1 is the initial state; the set of final states is denoted by F, the alphabet by $a, b, \dots$, and the transitions by $q_2 = \delta(q_1, a)$ or equivalently (q_1, a, q_2).

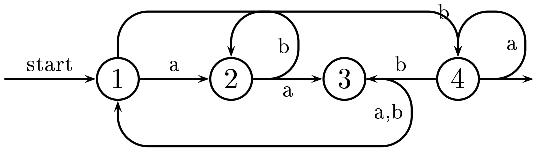


FIGURE 1. A group automaton.

Let us recall that two states q_1 and q_2 of an automaton are equivalent, notationally $q_1 \sim q_2$, if for every word u, the state $\delta(q_1, u)$ belongs to F if and only if $\delta(q_2, u)$ belongs to F. The automaton $\mathcal{A}$ is minimal if $\mathcal{A}$ has no distinct equivalent states. The structure properties of group automata are: the minimal automaton of a group automaton is a group automaton; the set of group automata is closed under union, intersection and complementation but it is not closed under star and product. As each letter acts like a permutation on the set of states, there cannot exist two transitions (q_1, a, q) and (q_2, a, q) with q_1 and q_2 distinct. This means that there is a "reversibility" property because when the automaton is in a state q after reading a word u, it is possible to retrace the path followed.

We are now interested in the connexity of an automaton. An automaton is connected if for any state q, there is a path joining the initial state to q. Because of the reversibility property, if a group automaton is connected then it is strongly connected, which means that for any states q and q', there is a path from q to q'. A group automaton is defined by the k permutations coding the transitions and by the set F, where k is the cardinality of the alphabet, so there are $(2^n - 1)n!^k$ group automata. We show that, if the alphabet has at least two letters, almost all group automata on n states are connected. In order to do this we first state the fact that given two permutations

σ and α the generated group $\langle\sigma,\alpha\rangle$ is almost surely transitive. This can be shown by a simple combinatorial argument. Take two letters a and b and consider σ_a the permutation related to a and σ_b the one related to b; then as $\langle\sigma_a,\sigma_b\rangle$ is almost always transitive the automaton is almost always connected. We even have an asymptotic estimate if the alphabet has exactly two letters:

$$\frac{\text{Card(not connected group automata)}}{\text{Card(group automata)}} \sim \frac{1}{n}.$$

2. Minimality

We now have to study the minimality of the automaton. An important theorem is that almost all connected group automata are minimal. The proof is partially based on the study of the one-letter case: if the automaton is connected, then as there is only one letter a, the permutation induced by a is a circular permutation. It is minimal if it is not stable under a rotation which is equivalent to saying that the word $u = 1\cdots\delta^k(1,a)\cdots\delta^{n-1}(1,a)$ is not a non-trivial factor of uu. Then in this case by counting the words corresponding to minimal circular permutations we show that almost all connected automata are minimal on a one-letter alphabet. If the alphabet has more than one letter, we observe that for almost all group automata, there is a letter a such that the permutation induced by a on the set of states has only one cycle of maximum length [3]. More precisely, we have the following lemma:

Lemma 1. *The probability that a permutation σ of size n has more than two cycles of maximum length is $o(1)$.*

Proof. Let $c_{n,m}$ be the probability that a permutation of size n has exactly two maximal cycles of size $m+1$. We note the generating function $C_m(z) = \sum_{n=0}^{\infty} c_{n,m} z^n$ and $c_n = \sum_{m\leq n/2} c_{n,m}$. The following equality holds:

$$C_m(z) = \frac{z^{2(m+1)}}{2(m+1)^2} e^z \cdots e^{\frac{z^m}{m}} = \frac{1}{1-z}\frac{z^{2(m+1)}}{2(m+1)^2}\exp\bigl(-r_m(z)\bigr)$$

where $r_m(z) = \sum_{n>m} z^n/n$ is the remainder of the generating function of the logarithm. In order to get the coefficient $c_{n,m}$ we apply Cauchy's formula:

$$c_{n,m} = \frac{1}{2i\pi}\int_{\mathcal{C}} \frac{1}{1-z}\frac{z^{2(m+1)}}{2(m+1)^2}\exp\bigl(-r_m(z)\bigr)\frac{dz}{z^{n+1}}$$

where $\mathcal{C}$ is a path around the origin. We choose for this path a circle around the origin defined by: $|z| = e^{-1/n}$ and we set $z = e^{-p/n}$ for a change of variable. So we have

$$c_{n,m} = \frac{1}{2i\pi}\int_{1-in\pi}^{1+in\pi} \frac{\exp\bigl(-r_m(e^{-p/n})\bigr)}{1-e^{-p/n}}\frac{e^{-p(2m+2)/n}}{2(m+1)^2}\frac{e^p}{n}\,dp$$

We now need to approximate some of the quantities in the integral, for this we use a technique and a few lemmas provided in [2]. We first have the relations

$$\text{(1)} \qquad r_m(e^{-p}) = E(mp) + O\left(\frac{e^{-mp}}{m}\right) \quad \text{and} \quad \frac{1}{n\left(1-e^{-p/n}\right)} = \frac{1}{p} + \frac{1}{n}\psi\left(\frac{p}{n}\right)$$

with $E(x) = \int_x^\infty \frac{e^{-v}}{v}\,dv$ and $\psi(z) = \frac{1}{1-e^{-z}} - \frac{1}{z}$, and where the error term $O\bigl(\exp(-mp)/M\bigr)$ is moreover uniform over $\Re(p) > 0$ and $\bigl|\Im(p)\bigr| \leq \pi$.

Property 1. *For all $a > 0$, the function $e^{-aE(u)}$ is bounded on $\Re(u) > 0$.*

The relations 1 allow us to write, after we set $\mu = m/n$:

$$c_{n,m} = \frac{1}{2i\pi}\int_{1-in\pi}^{1+in\pi} \exp\left(-E(\mu p) + O\left(\frac{1}{m}\right)\right)\left(\frac{1}{p}+\frac{1}{n}\psi\left(\frac{p}{n}\right)\right)\frac{e^p e^{-p(2m+2)/n}}{2(m+1)^2}\,dp$$
$$= \frac{1}{2i\pi}\int_{1-in\pi}^{1+in\pi} \exp\bigl(-E(\mu p)\bigr)\left(\frac{1}{p}+\frac{1}{n}\psi\left(\frac{p}{n}\right)+O\left(\frac{1}{pm}\right)\right)\frac{e^p e^{-p(2m+2)/n}}{2(m+1)^2}\,dp.$$

This rewrites as $c_{n,m} = I_1 + I_2 + I_3$ where

$$I_1 = \frac{1}{2i\pi}\int_{1-in\pi}^{1+in\pi} \exp\bigl(-E(\mu p)\bigr)\frac{1}{p}\frac{e^p e^{-p(2m+2)/n}}{2(m+1)^2}\,dp,$$
$$I_2 = \frac{1}{2i\pi}\int_{1-in\pi}^{1+in\pi} \exp\bigl(-E(\mu p)\bigr)\frac{1}{n}\psi\left(\frac{p}{n}\right)\frac{e^p e^{-p(2m+2)/n}}{2(m+1)^2}\,dp,$$
$$I_3 = \frac{1}{m}\frac{1}{2i\pi}\int_{1-in\pi}^{1+in\pi} \exp\bigl(-E(\mu p)\bigr)O\left(\frac{1}{p}\right)\frac{e^p e^{-p(2m+2)/n}}{2(m+1)^2}\,dp.$$

To study these three expressions, we use the fact that the quantities $\exp\bigl(-E(\mu p)\bigr)$ (Property 1) and $e^p e^{-p(2m+2)/n}$ are bounded uniformly on m. This helps us to give an upper bound for these three expressions: first,

$$I_1 = \int_{1-in\pi}^{1+in\pi} \frac{O(1)}{pm^2}\,dp = O\left(\frac{\log n}{m^2}\right)$$

and this approximation is uniform on m. Second

$$I_2 = \int_{1-in\pi}^{1+in\pi} O(1)\frac{1}{n}\psi\left(\frac{p}{n}\right)\frac{1}{2(m+1)^2}\,dp$$

as ψ is also bounded uniformly on m we have

$$I_2 = \frac{1}{nm^2}\int_{1-in\pi}^{1+in\pi} O(1)\,dp = O\left(\frac{1}{m^2}\right).$$

Third, as in the case of I_1, we obtain

$$I_3 = \frac{1}{m}\int_{1-in\pi}^{1+in\pi} O\left(\frac{1}{p}\right)\frac{1}{2(m+1)^2}\,dp = O\left(\frac{\log n}{m^3}\right).$$

Combining these estimates we obtain $c_{n,m} = O\left(\frac{\log n}{m^2}\right)$ uniformly on m. The approximation is going to be useful when m is greater than $\sqrt{n}$; otherwise we use the following lemma:

Lemma 2. *The probability that a permutation σ of size n has a maximal cycle of length smaller than $\sqrt{n}$ is $o(1)$.*

Proof. Let $p_{n,m}$ be the probability that a permutation of size n has all its cycles of size smaller than m. The saddle-point method gives us an upper bound for the quantity $p_{n,m}$. Then we have

$$p_{n,m} = [z^n]e^{l_m(z)} \le \frac{e^{l_m(r)}}{r^n} \quad \text{where} \quad l_m(z) = z + \cdots + \frac{z^m}{m}.$$

The saddle-point method drives us to apply this inequality to the value $r = n^{\frac{1}{3m}}$ chosen to fit the minimum, which gives

$$p_{n,m} \le \frac{\exp\left(n^{1/3}\log m\right)}{n^{n/3m}}, \quad \text{so} \quad p_{n,\sqrt{n}} \le e^{\left(\frac{n^{1/3}}{2}-\frac{\sqrt{n}}{3}\right)\log n} = o(1).$$

□

The probability that a permutation has two maximal cycles of size m is bounded by the probability that a permutation has one maximal cycle of size m. Therefore the probability that a permutation of size n has two maximal cycles of size smaller than $\sqrt{n}$ is $o(1)$. So $c_n = o(1) + \sum_{m=\sqrt{n}}^{m=n/2} c_{n,m} = o(1)$ by the approximation $c_{n,m} = O\left(\frac{\log n}{m^2}\right)$. Lemma 1 directly follows by showing that almost all permutations of size n having at least two maximal cycles have exactly two maximal cycles. □

We define $\mathcal{E}_n$ as the set of group automata $\mathcal{A}$ of size n that are connected and with the property that there exists one letter a such that the permutation induced by a has only one maximal cycle. By Lemma 1, we show that almost all connected group automata belong to $\mathcal{E}_n$. Furthermore, if $\mathcal{A}$ belongs to $\mathcal{E}_n$ then we can show that the maximal cycle of σ_a does not interfere with other cycles, because of their different cardinalities and so we can use the one-letter case, and say that this maximal cycle is almost always minimal. As the automaton considered is connected, this implies that the automaton is minimal. So we have the following result:

Theorem 1. *Almost all group automata are minimal.*

Proof. $\mathcal{E}_n \subset \mathrm{Minimal}_n \subset \mathrm{Connected}_n \subset \text{Group Automaton}_n$, and we have proved that almost every group automaton is in $\mathcal{E}_n$. □

3. Algorithm

This work naturally leads to an algorithm for generating uniformly at random a minimal connected group automata. Here the cardinality of the alphabet is bounded. The size of an automaton is the number n of states of its minimal automaton. The algorithm is:

- generate a random group automaton $\mathcal{A}$ using a function returning a random permutation for each letter of the alphabet. The cost is $O(n)$;
- test if $\mathcal{A} \in \mathcal{E}_n$, if not use Hopcroft's algorithm to check if it is minimal. Since Hopcroft is used rarely, the cost is $O(n)$;

this being done a constant number of time on average, because of the theorem above.

This yields a linear complexity in the average case, which is better than the best known algorithm by Hopcroft [4] which has complexity $n \log n$.

Bibliography

[1] Eilenberg (Samuel). – *Automata, languages, and machines. Vol. A.* – Academic Press, New York, 1974, xvi+451p. Pure and Applied Mathematics, Vol. 58.

[2] Gourdon (Xavier). – *Combinatoire, algorithmique et géométrie des polynômes.* – Thèse, École polytechnique, 1996.

[3] Gourdon (Xavier). – Largest component in random combinatorial structures. In *Proceedings of the 7th Conference on Formal Power Series and Algebraic Combinatorics (Noisy-le-Grand, 1995)*, vol. 180, pp. 185–209. – 1998.

[4] Hopcroft (John). – An $n \log n$ algorithm for minimizing states in a finite automaton. In *Theory of machines and computations (Proc. Internat. Sympos., Technion, Haifa, 1971)*. pp. 189–196. – Academic Press, New York, 1971.

[5] Hopcroft (John E.) and Ullman (Jeffrey D.). – *Introduction to automata theory, languages, and computation.* – Addison-Wesley Publishing Co., Reading, Mass., 1979, x+418p. Addison-Wesley Series in Computer Science.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 31–34.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Solving Discrete Initial- and Boundary-Value Problems

Marko Petkovšek
University of Ljubljana

October 4, 1999

Summary by Cyril Banderier

Abstract

Multivariate linear recurrences appear in such diverse fields of mathematics as combinatorics, probability theory, and numerical resolution of partial differential equations. Whereas in the univariate case the solution of a constant-coefficient recurrence always has a rational generating function, this is no longer true in the multivariate case where this generating function can be non-rational, non-algebraic, and even non-D-finite. Nevertheless, there are important cases where the solution can be computed exactly in terms of algebraic functions. Examples include many lattice-path problems such as the enumerations of Dyck, Motzkin, and Schroeder paths, determining the cardinality of various free algebras, and (in some cases) the enumeration of permutations with a forbidden pattern. This is joint work by Marko Petkovšek and Mireille Bousquet-Mélou (CNRS, Université de Bordeaux I).

1. Multivariate Linear Recurrences

Combinatorics are often synonymous of recurrences; whereas a quite impressive apparatus is available for univariate recurrences, multivariate recurrences are always a strange and mysterious world. Whereas a linear recurrence with constant coefficients necessarily leads to a rational generating function in one variable, this is no longer true in several variables (even with very regular boundary conditions). In fact, the set of multivariate generating functions with such a recurrence intersects almost all of the well-known classes of functions. Here are two examples leading to two kinds of generating functions.

A rational generating function: the chess king recurrence. On the square lattice, one performs a walk, beginning at $(0,0)$, made of a sequence of jumps (1,0), (1,1) or (0,1). Let $a_{n,k}$ be the number of ways to reach (n,k). Thus, one has the relation $a_{n,0} = a_{0,k} = 1$ (for $n,k \geq 0$) and the recurrence $a_{n,k} = a_{n-1,k} + a_{n,k-1} + a_{n-1,k-1}$ (for $n,k \geq 1$). Then, the generating function is

$$F(x,y) = \sum_{n,k=0}^{\infty} a_{n,k} x^n y^k = \frac{1}{1-(x+y+xy)}.$$

Of course there is an explicit formula for the coefficients (often referred to as Delannoy numbers[1]), namely $a_{n,k} = \sum_{i=0}^{n} \binom{n}{i}\binom{n+k-i}{n}$ which translates all the possible choices to perform i moves of the type $(1,1)$, $n-i$ moves of the type $(1,0)$ and $k-i$ moves of the type $(0,1)$.

[1] Henry Auguste Delannoy was born in 1833, graduated from the École polytechnique in 1853 and became a military intendant in the city of Orléans. He wrote a lot of contributions in recreative mathematics and combinatorics until 1895, the most remarkable being *How to use a chessboard in order to solve some probability problems.* After the death of his friend Lucas, he took in charge the publication of Lucas's last unachieved books.

An irrational generating function: the chess knight recurrence. This time, one performs jumps $(1,2)$ or $(2,1)$ and the $a_{n,k}$'s are known for $n \leq 1$ or $k \leq 1$. Petkovšek proved that $F(x,y) = \sum a_{n,k} x^n y^k$ is irrational [10] and a forthcoming article by Bousquet-Mélou and Petkovšek should show that F is in fact non D-finite.

A first problem which can arise in several variables is that initial conditions have to be correctly set in order to establish the uniqueness of the solution to a linear recurrence with constant coefficients. The second problem is what the nature of the solutions is, and how to compute them.

2. Existence and Uniqueness of the Solution

Henceforth, we view all the indices (and variables) as tuples of $\mathbb{Z}^d$ or $\mathbb{C}^d$, that is $\mathbf{n} = (n_1, \ldots, n_d)$ and $\mathbf{x} = (x_1, \ldots, x_d)$. Let $H = \{\mathbf{h_1}, \ldots, \mathbf{h_k}\}$ be the set of allowed jumps. Let $\mathbf{s}$ be the "true" *starting point* of the walk, that is the point after which all jumps are possible and where one does not care about the side conditions anymore. The kind of recurrence under study is formalized by

$$a_{\mathbf{n}} = \begin{cases} \phi(\mathbf{n}) & \text{for } \mathbf{n} \geq 0 \text{ and } \mathbf{n} \not\geq \mathbf{s}, \\ c_{\mathbf{h_1}} a_{\mathbf{n}+\mathbf{h_1}} + \cdots + c_{\mathbf{h_k}} a_{\mathbf{n}+\mathbf{h_k}} & \text{for } \mathbf{n} \geq \mathbf{s}. \end{cases} \tag{1}$$

The first part of the recurrence stands for the "initial" conditions (that is, the boundary values) and the second part reflects the different shifts (or jumps) allowed.

Definition 1 (Dependency relation). Define $\rightarrow$ by $p \rightarrow q \iff (p - q \in H \text{ and } q \geq s)$. Note $\xrightarrow{+}$ the transitive closure of $\rightarrow$.

Thus, $p \rightarrow q$ simply means that there is a "step" from p to q, with q outside of the "boundary value" area, and $p \xrightarrow{+} q$ means that there is a sequence of steps from p to q.

Theorem 1. *The following are equivalent:*

1. *the transitive closure* $\xrightarrow{+}$ *of the dependency relation* $\rightarrow$ *is well-founded in* $\mathbb{N}^d$*;*
2. *there exists* $\mathbf{u} > 0$ *such that* $\mathbf{u} \cdot \mathbf{h} < 0$ *for any "jump vector"* $\mathbf{h} \in H$*;*
3. *the convex hull of* H *does not intersect* $\mathbb{R}^d_+$.

The last point is the most efficient for proving uniqueness of the solution of recurrence (1) as it is easy to check. For example, for the chess king problem, one has a recurrence with starting point $\mathbf{s} = (1,1)$ and the set of allowed jumps is $H = \{(-1,0), (-1,-1), (0,-1)\}$, the intersection of the convex hull of H (a triangle in the lower left quarter) and of $\mathbb{R}^2_+$ is clearly the empty set; thus there is a unique solution to the recurrence. Considering now the recurrence $a_{n,k} = a_{n-1,k+2} + a_{n+2,k-1}$ (for $n, k \geq 1$), where the $a_{n,k}$'s are known (for $n = 0$ or $k = 0$), where the starting point is $\mathbf{s} = (1,1)$ and where the set of allowed jumps $H = \{(-1,2), (2,-1)\}$, gives an example for which the convex hull intersects $\mathbb{R}^2_+$; thus uniqueness does not hold. As a last example, one shows that the chess knight problem has a unique solution: the starting point is $\mathbf{s} = (2,2)$ and the set of allowed jumps is $H = \big\{(-2,1), (1,-2)\big\}$, whose convex hull does not intersect $\mathbb{R}^2_+$.

3. Nature of the Solution

Let $\mathbb{K}$ be a field of characteristic zero. Consider $F(\mathbf{x}) = \sum_{n \geq 0} a_n x^n$ with $a_n \in \mathbb{K}$ and $\mathbf{x}^n = x_1^n \cdots x_d^n$. A function $F(\mathbf{x})$ is called rational if there exist two polynomials P and Q in $\mathbb{K}[\mathbf{x}] \setminus \{0\}$ such that $QF - P = 0$. The function F is called algebraic if there exists $P \in \mathbb{K}[\mathbf{x}, t] \setminus \{0\}$ such that $P(\mathbf{x}, F(\mathbf{x})) = 0$. The function F is called D-finite if there exist polynomials $P_{i,j}$ in $\mathbb{K}[\mathbf{x}]$ such that

$$P_{i,k}(\mathbf{x}) \frac{\partial^k F(\mathbf{x})}{\partial x_i^k} + \cdots + P_{i,0}(\mathbf{x}) \frac{\partial^k F(\mathbf{x})}{\partial x_i^k} = 0$$

with $P_{i,j} \neq 0$ for at least one j for each $i = 1, 2, \ldots, d$. An equivalent definition states that the space spanned by all the derivatives of F is finite-dimensional over $\mathbb{K}(\mathbf{x})$. D-finite functions have nice closure properties and are related to a lot of combinatorial problems (see Stanley's article [12] and Lipshitz's article [7]).

Definition 2 (Apex)**.** The apex of H is the componentwise maximum of $H \cup \{0\}$.

For example, for the chess knight problem, one has $H = \{(-2,1),(1,-2)\}$ so the apex is $(1,1)$ (and the starting point is $(2,2)$). If $H = \{(-2,-1),(-1,2)\}$, then the apex is (0,2).

An important ingredient in the proof of the two theorems stated hereafter is the *kernel method.* Let us detail this point: one wants to make explicit the solution F_s of the recurrence (1), which rewrites $Q(\mathbf{x})F_s(\mathbf{x}) = K(\mathbf{x}) - U(\mathbf{x})$ where K stands for the *known* initial conditions and U stands for the *unknown* initial conditions. Q is called the kernel. The *kernel method* consists in cancelling the kernel $Q(\mathbf{x})$ by a choice of algebraic values $\mathbf{a}$ of $\mathbf{x}$, thus one gets a system of equations $K(\mathbf{a}) - U(\mathbf{a}) = 0$. Solving this system generally allows to make U explicit. This provides F_s for generic $\mathbf{x}$:

$$F_s(\mathbf{x}) = \frac{K(\mathbf{x}) - U(\mathbf{x})}{Q(\mathbf{x})}.$$

Typically, the function $U(\mathbf{x})$ is the sum of m unknown multivariate functions $F_i(x_1, \ldots, x_{d-1})$; thus cancelling the kernel with m different values for x_d (which then become functions of $(x_1, \ldots, x_{d-1})$) yields a system which allows to make explicit the F_i's. The kernel method has belonged to mathematical folklore since the 1970's; e.g., it has been used by combinatorialists [3][6, Sec. 2.2.1, Ex. 4 and 11] and probabilists [4]. There is also some recent work which makes a deep use of it [1, 2, 10, 11].

Theorem 2. *Assume the apex of H is 0. Then the generating function $F_s(\mathbf{x})$ of the unique solution of recurrence (1) is rational if and only if the known initial function $K(\mathbf{x})$ itself is rational.*

Theorem 3. *Take $\mathbb{K} = \mathbb{C}$. Assume the apex of H has at most one positive coordinate. Then the generating function $F_s(\mathbf{x})$ of the unique solution to the recurrence (1) is algebraic if and only if the known initial function $K(\mathbf{x})$ itself is algebraic.*

An algebraic example: Dyck paths. One performs steps $(1,1)$ or $(1,-1)$, the numbers $a_{i,j}$ of paths from $(0,0)$ to (i,j) satisfy the recurrence $a_{i,j} = a_{i-1,j-1} + a_{i-1,j+1}$ (for $m, n \geq 1$), $a_{0,0} = 1$ and $a_{i,j} = 0$ elsewhere. This leads to the functional equation $(y - x - xy^2)F_s(x,y) = y - U(x)$. Applying the kernel method yields

$$F_s(x,y) = \frac{y - \frac{1-\sqrt{1-4x^2}}{2x}}{y - x - xy^2}.$$

A transcendental and D-finite example: Young tableaux. The generating function of Young tableaux of height at most d is related to the numbers

$$a_{1,\ldots,1,n+1} = \prod_{i=1}^{d-1} i^{d-1} \frac{(dn)!}{\prod_{i=0}^{d-1}(n+i)!} \sim \prod_{i=1}^{d-1} i^{d-1} \frac{\sqrt{d}}{(2\pi)^{(d-1)/2}} \frac{d^{dn}}{n^{(d^2-1)/2}}.$$

Algebraicity would imply asymptotics of the type $C \cdot A^n / \big(\Gamma(1-r)n^r\big)$ with C and A algebraic numbers and r a rational number not in $\{1, 2, 3, \ldots\}$ (classical result from singularity analysis [5]). In our case, for odd $d > 1$, r is an integer and for even $d > 2$, $\Gamma\big(1 - (d^2-1)/2\big)$ is in $\mathbb{Q}\big(\sqrt{\pi}\big)$ but not in $\mathbb{Q}\big(\pi^{(d-1)/2}\big)$. Thus, the generating function of Young tableaux of height at most d is transcendental (for $d \geq 3$) and D-finite. Due to well-known one-to-one correspondences, this result extends from Young tableaux to ballot problems and involutions avoiding long increasing subsequences.

A non-D-finite and hypertranscendental example. The numbers $a_{m,n}$ defined by

$$a_{m,n} = a_{m+1,n-2} + a_{m-2,n+1} - a_{m-1,n-1}, \qquad \text{if } m,n \geq 2,$$

$a_{1,1} = -1$, and $a_{m,n} = 0$ elsewhere, actually belong to $\{0, 1, -1\}$ and correspond to a nice "fractal" lozenge pattern (see the "diamond figure" in [11]). Let $G(x) = \sum_{m \geq 2} a_{m,2}\, x^{m+1}$; one then has

$$F_s(x,y) = \sum_{m,n \geq 2} a_{m,n} x^{m-2} y^{n-2} = \frac{xy - G(x)G(y)}{(x-y^2)(y-x^2)}.$$

This equation gives $x^3 - G(x) - G(x^2) = 0$ which leads by iteration to $G(x) = x^3 \sum_{i \geq 0} (-i)^i x^{2^i}$. This kind of *lacunary* series cannot be D-finite. A stronger result gives that G is in fact hypertranscendental [8], which means that there exists no algebraic differential equation $P\left(z, G, G', \ldots, G^{(n)}\right) = 0$.

4. Conclusion

This talk gave a bestiary of solutions for linear multivariate recurrences with constant coefficients. In two dimensions, it covers the theory of Riordan arrays [9] (objects related to the Lagrange inversion formula). Even in two dimensions, it can be difficult to get the status of the generating function (algebraic?, D-finite?, ...). The main possible proofs are: in the algebraic case, the key point is the kernel method, note that this method also appears in two other summaries in these proceedings (see Schaeffer's and Banderier's talks); in the transcendental case, asymptotics allow to detect the nonalgebraicity, and for non D-finite functions, one generally tries bootstrapping and then obtaining an infinite number of singular points.

Marko Petkovšek has implemented some of the methods presented here in a Mathematica package MULTIVAR, available, as several author's articles, at `http://www.fmf.uni-lj.si/~petkovsek/`.

Bibliography

[1] Banderier (C.), Bousquet-Mélou (M.), Denise (A.), Flajolet (P.), Gardy (D.), and Gouyou-Beauchamps (D.). – Generating functions for generating trees. *Discrete Mathematics*. – 25 pages. To appear.

[2] Bousquet-Mélou (Mireille). – Multi-statistic enumeration of two-stack sortable permutations. *Electronic Journal of Combinatorics*, vol. 5, n° 1, 1998. – Research Paper 21, 12 pp. (electronic).

[3] Cori (Robert) and Richard (Jean). – Énumération des graphes planaires à l'aide des séries formelles en variables non commutatives. *Discrete Mathematics*, vol. 2, 1972, pp. 115–162.

[4] Fayolle (G.) and Iasnogorodski (R.). – Solutions of functional equations arising in the analysis of two-server queueing models. In *Performance of computer systems (Proc. Fourth Internat. Sympos. Modelling Performance Evaluation Comput. Systems, Vienna, 1979)*, pp. 289–303. – North-Holland, Amsterdam, 1979.

[5] Flajolet (Philippe). – Analytic models and ambiguity of context-free languages. *Theoretical Computer Science*, vol. 49, n° 2-3, 1987, pp. 283–309. – Twelfth international colloquium on automata, languages and programming (Nafplion, 1985).

[6] Knuth (Donald E.). – *The art of computer programming. Vol. 1: Fundamental algorithms.* – Addison-Wesley, 1968.

[7] Lipshitz (L.). – *D*-finite power series. *Journal of Algebra*, vol. 122, n° 2, 1989, pp. 353–373.

[8] Loxton (J. H.) and Van der Poorten (A. J.). – A class of hypertranscendental functions. *Aequationes Mathematicae*, vol. 16, n° 1-2, 1977, pp. 93–106.

[9] Merlini (Donatella) and Verri (M. Cecilia). – Generating trees and proper Riordan arrays. *Discrete Mathematics*, vol. 218, n° 1-3, 2000, pp. 167–183.

[10] Petkovšek (M.). – The irrational chess knight. In *Formal Power Series and Algebraic Combinatorics*, pp. 513–522. – 1998. Proceedings of FPSAC'98, June 1998, Toronto.

[11] Petkovšek (Marko) and Bousquet-Mélou (Mireille). – Linear recurrences with constant coefficients: the multivariate case. *Discrete Mathematics*, vol. 225, n° 1-3, 2000, pp. 51–75.

[12] Stanley (R. P.). – Differentiably finite power series. *European Journal of Combinatorics*, vol. 1, n° 2, 1980, pp. 175–188.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 35–38.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Classifying ECO-Systems and Random Walks

Cyril Banderier
Algorithms Project, INRIA Rocquencourt

September 27, 1999

Summary by Pierre Nicodème

Abstract

This talk presents a classification by rationality, algebraicity or transcendence of ECO-systems (Enumerating Combinatorial Objects) and of more general random walks. It is based on an article by Cyril Banderier, Mireille Bousquet-Mélou, Alain Denise, Philippe Flajolet, Danièle Gardy and Dominique Gouyou-Beauchamps [1].

1. Introduction

A *generating tree* is defined by a system (an axiom and a family of rewriting rules)

$$\Big((s_0), \big\{(k) \rightsquigarrow (e_1(k))(e_2(k))\dots(e_k(k))\big\}_{k\geq 0}\Big). \tag{1}$$

Here, the axiom (s_0) specifies the degree of the root, while the productions $e_i(k)$ (with $e_i(k) > 0$) list the degrees of the k descendants of a node labelled k (note the constraint on the number of descendants of a node). Such a system constitutes an *ECO-System.*

Example. 123-avoiding permutations. Consider the set $\mathfrak{S}_n(123)$ of permutations of length n that *avoid the pattern* 123: there exist no integers $i < j < k$ such that $\sigma(i) < \sigma(j) < \sigma(k)$. For instance, $\sigma = 4213$ belongs to $\mathfrak{S}_4(123)$ but $\sigma = 1324$ does not, since $\sigma(1) < \sigma(3) < \sigma(4)$.

Observe that if $\tau \in \mathfrak{S}_{n+1}(123)$, then the permutation σ obtained by erasing the entry $n+1$ from τ belongs to $\mathfrak{S}_n(123)$. Conversely, for every $\sigma \in \mathfrak{S}_n(123)$, insert the value $n+1$ in each place where this is compatible with the avoiding rule; this gives an element of $\mathfrak{S}_{n+1}(123)$. For example, the permutation $\sigma = 213$ gives 4213, 2413 and 2143, by insertion of 4 in first, second and third place respectively. The permutation 2134, resulting of the insertion of 4 in the last place, does not belong to $\mathfrak{S}_4(123)$. This process can be described by a tree whose nodes are the permutations avoiding 123: the root is 1, and the children of any node σ are the permutations derived as above (see Figure 1(a)).

Let us now label the nodes by their number of children: we obtain the tree of Figure 1(b). It can be proved that the k children of any node labelled k are labelled respectively $k+1, 2, 3, \dots, k$. Thus the tree we have constructed is the generating tree obtained from the following system:

$$\Big((2), \big\{(k) \rightsquigarrow (2)(3)\dots(k-1)(k)(k+1)\big\}_{k\geq 2}\Big). \tag{2}$$

Notations. We assume that all the values appearing in the generating tree are positive.

In the generating tree, let f_n be the number of nodes at level n and s_n the sum of the labels of these nodes. By convention, the root is at level 0, so that $f_0 = 1$. In terms of walks, f_n is the number of walks of length n. The generating function associated to the system is $F(z) = \sum_{n\geq 0} f_n z^n$.

Note that $s_n = f_{n+1}$, and that the sequence $(f_n)_n$ is nondecreasing.

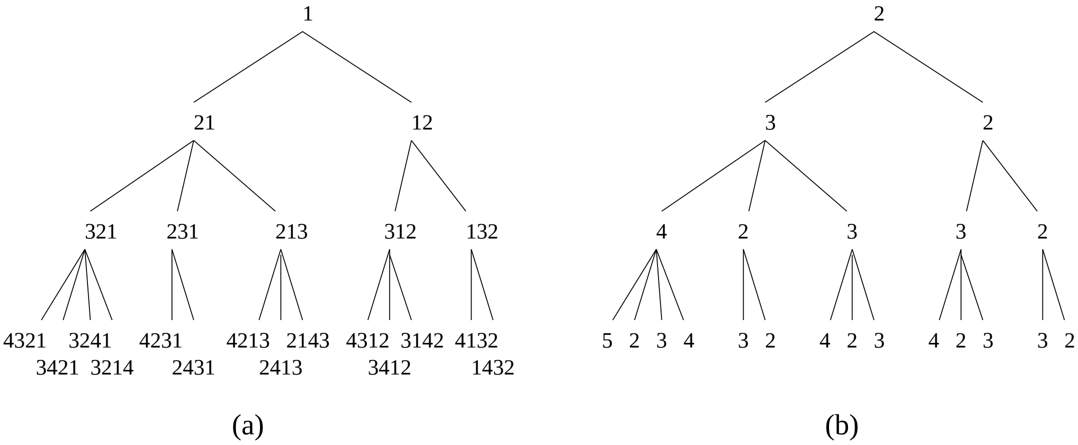


FIGURE 1. The generating tree of 123-avoiding permutations: (a) nodes labelled by the permutations; (b) nodes labelled by the numbers of children.

Now let $f_{n,k}$ be the number of nodes at level n having label k (or the number of walks of length n ending at position k). The following generating functions will be of interest:

$$F_k(z) = \sum_{n\geq 0} f_{n,k} z^n \qquad \text{and} \qquad F(z,u) = \sum_{n,k\geq 0} f_{n,k} z^n u^k.$$

We have $F(z) = F(z,1) = \sum_{k\geq 1} F_k(z)$. Furthermore, the F_k's satisfy the relation

$$F_k(z) = [k = s_0] + z \sum_{j\geq 1} \pi_{j,k} F_j(z), \tag{3}$$

where $[k = s_0]$ is 1 if $k = s_0$ and 0 elsewhere and $\pi_{j,k}$ denotes the number $\big|\{\, i \leq j \mid e_i(j) = k \,\}\big|$ of one-step transitions from j to k. This is equivalent to the recurrence $f_{n+1,k} = \sum_{j\geq 1} \pi_{j,k} f_{n,j}$ for the numbers $f_{n,k}$ (with $f_{0,s_0} = 1$), that results from tracing all the paths that lead to k in $n+1$ steps.

We refer to [1] for random generation using counting and generating trees.

2. Rational Systems

ECO-systems satisfying strong regularity conditions lead to rational generating functions. This covers systems that have a finite number of allowed degrees, as well as systems where the sum of the labels at level k depends linearly on k.

Proposition 1. *If finitely many labels appear in the tree, then $F(z) = F(z,1)$ is rational.*

Proof. Only a finite number of F_k's are nonzero; they are related by linear equations like Equation (3) above and therefore rational. $F(z)$ is a finite sum of these, and is also rational. □

Example. Fibonacci numbers are generated by the system $\big((1), \{(k) \rightsquigarrow (k)^{k-1}((k \bmod 2)+1)\}\big)$ that can also be written as $\big((1), \{(1) \rightsquigarrow (2), (2) \rightsquigarrow (1)(2)\}\big)$.

Proposition 2. *Let $\sigma(k) = e_1(k) + e_2(k) + \cdots + e_k(k)$. If σ is an affine function of k, say $\sigma(k) = \alpha k + \beta$, then the series $F(z)$ is rational. More precisely:*

$$F(z) = \frac{1 + (s_0 - \alpha)z}{1 - \alpha z - \beta z^2}.$$

Proof. Let $n \geq 0$ and let $k_1, k_2, \dots, k_{f_n}$ denote the labels of the f_n nodes at level n. Then

$$\begin{aligned} f_{n+2} = s_{n+1} &= (\alpha k_1 + \beta) + (\alpha k_2 + \beta) + \cdots + (\alpha k_{f_n} + \beta) \\ &= \alpha s_n + \beta f_n = \alpha f_{n+1} + \beta f_n. \end{aligned}$$

We know that $f_0 = 1$ and $f_1 = s_0$. The result follows. $\square$

Example. The system $\big((2), \{(k) \rightsquigarrow (2)^{k-1}(k+1)\}\big)$ produces the Fibonacci numbers of even index.

Proposition 2 can be adapted to apply to systems that "almost" satisfy its criterion (see [1]).

3. Algebraic Systems

Systems where a finite modification of the set $\{1, \dots, k\}$ is reachable from k lead to algebraic generating functions.

The possible moves from k are given by the rule:

$$(k) \rightsquigarrow \{(0), \dots, (k-1)\} \setminus \{(k-i) \mid i \in B\} \ \cup \ \{\{(k+j) \mid j \in A\}\}, \tag{4}$$

where $A \subset \mathbb{N}$ and $B \subset \mathbb{N}^+$ are a finite multiset (denoted $\{\{\dots\}\}$) and a finite set specifying respectively the *allowed forward jumps* (possibly coloured) and the *forbidden backwards jumps.*

Observe that these walk models are not necessarily ECO-systems, first because we allow labels to be zero—but a simple translation can take us back to a model with positive labels—, and second because we do not require (k) to have exactly k successors.

In this section $f_{n,k}$ is the number of walks of length n ending at point k and $f_n(u) = \sum_{k \geq 0} f_{n,k} u^k$ is the coefficient of z^n in $F(z,u)$.

We continue this section with the example $A = \{4, 15\}$ and $B = \{2\}$, axiom (0) and the corresponding family of rules

$$\big\{(k) \rightsquigarrow (0)(1)\dots(k-3)(k-1)(k+4)(k+15)\big\}.$$

This corresponds in generating functions to substituting u^k in

$$u^0 + \cdots + u^{k-1} - u^{k-2} + u^{k+4} + u^{k+15} = \frac{1-u^k}{1-u} - u^{k-2} + u^{k+4} + u^{k+15}$$

for $k \geq 2$. This gives the recurrence $f_{n+1}(u) = \frac{f_n(1) - f_n(u)}{1-u} + (u^4 + u^{15} - u^{-2}) f_n(u)$, and yields the functional equation

$$F(z,u) = 1 + z \left(\frac{F(z,1) - F(z,u)}{1-u} + P(u) F(z,u) - \{u^{<0}\} \sum_{n \leq 0} z^n L[f_n](u) \right). \tag{5}$$

Here $P(u) = \sum_{\alpha \in A} u^\alpha - \sum_{\beta \in B} u^{-\beta}$ and $L[g](u) = \frac{g(1) - g(u)}{1-u} + P(u) g(u)$. Equation (5) may be rewritten as

$$F(z,u) \left(1 + \frac{z}{1-u} - z P(u) \right) = 1 + \frac{z}{1-u} F(z,1) - z \sum_{j=0}^{b-1} c_j(u) \partial_u^j F(z,0),$$

where the $c_j(u)$ are Laurent polynomials. The kernel $K(z,u)$ of Equation (5) is the coefficient of $F(z,u)$ in the left-hand side of this equation. $F(z,u)K(z,u)$ is a linear combination of $b+1$ unknown functions. Solving $K(z,u) = 0$ in u gives $b+1$ convergent branches $u_i(z)$ which, in turn, give the $\partial_u^j F(z,0)$ through a $(b+1) \times (b+1)$ linear system, and from there $F(z,1)$, which is algebraic.

Proposition 3. *The generating function $F(z,1)$ counting the number of walks, starting from zero and irrespective of their endpoint is algebraic and $F(z,1) = -1/z \prod_{i=0}^{b}(1-u_i)$, where $b = \max B$ and $u_i(z)$ are the finite solutions at $z=0$ of the equation $K(z,u)=0$.*

Examples of algebraic systems are the Catalan numbers $\{(k) \rightsquigarrow (0)(1)\dots(k)(k+1)\}$, the Motzkin numbers $\{(k) \rightsquigarrow (0)\dots(k-1)(k+1)\}$, the Schröder numbers $\{(k) \rightsquigarrow (0)\dots(k-1)(k)(k+1)\}$ or the m-ary trees $\big((m), \{(k) \rightsquigarrow (m)\dots(k)(k+1)(k+2)\dots(k+m-1)\}\big)$.

4. Transcendental Systems

4.1. **Transcendence.** If the coefficients of a series grow too fast, its radius of convergence is zero.

Proposition 4. *Let b be a nonnegative integer. For $k \geq 1$, let $m_k = \big|\{\, i \mid e_i(k) \geq k-b \,\}\big|$. Assume that:*

1. *for all k, there exists a forward jump from k (i.e., $e_i(k) > k$ for some i),*
2. *the sequence $(m_k)_k$ is non-decreasing and tends to infinity.*

Then the generating function of the system has radius of convergence 0.

Proof. See [1]. □

However, there are ECO-systems or walks that are transcendental with positive radius of convergence such as $\{(k) \rightsquigarrow (2)(4)\dots(2k)\}$ or $\{(k) \rightsquigarrow (\lceil k/2\rceil)^{k-1}(k+1)\}$.

4.2. **Holonomy.** A subclass of transcendental functions is the class of holonomic functions. A series is said to be *holonomic* or *D-finite* if it satisfies a linear differential equation with polynomial coefficients in z. Equivalently, its coefficients f_n satisfy a linear recurrence relation with polynomial coefficients in n. Given a sequence f_n, the OGF (ordinary generating function) $\sum f_n z^n$ is holonomic if and only if the EGF (exponential generating function) $\sum f_n z^n/n!$ is holonomic.

The following table gives examples of holonomic and non-holonomic transcendental systems with references to the Encyclopedia of Integer Sequences (EIS) by Sloane and Plouffe [2, 3].

Axiom	Rewriting rules	Name	EIS Id.	Generating Function
	Holonomic OGF			**EGF**
(1)	$(k) \rightsquigarrow (k+1)^k$	Permutations	M1675	$1/(1-z)$
(2)	$(k) \rightsquigarrow (k)(k+1)^{k-1}$	Arrangements	M1497	$e^z/(1-z)$
(1)	$(k) \rightsquigarrow (k-1)^{k-1}(k+1)$	Involutions	M1221	$e^{z+z^2/2}$
(2)	$(k) \rightsquigarrow (k+1)^{k-1}(k+2)$	Partial permutations	M1795	$e^{z/(1-z)}/(1-z)$
	Nonholonomic OGF			**EGF**
(1)	$(k) \rightsquigarrow (k)^{k-1}(k+1)$	Bell numbers	M1484	e^{e^z-1}
(2)	$(k) \rightsquigarrow (k-1)(k)^{k-2}(k+1)$	Bessel numbers	M1462	—

Bibliography

[1] Banderier (C.), Bousquet-Mélou (M.), Denise (A.), Flajolet (P.), Gardy (D.), and Gouyou-Beauchamps (D.). – Generating functions for generating trees. *Discrete Mathematics.* – 25 pages. To appear.

[2] Encyclopedia of integer sequences. – Available from `http://www.research.att.com/~njas/sequences/`.

[3] Sloane (N. J. A.) and Plouffe (Simon). – *The encyclopedia of integer sequences.* – Academic Press Inc., San Diego, CA, 1995, xiv+587p.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), p. 39.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Combinatorics of Harmonic Polynomials

François Bergeron
LACIM, Université du Québec à Montréal

February 7, 2000

The space of totally harmonic polynomials in n variables (for the symmetric group) is "classically" defined as the set of solutions $y(x)$ to the system of PDE's:

$$\sum_{i=1}^{n} \partial_{x_i}^{k} y(x) = 0, \qquad 1 \leq k \leq n.$$

We recall an explicit description of this solution set before introducting the notion of diagonally harmonic polynomials. As we will see, this gives rise to many combinatorial problems.

Part II

Computer Algebra and Symbolic Methods

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 43–46.

Available online at the URL
http://algo.inria.fr/seminars/.

Efficient Algorithms on Numbers, Polynomials, and Series

Paul Zimmermann
Polka Project, INRIA Lorraine, F–54600 Villers-lès-Nancy, France

January 24, 2000

Summary by Frédéric Chyzak

Abstract

For a computer algebra system, it is crucial to optimize the arithmetical operations on basic objects—numbers, polynomials, series, ... In fact, two classes of objects can be distinguished: integers and polynomials, which require exact operations; floating-point numbers and series, for which only the most significant part of the exact result is needed. The best algorithms currently known for multiplication, division, and square root on integers and floating-point numbers are mostly recent. We present and analyse them using complexity models based on three different multiplication algorithms (naive, Karatsuba, and FFT).

The MPFR library developed by Guillaume Hanrot and Paul Zimmermann is a C library for multiprecision floating-point computations with exact rounding [6]. Its main purpose is to achieve efficiency with a well-defined semantics. Beside the elementary operations $+$, $-$, $\times$, and $/$, it provides routines for square root (with remainder in the integer case, without remainder in the floating-point case), logarithm and exponential. The longer-term goal is to integrate routines for the numerical evaluation of other elementary and special functions as well.

Paul Zimmermann's algorithm for square roots [8] originates in this work. It is reported on here, as well as other recent fast algorithms for multiplications, divisions, and square roots. They all base

Operation	Naive		Karatsuba		FFT	
Method	exact	truncated	exact	truncated	exact	truncated
Multiplication	1	1/2	1	1	1	1
Mulders				0.808		
Division	1	1/2				
Newton			7/2	5/2	5	4
Karp–Markstein			17/6	11/6	9/2	7/2
Jebelean, Burnikel–Ziegler			2	3/2		
Mulders				1.397		
Square root	1/2	1/4				
Newton			7/2	5/2	5	4
Karp–Markstein			17/6	11/6	9/2†	7/2†
Jebelean, Burnikel–Ziegler			3/2‡	1‡		
Mulders				0.966‡		

FIGURE 1. Complexity of division and square root algorithms in terms of exact multiplications for the three usual multiplication models. Algorithms marked '†', resp. '‡', were analysed, resp. designed and analysed, by Paul Zimmermann in [8].

on Newton's method, which essentially reduces division and square root to a few multiplications. Conversely, division cannot be performed faster than multiplication, for $ab = a/(1/b)$. Thus, once a model for multiplication is chosen, the best to hope is to lessen the constant in the computational complexity of inversion and square rooting. Several approaches to reduce this constant are described and combined in the following sections. To simplify the exposition, carries and their propagation are not taken into account, although they could be accomodated with no conceptual difficulty and no essential change of the complexities.

1. **The Three Classical Multiplication Models**

The naive multiplication algorithm computes a product by convolution between coefficients. Its arithmetical complexity is $N(n) = O(n^2)$. Karatsuba's recursive algorithm bases on the formula

$$uv = (u_1 b + u_0)(v_1 b + v_0) = u_1 v_1 b^2 + \big((u_1 + v_1)(u_0 + v_0) - u_1 v_1 - u_0 v_0\big) b + u_0 v_0, \tag{1}$$

where only three multiplications are required instead of four by the naive method, yielding the better complexity $K(n) = O(n^{\lg 3}) = O(n^{1.585\ldots})$. A refinement of this idea, splitting each term of the product into more and more parts as n goes to infinity, is the Toom–Cook approach [5]. The improved complexity is $O\left(n^{1+\sqrt{2}/\sqrt{\lg n}} \ln n\right)$. However this algorithm is only a theoretical one. Finally, the fastest known multiplication algorithm relies on FFT (fast Fourier transform) to achieve the complexity $F(n) = O(n \ln n \ln\ln n)$. FFT is a fast recursive method to compute the DFT (discrete Fourier transform) of a polynomial (i.e., its evaluation at each of the nth roots of unity, also called its Fourier coefficients). DFT exchanges product of polynomials—convolution of the coefficients—and point-wise product of the Fourier coefficients. A product of polynomials is thus essentially computed by two direct DFT, mulplication of the Fourier coefficients, and one reverse DFT. Note the following asymptotic relations between arithmetical complexities:

$$N(2n) \sim 4N(n), \quad K(2n) \sim 3K(n), \quad \text{and} \quad F(2n) \sim 2F(n). \tag{2}$$

2. **Newton's Scheme for Inverses and Square Roots**

Newton's schemes respectively given by $\iota(x) = x(2 - ax)$ and $\rho(x) = x(3 - ax^2)/2$ converge to $1/a$ and $1/\sqrt{a}$. This entails that inverses and square roots can be computed by additions and multiplications only, using $b/a = b \times (1/a)$ and $\sqrt{a} = a \times (1/\sqrt{a}\,)$. Both methods have a quadratic convergence rate since

$$\iota\left(\frac{1+\epsilon}{a}\right) = \frac{1-\epsilon^2}{a} \quad \text{and} \quad \rho\left(\frac{1+\epsilon}{\sqrt{a}}\right) = \frac{1 - 3\epsilon^2/2 - \epsilon^3/2}{\sqrt{a}}.$$

This means that the number of correct digits doubles at each step of the iteration.

For a of size n and x of size $n/2$, a naive calculation of $\iota(x)$ would take $5M(n/2)$ arithmetical operations, returning an output of size $2n$. The method is optimized by writing $\iota(x) = x + x(1 - ax)$ and noting that if the $n/2$ digits of x are correct, $1 - ax$ starts with $n/2$ zeroes and ends with a correction of size n, whose first $n/2$ digits only are useful. Thus, only the middle $n/2$ digits of ax are computed in $2M(n/2)$ arithmetical operations, then multiplied with x, then added to x by merely appending them. The overall cost $I(n)$ for inverting a of size n is therefore given by the recurrence $I(n) = 3M(n/2) + I(n/2)$. Unfolding it using (2) yields the asymptotics $2N(n)$ (no improvement), $3K(n)/2$, and $3F(n)$, depending on the multiplication model. Adding 1 for the final multiplications, this gives the constants for the truncated case. In the case of inversion with remainder, the latter is computed after the division as a correcting term, so that another 1 has to be added to the constant.

The same trick works to compute square roots, after writing $\rho(x) = x + x(1 - ax^2)/2$: x^2 is computed in $M(n/2)$ arithmetical operations, then $1 - ax^2$ in $M(n)$ arithmetical operations; the first $n/2$ digits are zero, and only the next $n/2$ ones are multiplied with x in $M(n/2)$ arithmetical operations. The overall cost $S(n)$ to compute $1/\sqrt{a}$ for a of size n is therefore given by the recurrence $S(n) = M(n) + 2M(n/2) + S(n/2)$, which once unfold yields the asymptotics $2N(n)$ (no improvement), $5K(n)/2$, and $4F(n)$, whence the constants for the truncated and exact cases.

3. Karp and Markstein's Modification of Newton's Method

Karp and Markstein's improvement is to incorporate the final multiplications $b \times (1/a)$ and $a \times (1/\sqrt{a})$, respectively, into the last step of Newton's method in the corresponding calculation [4].

In the case of the inverse, this corresponds to replacing the last step of the iteration with the computation of $y = bx$, then of $y + x(b - ay)$. Only the first $n/2$ digits of y are kept, and the convergence remains quadratic. As to the complexity, only $M(n/2)$ has been added to the iteration as a replacement for the arithmetical complexity $M(n)$ of a multiplication outside of it. The gain is thus $2K(n)/3$ or $F(n)/2$, depending on the multiplication model.

In the case of the square root, the last step of the iteration is replaced with the computation of $y = ax$, then of $y + x(a - y^2)/2$. Only the last $n/2$ digits of y are kept, the method remains quadratic, and the gains are the same as with inversion.

4. Burnikel and Ziegler's Division with Remainder

All the algorithms mentioned above base on Newton's method to reduce manipulations of objects of size $2n$ to manipulations of objects of size n. For a change, Burnikel and Ziegler's improvement of division [1, 3] consists of two mutually recursive algorithms for dividing an object of size $3n$ by an object of size $2n$ and for dividing an object of size $4n$ by an object of size $2n$. The division algorithm obtained in this way was then reused by Zimmermann for the computation of square roots [8].

Algorithm $D_{2/1}$ to divide $u_3b^3 + u_2b^2 + u_1b + u_0$ by $v_1b + v_0$ (where each u_i or v_i is a block of size n and where b is a suitable basis) first computes $(q_1, r_1b + r_0) = D_{3/2}(u_3b^2 + u_2b + u_1, v_1b + v_0)$, then $(q_0, s_1b + s_0) = D_{3/2}(r_1b^2 + r_0b + u_0, v_1b + v_0)$, to return $(q_1b + q_0, s_1b + s_0)$. The arithmetical complexity $D_{2/1}(n)$ to divide an object of size n by an object of size $n/2$ is thus twice the arithmetical complexity $D_{3/2}(n/2)$ to divide an object of size $3n/2$ by an object of size n. For its part, Algorithm $D_{3/2}$ to divide $u_2b^2 + u_1b + u_0$ by $v_1b + v_0$ first computes $(q, c) = D_{2/1}(u_2b + u_1, v_1)$, then $r = r_1b + r_0 = cb + u_0 - qv_0$; next, it decreases q by 1 while adding $v_1b + v_0$ to r until r is nonnegative, before returning (q, r). This 'while' loop is proved to cost little, so that the complexity $D_{3/2}(n)$ is just $D_{2/1}(n) + M(n)$.

Consequently, the complexity $D_{2/1}(n)$ is ruled by the recurrence $D_{2/1}(n) = 2D_{2/1}(n/2) + 2M(n/2)$. This makes no improvement in the case of FFT (complexity $2F(n)\ln n$), but provides a Karatsuba-based exact division of arithmetical complexity $2K(n)$, which is reduced to $3K(n)/2$ for truncated division. Indeed, the truncated variant of Algorithm $D_{2/1}$ calls the exact variant of Algorithm $D_{3/2}$ once, and its truncated variant once. Then, the exact $D_{3/2}$ only uses the exact $D_{2/1}$, while the truncated $D_{3/2}$ calls the truncated $D_{2/1}$. This variant saves as much as $M(n/2) + M(n/4) + \cdots$, that is to say $K(n)/2$ in the Karatsuba model.

Zimmermann's algorithm R to compute the square root of $u_3b^3 + u_2b^2 + u_1b + u_0$ first computes $(s', r') = R(u_3b + u_2)$, then $(q, u) = D_{2/1}(r'b + u_1, 2s')$; it next lets s and r be $s'b + q$ and $(ub + u_0) - q^2$, respectively; if r is nonnegative, it returns (s, r), else $(s, r + 2s - 1)$. The arithmetical complexity $R(n)$ to compute the square root of an object of size n is then given by the recurrence $R(n) = R(n/2) + D_{2/1}(n/2) + M(n/2)$. With multiplications by the Karatsuba algorithm, this reduces to $3K(n)/2$ for the exact case. In the truncated case, the algorithm is modified by calling the truncated variant of $D_{2/1}$ and by not substracting q^2 to define r. The recurrence becomes $R(n) = R(n/2) + D(n/2)$, which in the Karatsuba model delivers a complexity $K(n)$ for square roots without remainder.

5. Mulders' "Short Products"

Mulder's idea is a modification of Karatsuba's algorithm dedicated to the truncated case [7].

Each of the terms u_1v_1, $(u_1+v_1)(u_0+v_0)-u_1v_1-u_0v_0$, and u_0v_0 in Equation (1) has size $2n$ if the input u and v are of size $2n$. In view of a truncated product—or "short product"—, the same relation suggests to compute u_1v_1 exactly, only the most significant half of $(u_1+v_1)(u_0+v_0)-u_1v_1-u_0v_0$, and to save the calculation of u_0v_0. In fact, the simpler form $u_1v_0+u_0v_1$ is used: the product uv is thus reduced to an exact multiplication, u_0v_0, and two truncated multiplications, u_1v_0 and u_0v_1. Unfortunately, unfolding the recurrence $M(n) = K(n/2) + 2M(n/2)$ yields no optimization at all.

The idea is then to vary the sizes of the blocks in u and v: for blocks u_1 and v_1 of size βn, the recurrence becomes $M(n) = K(\beta n) + 2M\big((1-\beta)n\big)$, inducing $M(n) = cK(n)$ for $c = \beta^\alpha/\big(1 - 2(1-\beta)^\alpha\big)$, where $\alpha = \lg 3 = 1.585\ldots$ The optimum is obtained for $\beta \simeq 0.694$ and $c \simeq 0.808$.

The same idea applies to division, with an optimum for $\beta \simeq 0.542$ and $c \simeq 1.397$. Moreover, Zimmermann's algorithm reduces the computation of a truncated square root of an object of size n to an exact square root and a truncated division on objects of size $n/2$; this yields the arithmetical complexity $\simeq (3/2 + 1.397)K(n/2) \simeq 0.966K(n)$ for truncated square root.

6. Other Improvements

Other improvements for the Karatsuba model were announced in the talk: Hanrot and Zimmermann have obtained a better constant for inversion and division ($\simeq 1.212$), which was then used by Quercia to lessen the constant for division without remainder to roughly 1. These works have been further developed since then, with applications to square roots as well [2].

Bibliography

[1] Burnikel (Christoph) and Ziegler (Joachim). – *Fast recursive division.* – Research Report n° MPI-I-98-1-022, Max-Planck-Institut für Informatik, Saarbrücken, Germany, October 1998.

[2] Hanrot (Guillaume), Quercia (Michel), and Zimmermann (Paul). – *Speeding up the division and square root of power series.* – Research Report n° 3973, Institut National de Recherche en Informatique et en Automatique, July 2000. Available from `http://www.inria.fr/RRRT/RR-3973.html`.

[3] Jebelean (Tudor). – Practical integer division with Karatsuba complexity. In Küchlin (Wolfgang W.) (editor), *ISSAC'97 (July 21–23, 1997. Maui, Hawaii, USA).* pp. 339–341. – ACM Press, New York, 1997. Conference proceedings.

[4] Karp (Alan H.) and Markstein (Peter). – High-precision division and square root. *ACM Transactions on Mathematical Software*, vol. 23, n° 4, 1997, pp. 561–589.

[5] Knuth (Donald E.). – *The art of computer programming. Vol. 2.* – Addison-Wesley Publishing Co., Reading, Mass., 1981, second edition, xiii+688p. Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

[6] The MPFR library. – Available from `http://www.loria.fr/projets/mpfr/`.

[7] Mulders (Thom). – On short multiplications and divisions. *Applicable Algebra in Engineering, Communication and Computing*, vol. 11, 2000, pp. 69–88.

[8] Zimmermann (Paul). – *Karatsuba square root.* – Research Report n° 3805, Institut National de Recherche en Informatique et en Automatique, November 1999. 8 pages.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 47–52.

Available online at the URL
http://algo.inria.fr/seminars/.

Relax But Don't Be Too Lazy

Joris van der Hoeven
Laboratoire de Mathématique, Université Paris-Sud

January 24, 2000

Summary by Paul Zimmermann

Joris van der Hoeven's talk presents novel algorithms operating on formal power series. These new algorithms are based on fast multiplication methods (Karatsuba, Toom–Cook, FFT), and improve the best asymptotic complexities known, for example those obtained by Brent and Kung [1], while staying very efficient for the medium range (Karatsuba).

Most algorithms work with a linear space in the input size n, some of them require a space in $n \log n$. The basic idea of these new algorithms is what Joris van der Hoeven calls "the relaxed approach," intermediate between the zealous approach and the lazy approach. This relaxed approach was invented in 1997, with the presentation of two relaxed algorithms for the multiplication of formal power series at the ISSAC'97 conference [8]. The report [9] details these algorithms and their implantation, presents some other multiplication algorithms, shows how the relaxed approach extends naturally to other operations on formal power series, and finally offers several experimental comparisons between classical and relaxed algorithms.

1. The Zealous Approach

Let us consider the product of two formal power series, $f = f_0 + \cdots + f_n z^n$ and $g = g_0 + \cdots + g_n z^n$. The zealous approach consists in using *at the same time* every data $f_0, \ldots, f_n, g_0, \ldots, g_n$ to calculate the product $h = f \cdot g = h_0 + \cdots + h_n z^n + O(z^{n+1})$. So it corresponds to the classical or "off-line" approach. Several algorithms of different complexities implement this approach: the naïve multiplication in $O(n^2)$, Karatsuba's algorithm in $O(n^{\log_2 3})$ [6], and the multiplication by FFT in $O(n \log n \log\log n)$. The following table summarizes the complexity in time and space of the best known zealous algorithms for different operations on formal power series (to facilitate the reading, we omitted the $O(\cdot)$ terms):

Algorithm	Time	Space
Multiplication	$M(n) = n \log n \log\log n$	n
Division	$M(n)$	n
Differential equations	$M(n)$	n
Holonomic functions	n	n
Algebraic composition	$M(n) \log n$	n
General composition	$M(n)\sqrt{n \log n}$	$n \log n$
Composition in finite characteristic	$M(n) \log n$	n

Newton's method. Newton's method reduces several operations to elementary computations. For example, the logarithm of a formal power series is written:

$$\log f = \log f_0 + \int \frac{f'}{f}$$

and reduces to a division (f'/f) and an integration of linear complexity, whence a cost in $O\big(M(n)\big)$. Exponentiation reduces to logarithm by Newton's method. If g is such that $\log g - f = O(z^{n/2})$, i.e., g is an approximation to order $n/2$ of $\exp f$, then $\tilde{g} = g - g(\log g - f)$ will be an approximation to order n, whence an algorithm again in $O\big(M(n)\big)$. Functional inversion—given a series f, find g such that $f \circ g = z$—reduces to composition by:

$$\tilde{g} = g - \frac{f \circ g - z}{f' \circ g},$$

and so the complexity of inversion is that of composition.

Polynomial composition. The problem is as follows: given a polynomial f of degree p, a polynomial g with zero constant coefficient and of fixed degree q, and an integer $n \geq p$, compute $h = f \circ g$ to order n. The divide-and-conquer algorithm consists in writing:

$$f \circ g = (f_{\mathrm{lo}} + z^{p/2} f_{\mathrm{hi}}) \circ g = f_{\mathrm{lo}} \circ g + g^{p/2}(f_{\mathrm{hi}} \circ g),$$

and so on with $p/4$, $p/8$, $\ldots$, the powers of g being precomputed. It gives a complexity of $O\big((pq/n)M(n)\log n\big)$.

General composition. Given two formal power series $f = f_0 + \cdots + f_n z^n$ and $g = g_1 z + \cdots + g_n z^n$, we want to compute $h = f \circ g = h_0 + \cdots + h_n z^n + O(z^{n+1})$. Brent and Kung's algorithm [1] splits gf into two parts $g = g_{\mathrm{lo}} + g_{\mathrm{hi}}$:

$$\begin{aligned} g_{\mathrm{lo}} &= g_1 z + \cdots + g_q z^q \\ g_{\mathrm{hi}} &= g_{q+1} z^{q+1} + \cdots + g_n z^n, \end{aligned}$$

then writes the Taylor expansion of $f \circ (g_{\mathrm{lo}} + s)$ at $s = 0$:

$$f \circ g = f \circ g_{\mathrm{lo}} + (f' \circ g_{\mathrm{lo}}) g_{\mathrm{hi}} + \frac{1}{2}(f'' \circ g_{\mathrm{lo}})(g_{\mathrm{hi}})^2 + \cdots .$$

The computation of $f^{(n)} \circ g_{\mathrm{lo}}$ can be done by direct iteration:

$$f^{(i)} \circ g_{\mathrm{lo}} = \frac{\big(f^{(i-1)} \circ g_{\mathrm{lo}}\big)'}{g'_{\mathrm{lo}}}$$

or inverse iteration:

$$\frac{1}{(i-1)!} f^{(i-1)} \circ g_{\mathrm{lo}} = f_{i-1} + i \int \left(\frac{1}{i!} f^{(i)} \circ g_{\mathrm{lo}}\right) g'_{\mathrm{lo}}.$$

2. **The Lazy Approach**

Here, we regard the formal power series not as a list of coefficients given once and for all, but as a flow of coefficients. That corresponds to "in-line" computations. The lazy approach consists in calculating the coefficients one by one; at each stage, we only perform strictly necessary computations.

Let us consider for example the equation for the generating function $f(z)$ of binary trees counted according to their internal nodes:

$$f = 1 + zf^2.$$

Here the zealous or "off-line" approach does not apply because the coefficient f_n of order $n \geq 1$ of f depends on $f_0, f_1, \ldots, f_{n-1}$:

$$f_n = f_0 f_{n-1} + f_1 f_{n-2} + \cdots + f_{n-2} f_1 + f_{n-1} f_0.$$

Thus, for the multiplication at order 3 of $f = f_0 + f_1 x + f_2 x^2 + O(x^3)$ by $g = g_0 + g_1 x + g_2 x^2 + O(x^3)$ giving $h = f \cdot g = h_0 + h_1 x + h_2 x^2 + O(x^3)$, the lazy approach consists in calculating the value $h_0 = f_0 g_0$ at stage 0, then $h_1 = f_0 g_1 + f_1 g_0$ at stage 1 and $h_2 = f_0 g_2 + f_1 g_1 + f_2 g_0$ at stage 2, for a total of 6 multiplications. It is also possible to represent this computation graphically by the following table, where the value k at the intersection of line g_i and column f_j means that the value $f_j g_i$ is obtained at stage k:

g_2	2		
g_1	1	2	
g_0	0	1	2
$\times$	f_0	f_1	f_2

The major disadvantage of this approach is that the computation of all coefficients up to order n costs $O(n^2)$: we cannot use fast multiplication algorithms to reduce the complexity.[1]

Another example is the computation of the exponential $g = \exp f$ of a formal power series. By differentiation, we obtain $g' = g \cdot f'$, which reduces the exponentiation to a multiplication (the differentiation and the integration having linear complexity):

$$g = \int f' g.$$

However, here again, the series g appears in both members of the equation; with the lazy approach, we can calculate the product $f'g$ one term at a time only, here again giving a quadratic complexity.

The article [10] by Stephen Watt describes an implementation in Scratchpad II (former name of Axiom) of that approach, based on a lazy implementation of formal power series.

In conclusion, the lazy approach has the advantage on the zealous approach to apply to the case of implicit equations; in return it does not allow the use of fast multiplication algorithms, and therefore gives higher asymptotic complexities. It is precisely this drawback which the relaxed approach solves.

3. **The Relaxed Approach**

The relaxed approach tries to use fast algorithms from the zealous approach in cases where this approach is not applicable, i.e., when "off-line" computations are not possible, like for example for the computation of the coefficients of the generating function of binaries trees $f = 1 + zf^2$, or of the exponential of a series $g = \int f'g$.

The basic idea is the following: instead of performing the minimal computations at each stage as in the lazy approach, one performs a few more calculations at certain stages, which will allow the use of fast algorithms, and in the end a global gain. As all operations considered ultimately reduce to multiplications, it is enough to detail the relaxed approach for the multiplication of formal power series.

[1] By the way, this method is precisely that used in the COMBSTRUCT library for the enumeration of combinational structures.

Let us recall the above-mentioned example of the product of $f = f_0 + f_1x + f_2x^2 + O(x^3)$ by $g = g_0+g_1x+g_2x^2+O(x^3)$. The relaxed algorithm operates in the following way: at stage 1, instead of calculating h_1 by $f_0g_1+f_1g_0$, we obtain it by Karatsuba's formula $(f_0+f_1)(g_0+g_1)-f_0g_0-f_1g_1$, thus with two multiplications as well because $f_0g_0 = h_0$ has already been calculated. However, this already made it possible to compute part of h_2, namely f_1g_1. Then stage 2 has to compute f_0g_2 and f_2g_0 only, thus a gain of one multiplication compared to the lazy approach. The corresponding table is the following:

g_2	2		
g_1	1	1	
g_0	0	1	2
$\times$	f_0	f_1	f_2

where the square formed by the '0' and three '1' is obtained in three multiplications instead of four, thanks to Karatsuba's algorithm. Considering differently, we cut out the triangle of side 3 in two squares 1×1 and a square 2×2, for which we used a fast algorithm. More generally, any relaxed algorithm for the multiplication of formal power series of order n consists of a tiling of the triangle of side n by a set of squares. With each tiling corresponds a new algorithm. Each square is numbered by an integer from 0 to n, indicating the stage at which it is calculated; at stage n, only the coefficients of order less than or equal to n can be used.

The example above illustrates two significant points of relaxed algorithms:

1. at the end of stage 1, it is necessary to save the value of f_1g_1 which was computed in advance, for latter use at stage 2. The relaxed algorithms may thus require more memory than zealous algorithms. In most cases however, the memory used remains linear, but it can be in $n \log n$;
2. if we want to continue the calculation of $h = fg$ to a higher order, say order 4, the adopted strategy is not necessarily the best. Indeed, at stage 2 we could have calculated $(f_0 + f_2)(g_0 + g_2) - f_0g_0 - f_2g_2$ in two multiplications, which would give $f_0g_2 + f_2g_0$ in two multiplications as well, but would also give the term f_2g_2 of h_4.

Thus we can distinguish two cases: *(i)* the case where the maximum order n of calculations is known in advance and thus it is a question of optimizing the total number of operations up to this order n; *(ii)* the case where the maximum order is not known *a priori*, and one wants to optimize the "average" number of operations of the relaxed algorithm.

Joris van der Hoeven also shows that Karatsuba's algorithm for the multiplication of polynomials —we do not speak any more of formal power series here—is essentially relaxed, i.e., the formula giving the term h_k of the product only depends on $f_0, \ldots, f_k$ and $g_0, \ldots, g_k$. Consequently, Karatsuba's algorithm can directly be used for the relaxed multiplication. The table corresponding to the product of two polynomials of degree 3 is the following:[2]

g_3	3	3	3	3
g_2	2	3	2	3
g_1	1	1	3	3
g_0	0	1	2	3
$\times$	f_0	f_1	f_2	f_3

[2]Exercise: Find the operations carried out with each stage from 0 to 6 and check that one indeed performs 9 multiplications. Help: $9 = 1 + 2 + 2 + 3 + 1 + 0 + 0$.

The major disadvantage of the relaxed alternative of Karatsuba's algorithm is however the memory usage: on the one hand the memory required is in $O(n \log n)$, on the other hand the memory management is extremely complex, since for each stage it is necessary to know which values must be calculated, which should be reused—and among those, which can be destroyed—, finally which have to be saved for latter use.

Another algorithm proposed by Joris van der Hoeven consists in tiling the square $n \times n$ by a sequence of 'L' shapes of increasing width. That leads to a relaxed multiplication in $O\big(M(n) \log n\big)$. Several other alternatives are proposed in [9], both for complete products (polynomials) and truncated products (formal power series). The other operations (division, composition) are also "essentially relaxed." Finally we obtain the following complexities for the relaxed alternatives of the operations on formal power series:

Algorithm	Times	Space
Karatsuba's multiplication	$n^{\log_2 3}$	$n \log n$
Multiplication via FFT	$D(n) = M(n) \log n$	n
Division	$D(n)$	n
Differential equations	$D(n)$	n
Holonomic functions	n	n
Algebraic composition	$D(n) \log n$	n
General composition	$D(n)\sqrt{n \log n}$	$n^{3/2} \log n$
Composition in finished characteristic	$D(n) \log n$	$n \log n$

The time complexities are the same ones as for the zealous approach, while replacing $M(n)$ with $D(n)$. The memory complexity is identical, except when we use Karatsuba's multiplication algorithm (there is however a slower variant by a constant factor, but in space $O(n)$), or for the composition (general or in finite characteristic).

Joris van der Hoeven gives in his report [9] many experimental results for these new algorithms. Timings below correspond to an AMD processor at 200 MHz with 64 MB of main memory. Van der Hoeven's program calculates 500 terms of the Taylor expansion of $\exp(z \exp z)$ in 342 seconds against 1086 seconds for the zealous approach; it calculates the number of alcohols $C_n H_{2n+1} OH$ for $n = 5000$ in approximately 2300 seconds, whereas the naïve method does not allow this calculation in reasonable time and space; it calculates the expansion in $1/x$ of the differential-difference equation

$$f(x) = \frac{1}{x}\big(1 + f(x+1) + f'(x)^2\big)$$

to order 2000 in 1572 seconds.

4. **Conclusion**

Joris van der Hoeven presented us a whole panoply of algorithms which reduce the calculation of the first n coefficients of the majority of the formal power series defined by algebraic equations, differential equations or difference equations, to a quasi-linear complexity, whereas the best algorithms known before were almost quadratic (in the implicit case, i.e, where the zealous approach does not apply).

It would be nice if these algorithms were implemented in enumerative combinatorics softwares like COMBSTRUCT[3] or CS [3]. More generally, all computer algebra systems worthy of the name should implement these new algorithms, both for formal power series, polynomials, and integers. Indeed,

[3] `http://algo.inria.fr/libraries/software.html`

one of the by-products of Joris van der Hoeven's report is a division algorithm with remainder in $K(n)$ operations, whereas the best known algorithm was in $2K(n)$ [7, 2, 5].

Related work. For truncated division and square root, new algorithms based on Karatsuba's multiplication are detailed in the report [4].

Acknowledgement. People who don't read French may thank Gina Pierrelée-Grisvard who helped to translate this summary.

Bibliography

[1] Brent (Richard P.) and Kung (H. T.). – $O((n \log n)^{3/2})$ algorithms for composition and reversion of power series. In Traub (J. F.) (editor), *Analytic computational complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1975)*. pp. 217–225. – Academic Press, New York, 1976.

[2] Burnikel (Christoph) and Ziegler (Joachim). – *Fast recursive division.* – Research Report n° MPI-I-98-1-022, Max-Planck-Institut für Informatik, Saarbrücken, Germany, October 1998.

[3] Denise (Alain), Dutour (Isabelle), and Zimmermann (Paul). – CS: a MuPAD package for counting and randomly generating combinatorial structures. In *Formal Power Series and Algebraic Combinatorics*, pp. 195–204. – 1998. Proceedings of FPSAC'98, June 1998, Toronto. Sofware Demonstration.

[4] Hanrot (Guillaume), Quercia (Michel), and Zimmermann (Paul). – *Speeding up the division and square root of power series.* – Research Report n° 3973, Institut National de Recherche en Informatique et en Automatique, July 2000. Available from `http://www.inria.fr/RRRT/RR-3973.html`.

[5] Jebelean (Tudor). – Practical integer division with Karatsuba complexity. In Küchlin (Wolfgang W.) (editor), *ISSAC'97 (July 21–23, 1997. Maui, Hawaii, USA)*. pp. 339–341. – ACM Press, New York, 1997. Conference proceedings.

[6] Karatsuba (A. A.) and Ofman (Yu. P.). – Multiplication of multidigit numbers by automata. *Physics Doklady*, vol. 7, 1963, pp. 595–596. – Translated from *Doklady Akad. Nauk*, vol. 145, n° 2, 1962, pp. 293–294.

[7] Moenck (R.) and Borodin (A.). – Fast modular transforms via division. In *Proceedings of the 13th Annual IEEE Symposium on Switching and Automata Theory*, pp. 90–96. – October 1972.

[8] van der Hoeven (Joris). – Lazy multiplication of formal power series. In Küchlin (Wolfgang W.) (editor), *ISSAC'97 (July 21–23, 1997. Maui, Hawaii, USA)*. pp. 17–20. – ACM Press, New York, 1997. Conference proceedings.

[9] van der Hoeven (Joris). – *Relax, but don't be too lazy.* – Technical Report n° 78, Université de Paris-Sud, Mathématiques, Bâtiment 425, F-91405 Orsay, 1999. Submitted to the Journal of Symbolic Computation. Available from `http://www.math.u-psud.fr/~vdhoeven/`.

[10] Watt (Stephen M.). – A fixed point method for power series computation. In Gianni (Patrizia M.) (editor), *Symbolic and Algebraic Computation (International Symposium ISSAC'88, Rome, Italy, July 4-8, 1988). Lecture Notes in Computer Science*, vol. 358, pp. 206–217. – Springer Verlag, 1989. Conference proceedings.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 53–55.

Available online at the URL
http://algo.inria.fr/seminars/.

Threshold Phenomena in Random Lattices and Reduction Algorithms

Ali Akhavi
GREYC, Université de Caen

November 8, 1999

Summary by Philippe Flajolet

By a *lattice* is meant here the set of all linear combinations of a finite collection of vectors in $\mathbb{R}^n$ taken with integer coefficients,

$$\mathcal{L} = \mathbb{Z}e_1 \oplus \cdots \oplus \mathbb{Z}e_p.$$

One may think of a lattice as a regular arrangement of points in space, somewhat like atoms composing a crystal in $\mathbb{R}^3$. Given the generating family (e_j), there is great interest in finding a "good" basis of the lattice. By this is meant a basis that is "almost" orthogonal and is formed with vectors of "small" length. The process of constructing a "good" basis from a skewed one is referred to as lattice [basis] reduction.

Lattice reduction is of structural interest in various branches of number theory. For instance, reduction in dimension 2 is completely solved by a method due to Gauß. This entails a complete classification of binary quadratic forms with integer coefficients, a fact that has numerous implications in the analysis of quadratic irrationals and in the representation of integers by quadratic forms (cf. for example Pell's equation, $x^2 - dy^2 = 1$.)

The algorithmic and computational questions that stem from lattice reduction are of even greater applicability. In all generality, the *exact* optimization problem (i.e., finding the "best" basis, for instance, the one formed by vectors of strictly minimal lengths) is *NP*-complete, hence computationally intractable even in relatively low dimensions. However, as is usual in this range of optimization problems, *approximate* solutions may be found at a reasonable cost. In fact, a major advance in this area is due to Lenstra, Lenstra, and Lovász [4] who were the first to give a polynomial approximation algorithm (nicknamed the 'LLL' algorithm); this algorithm applies in all dimensions and is of polynomial time complexity. A spectacular consequence was to provide (for the first time) an algorithm that factorizes univariate polynomials over the rationals in polynomial time.[1] The LLL algorithm takes its inspiration from the classical Gram–Schmidt orthogonalization process, with the important modification that orthogonalization coefficients must be approximated by integers, while the algorithm strives to keep vectors of a "reasonable" length. This results both in a default of orthogonality and a default of minimality as regards the basis that is constructed.

Since 1982, the LLL algorithm has found innumerable consequences in various branches of computational number theory, computer algebra, cryptography, and combinatorial optimization.[2] The

[1] The authors of [4] proceed as follows. Let f be the initial polynomial (with integer coefficients) and h be an irreducible factor of $f \mod p^n$. The set of polynomials of degree one which reduce modulo p^n to a multiple of h is a lattice, and this lattice contains a vector of (relatively) short length if and only if it contains a multiple of the irreducible factor of f corresponding to h.

[2] An example of application at the crossroads of combinatorial optimization and cryptography is the Knapsack Problem.

superb book of von zur Gathen and Gerhard [5] devotes Chapters 16 and 17 to the question and offers a very readable account.

The talk presents two new notions of reduction that are structurally weaker than LLL reduction. These are called Gram reduction and Schmidt reduction. Regarding the algorithms associated to these reductions, not much gain is perceptible in the worst case when compared to LLL reduction. However, interesting differences start appearing in the average case. In contrast, the relaxation of constraints afforded by Gram or Schmidt reduction brings measurable benefits in many cases to be encountered in practice. We refer to Akhavi's Ph.D. thesis and especially to his paper [1] for a precise description of the algorithms involved. In what follows, we focus on modelling issues.

A simple and natural model of what a *random lattice* is can be described as follows: take a system of p vectors $(e_1, \ldots, e_p)$ choosen uniformly and independently inside the unit ball of $\mathbb{R}^n$ (with $n \geq p$). Let ℓ_j denote the length of the jth element of the orthogonalized version according to the classical Gram–Schmidt procedure (in the real domain). Daudé and Vallée have shown that each ℓ_j has a distribution that is asymptotically of the Beta type, with probability density proportional to $u^{n-j}(1-u^2)^{(j-1)/2}$; see [3]. A consequence of the estimates of [3] is the following upper bound for the expected number $E(K)$ of iterations of the LLL algorithm over inputs bounded from above by M,

$$E(K) \leq \frac{n^2}{\log t}\left(\frac{\log n}{2} + 3\right) + n + 3n^2 \frac{\log_t M}{M^{1/3}}.$$

(There $t \in (1,2)$ is a control parameter which influences the performance of the reduction algorithm.) This result implies an upper bound on the number of iterations of the order of $n^2 \log n$.

Akhavi improves the estimates of [3]. The noticeable fact here is the presence of *thresholds*. Consider a large dimension n together with the lengths of the ath and bth (standard Gram–Schmidt) orthogonalized vectors in $\mathbb{R}^n$. Then one has (Theorem 8 of [1]):

1. If $a = \alpha n + i$ and $b = \beta n + j$ with fixed $0 < \alpha < \beta < 1$, then the ratio ℓ_b/ℓ_a exhibits a sharp threshold: the random variable ℓ_b/ℓ_a is with high probability concentrated around its mean, namely $\theta_0 := \sqrt{1-\alpha}/\sqrt{1-\beta}$.
2. If $a = n - i$ and $b = n - j$, then the ratio ℓ_b/ℓ_a is governed by a modified Beta distribution (that admits a continuous density).

These results quantify precisely the "evolution" of the lengths of vectors during the orthogonalization process. They describe in fact two regimes, one with sharp thresholds is relative to the "initial" steps of the process while the other with continuous transitions describes what happens at the end.

Technically, the geometry of the problem leads to multidimensional integrals that one needs to estimate asymptotically. The method of choice here is the Laplace method for integrals as described for instance in [2]. The general method needs to be amended for the case at hand and Akhavi offers in [1] a valuable discussion of the asymptotics of 2-dimensional Laplace integrals when taken over polygonal domains. Naturally, the discussion bases itself on whether the maximum of the integrand lies inside, on the boundary, or outside of the integration domain. The net result is the precise quantification summarized above.

Finally, the estimates are put to use in order to analyse three reduction methods, in the sense of Siegel, Gram, and Schmidt. It turns out that, by relaxing the LLL conditions, the new reduced bases are obtained faster (see Theorem 9 of [1] for precise statements). An experimental study is conducted that supports the theoretical results. First, under the uniform model, there is little loss in the quality of the bases produced. Next the reduction of lattices associated with the "Subset

Sum" problem are considered: these are of cryptographic relevance (in connection with the Schnorr–Euchner system) and Akhavi reports computational gains by a factor in the range 2–5, while the new reduced bases obtained prove to be of a quality comparable to what classical reduction algorithms provide.

Bibliography

[1] Akhavi (Ali). – Threshold phenomena in random lattices and efficient reduction algorithms. In Nešetřil (Jaroslav) (editor), *Algorithms, ESA'99. Lecture Notes in Computer Science*, vol. 1643, pp. 476–489. – Springer, Berlin, 1999. Proceedings of the 7th Annual European Symposium, Prague, Czech Republic, July 1999.

[2] Bleistein (Norman) and Handelsman (Richard A.). – *Asymptotic expansions of integrals.* – Dover Publications Inc., New York, 1986, xvi+425p. A reprint of the second Holt, Rinehart and Winston edition, 1975.

[3] Daudé (Hervé) and Vallée (Brigitte). – An upper bound on the average number of iterations of the LLL algorithm. *Theoretical Computer Science*, vol. 123, n° 1, 1994, pp. 95–115. – Number theory, combinatorics and applications to computer science (Marseille, 1991).

[4] Lenstra (A. K.), Lenstra (H. W., Jr.), and Lovász (L.). – Factoring polynomials with rational coefficients. *Mathematische Annalen*, vol. 261, n° 4, 1982, pp. 515–534.

[5] von zur Gathen (Joachim) and Gerhard (Jürgen). – *Modern computer algebra.* – Cambridge University Press, New York, 1999, xiv+753p.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 57–63.

Available online at the URL
http://algo.inria.fr/seminars/.

Eigenring and Reducibility of Difference Equations

Raphaël Bomboy
Café Project, INRIA Sophia-Antipolis

March 6, 2000

Summary by Frédéric Chyzak and Pierre Nicodème

Abstract

The Galois theory for differential equations is now classical. We consider here a Galois theory for difference equations whose development is more recent. In analogy with the differential case, a concept of Liouvillian solutions of a difference equation is introduced, in relation to equations with solvable Galois group. In the first part of this talk, Bomboy presents the Galois theory for linear finite difference operators. Next he adapts the concept of eigenring introduced in the differential case by Singer [11] to suggest an algorithm searching for Liouvillian solutions of linear difference equations. This direct algorithm solves a subclass of the difference equations without using Petkovšek's algorithm [8].

Introduction

We review in Section 1 the basic notions of Galois theory for difference equations, following the presentation of [7]. As in the differential case, the Galois group is a linear algebraic group. In Section 2 we present the main properties of reducible and completely reducible systems, from the point of view of the structure of their associated matrices. In the differential case, a Liouvillian extension of a differential field is done by algebraic extensions and by the operations of exponentiation and integration of a function of the field. In Section 3 we define Liouvillian solutions in the difference case; these solutions are essentially interlacings of hypergeometric sequences. We describe the notion of eigenring in Section 4 and summarize relevant properties. We finish by presenting Bomboy's algorithm for searching Liouvillian solutions in Section 5, and by concluding comments.

1. Difference Galois Theory

A *difference ring* (k, ϕ) is a ring k with an automorphism ϕ. (Note that all rings considered here are rings with identity.) For example, let k be the ring $\mathbb{C}[z]$ of polynomials or the field $\mathbb{C}(z)$ of fractions, and ϕ the automorphism that substitutes $z+1$ for z. When $\phi(x) = x$ for $x \in k$, x is called a *constant* of (k, ϕ). The set $C(k)$ of constants is a subring of k.

From now on we assume that k is a field. A (scalar) difference equation has the form

$$L(y) = \phi^m(y) + a_{m-1}\phi^{m-1}(y) + \cdots + a_0 y = 0, \tag{1}$$

where the a_i's are in k and $L = \phi^m + a_{m-1}\phi^{m-1} + \cdots + a_0$ is the difference operator associated to the equation. The set of difference operators or skew polynomials in ϕ with multiplication $\phi a = \phi(a)\phi$ is a non-commutative ring $\mathcal{P}_k(\phi)$. Equation (1) can be transformed into the system $\phi(Y) = A_L Y$, where ϕ is applied componentwise to the vector Y and

$$A_L = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_0 & -a_1 & \dots & \dots & (-a_m - 1) \end{pmatrix}.$$

One sees that y is a solution of $L(y) = 0$ if and only if $\big(y, \phi(y), \dots, \phi^{m-1}(y)\big)^T$ is a solution of $\phi(Y) = A_L Y$.

More generally, we will consider systems of difference equations of the form

$$\phi(Y) = AY \tag{2}$$

for an element A of $\mathrm{GL}_n(k)$, the space of invertible matrices of dimension n over k. If R is a difference ring extension of k, a *fundamental matrix* for Equation (2) is an element $U = \big(u_{i,j}\big) \in \mathrm{GL}_n(R)$ such that $\phi(U) = AU$ where ϕ maps componentwise to matrices. A difference ring extension R of k is called a *Picard–Vessiot extension* of k for Equation (2) if R is a simple difference ring (the only ϕ-invariants ideals are (0) and R) and $R = k\big[u_{1,1}, \dots, u_{n,n}, (\det U)^{-1}\big]$ with U a fundamental matrix. The following theorem describes the structure of such extensions.

Theorem 1 ([12]). *If the set of constants $C(k)$ is algebraically closed, Picard–Vessiot extensions R of k exist and are unique up to isomorphism.*

The *Galois group* $\mathrm{Gal}(R/k)$ of R over k is the set of linear maps that are the identity on k and commute with ϕ. As in the differential case, it can be proved to have a structure of a linear algebraic group over $C(k)$. The set V of solutions of Equation (2) in R^n is an n-dimensional vector space over $C(k)$ that is invariant by $\mathrm{Gal}(R/k)$. This yields a representation of $\mathrm{Gal}(R/k)$ in $C(k)^n$.

Let $\phi(Y) = AY$ and $\phi(Y) = BY$ be two systems with A and B in $\mathrm{GL}_n(k)$ and let V_A and V_B be the corresponding solution spaces in Picard–Vessiot extensions R_A and R_B. Both systems are equivalent if there is a matrix $T \in \mathrm{GL}_n(k)$ such that $B = \phi(T)AT^{-1}$. Then, if U is a fundamental matrix of $\phi(Y) = AY$, it follows that TU is a fundamental matrix for $\phi(Y) = BY$; in this case, one can identify the rings R_A and R_B, and V_A and V_B are isomorphic as $\mathrm{Gal}(R/k)$-modules (defined as modules over the group algebra of $\mathrm{Gal}(R/k)$ with coefficients in $C(k)$). For a large class of difference fields, any system $\phi(Y) = AY$ is equivalent to the companion system of a scalar equation [7].

We conclude this section with an illustration on the ring $\mathcal{S}$ of germs of sequences over $\mathbb{C}$.

Definition 1. Consider two elements $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ of $C^{\mathbb{N}}$ (where $C \subset \mathbb{C}$ is a ring). We define the following equivalence relation: $(x_n) \equiv (y_n)$ if and only if (x_n) and (y_n) only differ by a finite number of terms. We now consider the quotient ring $\mathcal{S} = \big(C^{\mathbb{N}}/\equiv\big)$ where addition and multiplication are defined componentwise; an element of this ring is called a *germ*.

Note that this gives us a natural embedding ν of the rational function ring $\mathbb{C}(z)$ into $\mathcal{S}$, where for $F \in \mathbb{C}(z)$, $\nu(F)$ is given as the germ of any $(s_n)_{n\in\mathbb{N}}$ such that $s_n = F(n)$ for sufficiently large n.

Definition 2. The *shift* σ of $\mathcal{S}$ maps $\nu\big((x_0, \dots, x_n, \dots)\big)$ to $\nu\big((x_1, \dots, x_{n+1}, \dots)\big)$.

From now on, the ring C is an algebraically closed subfield of $\mathbb{C}$ and $k = \nu\big(C(z)\big)$.

Property 1 ([12]). *Let $C \subset \mathbb{C}$ be an algebraically closed field. There exists a Picard–Vessiot extension of the equation $\sigma(Y) = AY$ over $C(z) \subset \mathcal{S}$ that also lies in $\mathcal{S}$.*

Example. Consider $k = \nu\big(\mathbb{C}(z)\big)$ and the equation $\sigma(x) = -x$. The Picard–Vessiot extension R of k is the ring generated by k and the sequence $s = (1, -1, 1, -1, \dots)$. Note that if $t = s + (1, 1, \dots) = (2, 0, 2, \dots)$ then $t \times \sigma(t) = 0$. The Picard–Vessiot extension therefore has zero divisors and cannot be a field.

2. Reducibility

The following theorem gives a criterion of reducibility for operators.

Theorem 2 ([3]). *Consider an operator $L \in \mathcal{P}_k(\phi)$ with Picard–Vessiot extension R. The following statements are equivalent:*

1. *L is reducible (i.e., $L = L_1 L_2$ in $\mathcal{P}_k(\phi)$);*
2. *the solution space V has a strict subspace W that is stable under the action of the Galois group $G = \mathrm{Gal}(R/k)$;*
3. *the system $\phi(X) = A_L X$ is equivalent to a system with block upper triangular companion matrix.*

We also consider the class of completely reducible operators.

Definition 3. Let lclm stand for *least common left multiple.* An operator $L \in \mathcal{P}_k(\phi)$ is completely reducible if there exist $L_1, \dots, L_k$ such that $L = \mathrm{lclm}(L_1, \dots, L_k)$,

Beware that an irreducible operator L is completely reducible because $L = \mathrm{lclm}(L)$.

Property 2 ([3]). *The following statements are equivalent:*

1. *L is completely reducible;*
2. *the solution space V is expressible as a direct sum $V = V_1 \oplus \cdots \oplus V_k$ where V_i is a stable G-module for each i, and the corresponding operators are irreducible;*
3. *the system $\phi(X) = AX$ is equivalent to a system with block diagonal companion matrix where each block corresponds to an irreducible G-module.*

3. Liouvillian Solutions

We begin this section by defining *Liouvillian solutions* of an equation in terms of *interlacings* of sequences and *hypergeometric* sequences. Next we give the expected Galois-theoretic characterization of Liouvillian solutions of a difference equation, before giving another characterization in terms of interlacings of hypergeometric solutions.

Definition 4. The *interlacing* of sequences $x^1, \dots, x^l$ of $C^{\mathbb{N}}$ is the sequence $(x_0^1, x_0^2, \dots, x_0^l, x_1^1, \dots)$. This definition extends to interlacing of germs in a natural way.

Definition 5. *Hypergeometric sequences* are germs $x \in \mathcal{S}$ such that $\sigma(x) = ax$ for some $a \in k$.

Definition 6. The set $\mathcal{L}$ of *Liouvillian sequences* is the smallest subring of $\mathcal{S}$ such that:

1. constants belong to $\mathcal{L}$, where it is understood that $\gamma \in C(k)$ is identified to the germ $(\gamma, \gamma, \dots) \in \mathcal{S}$;
2. if x is hypergeometric, x belongs to $\mathcal{L}$;
3. if x is solution of $\sigma(x) = x + a$ with $a \in \mathcal{L}$, then x belongs to $\mathcal{L}$;
4. if x belongs to $\mathcal{L}$, the interlacings of x with zero germs (i.e., the interlacings of $x^1 = \cdots = x^{l-1} = 0$ and $x^l = x$) belongs to $\mathcal{L}$.

Example. Elements of k are hypergeometric, thus belong to $\mathcal{L}$; on the other hand, the germs $(2^n)_{n\in\mathbb{N}}$ and $(n!)_{n\in\mathbb{N}}$ are two examples of hypergeometric, thus Liouvillian, sequences that are not in k.

Example (Harmonic numbers). If $k = \mathbb{C}(z)$ and $x = \left(\sum_{j=1}^n 1/j\right)_{n\in\mathbb{N}}$ we have $\left(1/(n+1)\right)_{n\in\mathbb{N}} = \nu\left(1/(z+1)\right) \in k$ and $\sigma(x) = x + \left(1/(n+1)\right)_{n\in\mathbb{N}}$. The germ $\nu(x)$ thus belongs to $\mathcal{L}$.

Example. The sequence $(0, 1, 0, 1, \dots)$ is the interlacing of both constant sequences 0 and 1, and therefore belongs to $\mathcal{L}$.

The following theorem gives the expected Galois-theoretic characterization of Liouvillian sequences.

Theorem 3 ([7]). *A solution $x \in \mathcal{S}$ of Equation (1) is Liouvillian if and only if the Galois group of any Picard–Vessiot extension of this equation is solvable.*

We come to another characterization of Liouvillian sequences. Let Z be a fundamental system of $\sigma(X) = AX$. Then by iteratively applying σ to $\sigma(Z) = AZ$ we see that Z is solution of $\sigma^m(Z) = \Pi_\sigma^m Z$ where $\Pi_\sigma^m = \sigma^{m-1}(A)\dots A$. Let τ be the automorphism of $\mathbb{C}(z)$ substituting mz for z. Then $\tau \circ \sigma^m = \sigma \circ \tau$; for i from 0 to $m-1$, the ith m-section $\tau \circ \sigma^i(Z)$ of Z satisfies the equation $\sigma(O) = \left(\Pi_{\sigma,i}^m A\right) O$ in the unknown O, where $\Pi_{\sigma,i}^m A = \tau \circ \sigma^i\left(\Pi_\sigma^m A\right)$. This gives the following theorem and corollary.

Theorem 4 ([7]). *Let L be an operator of order n over k. The following statements are equivalent:*

1. *there is a Liouvillian solution for the equation $L(y) = 0$;*
2. *there exists an m less than or equal to n, such that the equation $L(y) = 0$ has a solution that is the interlacing of m hypergeometric series;*
3. *there exists an m such that, for all i between 0 and $m-1$, the equation $\sigma(y) = (\Pi_{\sigma,i}^m A_L)(y)$ has an hypergeometric solution;*
4. *there exist m and i, with $i \le m$, such that the equation $\sigma(y) = (\Pi_{\sigma,i}^m A_L)(y)$ has an hypergeometric solution.*

Corollary 1 ([7]). *Let L be an operator with coefficients in k. One can find operators $H_1, \dots, H_t$, R with coefficients in k such that*

1. *$L = RH_t \dots H_1$;*
2. *the solution space of each H_i is spanned by interlacings of hypergeometric sequences;*
3. *any Liouvillian solution of $L(y) = 0$ is a solution of $H_t \dots H_1(y) = 0$.*

4. Eigenrings and their Structure

We consider the non-commutative ring $A = \mathcal{P}_k(\sigma)$ and a difference operator $L \in A$ with Picard–Vessiot extension R. Let V be the space of solutions of L in R. We now describe isomorphisms between three classes of objects:

1. eigenrings, that are rings that essentially contain operators that follow some special commutation relation with L;
2. endomorphisms of V that commute with the Galois group $G = \mathrm{Gal}(R/k)$;
3. A-module homomorphisms of A/AL into A/AL.

Eigenring of L. Given an operator L, the elements $U + AL \in A/AL$ such that there exists $U' \in A$ satisfying $LU = U'L$ clearly form a ring. We call it the *eigenring* $E(L)$ of L. Note that $E(L)$ is never empty: $C(k)$ is always part of $E(L)$.

G-endomorphisms of the solution space V. For $P \in A$, consider the mapping η_P of R into R defined by $\eta_P(v) = P \cdot v$ for all v in R. This $C(k)$-linear mapping clearly commutes with G, since G commutes with σ. We are interested in the situation when the mapping η_P induces a linear map of $\mathrm{End}_G V$, the algebra of $C(k)$-linear mappings of V into V that commute with G. Take v in V; we have $L \cdot v = 0$. Consider $L \cdot \eta_P(v) = LP \cdot v$. This is zero if and only if $P + AL$ belongs to $E(L)$, for then there is P' such that $LP = P'L$. In this latter case, η_P induces a G-endomorphism of V.

A-linear endomorphisms of A/AL. Consider the $C(k)$-algebra $\mathrm{End}_A(A/AL)$ of A-linear endomorphisms of A/AL, and λ an element of this algebra. Recall that the module A/AL can be viewed as the A-module generated by any "generic solution" of L; the linear map λ is thus completely described by the image of the generator $1 + AL$ of A/AL. The map λ is well-defined as an A-linear map if and only if the image $\lambda(1 + AL) = U + AL$ abides by the relation

$$L(U + AL) = L\lambda(1 + AL) = \lambda\big(L(1 + AL)\big) = \lambda(0) = 0,$$

which implies that there exists U' such that $LU = U'L$; in other words, $U + AL$ is in the eigenring. The converse property is proved similarly.

With a closer look on the bijections above, one gets the following result.

Proposition 1. *The three rings $E(L)$, $\mathrm{End}_G V$, and $\mathrm{End}_A(A/AL)$ are isomorphic.*

The classical representation theory for semi-simple modules [6] applies to the study of the structure of eigenrings, yielding the following proposition and corollary.

Proposition 2 ([4]). *For an operator L with Galois group G and space of solutions V, there are ring isomorphisms between:*

1. *the eigenring $E(L)$;*
2. *the endomorphism algebra $\mathrm{End}_G V$;*
3. *the set of matrices $P \in M_n(k)$ satisfying $A_L P = \sigma(P) A_L$.*

Proposition 3 ([4]). *Let L be a completely reducible operator with solution space V. Then V is isomorphic to a direct sum $V_1^{n_1} \oplus \cdots \oplus V_l^{n_l}$ where no V_i and V_j are isomorphic for $i \neq j$; the eigenring $E(L)$ is isomorphic to the direct sum $\bigoplus_{i=1}^{l} M_{n_i}\big(C(k)\big)$.*

Corollary 2 ([4]). *Let L be a difference operator with eigenring $E(L)$. Then:*

1. *L is irreducible implies that $E(L)$ is isomorphic to $C(k)$;*
2. *L is completely reducible and $E(L)$ is isomorphic to $C(k)$ imply that L is irreducible.*

5. Algorithms

Eigenring. An algorithm to compute the eigenring of a differential operator was given by Singer [11]. A similar algorithm computes the eigenring in the difference case. The method proceeds by undetermined coefficients: an element of the eigenring of an operator L of order n is viewed as a residue U modulo L; it is thus represented by n undetermined rational function coefficients. One then performs the multiplication by L on the left, then the Euclidean division by L on the right. This yields a first-order linear difference system in the n unknowns. This system is then solved for rational function solutions by algorithms based on Abramov's algorithm [1].[1]

Linear Difference Equations of Order 2. We consider the search for Liouvillian solutions of linear difference operators in the case of order 2. As follows from the analysis in Section 3, the search for Liouvillian solutions reduces to searching for hypergeometric solutions of associated equations. Petkovšek gave an algorithm for this purpose [8], but with exponential complexity. Bomboy's algorithm proceeds by determining hypergeometric solutions from the computation of successive eigenrings, so as to derive the shape of the Galois group G little by little, while avoiding Petkovšek's algorithm as much as possible.

In order to help to solve for hypergeometric solutions, note that each non-trivial element $U + AL$ of $E(L)$ yields a right factor of L. Indeed, viewed as an element of $\mathrm{End}_G V$, it necessarily has an eigenvalue λ and a corresponding eigenvector v. The right gcd G of $U - \lambda$ and L can be expressed by a Bézout relation and satisfies $G \cdot v = 0$. It is therefore a non-constant right-hand factor of L.

[1]Note that the same idea was used in the context of symbolic summation/integration in Chyzak's work [5].

Let x be a hypergeometric solution: there exists $a \in \mathbb{C}(z)$ such that $\sigma(x) = a \cdot x$. For all g in the Galois group G we have

$$\sigma\big(g(x)\big) = g\big(\sigma(x)\big) = g(a \cdot x) = a \cdot g(x).$$

Therefore the subspace $\mathbb{C}x$ is globally invariant under the action of G. This entails that the space of hypergeometric solutions is a G-module, as is the total solution space of L. From this and Proposition 3, it follows that the eigenring is either not a semi-simple G-module, or has dimension 1, 2, or 4.

If the space of hypergeometric solutions is 2-dimensional, G is isomorphic to the group of diagonal matrices with two independent non-zero entries, and $E(L)$ has dimension 2 or 4. If there is only a 1-dimensional space of hypergeometric solutions, a classification of the algebraic subgroups of $\mathrm{GL}_2(\mathbb{C})$ then shows that G is isomorphic to the group of upper triangular matrices $\binom{a\ b}{0\ a}$; moreover, either the solution space V is semi-simple as G-module and the eigenring $E(L)$ has dimension 2, or it is not semi-simple, and in view of $E(L) \simeq \mathrm{End}_G(V)$, $E(L)$ consists of matrices that commute with all the upper triangular matrices above, and has dimension 1 or 2. If there are no hypergeometric solutions, the same classification shows that the Galois group G contains the special linear group $\mathrm{SL}_2(\mathbb{C})$ of matrices of determinant 1, and $E(L)$ has dimension 1.

If L has a Liouvillian solution, it also has a one that is either hypergeometric or the interlacing of two hypergeometric sequences. Bomboy's algorithm to decide the existence of Liouvillian solutions and compute a basis of their vector space therefore first computes the eigenring $E(L)$. If it is not trivial (i.e., does not reduce to homotheties), it provides all hypergeometric solutions, then all Liouvillian solutions; otherwise, the eigenring corresponding to the system $\Pi_\sigma^2 A_L$ is computed and:

1. if it is not trivial, we obtain an hypergeometric solution of this system, which gives a solution of L by interlacing of hypergeometric sequences;
2. otherwise, the classification of algebraic groups shows that either L has a unique hypergeometric solution, and it is necessary to search this solution by Petkovšek's algorithm, or L has no hypergeometric solutions, and therefore L provedly has no Liouvillian solution.

6. Conclusion

Finally, the authors of this summary wish to do full justice to Petkovšek, and want to emphasize that the search for Liouvillian solutions can be entirely performed by means of (variants of) algorithms by Petkovšek, and with no need of Galois theory.[2]

Indeed, Petkovšek showed in an unpublished work [9][3] how to use his algorithm for finding hypergeometric solutions [8] in a recursive fashion and in combination with reduction of order so as to produce all *Alembertian solutions* of an operator. (The class of Alembertian sequences is obtained by the same closure operations as the Liouvillian case, except for interlacings.) This algorithm corresponds to factorizations into first-order operators H_i in Corollary 1.

In fact, Petkovšek's hypergeometric algorithm extends in a simple way to an algorithm for finding the solutions of a recurrence

$$a_0(n)u_n + \cdots + a_{m-1}u_{n+m-1} + u_{n+m} = 0$$

that are interlacings of hypergeometric sequences:

1. derive a recurrence on u_n in which the index is shifted by multiples of m: since we know that the $\mathbb{C}(n)$-vector space generated by u_n is finite-dimensional with basis $(u_n, u_{n+1}, \dots, u_{n+m-1})$,

[2] This section is the result of stimulating discussions with Bruno Salvy.

[3] seemingly subsumed by [2],

the particular shifts u_n, u_{n+m}, u_{n+2m}, ... rewrite onto this basis, and a linear dependency can be found by Gaussian elimination;
2. for each i between 0 and $m-1$, derive a recurrence on $v_p^{(i)} = u_{mp+i}$ by substituting $mp+i$ for n in the obtained recurrence, and solve it for hypergeometric solutions;
3. return the interlacing of the sequences $v_p^{(0)}$, $v_p^{(1)}$, ..., $v_p^{(m-1)}$.

A variant algorithm (corresponding to Steps 1. and 2. above) is derived in [10] by a different approach.

Corollary 1, or equivalently a direct analysis mimicking that in [9], can now be used to derive an algorithm for finding all Liouvillian solutions of a recurrence. This algorithm is essentially Petkovšek's algorithm for Alembertian solutions where searches for hypergeometric solutions—and first-order right-hand factors—is replaced with searches for interlacings of hypergeometric solutions—and higher-order right-hand factors. The main difference is that reduction of order is simultaneously performed by as many independent particular solutions as the order of the interlacings, instead of by just 1.

One can thus view Bomboy's contribution as providing a variant algorithm in terms of eigenrings. A complexity of both approaches still has to be performed so as to compare them conclusively.

Bibliography

[1] Abramov (S. A.). – Rational solutions of linear difference and q-difference equations with polynomial coefficients. In Levelt (A. H. M.) (editor), *Symbolic and Algebraic Computation.* pp. 285–289. – ACM Press, New York, 1995. Proceedings of ISSAC'95, Montreal, Canada.

[2] Abramov (Sergei A.) and Petkovšek (Marko). – D'Alembertian solutions of linear differential and difference equations. In *FPSAC/SFCA'94 (Rutgers, 1994)*, pp. 169–174. – 1994. Conference proceedings. Available from `http://www.fmf.uni-lj.si/~petkovsek/papers.html`.

[3] Barkatou (M. A.). – Rational solutions of matrix difference equations: Problem of equivalence and factorisation. In Dooley (Sam) (editor), *ISSAC'99 (July 29–31, 1999).* pp. 277–282. – ACM Press, 1999. Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation.

[4] Bomboy (R.). – *Réductibilité des opérateurs aux différences finies : une approche Galois-théorique.* – Research Report n° 3735, Institut National de Recherche en Informatique et en Automatique, July 1999.

[5] Chyzak (Frédéric). – An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Mathematics*, vol. 217, n° 1-3, 2000, pp. 115–134. – Formal power series and algebraic combinatorics (Vienna, 1997).

[6] Curtis (Charles W.) and Reiner (Irving). – *Methods of representation theory. Vol. I.* – John Wiley & Sons Inc., New York, 1990, xxiv+819p. With applications to finite groups and orders, Reprint of the 1981 original, A Wiley-Interscience Publication.

[7] Hendriks (Peter A.) and Singer (Michael F.). – Solving difference equations in finite terms. *Journal of Symbolic Computation*, vol. 27, n° 3, 1999, pp. 239–259.

[8] Petkovšek (Marko). – Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, vol. 14, n° 2-3, 1992, pp. 243–264.

[9] Petkovšek (Marko). – Nested indefinite hypergeometric sums. – 1992. Unpublished article.

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

[11] Singer (Michael F.). – Testing reducibility of linear differential operators: a group-theoretic perspective. *Applicable Algebra in Engineering, Communication and Computing*, vol. 7, n° 2, 1996, pp. 77–104.

[12] van der Put (Marius) and Singer (Michael F.). – *Galois theory of difference equations.* – Springer-Verlag, Berlin, 1997, *Lecture Notes in Mathematics*, viii+180p.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 65–70.

Available online at the URL
http://algo.inria.fr/seminars/.

Difference Equations with Hypergeometric Coefficients

Manuel Bronstein
Café Project, INRIA Sophia-Antipolis

March 3, 2000

Summary by Anne Fredet

Abstract

Let k be a difference field with automorphism σ. Let b be an element of k, and L be a linear ordinary difference operator with coefficients in k. A classical problem in the theory of difference equations is to compute all the solutions in k of the equation $L(y) = b$. If C denotes a constant field and if $k = C(n)$ and $\sigma n = n + 1$ or $\sigma n = qn$, there are known algorithms (see [2] for example). Manuel Bronstein presents here a generalization to monomial extensions of $C(n)$ (see [5] for details and generalization).

1. Historical Context

The rational solutions of linear differential equations (equations of the form $\sum_{i=0}^{n} a_i y^{(i)}$) have been first studied a long time ago, for example by Beke and Schlesinger at the end of the last century. In the middle of this century, R. H. Risch gave an algorithm to compute elementary integrals (see [11, 12, 13]). In [8], M. Karr considered difference equations (equations of the form $\sum_{i=0}^{n} a_i y(x+i)$). The link between the linear differential equations and the linear difference equations is now clear, and in [1], an algorithm to compute the rational solutions of this two types of equations with coefficients in $C(x)$ is given. In [2], the author extends the previous algorithm to q-linear difference equations (equations of the form $\sum_{i=0}^{n} a_i y(q^i x)$).

Algorithms to compute the rational solutions of linear differential, difference and q-difference equations with coefficients in $C(x)$ are now available, and extensions of $C(x)$ have been considered. In [14], M. F. Singer gives an algorithm to compute the rational solutions of linear differential equations with coefficients in almost all the Liouvillian extensions of $C(x)$, i.e., the extensions built up using integral, exponential of integral, and algebraic functions. In [7], the authors improve the algorithm for the rational solutions of linear differential equations with coefficients in an exponential extension of $C(x)$. In [6], M. Bronstein adapts the algorithm given in [1] to monomial extensions, and in [5], the author uses the methods given in [2, 6] to find the solutions of linear difference equations in their coefficient field.

2. Introduction

In [6], the author introduced the splitting factorization: he decomposed a polynomial in two factors, the *normal* part where every irreducible factor is coprime with its derivative, and the *special* part where every irreducible factor divides its derivative. He then gave an algorithm to compute the normal part of the denominator of rational solutions of a linear differential equation with coefficients in a monomial extension. In [2], S. Abramov proposed an algorithm to compute a polynomial which is divisible by the denominator of any rational solution of a linear difference equation with coefficients in $C(n)$, where $\sigma n = n+1$ or $\sigma n = qn$. In [7], a method to compute the numerator of the rational solution of a linear differential equation with coefficients in an exponential extension of $C(x)$ is given. Manuel Bronstein now considers difference equations with hypergeometric terms in the coefficients (a term $h(n)$ is hypergeometric if $h(n+1)/h(n)$ is in $C(n)$). He adapts the previous methods to difference equations with coefficients in an hypergeometric extension of $C(n)$, and this gives an efficient algorithm to compute the rational solutions of such equations. Remark that an algorithm to compute the hypergeometric solutions of linear difference equation with coefficients in $C(n)$ is given in [10] and in [4] for q-hypergeometric solutions of q-difference equations.

3. Difference Equations and Hypergeometric Extensions

Let R be a commutative ring of characteristic 0. Let σ be an automorphism of R. Define

- $R_\sigma = \{ x \in R$ such that $\sigma x = x \}$ (the set of invariant elements of R);
- $R_{\sigma^\star} = \{ x \in R$ such that $\sigma^n x = x$ for some $n > 0 \}$ (the set of periodic elements);
- $R^\sigma = \{ x \in R$ such that $\sigma x = ux$ for some $u \in R^\star \}$ (the set of semi-invariant elements);
- $R^{\sigma^\star} = \{ x \in R$ such that $\sigma^n x = ux$ for some $n > 0,\ u \in R^\star \}$ (the set of semi-periodic elements).

It is clear that we have the inclusion $R_\sigma \subseteq R^\sigma \subseteq R^{\sigma^\star}$. If R is a unique factorization domain then $R^{\sigma^\star}$ is closed under taking factors, i.e., for any polynomial q in $R^{\sigma^\star}$, each factor p of q is in $R^{\sigma^\star}$. This property is false for R_σ and R^σ, as shown by the example $R = \mathbb{Q}[t]$ and $\sigma(t) = 1-t$: $\sigma(1-t) = t$ and $\sigma(t-t^2) = t - t^2$ is in R_σ (and then in R^σ and in $R^{\sigma^\star}$), whereas t and $1-t$ are in $R^{\sigma^\star}$, but neither in R^σ nor in R_σ.

3.1. Monomial extensions.

Let k be a difference field with automorphism σ. Let (K,σ) be an extension of (k,σ).

Definition 1. t in K is a monomial over k if t is transcendental over k with σt in $k[t]$.

Let σ be an automorphism of K such that $\sigma(t)$ is in $k[t]$. Then σ induces an automorphism of $k(t)$, an automorphism of $k[t]$, and thus $\sigma(t) = at + b$ for some a in $k^\star$ and b in k

Proposition 1 ([9])**.** *If for all w in $k^\star$ we have $\sigma w \neq aw + b$, then t is transcendental over k and the following equalities hold: $k(t)_\sigma = k_\sigma$ and $k[t]^\sigma = k[t]^{\sigma^\star} = k$.*

3.2. Hypergeometric extensions.

Let σ be such that $\sigma t = at$ for some $a \in k^\star$.

Proposition 2 ([9])**.** *If for all w in $k^\star$ and $n > 0$ we have $\sigma w \neq a^n w$, then t is transcendental over k and the following equalities hold: $k(t)_\sigma = k_\sigma$ and $k[t]^\sigma = k[t]^{\sigma^\star} = \{ ct^m \mid c \in k,\ m \geq 0 \}$.*

For example, in $C[n]$, let σ be such that $\sigma n = qn$ for some $q \in C^\star$. The property holds whenever q is not a root of unity. Or we can consider $C[n,t]$, with σ such that $\sigma_{|C} = id_C,\ \sigma n = n+1$ and $\sigma t = (n+1)t$; in other words t represents $n!$.

4. Dispersion

Definition 2. Let K be a field of characteristic 0. Let $\phi : K[X] \to K[X]$ be a function. Let p and q be non-zero polynomials in $K[X]$. One defines

- the spread of p and q with respect to ϕ:

$$\mathrm{Spr}_\phi(\mathrm{p},\mathrm{q}) = \{ m \geq 0 \text{ such that } p \text{ and } \phi^m q \text{ have a non trivial gcd} \}$$

– the dispersion of p and q with respect to ϕ:

$$\mathrm{Dis}_\phi(\mathrm{p},\mathrm{q}) = \begin{cases} -1 & \text{if } \mathrm{Spr}_\phi(\mathrm{p},\mathrm{q}) \text{ is empty;} \\ \max(\mathrm{Spr}(p,q)) & \text{if } \mathrm{Spr}_\phi(p,q) \text{ is a finite nonempty set;} \\ +\infty & \text{if } \mathrm{Spr}_\phi(\mathrm{p},\mathrm{q}) \text{ is an infinite set.} \end{cases}$$

These definitions are specialized to the case $p = q$: $\mathrm{Spr}_\phi(p) = \mathrm{Spr}_\phi(p,p)$ and $\mathrm{Dis}_\phi(p) = \mathrm{Dis}_\phi(p,p)$.

Examples are:
– $\mathrm{Dis}_{d/dx}\big(p(x)\big)$ is the maximum of the multiplicity of a root of p minus 1;
– $\mathrm{Spr}_{n\to n+1}\big(p(n)\big)$ is finite (and then $\mathrm{Dis}_{n\to n+1}\big(p(n)\big) < +\infty$);
– $\mathrm{Dis}_{n\to qn}(n)$ is infinite.

Let σ be an automorphism of $k[t]$ such that $\sigma k \subseteq k$. Then the dispersion $\mathrm{Dis}_\sigma(q)$ is infinite if and only if there exists p in $k[t]^{\sigma^\star}\backslash k$ such that p divides q. Also, the dispersion $\mathrm{Dis}_\sigma(h,q)$ is infinite if and only if there exists p in $k[t]^{\sigma^\star}\backslash k$ such that p divides q and $\sigma^n p$ divides h.
Example. Let $a = 2n^7 + 19n^6 + 63n^5 + 81n^4 + 27n^3$ be in $\mathbb{Q}[n]$ and ϕ be the automorphism of $\mathbb{Q}[n]$ over $\mathbb{Q}$ that maps n to $n+1$. The resultant of a and $\phi^m a$ is

$$4m^{19}(2m+5)^3(2m+1)^3(2m-1)^3(2m-5)^3(m-3)^9(m+3)^9,$$

implying that $\mathrm{Spr}_\phi(a) = \{0,3\}$ and $\mathrm{Dis}_\phi(a) = 3$

4.1. **Splitting factorization.** One now extends the splitting factorization of polynomials to difference field: let q in $k[t]$ be decomposed into two factors $q = q_\infty \overline{q}$ such that
– the gcd of q_∞ and $\overline{q}$ is equal to 1,
– for all irreducible factor p of q, p divides q_∞ if p is in $k[t]^{\sigma^\star}$,
– and for all irreducible factor p of q, p divides $\overline{q}$ if p is not in $k[t]^{\sigma^\star}$.

The polynomial q_∞ is the *infinite part* of q, and $\overline{q}$ is its *finite part.* We note that the dispersion $\mathrm{Dis}_\sigma(\overline{q})$ is finite, the dispersion $\mathrm{Dis}_\sigma(q_\infty)$ is infinite, and for all h the dispersion $\mathrm{Dis}_\sigma(h,\overline{q})$ is finite.

4.2. **σ-Orbits.** Given α and β in a field K, the problem of the orbit is to find $m \geq 0$ such that $\alpha^m = \beta$. A bound for the smallest m such that $\alpha^m = \beta$ is given in [3]. The main ideas are as follows: if there exists d such that $\alpha^d = 1$ then one can test whether $\alpha^i = \beta$ for $0 \leq i \leq d$. If it is not the case, then the orbit problem has no solution, otherwise its solutions consist of all the integers of the form $i_0 + kd_0$ where $k \geq 0$, i_0 is the smallest $i \geq 0$ such that $\alpha^i = \beta$ and d_0 is the smallest $d > 0$ such that $\alpha^d = 1$. One can now assume that α is not a root of unity, which implies that the orbit problem has at most one solution. If α is transcendental over $\mathbb{Q}$, the orbit problem has a solution if and only if β is algebraic over $\mathbb{Q}(\alpha)$. Looking at the degree at which α appears in β gives at most one candidate solution for the orbit problem. One can now assume that α is algebraic over $\mathbb{Q}$. This generalizes to find $m \geq 0$ such that $\alpha^{m,\sigma} = \alpha(\sigma\alpha)\ldots(\sigma^{m-1}\alpha) = \beta$ (see [3]).

4.3. **Computation of the dispersion.** Let $\sigma : K[X] \to K[X]$ be an automorphism such that $\sigma K \subseteq K$. Then

$$\mathrm{Spr}_\sigma\Big(\prod_i p_i^{e_i}, \prod_j q_j^{f_j}\Big) = \bigcup_{i,j} \mathrm{Spr}_\sigma(p_i,q_j) \quad \text{and} \quad \mathrm{Dis}_\sigma\Big(\prod_i p_i^{e_i}, \prod_j q_j^{f_j}\Big) = \max_{i,j} \mathrm{Dis}_\sigma(p_i,q_j).$$

The computation of the dispersion reduces to the computation of the dispersion of two irreducible polynomials.

Let p and q be irreducible polynomials. Let m be in $\mathrm{Spr}_\sigma(p,q)$. This means that the greatest common divisor of p and $\sigma^m q$ is not trivial. The polynomials being irreducible, this is equivalent to the existence of u in $K^\star$ such that $\sigma^m q = up$. This implies that $\deg p = \deg q$. One just has to consider irreducible polynomials with common degree.

Let p and q be monic irreducible polynomials of $k[t]$ with degree n: $p = t^n + \sum_{i=0}^{n-1} p_i t^i$ and $q = t^n + \sum_{i=0}^{n-1} q_i t^i$. Assume that $\sigma t = at$ for some $a \in k^\star$. Then m is in $\mathrm{Spr}_\sigma(p,q)$ implies $\alpha_i^{m,\sigma} = \beta_i$ for all i such that $p_i q_i \neq 0$, where $\beta_i = q_i/p_i$ and $\alpha_i = a^{n-i} q_i/\sigma q_i$. Therefore, if $\mathrm{Spr}_\sigma(p,q)$ is not empty then p_i and q_i vanish simultaneously. If $p = q = t$ then $\mathrm{Dis}(p,q) = +\infty$. Otherwise, this reduces to the orbit problem $\alpha^{m,\sigma} = \beta$ for α, β in $k^\star$ and m in $\mathrm{Spr}(p,q)$. Remark that if $\sigma w \neq a^d w$ for all w in $k^\star$ and $d > 0$ then $\alpha^d \neq 1$ for all $d > 0$. So, the orbit problem has at most one solution and then $\mathrm{Spr}_\sigma(p,q)$ has at most one element.

One can extend the computation of the dispersion to rational functions: let $f = p/q$ with relatively prime p and q in $C[n]$. Let $\mathrm{Dis}_\sigma(f) = \max\big(\mathrm{Dis}_\sigma(p), \mathrm{Dis}_\sigma(p,q), \mathrm{Dis}_\sigma(q,p), \mathrm{Dis}_\sigma(q)\big)$ and $\nu_\infty(f) = \deg q - \deg p$. Then $\nu_\infty(f^{m,\sigma}) = m\nu_\infty(f)$. And if f is not in C then $\mathrm{Dis}_\sigma(f^{m,\sigma}) = \mathrm{Dis}_\sigma(f) + m - 1$.

This last equality allows us to reduce orbit problems to dispersions whenever α is not constant.

5. Rational Solutions of Difference Equations

Let t be a monomial over $k = C(n)$. Let σ be such that $\sigma n = n+1$ and $\sigma t = at$ for some a in k such that $\sigma w \neq a^d w$, for all w in $k^\star$ and $d > 0$. Let $L = \sum_{i=0}^N a_i \sigma^i$ be a linear difference operator, with the a_i's in $k[t]$ and both a_0 and a_N not equal to 0. Let b be in $k[t]$. The aim of this section is to described an algorithm to find y in $k(t)$ such that $L(y) = b$ (if there exists such a y).

5.1. **Denominator of a rational solution.** The first problem is to find a bound for the finite part of any y in $k(t)$ such that $L(y) = b$. This means to compute a polynomial q in $k[t]$ such that if $L(y) = b$ then $yq = p/d_\infty$ where p is in $k[t]$ and d_∞ in $k[t]^{\sigma\star}$. We outline the ideas here, proofs and technical details are given in [5].

Let a_0 be decomposed: $a_0 = a_{0,\infty}\overline{a_0}$. Let y be in $k(t)$ such that $L(y) = b$, where $y = p/d$ and $d = d_\infty \overline{d}$. Then $\mathrm{Dis}_\sigma(\overline{d}) \leq \max\big(-1, \mathrm{Dis}_\sigma(a_N, \overline{a_0}) - N\big)$. Let $h > 0$ be an integer. One can compute an operator $L_h = b_s\sigma^{sh} + b_{s-1}\sigma^{(s-1)h} + \cdots + b_0$ such that $L_h = RL$ for some R in $k(t)[\sigma]$. It follows that $L_h(y) = Rb$ for any b in $k[t]$ and any solution y in $k(t)$ of $L(y) = b$. We get that every solution y in $k(t)$ of $L(y) = b$ satisfies an equation of the form

$$c_s\sigma^{hs}(y) + \cdots + c_1\sigma^h(y) = d_h$$

where $c_0, \ldots, c_s, d_h$ are in $k[t]$ and $c_s \neq 0$. If h was chosen such that $\mathrm{Dis}_\sigma(\overline{d}) < h$ then $\overline{d}$ divides $\gcd_{0\leq i\leq s}(\sigma^{-ih}c_i)$. This gives us a polynomial q such that if $L(y) = b$ then $qy = p/d_\infty$ with p in $k[t]$ and d_∞ in $k[t]^{\sigma\star}$

Example. Consider $y(n+2) - (n!+n)y(n+1) + n(n!-1)y(n) = 0$. If we define σ by $\sigma n = n+1$ and $\sigma t = (n+1)t$ then the associated difference operator is $\sigma^2 - (t+n)\sigma + n(t-1)$. $a_N = 1$, $a_0 = \overline{a_0} = n(t-1)$ and $\mathrm{Dis}_\sigma(a_N, \overline{a_0}) = -1$. Then $\mathrm{Dis}_\sigma(\overline{d}) \leq -1$ and $\overline{d} \in C(n)$. So, if there exists $y \in C(n)(t)$ such that $L(y) = b$ then y is in $C(n)[t, t^{-1}]$.

Remark. The same results holds for the q-difference equation: let q be transcendental over $\mathbb{Q}$. Let σ be such that $\sigma x = qx$. Consider the q-difference equation

$$q^3(qx+1)y(q^2x) - 2q^2(x+1)y(qx) + (x+q)y(x) = 0 \qquad (1)$$

We have $\overline{a_0} = x+q$, $a_2 = q^3(qx+1)$. The resultant of a_2 and $\sigma^m(\overline{a_0})$ is $q^3(q^2 - q^m)$, which implies that $\mathrm{Dis}_\sigma(a_2, \overline{a_0}) = 2$ hence that any solution of (1) has a denominator of the form $x^n\overline{d}$ where $\mathrm{Dis}_\sigma(\overline{d}) \leq 0$. Using the bound $h = 1$, we get $L_h = L$ and $\overline{d}$ divides the greatest common divisor of $\gcd_{0\leq i\leq 2}(\sigma^{-i}a_i) = \gcd\big(x+q, \sigma^{-1}(q^2(x+1)), \sigma^{-2}(q^3(qx+1))\big) = \gcd\big(x+q, q(x+q), q^2(x+q)\big) = x+q$.

Therefore, any rational solution of (1) can be written as $y = p/\big(x^n(x+q)\big)$ where $n \geq 0$ and p is in $\mathbb{Q}[x]$.

The indicial equation at $x=0$ is $qZ^2 - 2q^2Z + q^3 = 0$ (see [2]). Its only solution of the form $Z = q^n$ is for $n=1$, which implies that any rational solution of (1) can be written as $y = p/\big(x(x+q)\big)$. Replacing y by this form, we get $p(q^2x) - 2p(qx) + p(x) = 0$ (whose solution space is $\mathbb{Q}(q)$, which implies that the general rational solution of (1) is $y = C/\big(x(x+q)\big)$ for any C in $\mathbb{Q}(q)$).

5.2. **Laurent polynomial solution.** The problem of finding rational solutions y of $L(y) = b$ is reduced to finding y in $k[t, t^{-1}]$ such that $L(y) = b$, where b is in $k[t, t^{-1}]$ and $L = \sum_{i=0}^{N} a_i \sigma^i$ is a difference operator, with $a_i \in k[t]$ and non-zero a_0 and a_N. This decomposes in two steps:

1. find a bound for the degree and the order in t of y;
2. compute the coefficients of y, seen as a Laurent polynomial in t.

5.2.1. *Bound for the degree and order of a polynomial solution.* One rewrites L as $\sum_{j=\nu}^{d} t^j L_j$ where the L_j's are in $k[\sigma]$ and L_ν and L_d are not equal to zero. Let $y = y_\delta t^\delta + \cdots + y_\gamma t^\gamma$ be in $k[t, t^{-1}]$ for integers γ and δ satisfying $\gamma \geq \delta$ and such that neither y_δ nor y_γ is equal to zero. Let b be in $k[t, t^{-1}]$. If $L(y) = b$, then

1. either $\delta \geq \nu(b) - \nu$, or $L_\nu(y_\delta t^\delta) = 0$;
2. either $\gamma \leq \deg b - d$, or $L_d(y_\gamma t^\gamma) = 0$.

The problem is reduced to considering difference operators $T = \sum_{i=m}^{M} A_i \sigma^i$ with $A_i \in C[n]$ for non-zero A_m and A_M, and to searching bounds for $\gamma \in \mathbb{Z}$ such that $T(zt^\gamma) = 0$ for some z in $C(n)$. Let $e = -\nu_\infty(\sigma t/t) = \nu_\infty(a)$. There are three possibilities:

- if $e > 0$ then $(\deg_n A_m - \deg_n T)/e \leq \gamma \leq (\deg_n T - \deg_n A_M)/e$;
- if $e < 0$ then $(\deg_n T - \deg_n A_m)/e \leq \gamma \leq (\deg_n A_m - \deg_n T)/e$;
- if $e = 0$ then $\alpha = a(\infty) \in C^\star$. We decompose $A_i = a_{i,\alpha_i} n^{\alpha_i} + \cdots$. We define $Q(z) = \sum_{i | \alpha_i = \max_j(\alpha_j)} a_{i,\alpha_i} z^i$. We have $Q(\alpha^\gamma) = 0$. This problem can be solved if $\alpha^d \neq 1$ for all $d \geq 0$ (see section 4.2).

5.2.2. *Coefficients of a Laurent polynomial solution.* This is a generalization of the specialization given in [7].

We have found γ and δ such that if y is in $k[t, t^{-1}]$ with $L(y) = b$ then $\deg_t(y) \leq \gamma$ and $\mathrm{val}_t(y) \geq \delta$. Let $z = t^\delta y$. Note that $\deg_t(z) \leq \gamma - \delta = J$. One has to consider the problem $L(z) = b$ where L is in $k[t][\sigma]$ and b in $k[t]$. Let $L = \sum_{j=0}^{d} t^j L_j$ with L_j in $k[\sigma]$, and L_0, L_d not equal to zero.

- if $J = 0$ then $L(z) = \sum_{j=0}^{d} t^j (L_j z)$. But $L_j(z)$ is in k so $L(z) = b$ implies $L_j(z) = b_j$ for all j and this reduces to difference equations with coefficients in $C[n]$;
- if $J > 0$ then one decomposes $z = z_0 + t\overline{z}$ where $z_0 = z(0)$ is in $C(n)$. Then $L_0(z_0) = b_0$ and one can find z_0. So, $L(z) = (L - L_0)(z_0) + L(t\overline{z}) + L_0(z_0)$ and $L(z) = b$ implies
$$\begin{aligned} L(t\overline{z}) &= b - b_0 - (L - L_0)z_0 \\ t\tilde{L}(\overline{z}) &= t\left(\frac{b - b_0}{t}\right) - t\frac{(L - L_0)z_0}{t} \end{aligned}$$
This gives us a new difference equation with a solution $\overline{z}$ of degree strictly less than J. By induction, one can find $\overline{z}$.

Example. Consider $y(n+2) - (n! + n)y(n+1) + n(n! - 1)y(n) = 0$, which is associated to the difference operator
$$L = \sigma^2 - (t+n)\sigma + n(t-1) = t(n - \sigma) + (\sigma^2 - n\sigma - n) = tL_1 + L_0$$

Using the same notations as previously, $e = -\nu_\infty(\sigma t/t) = -\nu_\infty(n+1) = 1$ and then $y = y_0 + y_1 t$. One first considers $L_0(y_0) = \sigma^2 y_0 - n\sigma y_0 - n y_0 = 0$, and finds that $y_0 = 0$. Then:

$$L(tz_1) = (n+2)(n+1)t\sigma^2(z_1) - (n+1)(t+n)t\sigma(y_1) + n(t-1)ty_1,$$

from which follows that

$$\tilde{L}(y_1) = (n+2)(n+1)\sigma^2(y_1) - (n+1)(t+n)\sigma(y_1) + n(t-1)y_1 = 0.$$

This implies that $y_1 = c/n$. Then $y = y_1 t = (c/n)n! = c(n-1)!$.

Bibliography

[1] Abramov (S. A.). – Rational solutions of linear differential and difference equations with polynomial coefficients. *Computational Mathematics and Mathematical Physics*, vol. 29, n° 11, 1989, pp. 1611–1620, 1757. – Transl. from Akademiya Nauk SSSR. Zhurnal Vychislitelc'noĭ Matematiki i Matematicheskoĭ Fiziki.

[2] Abramov (S. A.). – Rational solutions of linear difference and q-difference equations with polynomial coefficients. In Levelt (A. H. M.) (editor), *Symbolic and Algebraic Computation*. pp. 285–289. – ACM Press, New York, 1995. Proceedings of ISSAC'95, Montreal, Canada.

[3] Abramov (Sergei) and Bronstein (Manuel). – Hypergeometric dispersion and the orbit problem. In *Proceedings of ISSAC'00, St Andrews (Scotland)*. – ACM Press, 2000.

[4] Abramov (Sergei A.), Paule (Peter), and Petkovšek (Marko). – q-hypergeometric solutions of q-difference equations. In *Proceedings of the 7th Conference on Formal Power Series and Algebraic Combinatorics (Noisy-le-Grand, 1995)*, vol. 180, pp. 3–22. – 1998.

[5] Bronstein (Manuel). – On solutions of linear ordinary difference equations in their coefficient field. *Journal of Symbolic Computation*. – To appear. Preliminary version available as INRIA Research Report n° 3797.

[6] Bronstein (Manuel). – On solutions of linear ordinary differential equations in their coefficient field. *Journal of Symbolic Computation*, vol. 13, n° 4, 1992, pp. 413–439.

[7] Bronstein (Manuel) and Fredet (Anne). – Solving linear ordinary differential equations over $C(x, \exp \int f(x)dx)$. In Dooley (Sam) (editor), *ISSAC'99 (July 29–31, 1999)*. pp. 173–179. – ACM Press, 1999. Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation.

[8] Karr (Michael). – Summation in finite terms. *Journal of the ACM*, vol. 28, n° 2, 1981, pp. 305–350.

[9] Karr (Michael). – Theory of summation in finite terms. *Journal of Symbolic Computation*, vol. 1, n° 3, 1985, pp. 303–315.

[10] Petkovšek (Marko). – Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, vol. 14, n° 2-3, 1992, pp. 243–264.

[11] Risch (R. H.). – *On the integration of elementary functions which are built up using algebraic operations*. – Report n° SP-2801/002/00, Sys. Dev. Corp., Santa Monica, CA, 1968.

[12] Risch (Robert H.). – The problem of integration in finite terms. *Transactions of the AMS*, vol. 139, 1969, pp. 167–189.

[13] Risch (Robert H.). – The solution of the problem of integration in finite terms. *Bulletin of the AMS*, vol. 76, 1970, pp. 605–608.

[14] Singer (Michael F.). – Liouvillian solutions of linear differential equations with Liouvillian coefficients. *Journal of Symbolic Computation*, vol. 11, n° 3, 1991, pp. 251–273.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 71–74.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Attribute Grammars and Automatic Complexity Analysis

Marni Mishna
Algorithms Project, INRIA Rocquencourt

June 19, 2000

Summary by Marianne Durand

Abstract

Starting from combinatorial structures, one can study some of their characteristics by means of attribute grammars [1, 2]. This leads to multivariate generating functions that permit us to study the distribution of these characteristics, part of it automatically.

1. Attribute Grammars

The grammars considered here are built from atoms, Z, Z_1,... of weight 1 and from an ϵ of weight 0. The production rules are described in terms of a few constructors: union, cartesian product, set, sequence and cycle. These constructors can take place in a labelled world (permutations) or unlabelled (trees) and they are already present in the COMBSTRUCT package. A grammar is composed of production rules of the type $T = \Phi(T_1, \ldots, T_n)$; T is said to be an ancestor of each T_i and each T_i is a descendant of T. The attributes on these grammars are values on the objects produced by the grammar, here on combinatorial structures, like for example the size or the internal path length on a binary search tree. An attribute is *synthesized* if it is a function of his descendants (size of a tree) and *inherited* if it is a function of his ancestors. An example of an inherited attribute is the depth of a tree. The depth is defined by : the depth of the root is zero and the depth of a subtree is the depth of its father plus one. An attribute is *well-defined* if there are no circular dependencies amongst the attributes, which can be checked algorithmically [5]. The attribute is *linear* if it is a linear function of the attributes of the descendants. The size of a tree is a linear attribute, but the height of a tree defined by the maximum of the height of the subtrees plus one is not.

We now consider linear synthesized and well-defined attributes. The general specification of a structure is:

$$B = \Phi_1(B_1^1, \ldots, B_{k_1}^1) \mid \ldots \mid \Phi_n(B_1^n, \ldots, B_{k_n}^n). \tag{1}$$

where Φ_i is a standard constructor, like cartesian product, set, sequence, or cycle, or a terminal. The general form of the definition of an attribute F_i then is

$$F_i(B) = \bigcup_{1 \le m \le n} \phi\left(\delta_i^m + \sum_{j,k} \alpha_{i,j}^m F_j\left(B_k^m\right)\right) + \gamma_i$$

where lower case indexed greek letters indicate integer constants, and F_j corresponds to other attributes. The letter ϕ stands for a general iterative operator coding the fact that all the subelements of the structure are considered. For example if Φ is the sequence constructor, each element of the sequence is considered recursively. The non-planar trees are defined by $T = N \cdot \mathrm{Set(T)}$ and there the internal path length is specified by $\mathrm{ipl}(T) = \phi(\mathrm{size}(T) + \mathrm{ipl}(T))$. Other examples are the area below Dyck paths, the number of cycles in a permutation or the number of parts in a partition.

All these attributes can be encoded in multivariate generating functions as follows. If the attributes are named F_i and the structure is defined as in equation (1), the generating function in an unlabelled world is

$$B(z_0, \ldots, z_k) = \sum_{b \in B} z_0^{|b|} z_1^{F_1(b)} \ldots z_k^{F_k(b)}. \tag{2}$$

Let $\mathbf{z}$ be the vector $(z_1, \ldots, z_k)$, α^m be the matrix $[\alpha^m_{i,j}]$, γ^m and δ^m be vectors, where m is an index indicating the related constructor Φ_m. We use the following notations: $\mathbf{z}^\delta = (z_1^{\delta_1}, \ldots, z_k^{\delta_k})$ and $\mathbf{z}^\alpha = \left(z_1^{\alpha_{1,1}} \ldots z_k^{\alpha_{1,k}}, \ldots, z_1^{\alpha_{1,k}} \ldots z_k^{\alpha_{k,k}}\right)$. This allows us to state the Attribute Grammars Generating Function theorem.

Theorem 1. [6] *Given the grammar specification $B = \Phi_1(B^1_1, \ldots, B^1_{k_1}) \mid \ldots \mid \Phi_n(B^n_1, \ldots, B^n_{k_n})$ where each Φ_i is a grammar constructor or a terminal and given the set of attribute productions $F_i(B) = \bigcup_{1 \le m \le n} \phi\left(\delta^m_i + \sum_{j,k} \alpha^m_{i,j} F_j\left(B^m_k\right)\right) + \gamma_i$ the multivariate generating function $B(\mathbf{z})$ satisfies*

$$B(\mathbf{z}) = \sum_m \mathbf{z}^{\gamma^m} \mathcal{G}_{\Phi_m}\left(\mathbf{z}^{\delta^m} B^m_k(\mathbf{z}^{\alpha^m})\right)$$

where $\mathcal{G}_{\Phi_m}$ is the classical generating function transformation on structures.

Proof. The proof requires a study of each constructor. We give here a simplified proof where $B = \Phi(C)$. As in equation (2) the generating function is defined by

$$B(\mathbf{z}) = \sum_{b \in B} z_1^{F_1(b)} \ldots z_k^{F_k(b)}.$$

By replacing with the definition of F_i, i.e., $F_i(B) = \phi\big(\delta_i + \sum_{j,k} \alpha_{i,j} F_j(B_k)\big) + \gamma_i$, we obtain

$$B(\mathbf{z}) = \sum_{b \in B} \prod_l z_l^{\gamma_l} \cdot \prod_{a \in b} \prod_i z_i^{\delta_i + \sum_{j=1}^k \alpha_{ij} F_j(b)}, \tag{3}$$

which simplifies into

$$B(\mathbf{z}) = \mathbf{z}^\gamma \sum_{b \in B} \mathbf{z}^\delta \prod_{a \in b} \prod_j \left(\prod_i z_i^{\alpha_{ij}}\right)^{F_j(b)}.$$

In view of $C(\mathbf{z}) = \sum_{c \in C} \prod_j z_j^{F_j(c)}$ and $B(z) = \sum_{b \in B} \prod_{a \in b} z^{|b|} = \mathcal{G}\big(B(z)\big)$, we now have the final result

$$B(\mathbf{z}) = \mathbf{z}^\gamma \mathcal{G}_\Phi\left(\mathbf{z}^\delta C(\mathbf{z}^\alpha)\right).$$

We obtain a simple formula to express the generating function of a structure given the type of its attributes.

□

2. Automatic Complexity Analysis

The idea of working on combinatorial properties is not new, it has already been exploited in ΛΥΩ [3, 7], part of which is implemented in the COMBSTRUCT package. Given a combinatorial structure and a class of algorithms based on programming primitives like sequence of programs, test on unions, partial program descent and full component iteration, ΛΥΩ returns the asymptotic value of the cost of the program on all structures of size n. It is then possible to get the average value of the cost of the considered program. The programs analysed by ΛΥΩ can be viewed as attributes

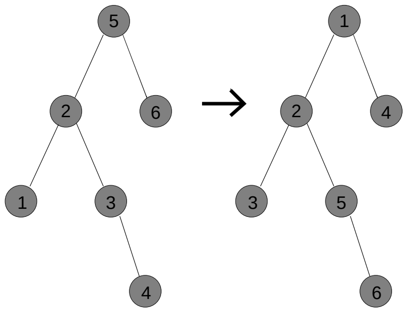


FIGURE 1. The binary search tree and increasing tree associated with [521634].

on a grammar corresponding to the structure. In fact the expressivity of $\Lambda\Upsilon\Omega$ is encompassed by the attribute grammar system. The attribute grammars are well implemented and will be in the COMBSTRUCT package soon. For example it is possible to compute the cost of differentiating a regular expression based on plus, times and exp and to get the average and the variance of this cost, which is not possible in $\Lambda\Upsilon\Omega$.

These techniques can also be applied to other constructors, if their translation into generating function is known. For instance the Quicksort algorithm can be studied using attribute grammars. The Quicksort algorithm takes as input a random permutation, chooses a pivot, sorts the elements according to their position with respect to the pivot and then sorts recursively the two subarrays. The run of the algorithm can be visualised by a binary search tree, the root being the pivot, and the two sons being the two subarrays. The complexity is the number of comparisons done, which corresponds to the internal path length of the binary search tree. This correspondance between executions of the algorithm and binary search trees is not a bijection, because the inputs 231 and 213 yield the same tree. The solution to this problem is to keep the shape of the tree and to label it with the order in which the nodes are filled, as shown on Figure 1. This gives a bijection between runs of Quicksort and increasing trees. To describe increasing trees with attribute grammars, we need to introduce the *Greene operator* also called *box operator* [4]. In a labelled structure, the Greene operator specifies where the minimum label is to be. For example the increasing trees are defined by $T = \epsilon \mid T_1 \cdot \mathrm{Min}(N) \cdot T_2$ which specifies that the minimum is in the root N. The generating function has been determined by Greene:

$$T(z) = \int_0^z T^2(x) \frac{\partial N(x)}{\partial x}\, dx.$$

It is now possible to define the internal path length as an attribute on the increasing tree structure by the relation

$$\mathrm{ipl}(T) = 0 \mid \mathrm{ipl}(T_1) + \mathrm{size}(T_1) + \mathrm{ipl}(T_2) + \mathrm{size}(T_2),$$

assuming that the internal path length of a node is 0, which is coherent with the complexity model of the number of comparisons. The multivariate generating function is

$$T(z,u) = 1 + \int_0^z \left(\frac{\partial}{\partial x} x\right) T(xu,u)^2\, dx.$$

The average is therefore

$$\frac{[z^n]T_u(z,u)|_{u=1}}{[z^n]T(z,1)} = 2H_n - 3 + \frac{H_n}{n} \quad \text{with} \quad T(z,1) = \frac{1}{1-z}$$

where H_n is the nth harmonic number.

All this work has been implemented in Maple in such a way that the syntax of attributes grammars use the same basic functions as COMBSTRUCT. For example if a grammar rule is $A = B \mid C$ then an attribute for A follows the equation, in COMBSTRUCT syntax,

```
F(A)=Union(b_1*F_1(B)+...+b_k*F_k(B),c_1*F_1(C)+...+c_k*F_k(C))
        + a_1*F_1(A)+...+a_k*F_k(A)+a_0.
```

Similar rules apply for product and set. Since COMBSTRUCT can verify if a grammar is well defined, the same algorithm can tell if an attribute grammar is linear and synthetic. For example if one looks again at the internal path length but this time of a binary Catalan tree, using two lines to define the grammar ($B = \epsilon + zB^2$) and the attribute coding internal path length ($\mathrm{ipl} = \mathrm{size}(B) + \mathrm{ipl}(B_1) + \mathrm{ipl}(B_2)$) and five to compute the generating functions and the first moments, one gets automatically that the average equals $\sqrt{\pi}n^{3/2}+O(n)$ and the variance equals $(10/3-\pi)n^3+O(n^{5/2})$. This computation can also be done on examples like the grammar defining the expressions based on zero, one, x, sum, product and exponentiation. It is possible to define the attribute coding the size of an expression after differentiation. This leads to an automatic proof that on average differentiating an expression of size n yields an expression of size $0.8\,n^{3/2}$.

Attribute grammars provide a good way of describing recursive properties of decomposable structures; a structure is decomposable if it can be expressed with basic atoms (ϵ, Z) and basic constructors (union, product, set, sequence, ...). The work that has been done on this subject can be used to obtain algorithms for random generation of structures with given attribute value, and also to obtain the distribution of the attribute. It can be continued on other attribute types for example heads or tails of sequences. From the aspect of attribute grammar research, some theory has been developed on the idea of coupling grammars. This simulates repeated application of a function. This, for example, would allow a simple analysis of repeated differentiation, and other composed functions. This requires a system where the attributes may be more than constants, but rather further structures.

Bibliography

[1] Delest (M.-P.) and Fedou (J. M.). – Attribute grammars are useful for combinatorics. *Theoretical Computer Science*, vol. 98, n° 1, 1992, pp. 65–76. – Second Workshop on Algebraic and Computer-theoretic Aspects of Formal Power Series (Paris, 1990).

[2] Dutour (I.) and Fédou (J. M.). – Object grammars and random generation. *Discrete Mathematics and Theoretical Computer Science*, vol. 2, 1998, pp. 47–61.

[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] Green (D.). – *Formal languages and their uses.* – Ph. D. Thesis, Stanford University, 1985.

[5] Knuth (D. E.). – Semantics of context-free languages. *Mathematical Systems Theory*, vol. 2, n° 2, 1968, pp. 127–145.

[6] Mishna (Marni). – *Attribute grammars and automatic complexity analysis.* – Research Report n° 4021, Institut National de Recherche en Informatique et en Automatique, October 2000. 20 pages.

[7] Zimmermann (P.). – *Series génératrices et analyse automatique d'algorithmes.* – Ph. D. Thesis, École polytechnique, Palaiseau, 1991.

Part III

Analysis of Algorithms and Data Structures

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 77–80.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Average Bit-Complexity of Euclidean Algorithms

Brigitte Vallée
GREYC, Université de Caen

May 22, 2000

Summary by Marni Mishna

Abstract

The complexity the Euclidean algorithm and its variants is well studied. This work refines the problem further by considering precise average bit-complexity. The technique is sufficiently general as to apply to a wide class of gcd-type algorithms. It determines elementary transformations for each algorithm and derives asymptotic information from their analytic behaviour. The methods rely on properties of transfer operators adapted from dynamical systems theory. The use of Ergodic Theorems in the continuous case (continued fraction algorithms) foreshadows the results, which use Tauberian Theorems as replacement. This is joint work with Ali Akhavi [1].

1. Why the Bit Case?

Since the initial average case analysis of the Euclidean algorithm in 1969 by Heilbronn and Dixon a wide variety of approaches have been used to examine variants, the most recent of which is the method of using transfer operators [3, 4].

The technique involves viewing the algorithm as a dynamical system and each iterative step as a linear fractional transformation (LFT). Previous talks by the speaker [2] shed some light on this technique, how several classes of GCD algorithms fell under a unified approach and furthermore, why they were naturally divided into two categories: slow ($\Theta(\log^2 n)$) and fast ($\Theta(\log n)$).

This same technique will now aid in the study of bit-wise complexity. The motivation for this refinement is the following. It is not a priori evident whether the properties which make the slow algorithms slow extend to the bit case. It is true that there are more iterations, but what of the size of each iteration? This work answers the question definitively, yielding the same divisions between slow and fast algorithms, however with new complexity descriptions, $\Theta(\log^3 n)$ and $\Theta(\log^2 n)$. Furthermore, it is of interest to consider a practical complexity measure. The method offers precise insights on the distribution of costs. This enables a further refinement on the classification between the fast and slow algorithms.

1.1. **Standard algorithm.** The standard Euclidean algorithm determines the gcd of v_0 and v_1 by a finite number of steps of the form $v_i = m_i v_{i-1} + v_{i+1}$, with final step $v_k = 0$. Define $l(x) = \lfloor \log_2 x \rfloor + 1$, the binary length of x. At each step there is one "naive" division with bit cost $l(v_i)l(m_i)$, and two assignment steps involving v_i and v_{i+1}. The total bit-complexity of one iteration is $l(v_i)l(m_i) + l(v_i) + l(v_{i+1})$. The cost for the standard algorithm is then

$$C(v_1, v_0) = \sum_{i=1}^{k} l(v_i) \cdot \bigl(l(m_i) + 2\bigr).$$

2. Main Result: Bit-Wise Complexity

The following two sets are valid input to the Euclidean algorithm:

$$\Omega = \{\, (u,v) \mid \gcd(u,v) = 1,\ 1 \le u < v \,\} \quad \text{and} \quad \Omega_N = \{\, (u,v) \mid (u,v) \in \Omega,\ v \le N \,\}.$$

The goal is to estimate the mean value of a cost $C : \Omega \to R$ on Ω_N. More precisely, to determine the asymptotic value as $N \to \infty$ of the mean value $E_N[C]$ satisfying $E_N[C] = C_N/|\Omega_N|$, where $C_N = \sum_{(u,v)\in\Omega_N} C(v,u)$.

The function of interest in this presentation is the bit-cost of the standard Euclidean algorithm, and consequently the cost is as defined in the previous section, but the methods are sufficiently general as to apply to a number of cases. The technique views the algorithm as a dynamical system with each iterative step a LFT. Modifying the LFT yields the variants. The continued fraction expression of the problem motivates the use of the transformations $U(x) = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$ and $M(x) = \left\lfloor \frac{1}{x} \right\rfloor$. Notice that $m_{i+1} = M\big(U^i(v_1/v_0)\big)$. The value of k in the continued fraction to the right is the depth.

$$\frac{v_1}{v_0} = \cfrac{1}{m_1 + \cfrac{1}{m_2 + \cfrac{1}{m_3 + \cdots + \cfrac{1}{m_k}}}}$$

2.1. Ergodic theory estimates.

Gauss observed that the iteration of the transformation U has invariant density $\Psi(t) = \frac{1}{\log 2}\frac{1}{1+t}$. For any $A : N \to R$ such that $\sum A(m)m^{-2} < \infty$, define $E_\infty\big[A(m)\big] = \int_0^1 A\big[m(t)\big]\Psi(t)\,dt$. This is equal to

$$E_\infty\big[A(m)\big] = \sum_{m\ge 1} A(m)\left(\log_2\left(1+\frac{1}{m}\right) - \log_2\left(1+\frac{1}{m+1}\right)\right).$$

For example, when applied to $l(m)$: $E_\infty\big[l(m)\big] = (1/\log 2)\,\log\big(\prod_{k\ge1} 1+2^{-k}\big)$.

In the continuous case, ergodic theory is applicable and gives the result that the expected value $E_N\big[\sum_{k=1}^{m} A(U^k(x))\big]$ approaches $E_\infty[A]$ almost everywhere. Although ergodic theory does not apply in the discrete case, it does give plausible estimates as to what to expect. The assignment $A(m) = l(m)$ gives the expected size of m_i in bits. The discrete version is formulated as $E_N\big[\sum_{k=1}^{p(x)} A(U^k(x))\big]$, where $p(x)$ is the depth of the necessarily finite continued fraction expansion of the rational x. In this framework one can study the aymptotic behaviour of several functions on Ω_N, such as: $\tilde{A}(x) = \sum_{k=1}^{p(x)} A\big(m_k(x)\big)$ and $\tilde{C}(x) = \sum_{k=1}^{p(x)} l\big(m_k(x)\big)\cdot\log_2 v_k(x)$.

One might anticipate that the value of $E_N\big[\tilde{A}\big]$ under certain conditions should relate to the expected depth and the expected size of an iteration. The expected depth, $\mathbf{E}[p]$, corresponds to the number of iterations of the Euclidean algorithm on input Ω_N, and is asymptotic to $6/\pi^2\log^2 N$. So, in the case of $A(m) = l(m)$,

$$E_N\big[\tilde{A}\big] \sim E_N[p] \times E_\infty\big[A(m)\big] = \left(\frac{12}{\pi^2}\log\prod_{k\ge0}\left(1+\frac{1}{2^k}\right)\right)\log_2 N.$$

This is the mean size of the continued fraction encoding of a rational number. A similar heuristic analysis of $E_N[\tilde{C}]$ shows the relation

$$E_N\big[\tilde{C}\big] \sim E_N[p]\frac{1}{2}\log_2 N \cdot E_\infty\big[l(m)\big].$$

These observations give a context for the main result.

Theorem 1. *The average bit-complexity of the standard Euclidean algorithm on the set of valid inputs of denominator less than N is asymptotically of log-squared order:*

$$E_N[C] \sim \left(\frac{6 \log 2}{\pi^2} \log \prod_{k \geq 1} \left(1 + \frac{1}{2^k}\right) \right) \log_2^2 N.$$

This agrees with the heuristic argument. Numerically, this values satisfies $E_N[C] \sim 1.24237 \log_2^2 N$.

3. Summary of Methods

The general method for obtaining this result is similar to the speaker's analysis of gcd-type functions. The average is expressible by partial sums of coefficients of Dirichlet series. Tauberian theory transfers the analytic behaviour of the series near singularities into asymptotic behaviour of coefficients. When seen as a dynamical system the generating functions of bit-cost relate to the Ruelle operators associated to the algorithm. The singularities of the Dirichlet series are related to spectral projections of the operators and are easy to describe.

3.1. **Dirichlet generating functions.** Define ω_n to be the set of all pairs (u, v) in Ω with $v = n$ and C_n as the cumulative value of C over ω_n. Then the corresponding encoding into Dirichlet generating functions is

$$F_{\langle c \rangle}(s) = \sum_{n \geq 1} \frac{C_n}{n^s} = \sum_{(v_0, v_1) \in \Omega} \frac{C(v_1, v_0)}{v_0^s}.$$

Thus the expected average cost is $E_N[C] = \left(\sum_{n \leq N} C_n\right) / \left(\sum_{n \leq N} |\omega_n|\right)$.

3.2. **Tauberian theorem.** The Tauberian theorems are a natural tool to consider as they give asymptotic information about the partial sums of coefficients of Dirichlet series. They rely on the nature and position of the singularities of $F(s) = \sum a_n n^{-s}$.

Theorem 2 (Delange). *Let $F(s)$ be a Dirichlet series with non-negative coefficients such that $F(s)$ converges for $\Re(s) > \sigma > 0$. Assume that:*

1. *F is analytic on $\Re(s) = \sigma$, where $s \neq \sigma$;*
2. *$F(s) = A(s)(s - \sigma)^{-\gamma - 1} + C(s)$ for some $\gamma \geq 0$, and $A(s)$ and $C(s)$ analytic with $A(\sigma) \neq 0$.*

Then, as $N \to \infty$, the partial sum of coefficients is

$$\sum_{n \leq N} a_n = \frac{A(\sigma)}{\sigma \Gamma(\gamma + 1)} N^\sigma \log^\gamma N \, \big(1 + \epsilon(N)\big), \quad \textit{where} \quad \epsilon(N) \to 0.$$

However, the conditions are difficult to verify for $F_{\langle c \rangle}(s)$ in its present form. A transformation gives the required information about the singularities.

3.3. **Ruelle operators.** The classical operator is

$$G_s[F](x) = \sum_{m \geq 1} \frac{1}{(m + x)^s} F\left(\frac{1}{m + x}\right).$$

Let $\mathcal{H} = \{\, h \mid h(x) = (m + x)^{-1}, m \geq 1 \,\}$, the set of inverse branches of U. If $D[h]$ is the denominator of the LFT $h(x)$, then since $D\big[h \circ g\big](x) = D[h]g(x) \cdot D[g](x)$, the iterates of G_s are given by

$$G_s^k[F](x) = \sum_{h \in \mathcal{H}} \frac{1}{D[h](x)^s} F \circ h(x).$$

Rationals of Ω can be written $x = h(0)$ for some h in $\mathcal{H}^k$ where $k \geq 0$. Then the Dirichlet generating function for $|\omega_n|$ is equal to $\sum_{n\geq 1} |\omega_n| n^{-s} = \sum_{h\in\mathcal{H}^*} D[h](0)^{-s} = (I - G_s)^{-1}[1](0)$. A cost version of $R_{s,h}[F](x) = D[h](x)^{-1} F \circ h(x)$ is defined as $R^{[c]}_{s,h}[F](x) = c(h) D[h](x)^{-1} F \circ h(x)$. Similarly the cost companion to $G_s = \sum_{h\in\mathcal{H}} R_{s,h}$ is $G^{[c]}_s = \sum_{h\in\mathcal{H}} R^{[c]}_{s,h}$.

Recall that $C(v_0, v_1) \sim \sum_{i=1} \log_2(v_i) c(m_i)$. If $x = v_1/v_0 = h_1 \circ h_2 \circ \cdots \circ h_k(0)$, then $c(m_i)$ only depends on h_i and v_i only depends on $h_{i+1} \circ \cdots \circ h_k(0)$. That is, the function can be expressed as

$$h = (h_1 \circ \cdots \circ h_{i-1}) \circ h_i \circ (h_{i+1} \circ \cdots \circ h_k) = b_i(h) \circ h_i \circ e_i(h).$$

Defining $C_{s,h} = -\sum_{i=1}^{k} \frac{\partial}{\partial s} R_{s,e_i(h)} \circ R^{[c]}_{s,h_i} \circ R_{s,b_i(h)}$ yields $F_{\langle c\rangle}(s) = \sum_{h\in\mathcal{H}^*} C_{s,h}[1](0)$.

3.4. **Functional analysis.** The singularities of the cost function can now be described in terms of the singularities of the $C_{s,h}$, and subsequently of $(I - G_s)^{-1}$. Analysis of $(I - G_s)^{-1}$ determines the values for the Tauberian theorem to be $\sigma = 2$ and $\gamma = 2$. Using this, Theorem 1 now follows. In the case of the operators related to the slow algorithms, the corresponding result is $\gamma = 3$, accounting for the log-cubed behaviour.

4. **Variants and Encoding**

As before, the technique applies to a family of variants. For example, the bit-complexity of the centred algorithm is asymptotic to

$$\frac{6\log 2}{\pi^2} \log\left(\phi^2 \prod_{k=3}^{\infty} \frac{\phi^2 + \frac{2\phi}{2^k-1}}{\phi^2 - \frac{2}{2^k-1}}\right) \log_2^2 N, \quad \text{where} \quad \phi = (\sqrt{5}+1)/2.$$

Finally, the average length of a continued fraction encoding is computable. This is the room occupied in memory by $(m_1, m_2, \ldots, m_k, v_k)$. The encoding uses the same principles as Fano–Shannon.

Theorem 3. *The average Fano–Shannon code-length D_N of the continued fraction expansion produced by the standard algorithm on valid inputs with denominator size N satisfies*

$$D_N \sim \frac{12\log^2}{\pi^2} \left(1 + \frac{2}{\log 2} \log \prod_{k=1}^{\infty} \left(1 + \frac{1}{2^k}\right)\right) \log_2 N.$$

The numerical value is $2.04868 \log_2 N$, which is close to the optimal $2\log_2 N$.

Bibliography

[1] Akhavi (A.) and Vallée (B.). – Average bit-complexity of Euclidean algorithms. In Montanari (Ugo), Rolim (José D. P.), and Welzl (Emo) (editors), *Automata, languages and programming. Lecture Notes in Computer Science*, vol. 1853, pp. 374–387. – Springer, New York, 2000. Proceedings of the 27th ICALP Conference, Geneva, Switzerland, July 2000.

[2] Salvy (B.). – *Algorithms Seminar, 1998–1999.* – Research Report n° 3830, Institut National de Recherche en Informatique et en Automatique, December 1999.

[3] Vallée (B.). – Dynamics of the binary Euclidean algorithm: functional analysis and operators. *Algorithmica*, vol. 22, n° 4, 1998, pp. 660–685. – Average-case analysis of algorithms.

[4] Vallée (Brigitte). – A unifying framework for the analysis of a class of Euclidean algorithms. In Gonnet (Gastón H.), Panario (Daniel), and Viola (Alfredo) (editors), *LATIN 2000: Theoretical Informatics. Lecture Notes in Computer Science*, vol. 1776, pp. 343–354. – Springer, Berlin, 2000. Proceedings of the 4th Latin American Symposium, Punta del Este, Uruguay, April 2000.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 81–82.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Continued Fractions, Comparison Algorithms and Fine Structure Constants

Philippe Flajolet
Algorithms Project, INRIA Rocquencourt

November 8, 1999

Summary by Cyril Banderier

Abstract

The simple problems of comparing fractions (Gosper's algorithms for continued fractions from the Hacker's Memorandum) and of deciding the orientation of triangles in computational geometry lead to a complexity analysis with an incursion into a surprising variety of domains: dynamical systems (symbolic dynamics), number theory (continued fractions), special functions (multiple zeta values), functional analysis (transfer operators), numerical analysis (series acceleration), and complex analysis (Riemann hypothesis). These domains all eventually contribute to a detailed characterization of the complexity of comparison and sorting algorithms, either on average or in probability. (Joint work with Brigitte Vallée.)

1. Introduction

To compare two rational numbers (or similarly, to determine the sign of a 2×2 determinant, a relevant problem in computational geometry) is a delicate problem when you have to work with a numerical calculator limited to a given number of digits. For example, since $\frac{312689}{99532}-\frac{833719}{265381}\approx 3\times 10^{-11}$, a computer with 10-digit accuracy cannot compare "naively" the two rational numbers.

In the "Hacker's Memorandum" [2], it is showed that it is always possible to solve this comparison problem without exceeding the accuracy of the calculator. The algorithm consists in expanding both rational numbers in continued fractions, but stopping as soon as one gets two different coefficients. This algorithm is easily generalized to any number representation system (binary, d-ary, centered or classical continued fraction, etc.) and also to the comparison of n rational numbers.

2. Results

The functions $\bar{U}(x)=\{1/x\}$ and $\hat{U}(x)=\{\{1/x\}\}$ (where $\{\{y\}\}$ stands for the distance to the nearest integer from y) are the maps of classical continued fractions and centred continued fractions respectively. Under a uniform probabilistic model (over the set of legal inputs, that is, an interval of the shape $[\,0,\alpha\,]$), the number L of iterations needed to compare two numbers satisfies $\mathbf{P}(L\geq k+1)=\sum_{|h|=k}\left|\frac{h(0)-h(\alpha)}{\alpha}\right|^2$ and the moment sums of order l satisfy $\rho^{(l)}=\sum_h\left|\frac{h(0)-h(\alpha)}{\alpha}\right|^l$. These sums are over the inverse branches of U, which appear to be linear fraction operators of a specific shape [6], thus:

Theorem 1. *The expected cost of the basic ($\bar{\rho}$) and centred ($\hat{\rho}$) comparison algorithms are expressible as sums over lattice points in $\mathbb{N}^2$*

$$\bar{\rho}^{(l)}=1+\frac{1}{2^l}+\frac{2}{\zeta(2l)}\sum_{d<c<2d}\frac{1}{c^l d^l}\quad\textit{and}\quad \hat{\rho}^{(l)}=\frac{2^l}{\zeta(2l)}\sum_{d\phi<c<d\phi^2}\frac{1}{c^l d^l}\qquad\left(\phi=\left(1+\sqrt{5}\,\right)/2\right).$$

With the help of double zeta values (also known as Euler–Zagier sums), defined as

$$\zeta^{++}(s,t)=\sum_{n=1}^{\infty}\sum_{q=1}^{n-1}\frac{1}{n^s q^t} \quad \text{and} \quad \zeta^{-+}(s,t)=\sum_{n=1}^{\infty}\sum_{q=1}^{n-1}\frac{(-1)^n}{n^s q^t},$$

it is possible to rewrite $\bar{\rho}^{(2)}$ as a peculiar value of these multiple zeta values (this implies that $\bar{\rho}^{(2)}$ can be computed to any precision in polynomial time):

Theorem 2. *The mean number $\bar{\rho}^{(2)}$ of comparisons in the classical continued fraction can be expressed in terms of double zeta values as*

$$\bar{\rho}^{(2)}=\frac{3}{4}+\frac{360}{\pi^4}\zeta^{-+}(2,2)=17-\frac{60}{\pi^4}\left(24\operatorname{Li}_4(1/2)-\pi^2(\ln 2)^2+21\zeta(3)\ln 2+(\ln 2)^4\right)=1.35113157...$$

There exists a lot of alternative expressions, due to intriguing relations between multiple zeta values, which is a topic of active research (see [1, 5, 7]), nowadays relevant to knot invariants, Feynman diagrams and even the theory of perverse sheaves. For all the other moment sums, polynomial time computations are also possible, via some nice series/integral representations [6].

For the comparison of n real numbers, the cost of sorting n numbers depends on the position of the nontrivial zeroes of the Riemann zeta function (see [3] for an approach by Dirichlet depoissonization and Mellin transform, using the Vallée secant operator or [6] for an approach by Nörlund–Rice integrals, via a complex lifting of the moment sums):

Theorem 3. *The expected cost of sorting n uniform real numbers given by their classical continued fraction representations satisfies*

$$\bar{P}(n)=n\sum_{l-1}^{n-1}(-1)^{l-1}\binom{n-1}{l}\bar{\rho}^{(l+1)}=K_0 n\ln n+K_1 n+Q(\ln n)+O(1),$$

where K_0 is Lévy's entropy constant and K_1 is a Porter-like constant (see [4]):

$$K_0=\frac{6\ln 2}{\pi^2} \quad \textit{and} \quad K_1=18\frac{\gamma\ln 2}{\pi^2}+9\frac{(\ln 2)^2}{\pi^2}-72\frac{\ln 2\,\zeta'(2)}{\pi^4}-\frac{1}{2}.$$

The function $Q(u)$ is an oscillating function with mean value 0 that satisfies $Q(n)=O(u^{\delta/2})$, where δ is any number such that $\delta>\sup\{\Re(s)\mid\zeta(s)=0\}$.

For more details, we refer to Flajolet and Vallée's articles, available on their web pages:

`http://algo.inria.fr/flajolet` and `http://www.info.unicaen.fr/~brigitte`.

Bibliography

[1] Bailey (David H.), Borwein (Jonathan M.), and Girgensohn (Roland). – Experimental evaluation of Euler sums. *Experimental Mathematics*, vol. 3, n° 1, 1994, pp. 17–30.

[2] Beeler (M.), Gosper (R. W.), and Schroeppel (R.). – *HAKMEM*. – Artificial Intelligence Memorandum n° 239, Massachusetts Institute of Technology, A. I. Laboratory, February 1972. Retyped version available from `http://www.inwap.com/pdp10/hbaker/hakmem/hakmem.html`.

[3] Clément (J.), Flajolet (P.), and Vallée (B.). – Dynamical sources in information theory: A general analysis of trie structures. *Algorithmica*. – 61 pages. To appear.

[4] Finch (S.). – Favorite mathematical constants. – `http://www.mathsoft.com/asolve/constant/constant.html`.

[5] Flajolet (Philippe) and Salvy (Bruno). – Euler sums and contour integral representations. *Experimental Mathematics*, vol. 7, n° 1, 1998, pp. 15–35.

[6] Flajolet (Philippe) and Vallée (Brigitte). – Continued fractions, comparison algorithms, and fine structure constants. In Théra (M.) (editor), *Analysis and Applications, Canadian Mathematical Society*. – To appear.

[7] Zagier (Don). – Values of zeta functions and their applications. In *First European Congress of Mathematics, Vol. II (Paris, 1992)*, pp. 497–512. – Birkhäuser, Basel, 1994.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 83–88.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Continued Fractions and Modular Forms

Ilan Vardi
IHES

April 3, 2000

Summary by Cyril Banderier

Abstract

This incursion into the realm of elementary and probabilistic number theory of continued fractions, via modular forms, allows us to study the alternating sum of coefficients of a continued fraction, thus solving the longstanding open problem of their limit law.

1. Introduction

For the readers of these proceedings,[1] it is not a secret anymore that the continued fraction expansion of p/q, the Jacobi symbol $\left(\frac{p}{q}\right)$, or the gcd of two integers (p, q), or even Gauss' lattice reduction algorithm cover phenomena of similar computational complexity. However for continued fractions, two distinct cases have to be considered: the continuous and the discrete case. The discrete case deals with continued fraction expansions of rational numbers whereas the continuous case deals with continued fractions of real (irrational) numbers.

For the continuous model, given the apparatus of ergodic theory, many basic results on continued fractions fall as application of more general theorems (see Chapter 9 in Paul Lévy's book [11]). Ergodic theory, which was guessed by Maxwell and formulated by Boltzmann, concerns itself principally with quantifying how points in a continuous space evolve under iteration of a transformation. The ergodic theorem (due to Birkhoff in 1931) states that for almost all initial points x_0 of the continuous space E with measure λ,

$$\lim_{n\to+\infty} \frac{1}{n} \sum_{j=1}^{n} f\left(T^j(x_0)\right) = \int_E f(y)\, d\lambda(y).$$

For the discrete model, there is some kind of effect that precludes the use of ergodic theory. At least, results from the continuous model may serve as a heuristic for guessing corresponding facts about the discrete world.

What one would need in order to make this heuristic rigorous is a kind of "ergodic theory with an error term." This is to some extent afforded by the introduction of Ruelle operators (see the works by Brigitte Vallée in 1995) and of modular forms (see the works by Ilan Vardi in 1987–1993) in the discrete model.

The main object of this lecture is the alternating sum of coefficients of a continued fraction. A motivation for studying this is the evaluation of Legendre symbol $\left(\frac{d}{c}\right)$, which essentially expresses whether d is a perfect square modulo c. This symbol can be evaluated using the Euclidean reduction

[1]For newcomers, I highly recommend the reading of summaries of previous talks by I. Vardi [3] and B. Vallée [1]!

$\left(\frac{d}{c}\right) = \left(\frac{d \bmod c}{c}\right)$ (where c, d are odd) and the quadratic reciprocity law

$$\left(\frac{c}{d}\right) = (-1)^{(c-1)(d-1)/4}\left(\frac{d}{c}\right).$$

In fact, it was shown by Rademacher [12] that

$$\left(\frac{d}{c}\right) = (-1)^{(3-a-d+c\sum(-1)^i a_i)/4},$$

where $\frac{d}{c} = [0, a_1, \ldots, a_r]$ (brackets stand for the expansion in continued fraction) and $0 < a, d < c$, $ad \equiv 1 \mod c$ with c and r odd. Note that $d^{-1} \mod c$ can itself be computed using the Euclidean algorithm.

There is also a geometrical motivation: the alternating sum expresses the number of times that a geodesic winds around the cusp of a modular surface.

In the continuous case, Guivarc'h and Le Jan [7] established that the average alternating sum converges to a Cauchy distribution with characteristic function $\exp\left(-\pi|t|/(2\ln 2)\right)$. For the discrete case, the stumbling block is that even the expected asymptotics estimated $\sigma(d,c) \sim \frac{12}{\pi^2}\ln c \ln\ln c$ is unproved ($\sigma(d,c)$ stands for the sum of the coefficient of the continued fraction of $\frac{d}{c}$). The problem remained open until Vardi found another approach (see [15]) via Dedekind sums.

2. Dedekind Sums

The Dedekind sum appears to be have been mistakenly defined and instead should have been defined as the alternating sum of continued fraction coefficients. Historically, the Dedekind sum is defined for relatively prime integers d and c as

$$s(d,c) = \sum_{h=1}^{c-1} ((hd/c))((h/c)),$$

with the notation $((x)) = \begin{cases} 0, & \text{if } x \text{ is an integer,} \\ x - \lfloor x \rfloor - 1/2, & \text{otherwise.} \end{cases}$

This sum was introduced by Dedekind in 1876 while editing Riemann's collected works. He used this sum to express the functional equation of the Dedekind η function

$$\eta(z) = e^{\pi i z/12}\prod_{n=1}^{\infty}(1 - e^{2\pi i n z})$$

which, he proved, satisfies

$$\ln\eta\left(\frac{az+b}{cz+d}\right) = \begin{cases} \ln\eta(z) + \frac{1}{2}\ln(cz+d) + \frac{\pi i}{12}\left(-3 + \frac{a+d}{c} - 12s(d,c)\right), & \text{for } c > 0, \\ \ln\eta(z) + \frac{\pi i b}{12}, & \text{when } c = 0, \end{cases} \tag{1}$$

where $\Im(z) > 0$, $g = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, and a, b, c, d are integers satisfying $ad - bc = 1$. Note that

$$\ln\eta(z) = \frac{\pi i z}{12} - \sum_{m,n=1}^{\infty}\frac{e^{2\pi i m n z}}{m},$$

so $\ln\eta$ is holomorphic for $\Im(z) > 0$. Using the functional equation (1) Dedekind proved a fundamental identity for Dedekind sums, namely the reciprocity law

$$s(c,d) = \frac{c}{d} + \frac{d}{c} + \frac{1}{cd} - s(d,c).$$

Note that the definition of the Dedekind sum gives that $s(d,c) = s(d \bmod c, c)$ and the reciprocity law relates the value of $s(d,c)$ to $s(c,d)$. It follows that $s(d,c)$ can be computed by using the Euclidean algorithm, so it should be expressible in terms of the continued fraction expansion of d/c. In fact, this is the statement of a result found independently by three authors in 1977:

Theorem 1 (Barkan [2], Hickerson [9], Knuth [10]). *If $[0, a_1, a_2, \ldots, a_r]$ is the regular continued fraction expansion of d/c with r odd (with $d < c$ and $0 < a < c$ such that $ad \equiv 1 \mod c$) then*

$$s(d,c) = \frac{1}{12}\Big(-3 + \frac{a+d}{c} - \sum_{i=1}^{r}(-1)^i a_i\Big).$$

It remains to find the distribution of the values of the Dedekind sums when c and d range over large intervals. Vardi did so by using Paul Lévy's theorem (details in Section 4) and for this, needed to justify several approximations. To this aim, let us recall a few facts about modular forms and Kloosterman sums, since these objects appeared to be the key to the asymptotic analysis.

3. Modular Forms

The group $\mathrm{SL}(2,\mathbb{Z})$ acts on the upper half complex plane $\mathbf{H}$ by $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) z = \frac{az+b}{cz+d}$. One now considers subgroups G of $\mathrm{SL}(2,\mathbb{Z})$ containing every matrices of $\mathrm{SL}(2,\mathbb{Z})$ congruent to the identity matrix in $\mathrm{SL}(2,\mathbb{Z}/N\mathbb{Z})$. Every such group G has a fundamental domain: an open set $D \subset \mathbf{H}$ such that for all $z \in \mathbf{H}$ there is at most one $g \in G$ with $g(z) \in D$ and at least one $g \in G$ with $g(z) \in \overline{D}$.

Definition. A *modular form* of weight k is a holomorphic function on $\mathbf{H}$ satisfying:

1) Modularity condition: $f(gz) = (cz+d)^k f(z)$ for $g \in G$,
2) Meromorphy condition: $f(z)$ is bounded in the cusps (i.e., parts of D going off to infinity).

Definition. A *non-holomorphic modular form* of *weight* r and *multiplier system* χ is a function $f(z)$ on $\mathbf{H}$ satisfying:

$$1')\ f(gz) = \chi(g)\left(\frac{cz+d}{|cz+d|}\right)^r f(z) \quad (\text{for } g \in G), \quad \text{and} \quad 2')\ \iint_D |f(x+iy)|^2 \frac{dxdy}{y^2} < \infty.$$

Condition 2') shows that the non-holomorphic modular forms form a Hilbert space $L^2(D,\chi,r)$ under the Petersson inner product

$$\langle f, g\rangle = \iint_D f(z)\overline{g(z)} y^r \frac{dxdy}{y^2}.$$

The Kloosterman sum (introduced by Kloosterman in 1927 in a refinement of the Hardy–Littlewood circle method) is defined by

$$S(m,n,c) = \sum e^{2\pi i(ma+nd)/c},$$

where the sum ranges over $d < c$ for $ad \equiv 1 \mod c$ and $\gcd(d,c) = 1$.

In 1948, André Weil proved the estimate $S(m,n,c) = O(c^{1/2+\epsilon})$. Asymptotics of sums of Kloosterman sums is a vivid subject, e.g., this "kloostermania" recently succeeded [5] to prove that there are infinitely many numbers of the form $x^2 + y^4$. Sums of Kloosterman sums exhibit strong cancellations that can be estimated by making use of modular forms.

Generalized Kloosterman sums (for some subgroup G of $\mathrm{SL}(2,\mathbb{Z})$) are defined by

$$S(m,n,c,\chi,G) = \sum \overline{\chi(g)} e^{2\pi i((m-\alpha)a+(n-\alpha)d)/(qc)},$$

where the sums ranges over $g = \left(\begin{smallmatrix} a & c \\ c & d \end{smallmatrix}\right) \in G$ with $0 < a < qc$ and $0 < d < qc$. In the sum, q is the smallest integer such that $\left(\begin{smallmatrix} 1 & q \\ 0 & 1 \end{smallmatrix}\right) \in G$ and α is defined by $e^{-2\pi i\alpha} = \chi\left(\begin{smallmatrix} 1 & q \\ 0 & 1 \end{smallmatrix}\right)$ with $0 \le \alpha < 1$.

Goldfeld and Sarnak's formulation (see [6]) of Kuznetsov's trace formula gives

$$\sum_{c<N} S(m,n,c,\chi,G) = \sum_{1/2<s_j<1} \tau_j(m,n,\chi,G)\frac{N^{2s_j-1}}{2s_j} + O(N^{\beta/3+\varepsilon}), \tag{2}$$

where β is the best constant that can be put in the estimate $S(m,n,c,\chi,G) = O(c^{\beta+\varepsilon})$, and the sum is over exceptional eigenvalues s_j (defined hereafter) of the operator $\Delta_r = y^2\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) - iry\frac{\partial}{\partial x}$. Since Δ_r has a self-adjoint extension to $L^2(D,\chi,r)$, its spectrum is discrete and real: there is a sequence of eigenvalues going to infinity, with only a finite set of negative eigenvalues which correspond to holomorphic modular forms if r is an even integer. The non-negative eigenvalues are simple except the case $\lambda = 1/4$, which could have multiplicity 2.

According to Selberg's notation, one writes an eigenvalue as $\lambda = s(1-s)$, with $\Re(s) \geq 1/2$. It follows that there is a finite number of *exceptional* eigenvalues for which $\lambda < 1/4$. An exceptional eigenvalue corresponds to $s > 1/2$, while the other eigenvalues have $\Re(s) = 1/2$ (note the analogy with the Riemann hypothesis).

For a given non-holomorphic Poincaré series P_m (see [4]), the Petersson product $\langle P_m, u_j\rangle$ gives the mth Fourier coefficient $\rho_j(m)$ of the eigenfunction u_j (which is a modular form) associated to the eigenvalue $s_j(1-s_j)$. This allows to make explicit the τ_j's of the formula (2):

$$\tau_j(m,n,\chi,G) = \frac{q^2\overline{\rho_j(m)}\rho_j(n)\left(\pi^2(m-\alpha)(n-\alpha)/q^2\right)^{1-s_j}\Gamma(s_j+r/2)\Gamma(2s_j-1)}{(-i)^r\pi\Gamma(s_j-r/2)}$$

Finally, the following theorem provides the link between a sum of generalized Kloosterman sums and a sum whose asymptotics allows the application of Lévy's theorem:

Theorem 2 (Vardi [14]).

$$e^{\pi ir/2}\sum_{\substack{0<d<c\\ \gcd(d,c)=1}} e^{2\pi irs(d,c)} = S\big(1,1,c,\chi_r,\mathrm{SL}(2,\mathbb{Z})\big),$$

where $\chi_r = e^{2\pi ir\left(\frac{a+d}{c}-3-12s(d,c)\right)}$.

4. Limiting Distribution

One says that an arithmetic function $f(n)$ has a limiting distribution $F(x)$ if

$$\lim_{N\to\infty}\frac{1}{N}\big|\{\, n<N : f(n)<x \,\}\big| = F(x).$$

In other words, one takes a histogram of values of the function $f(n)$ and looks at its shape. One method of showing that an arithmetic function has a limiting distribution is due to Paul Lévy (see examples in [13]).

Theorem 3 (Paul Lévy). *If there exists a function $g(t)$ continuous in 0 such that*

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n<N} e^{itf(n)} = g(t),$$

then $f(n)$ has a limiting distribution $F(x)$ satisfying $g(t) = \int_{-\infty}^{\infty} e^{itx}\,dF(x)$.

This is simply the probabilist's terminology for the Fourier transform; g is the *characteristic function* of the distribution.

In order to prove the limiting distribution result for Dedekind sums (and thus for alternating sums of continued fraction coefficients) one applies Lévy's theorem to $s(d,c)/\ln c$. What we want to prove is the estimate

$$\lim_{N\to\infty} \frac{\displaystyle\sum_{\substack{0<d<c<N\\ \gcd(d,c)=1}} e^{its(d,c)/\ln c}}{\left|\{\,0<d<c<N : \gcd(d,c)=1\,\}\right|} = e^{-|t|/(2\pi)}, \tag{3}$$

where the right-hand side corresponds to the characteristic function of the Cauchy distribution

$$\int_{-\infty}^{\infty} \frac{\alpha e^{ity}}{\alpha^2+y^2}\,dy = e^{-\alpha|t|},$$

with $\alpha = 1/(2\pi)$. The well-known estimate [8]

$$\left|\{\,0<d<c<N : \gcd(d,c)=1\,\}\right| = \frac{3N^2}{\pi^2} + O(N\ln N)$$

shows that the sought estimate (3) can be rewritten as

$$\sum_{\substack{0<d<c<N\\ \gcd(d,c)=1}} e^{its(d,c)/\ln c} \sim e^{-|t|/(2\pi)}\frac{3N^2}{\pi^2}.$$

Proving such a formula presents a number of technical difficulties. For example, one would like to remove the absolute values on the right hand side, and the bothersome $1/\ln c$ term in the exponential. The first point is solved since $s(c-d,c) = -s(d,c)$ so that the left-hand side is independent of the sign of t. Consequently, one may restrict attention to $t>0$. The second point is solved noting that the ln function does not vary very much, and that for most values of $c<N$, $\ln c$ is almost equal to $\ln N$. An estimate obtained by the continued fraction formula for Dedekind sums and the subsequent upper bound of $S(d/c) \le (\ln N)^{3/2+\varepsilon}$ for almost all $d<c<N$, shows that

$$\sum_{\substack{0<d<c<N\\ \gcd(d,c)=1}} e^{its(d,c)/\ln c} = \sum_{\substack{0<d<c<N\\ \gcd(d,c)=1}} e^{its(d,c)/\ln N} + O\left(N^2(\ln N)^{-1/5+\varepsilon}\right).$$

See [15] for details. The problem is therefore reduced to showing that

$$\sum_{\substack{0<d<c<N\\ \gcd(d,c)=1}} e^{its(d,c)/\ln N} \sim e^{-t/(2\pi)}\frac{3N^2}{\pi^2}, \quad t>0. \tag{4}$$

Summing the relation of Theorem 2 leads to

$$\sum_{\substack{0<d<c<N\\ \gcd(d,c)=1}} e^{2\pi i r s(d,c)} = e^{-i\pi r/2} \sum_{0<c<N} S\big(1,1,c,\chi_r,\mathrm{SL}(2,\mathbb{Z})\big),$$

so it suffices to obtain asymptotics of the last right-hand side when $r = t/(2\pi\ln N)$. This is given by the specialization of Kuznetsov's trace formula (2) with $\beta = 1$ (the trivial bound) which yields

$$\frac{(1/4)^r}{\pi A_r(1-r/2)} N^{2-r} + O(N^{4/3+\varepsilon}) = \frac{1}{\pi A_0} e^{-t/(2\pi)} N^2 + O(N^2/\ln N)$$

where $A_r = \iint_D y^r |\eta(x+iy)|^{4r} \frac{dx\,dy}{y^2}$ is the Petersson norm of $y^{r/2}\eta^{2r}(x+iy)$ (the eigenfunction of the only exceptional eigenvalue $\frac{r}{2}(1-\frac{r}{2})$).

As one easily computes $A_0 = \pi/3$, Equation (4) is satisfied.

5. Conclusion

This talk is based on [4], for more details, refer to Ilan Vardi's articles, available from

`http://www.ihes.fr/~ilan/publications.html`.

The alternating sum of coefficients of a continued fraction seems to be the first example where one needs not only upper bounds for sums of Kloosterman sums, but also their precise asymptotics.

The following fact is noteworthy. Euclidean algorithms are fundamental is several branches of science while counting amongst the oldest known algorithms. It is another testimony of the "unreasonable effectiveness of mathematics" (a phrase due to Eugene Wigner [16]) that they reveal their finest secrets only with our recent knowledges of dynamical systems and of analytical number theory. Long live applied mathematics!

Bibliography

[1] Banderier (Cyril). – Unified analysis of Euclidean algorithms [summary of a talk by Brigitte Vallée]. In Salvy (Bruno) (editor), *Algorithms Seminar, 1998–1999*, pp. 53–56. – Institut National de Recherche en Informatique et en Automatique, December 1999. Research Report n° 3830.

[2] Barkan (Philippe). – Sur les sommes de Dedekind et les fractions continues finies. *Comptes rendus hebdomadaires des Séances de l'Académie des Sciences. Séries A et B*, vol. 284, n° 16, 1977, pp. A923–A926.

[3] Flajolet (Philippe). – Continued fractions from Euclid till present [summary of a talk by Ilan Vardi]. In Salvy (Bruno) (editor), *Algorithms Seminar, 1998–1999*, pp. 89–96. – Institut National de Recherche en Informatique et en Automatique, December 1999. Research Report n° 3830.

[4] Flajolet (Philippe), Vallée (Brigitte), and Vardi (Ilan). – Continued fractions from Euclid to the present day. – In preparation.

[5] Friedlander (John) and Iwaniec (Henryk). – The polynomial X^2+Y^4 captures its primes. *Annals of Mathematics. Second Series*, vol. 148, n° 3, 1998, pp. 945–1040.

[6] Goldfeld (D.) and Sarnak (P.). – Sums of Kloosterman sums. *Inventiones Mathematicae*, vol. 71, n° 2, 1983, pp. 243–250.

[7] Guivarc'h (Y.) and Le Jan (Y.). – Asymptotic winding of the geodesic flow on modular surfaces and continued fractions. *Annales scientifiques de l'École normale supérieure. Quatrième Série*, vol. 26, n° 1, 1993, pp. 23–50.

[8] Hardy (G. H.) and Wright (E. M.). – *An introduction to the theory of numbers.* – The Clarendon Press Oxford University Press, New York, 1979, fifth edition, xvi+426p.

[9] Hickerson (Dean). – Continued fractions and density results for Dedekind sums. *Journal für die reine und angewandte Mathematik*, vol. 290, 1977, pp. 113–116.

[10] Knuth (Donald E.). – Notes on generalized Dedekind sums. *Acta Algorithmica*, vol. 33, n° 4, 1977, pp. 297–325.

[11] Lévy (P.). – *Théorie de l'addition des variables aléatoires.* – Gauthier–Villars, 1937.

[12] Rademacher (Hans). – Generalization of the reciprocity formula for Dedekind sums. *Duke Mathematical Journal*, vol. 21, 1954, pp. 391–397.

[13] Tenenbaum (Gérald). – *Introduction to analytic and probabilistic number theory.* – Cambridge University Press, Cambridge, 1995, xvi+448p. Translated from the second French edition (1995) by C. B. Thomas.

[14] Vardi (Ilan). – A relation between Dedekind sums and Kloosterman sums. *Duke Mathematical Journal*, vol. 55, n° 1, 1987, pp. 189–197.

[15] Vardi (Ilan). – Dedekind sums have a limiting distribution. *International Mathematics Research Notices*, n° 1, 1993, pp. 1–12.

[16] Wigner (Eugene P.). – The unreasonable effectiveness of mathematics in the natural sciences. *Communications in Pure and Applied Mathematics*, vol. 13, n° 1, 1960, pp. 1–14. – Also available at the URL `http://www.txwesleyan.edu/aegis/aegistwo/Unreasonable.html`.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 89–92.

Available online at the URL
http://algo.inria.fr/seminars/.

Transcendence of Numbers whose Expansion in Base b or into Continued Fractions is "Too Regular"

J.-P. Allouche
LRI, Université Paris-Sud

February 7, 2000

Summary by Philippe Flajolet

1. Normality and Transcendence

Émile Borel introduced the concept of *normal numbers*: a real is normal in base b if its expansion in this base contains each k-block a "normal" number of times, that is, with a frequency asymptotic to $1/b^k$. This concept of normality is closely related to the famous Borel–Cantelli lemma, a consequence of which is that almost all numbers (in a measure-theoretic sense) are normal [3]. Borel himself returned to the subject towards the end of his life and conducted detailed statistical studies [4] on the first two thousand digits of $\sqrt{2}$ as well as on other numbers like e or π. For instance the frequencies of appearance of 0–9 amongst the first 50 digits of the decimal representation of π,

$$\pi = 3.14159\,26535\,89793\,23846\,26433\,83279\,50288\,41971\,69399\,3751\ldots$$

are respectively 1, 5, 5, 9, 4, 5, 4, 4, 5, 8, and irregularities tend to be much smoothed out when more digits are considered. Every mathematician *believes* that numbers like $\sqrt{2}$ or π are normal in any base. However, such conjectures, tested nowadays to billions of digits, seem well beyond the reach of current mathematical knowledge.

A similar notion of normality can be defined for continued fraction expansions. Every number has a continued fraction expansion, for instance,

$$\gamma := \lim_{n\to\infty}(H_n - \log n) = \cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{2+\cfrac{1}{\ddots}}}}$$

$$= /1,1,2,1,2,1,4,3,13,5,1,1,8,1,2,4,1,1,40,1,11,3,7,1,7,1,1,5,1,49,4,1,65,\ldots/.$$

The "law of Gauss" predicts the asymptotic frequency of digit k to be $\log_2\big((k+1)^2/(k(k+2))\big)$ for a random real number, say, uniform over $(0,1)$; see [8, Sec. 4.5.3] for an agreeable introduction. Though it is observed numerically on extensive data that many classical constants like $\sqrt[3]{2}$, π, or γ obey the law of Gauss, proofs are currently not in sight. (E.g., it is not even known whether the continued fraction expansion of Euler's constant γ terminates, i.e., whether the constant γ is irrational).

Very roughly, two conjectures are believed by most to be true:

Conjecture 1. *The base b expansion of any irrational algebraic number is normal.*

Conjecture 2. *The continued fraction expansion of any algebraic irrational number that is not a quadratic number is normal. In particular the continued fraction digits of any such number should be unbounded.*

Given these conjectures, one may then expect the following: base expansions or continued fraction expansions that are in a sense "too regular" (hence fail to satisfy the strong normality condition) should determine transcendental numbers. The research described in this talk proceeds along these lines; see [1] to which we refer for an extensive bibliography.

Since transcendence of numbers is at stake it may be appropriate to start with a few basic facts; see Gel'fond's book [7] for a pleasant introduction. Liouville was the first in 1844 to observe that algebraic numbers are not well approximated by rationals: if α is algebraic of degree ν, then the inequality (a one-liner),

$$\left|\alpha - \frac{p}{q}\right| > \frac{C}{q^{\kappa}}, \qquad C > 0, \tag{1}$$

is satisfied for all integers p, q with $\kappa = \nu$. By the converse implication, a transcendence criterion results and, in particular, Liouville deduced that numbers with "very sparse" non-zero digits in some base representation, for instance,

$$\eta := \sum_{n=0}^{\infty} \frac{1}{10^{n!}},$$

must be transcendental. Thue, Siegel, and Roth in the twentieth century refined Liouville's estimate (1) by showing successively that one could take $\kappa > \frac{1}{2}n + 1$, $\kappa > 2\sqrt{n}$, and finally any $\kappa > 2$ (Roth, 1955); see the insightful description of the story in [2, Ch. 7]. Such improvements considerably enlarge the class of numbers recognized to be transcendental. For instance, the "sparse" number

$$\xi := \sum_{n=0}^{\infty} \frac{1}{10^{\lfloor \beta^n \rfloor}}, \qquad \beta > 1,$$

is now known to be transcendental (its nonzero digits are denser than those of η). These classical examples thus provide a first class of numbers with explicit base representations (but very sparse non-zero digits, though!) that are provably transcendental. They also entail that continued fraction whose digits grow "too fast" lead to transcendental numbers.

For base representations and for continued fraction expansions, transcendence thus becomes accessible to proof whenever one can derive rational approximations that are "too good". This will be the case, in connection with the results mentioned above, as soon as enough combinatorial regularities of sorts happen to be present in number representations.

2. Base Representations and Transcendence

In 1997, Ferenczi and Mauduit [5] proved the following:

Theorem 1. *Assume that the base b representation of α is for each n of the form $0.U_nV_nV_nV_n'\cdots$, where V_n' is a prefix of V_n, and the following length conditions are satisfied:*

$$|V_n| \to \infty; \quad \liminf_{n\to\infty} \frac{|V_n'|}{|V_n|} > 0; \quad \limsup_{n\to\infty} \frac{|U_n|}{|V_n|} < \infty.$$

Then, the number α is transcendental.

This theorem states that a number is transcendental if its base representation contains "near-cubes" ($V_nV_nV_n'$) that are "not too far" from the beginning and long enough (the length conditions). Roughly, such numbers turn out to be too well approximated by numbers that are "close" to b-adic rationals (i.e., rationals whose denominator is a power of b). They are proved to be transcendental by virtue of a theorem established by Ridout in 1957 (see [2, p. 68]) that constitutes a generalization of the Liouville and Roth theorems to the p-adic domain.[1] Allouche [1] noticed that the methods of [5] give a bit more. First define the *complexity* of a sequence $\{u_n\}$ of digits as the function $k \mapsto p(k)$ that counts the number of distinct blocks of length k appearing in the sequence. A normal number (in base b) certainly has $p(k) = b^k$. Thus, we might expect in view of Conjecture 1 that any number with $p(k) < b^k$ is transcendental. A step in this direction is provided by the following theorem:

Theorem 2. *Assume that $p(k)$ is for k large enough dominated by a function of the form $k + a$. Then x is either rational or transcendental.*

The proof relies on combinatorial properties of sequences of low complexity. The case is reduced by a suitable morphism[2]) to that of Sturmian sequences, that is, binary sequences such that $p(k) = k + 1$. For these a suitable version of Theorem 1 can be applied.

Extending Theorem 2 to sequences of complexity $p(k) = O(k)$ seems to be hard. Cases of special interest amongst sequences of complexity $O(k)$ are those that are determined by iteration of morphisms[3] that are "simple enough". For example:

1. the Fibonacci sequence, i.e., the fixed point of the morphism $0 \mapsto 01$, $1 \mapsto 0$ that starts as
$$0\,1\,0\,0\,1\,0\,1\,0\,0\,1\,0\,0\,1\,0\,1\,0\,0\,1;$$
2. the Thue–Morse sequence defined by the morphism $0 \mapsto 01$, $1 \mapsto 10$, that starts as
$$0\,1\,1\,0\,1\,0\,0\,1\,1\,0\,0\,1\,0\,1\,1\,0\,1\,0\,0\,1.$$

(Note: there seems to be gaps in technical results of Loxton and van der Poorten concerning the transcendence of automatic sequences.) Zamboni and Allouche proved recently:

Theorem 3. *If the binary expansion of a real number is the fixed point of a morphism that is either "primitive" (e.g., the Fibonacci sequence) or of fixed length (e.g., the Thue–Morse sequence), then this number is either rational or transcendental.*

There, the notion of primitivity is the one familiar from the theory of positive matrices and Markov chains [6].

3. Continued Fraction Expansions and Transcendence

Somewhat similar results have been established for continued fractions (abbreviated as CF) whose digits—one also says quotients—are too regular. Results here are due to Davison, Queffélec, Zamboni and Allouche. A special rôle is played in this context by quadratic irrationals whose CF expansion is eventually periodic. A theorem of Schmidt relates approximability by quadratic irrationals to transcendence. (It is in a sense the analogue of the refinements of Liouville's criterion.) Roughly, like what happens with base representations, too much combinatorial regularity is shown to imply transcendence.

---

[1]Ridout's theorem is: *If α is an algebraic number and $\epsilon > 0$ is arbitrary, then there exist only finitely many integers p, q comprised solely of a fixed set of primes such that $|\alpha - p/q| < q^{-\epsilon}$.*

[2]A morphism here is a substitution of letters by words.

[3]Note that a general sequence defined by iteration of a morphism may have complexity of the order of k^2.

We shall only quote here two typical results surveyed in [1] that are relative to CF digit sequences of complexities $(k+1)$ and $O(k)$.

Theorem 4. 1. *If the sequence of CF digits of a number α is a Sturmian sequence (i.e., a binary sequence of complexity $k+1$), then the number α is transcendental.*

2. *Let θ be irrational and let the sequence of CF digits of a number α be defined as*

$$a_n = 1 + \big(\lfloor n\theta \rfloor \bmod 2\big),$$

Then, the number α is transcendental.

Thus CF representations corresponding to digit sequences of low complexity produce transcendental numbers. This is supplemented by other results (see [1, 9]) implying for instance that the numbers (in CF representation) defined by any nontrivial rewriting of the Thue–Morse sequence is transcendental.

Bibliography

[1] Allouche (Jean-Paul). – Nouveaux résultats de transcendance de réels à développement non aléatoire. *Gazette des Mathématiciens*, n° 84, 2000, pp. 19–34.

[2] Baker (Alan). – *Transcendental number theory.* – Cambridge University Press, Cambridge, 1990, second edition, x+165p.

[3] Billingsley (Patrick). – *Probability and measure.* – John Wiley & Sons Inc., New York, 1986, second edition, xiv+622p.

[4] Borel (Émile). – Sur les chiffres décimaux de $\sqrt{2}$ et divers problèmes de probabilités en chaîne. *Comptes rendus de l'Académie des Sciences de Paris*, vol. 230, 1950, pp. 591–593.

[5] Ferenczi (Sébastien) and Mauduit (Christian). – Transcendence of numbers with a low complexity expansion. *Journal of Number Theory*, vol. 67, n° 2, 1997, pp. 146–161.

[6] Gantmacher (F. R.). – *Matrizentheorie.* – VEB Deutscher Verlag der Wissenschaften, Berlin, 1986, 654p. Translated from the Russian original (1966) by Helmut Boseck, Dietmar Soyka and Klaus Stengert.

[7] Gel'fond (A. O.). – *Transcendental and algebraic numbers.* – Dover Publications Inc., New York, 1960, vii+190p. Translated from the first Russian edition (1952) by Leo F. Boron.

[8] Knuth (Donald E.). – *The art of computer programming.* – Addison-Wesley Publishing Co., Reading, Mass., 1997, third edition, xiv+762p. Volume 2: Seminumerical algorithms.

[9] Queffélec (M.). – Transcendance des fractions continues de Thue–Morse. *Journal of Number Theory*, vol. 73, n° 2, 1998, pp. 201–211.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 93–96.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Routing Permutations on Trees

Sylvie Corteel
PRISM, Université de Versailles - Saint-Quentin-en-Yvelines

June 19, 2000

Summary by Dominique Gouyou-Beauchamps

Abstract

We study the problem of routing permutations on trees. We show that this problem is NP-hard but that it is 5/3-approximable. For a linear network or for a star tree network, the problem is polynomial and we give its average complexity. We extend these results and obtain an upper bound for arbitrary trees. This talk is based on a joint work with Mario Valencia-Pabon, Danièle Gardy, Dominique Barth, and Alain Denise [4].

1. Introduction

The routing problem on communication networks consists in the efficient allocation of resources to connection requests. In the case of all-optical networks, data is transmitted on lightwaves through optical fiber, and several signals can be transmitted through a fiber link simultaneously provided that different wavelengths are used in order to prevent interferences [3]. As the number of wavelengths is a limited resource, it is desirable to establish a given set of connection requests with a minimum number of wavelengths. Then the routing problem for all-optical networks can be viewed as a path coloring problem: it consists in finding a desirable collection of paths on the network associated with the collection of connection requests in order to minimize the number of colors needed to color these paths in such a way that any two different paths sharing a same link are assigned different colors. For simple networks, such as trees, the routing problem is simpler, as there is a unique path for each communication request.

Clearly, such a routing problem can be modeled as a permutation-path coloring problem on trees. An instance of the permutation-path coloring problem on trees is given by a directed symmetric tree graph T on n nodes and a permutation σ of the node set of T. Moreover, we associate with each pair $\big(i,\sigma(i)\big)$, $i \neq \sigma(i)$, $1 \leq i \leq n$, the unique directed path on T from node i to node $\sigma(i)$. Thus, the permutation-path coloring problem for this instance consists in assigning the minimum number of colors to such a permutation-set of paths in such a way that any two paths sharing a same arc of the tree are assigned different colors.

2. Definitions

We model the tree network as a rooted labeled symmetric directed tree $T = (V, A)$ on n vertices, where processors and switches are vertices and links are modeled by two arcs in opposite directions. Let P be a collection of directed paths on T. We assume that the vertices of T are arbitrarily labeled by different integers $\{1, 2, \ldots, n\}$ and that the vertex labeled n is the root vertex of T. We denote $i \rightsquigarrow j$ the unique directed path from vertex i to vertex j. The arc from vertex i to its father (resp.

from the father of i to i), $1 \leq i \leq n-1$, is labeled by i^+ (resp. i^-). We call $T(i)$ the subtree of T rooted at vertex i, $1 \leq i \leq n$.

For any i, $1 \leq i \leq n-1$, the load of an arc i^+ (resp. i^-) of T, denoted by $L_T(P, i^+)$ (resp. $L_T(P, i^-)$), is the number of paths in P using such an arc, and the maximum load among all arcs of T is denoted by $L_T(P)$. We call the coloring number and we denote by $R_T(P)$, the minimum number of colors needed to color the paths in P such that any two paths sharing a same arc in T are assigned different colors. Trivially, we have that $R_T(P) \geq L_T(P)$.

We say that P is a permutation-path set on T if P represents a permutation $\sigma \in S_n$ of the vertex set of T, where $\sigma(i) = j$, $i \neq j$, if and only if $i \rightsquigarrow j \in P$. In the sequel we talk indifferently of a permutation-path set P or of the permutation $\sigma \in S_n$ that P represents. Thus, given a permutation $\sigma \in S_n$ and a tree T on n vertices, the load of the arc i^+, resp. i^-, $1 \leq i \leq n-1$, can be expressed by $L_T(\sigma, i^+) = \left|\{\, j \in T(i) \mid \sigma(j) \notin T(i) \,\}\right|$, resp. $L_T(\sigma, i^-) = \left|\{\, j \notin T(i) \mid \sigma(j) \in T(i) \,\}\right|$.

Let T be a tree on n vertices. The average load of all permutations $\sigma \in S_n$ on T, denoted by $\bar{L}_T$, is defined as $\bar{L}_T = (n!)^{-1} \sum_{\sigma \in S_n} L_T(\sigma)$.

Proposition 1 ([7])**.** *There is a polynomial time algorithm to color any collection P of paths on any tree such that $L_T(P) \leq R_T(P) \leq \lceil (5/3) L_T(P) \rceil$.*

Let T be a tree on n vertices. We denote by $\bar{R}_T$ the average number of colors needed to color all permutations in S_n on T.

Proposition 2. *Let T be a tree on n vertices. Then $\bar{L}_T(P) \leq \bar{R}_T(P) \leq (5/3)\bar{L}_T(P) + 1$.*

Let T be a tree on $2n$ vertices. We denote by $\tilde{R}_T$ the average number of colors needed to color all involutions in I_{2n} on T.

Proposition 3. *Let T be a tree on $2n$ vertices and let $\tilde{L}_T$ be the average load of all involutions in I_{2n} on T. Then $\tilde{L}_T \leq \tilde{R}_T \leq (3/2)\tilde{L}_T$.*

3. Complexity of Computing the Coloring Number

We show the NP-hardness of the symmetric-path coloring problem on binary trees, answering an open question in [2]. For this, we use a reduction similar to the one used in [6, 10] for proving the NP-hardness of the general path coloring problem on binary trees. We extend this reduction to obtain NP-hardness results on very restrictive instances like involutions on both binary trees and trees having only two vertices with degrees greater than two.

Theorem 1. *Let T be a directed symmetric tree and let P be a collection of directed paths on T. Then, computing $R_T(P)$ is NP-hard in the following cases:*

- *T is a binary tree and P is a collection of symmetric paths on T.*
- *T is a binary tree and P represents an involution of the vertices of T.*
- *T is a tree with maximum degree greater or equal to 4, and P represents a circular permutation of the vertices of T.*
- *T is a tree having only two degrees greater than two and P represents an involution of the vertices of T.*

4. A Lower Bound for the Average Coloring Number

Let $G = (V, A)$ be a directed symmetric graph on n vertices and r a routing function in G which assigns a set of paths on G to route any permutation $\sigma \in S_n$. Let $\bar{L}_{G,r}$ be the average load of all permutations in S_n induced by the routing function r, and let $U \subseteq V$ be a subset of the vertex set of G. We denote by $c(U)$ the cut $(U, \bar{U})$, i.e., the set of arcs $\{\, (u,v) \in A \mid u \in U,\ v \in V \setminus U \,\}$.

Proposition 4. *For any graph $G=(V,A)$ on n vertices, and any routing function r in G,*

$$\bar{L}_{G,r} \geq \frac{1}{n} \max_{U \subseteq V} \left(\frac{|U|(n-|U|)}{|c(U)|} \right).$$

Let T be a tree on n vertices. By the previous proposition, we can deduce that the average load of any arc i^+ of T, $1 \leq i \leq n-1$, denoted by $\bar{L}_T(i)$, satisfies $\bar{L}_T(i) = |T(i)|\bigl(n-|T(i)|\bigr)/n$. Moreover, for any vertex i of T, let $v_T(i) = |T(i)|/n$ and $\tilde{v}_T(i) = \min\bigl(v_T(i), 1-v_T(i)\bigr)$. Let $\tilde{v}_T = \max_i \tilde{v}_T(i)$.

Proposition 5. *Both inequalities $\bar{L}_T \geq n\tilde{v}_T(1-\tilde{v}_T)$ and $\bar{R}_T \geq n\tilde{v}_T(1-\tilde{v}_T)$ hold.*

5. Average Coloring Number on Linear Networks

The main result is the following:

Theorem 2. *The average coloring number of the permutations in S_n to be routed on a linear network on n vertices is $n/4 + (\lambda/2)n^{1/3} + O(n^{1/6})$ where $\lambda = 0.99615\ldots$*

To prove this result, we use enumerative and asymptotic combinatorial techniques (Theorems 3 and 4 below and results of Louchard [12] and Daniels and Skyrme [5]). Our approach uses the same methodology as Lagarias et al. [11] who studied involutions with no fixed point routed on the linear network.

Let W_n be the set of Motzkin walks of length n labeled as follows:

- each South-East step of height i is labeled by an integer between 1 and $(i+1)^2$,
- each East step of height i is labeled by an integer between 1 and $2i+1$.

Theorem 3. [9] *There is a one-to-one correspondence between the elements W_n and those of S_n.*

We use Biane's bijection [1] because it preserves the height of our objects, i.e., the height of a labeled Motzkin walks is equal to the height of the corresponding permutation. Moreover, the height of a permutation is equal to its load.

Let $S_{n,\leq k}$ be the number of permutations in S_n of height at most k and let $S_{n,k}$ be the number of permutations in S_n of height exactly k.

Theorem 4. [8, 13] *We have the identities* $H_k(z) = \sum_{n \geq 0} \sum_{\sigma \in S_{n,k}} z^n = \dfrac{(k!)^2 z^{2k}}{P^\star_{k+1}(z) P^\star_k(z)}$ *and*

$$H_{\leq k}(z) = \sum_{n \geq 0} \sum_{\sigma \in S_{n,\leq k}} z^n = \cfrac{1}{1 - \cfrac{z^2}{1-3z - \cfrac{4z^2}{1-5z - \cfrac{\vdots}{1-(2k-1)z - \cfrac{k^2 z^2}{1-(2k+1)z}}}}},$$

with $P_0(z) = 1$, $P_1(z) = z-1$ and $P_{n+1}(z) = (z-2n-1)P_n(z) - n^2 P_{n-1}(z)$ for $n \geq 1$, where $P^\star$ is the reciprocal polynomial of P, that is $P^\star_n(z) = z^n P_n(1/z)$ for $n \geq 0$.

6. Average Coloring Number on Arbitrary Tree Networks

We can extend the average complexity results on linear networks to arbitrary tree networks.

Theorem 5. *The average load induced by all permutations of S_n on T is $\bar{L}_T = n\tilde{v}_T(1-\tilde{v}_T) + O(n^{1/2})$.*

Theorem 6. *For all ϵ, there exists $n_0 = n_0(\epsilon)$ such that, for all $n \geq n_0$ and any tree T on n vertices, the average number of colors $\bar{R}_T$ needed to color any permutation $\sigma \in S_n$ on T satisfies $\bar{R}_T \leq (5/3 + \epsilon) n \tilde{v}_T (1 - \tilde{v}_T)$.*

Let $ST(n)$ denote the directed symmetric star graph on n vertices (i.e., the tree having only one internal vertex connected to $n-1$ leaves). We call generalized star graph that we denote by $GST(\lambda)$, a directed symmetric tree on n vertices having k branches connected to each other by one vertex, where $\lambda = (\lambda_1, \ldots, \lambda_k)$ is a partition of the integer $n-1$ into k parts ($k > 2$) and where λ_i denotes the length of the ith branch (i.e., a branch of length λ_i is a path graph on $\lambda_i + 1$ vertices). We can also obtain the same type of results for generalized star trees and involutions instead of permutations.

Theorem 7. *Let k be a fixed integer greater than 2. The average number of colors needed to color any permutation $\sigma \in S_{nk+1}$ on a generalized star tree $GST(\lambda)$ having $nk+1$ vertices and k branches of length n is $n(k-1)/k + O(n^{1/2})$.*

Theorem 8. *Let T be a tree on $2n$ vertices. The average load induced by all involutions with no fixed points $\sigma \in I_{2n}$ on T is $\bar{L}_T = 2n\tilde{v}_T(1 - \tilde{v}_T) + O(n^{1/2})$.*

Bibliography

[1] Biane (Philippe). – Permutations suivant le type d'excédance et le nombre d'inversions et interprétation combinatoire d'une fraction continue de Heine. *European Journal of Combinatorics*, vol. 14, n° 4, 1993, pp. 277–284.

[2] Caragiannis (I.), Kaklamanis (C.), and Persiano (P.). – Wavelength routing of symmetric communication requests in directed fiber trees. In *Proceedings of SIROCCO'98*, pp. 221–222. – 1998.

[3] Cheung (N. K.), Nosu (K.), and Winzer (G.) (editors). – *Dense wavelength division multiplexing techniques for high capacity and multiple access communication systems.* – vol. 8, August 1990. Special issue of *IEEE Journal on Selected Areas in Communications.*

[4] Corteel (S.), Valencia-Pabon (M.), Gardy (D.), Barth (D.), and Denise (A.). – *The permutation-path coloring problem on trees.* – Rapport de recherche du LRI n° 1256, Université de Paris Sud, 2000.

[5] Daniels (H. E.) and Skyrme (T. H. R.). – The maximum of a random walk whose mean path has a maximum. *Advances in Applied Probability*, vol. 17, n° 1, 1985, pp. 85–99.

[6] Erlebach (T.) and Jansen (K.). – Call scheduling in trees, rings and meshes. In *Proceedings of the 30th Hawaii International Conference on System Sciences HICSS-30.* – IEEE CS Press, 1997.

[7] Erlebach (Thomas), Jansen (Klaus), Kaklamanis (Christos), Mihail (Milena), and Persiano (Pino). – Optimal wavelength routing on directed fiber trees. *Theoretical Computer Science*, vol. 221, n° 1-2, 1999, pp. 119–137. – Proceedings of ICALP '97 (Bologna).

[8] Flajolet (P.). – Combinatorial aspects of continued fractions. *Discrete Mathematics*, vol. 32, n° 2, 1980, pp. 125–161.

[9] Françon (Jean) and Viennot (Gérard). – Permutations selon leurs pics, creux, doubles montées et double descentes, nombres d'Euler et nombres de Genocchi. *Discrete Mathematics*, vol. 28, n° 1, 1979, pp. 21–35.

[10] Kumar (S. Ravi), Panigrahy (Rina), Russell (Alexander), and Sundaram (Ravi). – A note on optical routing on trees. *Information Processing Letters*, vol. 62, n° 6, 1997, pp. 295–300.

[11] Lagarias (J. C.), Odlyzko (A. M.), and Zagier (D. B.). – On the capacity of disjointly shared networks. *Computer Networks and ISDN Systems*, vol. 10, n° 5, 1985, pp. 275–285.

[12] Louchard (G.). – Random walks, Gaussian processes and list structures. *Theoretical Computer Science*, vol. 53, n° 1, 1987, pp. 99–124.

[13] Viennot (Gérard). – A combinatorial theory for general orthogonal polynomials with extensions and applications. In *Orthogonal polynomials and applications (Bar-le-Duc, 1984)*, pp. 139–157. – Springer, Berlin, 1985. n° 1171 in Lecture Notes in Mathematics.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 97–107.

Available online at the URL
http://algo.inria.fr/seminars/.

Synchronous Decision Diagrams: a Data Structure for Representing Finite Sequential Digital Functions

Jean Vuillemin

DMI, École normale supérieure

May 22, 2000

Summary by Philippe Dumas and Philippe Flajolet

Abstract

Binary Diagrams (BDD's) are an important way to represent boolean functions, that is, combinational circuits. Vuillemin proposes Synchronous Decision Diagrams (SDD's) that are capable of representing all causal circuits with finite memory. The framework provides a general basis for the analysis and synthesis of digital circuits. On the mathematical side, it provides unexpected connections between hardware design and the classical notion of automatic sequences in number theory.

Researchers working in circuit theory are concerned with design (given a function, how can it be realized efficiently?) and analysis (what is the function computed by a given circuit?). This talk presents a mathematical framework for the design and analysis of boolean circuits, either combinational (i.e., without memory) or sequential (i.e., with memory). It is superbly elegant as well as conceptually simple. We shall start here with a review of Binary Decision Diagrams (BDD's) that constitute a canonical way to represent boolean functions and serve the purpose of a gentle introduction to the subject. Then, we shall proceed with Synchronous Decision Diagrams (SDD's) that can represent any type of circuit likely to be encountered in practice (i.e., circuits with finite memory of the past whose output does not depend on the future). Due to severe time constraints imposed by the editor of the seminar proceedings,[1] the authors of this summary regret that they cannot do full justice to the work presented and refer to the paper [8] for an introduction to the main ideas.

1. Binary Decision Diagrams

Let $\mathcal{B}$ be the boolean domain $\mathcal{B} = \{0, 1\}$. A boolean function of n variables is a function from $\mathcal{B}^n$ into $\mathcal{B}$. Such a function may be specified by its truth table that is the sequence of its values on its 2^n possible inputs. Let Φ_n be the set of n-ary functions and ϕ_n the corresponding cardinality. Clearly, one has $\phi_n = 2^{2^n}$, hence the identity

$$\phi_{n+1} = 2^{2^{n+1}} \equiv \left(2^{2^n}\right)^2 = (\phi_n)^2.$$

This trivial identity suggests the existence of a fundamental isomorphism

$$\Phi_{n+1} \simeq \Phi_n \times \Phi_n.$$

[1] *Editor's Note.* I acknowledge the promptitude of the authors of the summary. Especially their promptitude to renegociate deadlines.

Indeed any boolean function $f(x_1, \ldots, x_n, x_{n+1})$ with $f \in \Phi_{n+1}$ can be specified by a pair (f_0, f_1), where $f_0 \in \Phi_n$ and $f_1 \in \Phi_n$ are "specializations" of f,

$$f_0(x_1, \ldots, x_n) := f(x_1, \ldots, x_n, 0), \qquad f_1(x_1, \ldots, x_n) := f(x_1, \ldots, x_n, 1).$$

Consequently, when the decomposition is iterated, any boolean function of n variables becomes representable as a perfect binary tree, the *binary decision tree* $\mathrm{bdt}(f)$, whose height is n, whose internal nodes correspond to partial specializations of f, and whose external nodes are either the constant function 0 or the constant function 1. Observe that reading the $\mathcal{B}$ labels of the extrenal nodes of $\mathrm{bdt}(f)$ from left to right produces precisely the truth table of f.

The *binary decision diagram*[2] of f, $\mathrm{bdd}(f)$, is then nothing but the directed acyclic graph (dag) representation of this tree obtained in the usual way by sharing repeated subtrees and representing them only once. It is classically known that such a dag representation of a tree of size N can always be constructed in time $O(N)$; see for instance [5] for a discussion. Here, one has $N = 2^n$ for functions in Φ_n, so that the sharing algorithm approach is of exponential time complexity when f is given by its truth table or, equivalently, by its tree $\mathrm{bdt}(f)$. In many cases, fortunately, one can operate with polynomial time complexity.

Here is an example. Consider the adder function on three variables,

$$f(a, b, c) = a \oplus b \oplus c.$$

We purposely refrain from operating with the truth-table description of f in order to emphasize that BDD's are directly accessible via a symbolic calculus on boolean functions. Here, two "sub-functions" are first obtained upon setting the variable c to either 0 or 1:

$$f_0(a, b) = a \oplus b, \qquad f_1(a, b) = a \oplus b \oplus 1.$$

Next, specialize b, which yields here *only two* (and not four!) distinct functions, namely,

$$f_{00}(a) = a, \qquad f_{10}(a) = a \oplus 1 \qquad (\text{with } f_{01} \equiv f_{10} \text{ and } f_{11} \equiv f_{00}).$$

Finally, specialize a, which eventually leads to a reduction to the two constant functions

$$f_{000}() = 0, \qquad f_{010}() = 1 \qquad (\text{with } f_{100} \equiv f_{010} \text{ and } f_{110} \equiv f_{000}).$$

This example shows, more generally, that the BDD of the n-fold adder $f(x_1, \ldots, x_n)$ can be determined in time linear in n via basic boolean algebra alone, this despite the fact that the truth table has size 2^n. The construction in the case of the function $f(a, b, c) = a \oplus b \oplus c$ is described in Figure 1.

Bryant has invented the BDD concept in 1986 (see [1, 2]). The BDD of an n-ary function can often be computed in time much less than $O(2^n)$ (cf. the adder example), since it captures the regularities that are likely to be present in most functions occurring in practice.[3] Also, given the BDD's of f and g it is possible, in low polynomial time, to determine BDD's for various compositions of f and g like $f \oplus g$, $f \circ g$, etc. Finally, once an ordering on variables has been fixed,[4] the BDD

[2] The BDD's described here are sometimes called *OBDD's*, where the 'O' stands for "ordered" and refers to a fixed ordering on boolean variables.

[3] In the worst case, a BDD contains up to $O(2^n/n)$ nodes. A similar bound [6] even holds on average, for a random boolean function. Such properties are also related to a famous theorem of Shannon and Muller [7, p. 763] to the effect that almost all boolean functions have minimal circuit complexity of the order of $2^n/n$. This theoretical discussion is however to be counterbalanced by the fact that functions destined to be realized in silicon are seldom chosen at random!

[4] The structure and size of a BDD depends on the ordering of variables. Several heuristics have been developed in order to try and come up with "good" orders.

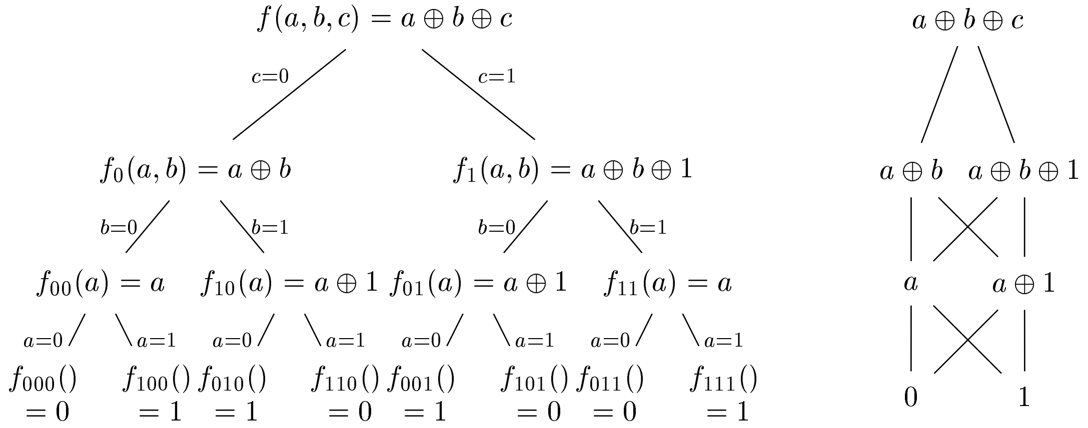


FIGURE 1. The adder function, $f(a,b,c) = a \oplus b \oplus c$: its Binary Decision Tree (left) and its Binary Decision Diagram (right).

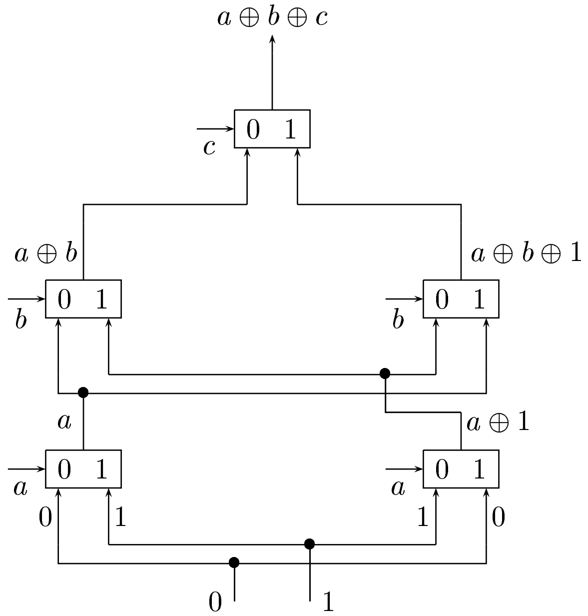


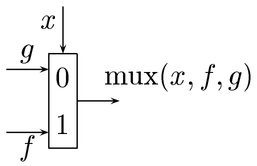
FIGURE 2. A realization of the adder function based on the BDD representation and multiplexers.

becomes a canonical representation of the function it represents, so that equivalence of boolean functions becomes decidable in time linear in the sizes of the compact BDD representations. In particular, this observation makes it possible to compare any combinational circuit design against a canonical specification (the "semantics" of the function) in a computationally efficient manner. This constitutes one of the powerful implications of the BDD concept.

Finally, we mention that once the BDD form of a boolean function has been obtained, a circuit realization of proportional size is immediate: all that is needed is "Shannon's switch" also known as "multiplexer,"

$$\mathrm{mux}\,(x, f, g) := \text{‘}\mathbf{if}\ x\ \mathbf{then}\ f\ \mathbf{else}\ g\text{’} = (x \wedge f) \vee (\overline{x} \wedge g),$$

together with entries grounded at 0 and 1. A diagrammatic representation is as follows:



The way the BDD of the adder function "compiles" into a circuit based on multiplexers is displayed in Figure 2.

2. Polynomial Representations of BDD's

As a preparation for the treatment of synchronous decision diagrams, we now introduce a representation of boolean functions by means of univariate polynomials with coefficients in the binary field $\mathbb{F}_2$. Let f be a boolean function in n variables. Its *truth-table polynomial* $F = \mathbf{T}f$ is defined as follows: interpret each n-tuple $(x_1, x_2, \ldots, x_m)$ of boolean values as the binary representation of an integer,

$$\beta(x_1, \ldots, x_n) := (x_1 x_2 \ldots x_n)_2 = x_1 2^{n-1} + \cdots + x_n,$$

(observe the convention that lower order bits are on the right), and set

$$\mathbf{T}f(z) = \sum_{x_1, \ldots, x_n \in \mathcal{B}} f(x_1, x_2, \ldots, x_n) z^{\beta(x_1, \ldots, x_n)}.$$

For instance the adder function $f(a, b, c) = a \oplus b \oplus c$ has the standard truth table

$x_1 x_2 x_3$	000	001	010	011	100	101	110	111
$\beta(x_1, x_2, x_3)$	0	1	2	3	4	5	6	7
$f(x_1, x_2, x_3)$	0	1	1	0	1	0	0	1

so that its truth-table polynomial is

$$\mathbf{T}f(z) = z + z^2 + z^4 + z^7.$$

The BDD algorithm is amenable to interpretation in this formalism. Define the two "sectioning" operators on polynomials $\mathbb{F}_2[z]$ by

$$S_0\Big(\sum_k f_k z^k\Big) = \sum_k f_{2k} z^k, \qquad S_1\Big(\sum_k f_k z^k\Big) = \sum_k f_{2k+1} z^k.$$

(The definition is also valid for power series of $\mathbb{F}_2[[z]]$, a fact to be used later.) The specialization of the last bit in a function $f(x_1, \ldots, x_n)$ is then seen to be isomorphic to sectioning. Indeed, a simple calculation shows that

$$\begin{aligned} S_0\mathbf{T}\big(f(x_1, \ldots, x_{n-1}, x_n)\big) &= \mathbf{T}\big(f(x_1, \ldots, x_{n-1}, 0)\big) \\ S_1\mathbf{T}\big(f(x_1, \ldots, x_{n-1}, x_n)\big) &= \mathbf{T}\big(f(x_1, \ldots, x_{n-1}, 1)\big). \end{aligned}$$

Consequently, the BDD construction can be regarded as being equivalent to decomposing a polynomial by means of S_0, S_1 until an eventual reduction to the constants 0 and 1 is attained. In this framework, the BDD algorithm applied to the adder example corresponds to the tree and the diagram of Figure 3.

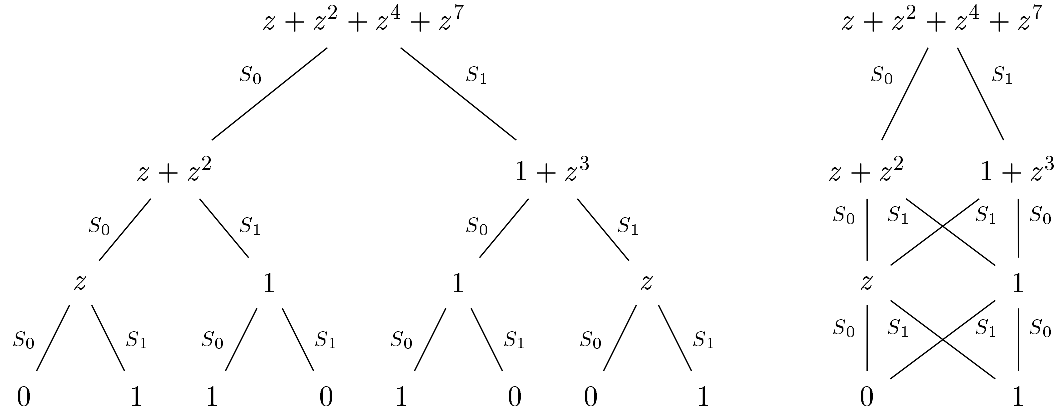


FIGURE 3. Polynomial representations of the Binary Decision Tree and the Binary Decision Diagram of the ternary adder.

3. Synchronous Decision Diagrams

In all generality, a *sequential function* maps infinite sequences of binary inputs into infinite sequences of binary outputs. It thus takes as input a "stream" of bits $(x_t)_{t\geq 0}$ and produces another "stream" $(y_t)_{t\geq 0}$. In other words, a sequential function is a mapping from $\mathcal{B}^\infty$ to $\mathcal{B}^\infty$. For practical purposes, additional constraints must clearly be imposed on the sequential functions considered.

First, we say that a function f from $\mathcal{B}^\infty$ to $\mathcal{B}^\infty$ is *causal* when the output at time t depends exclusively upon the input values from times 0 through t. In what follows, only causal functions are considered. (For the mathematically inclined reader, we note that causal functions are particular continuous functions on the set $\mathcal{B}^\infty$ endowed with the topology induced by the metric $d(a,b) = 2^{-\min\{\,t\,|\,a_t\neq b_t\,\}}$.)

For f causal, we let f_t be the output at time t:

$$y_t = f_t(x_0,\ldots,x_t).$$

By analogy with the specialization of combinational functions, we define the *predictors*, $\varpi_0 f$ and $\varpi_1 f$, by the properties:

$$(\varpi_0 f)_{t+1} = f_t(x_0,\ldots,x_t,0), \qquad (\varpi_1 f)_{t+1} = f_t(x_0,\ldots,x_t,1).$$

These predictors tabulate which value of f will be taken when the input bit to arrive next is specialized to 0 or 1. For $b_0\ldots b_r$ a sequence of bits, we then have the (generalized) predictor of order $r+1$,

$$\varpi_{b_0\ldots b_r} f = \varpi_{b_r}\ldots\varpi_{b_0} f.$$

By infinite iteration, we can then construct the *synchronous decision tree* (SDT) denoted by $\mathrm{sdt}(f)$ as the tree where the nodes are the quantities $\nu = \varpi_w(f)$ and the descendents of node ν are $\varpi_0(\nu)$, $\varpi_1(\nu)$. The tree $\mathrm{sdt}(f)$ can be realized by an infinite tree circuit using only multiplexers and registers (i.e., circuits capable of storing one binary value), much in the same way as combinational circuits are realized by finite tree circuits. See Figure 4 for an illustration.

Next, in order to be computable by some physical device, a digital function must be causal, but also representable by some finite system. To formalize this, we introduce the notion of *on-line computable function*: by this is meant a function such that the collection of all predictors of all

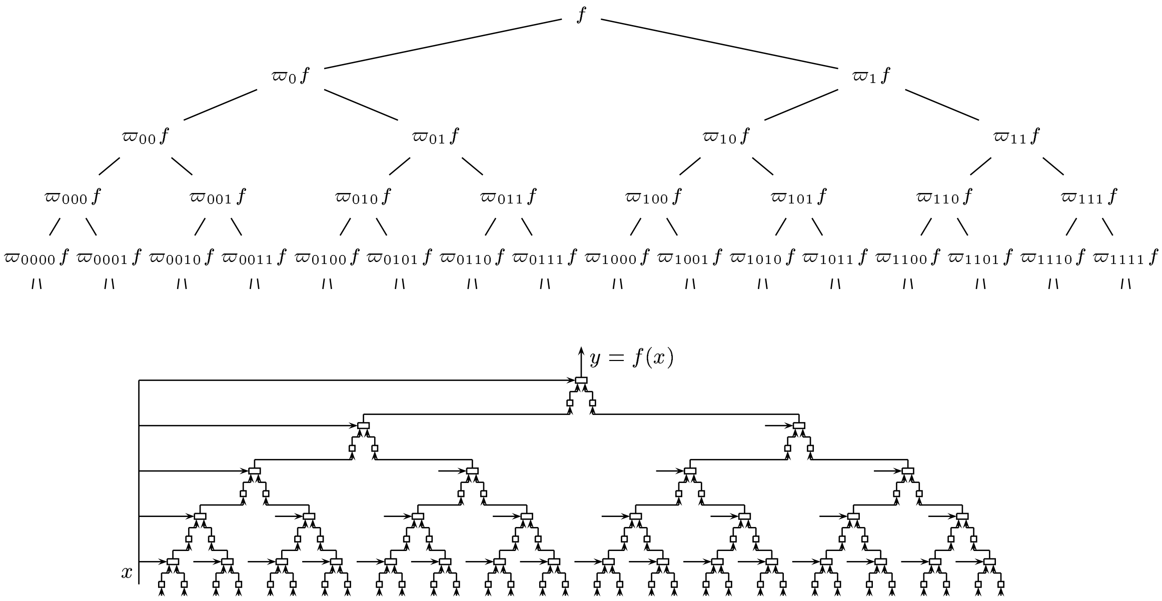


FIGURE 4. The infinite synchronous decision tree (top) and its circuit realization (bottom).

orders forms a finite set. In this case, the (infinite) tree can be converted to a (finite) graph[5] by identifying nodes of the SDT associated with functions that are equal. The resulting graph is called the *synchronous decision diagram* (SDD) and it is obtained by a simple algorithm: *(i)* build the infinite SDT for f; *(ii)* systematically share all the subexpressions generated during this process. (Optionally, one may also consider functions f, g to be isomorphic if either $f = g$ or $f = \neg g$; in that case the SDD will also involve logical negation gates but will be more compact.)

When presented as above, the SDD algorithm looks like an infinite process. However, it can be seen [8] that if a function is realizable by a finite transducer (i.e., an automaton with output), then the SDD algorithm terminates in finite time. In fact, the SDD algorithm provides an integrated alternative to the classical design of sequential circuits.[6]

In order to illustrate the SDD concept, we apply it now to the design of a circuit that takes as input a stream of bits (x_t) meant to represent the real number $\xi = \sum_{t \geq 0} x_t 2^{-t}$ and produces as output the stream (y_t) where the real number $\eta = \sum_{t \geq 0} y_t 2^{-t}$ satisfies $\eta = (1/3)\ \xi$. Introduce the integers

$$x_{(t)} = 2^t \sum_{s \leq t} \frac{x_s}{2^s}, \quad y_{(t)} = 2^t \sum_{s \leq t} \frac{y_s}{2^s}, \quad t \geq 0,$$

and the carry r_t defined by

$$x_{(t)} = 3y_{(t)} + r_t, \quad 0 \leq r_t < 3, \quad t \geq 0.$$

An easy calculation that mimics high school arithmetics yields

$$2r_t + x_{t+1} = 3y_{t+1} + r_{t+1}, \quad t \geq 0.$$

[5]Observe that, as opposed to the case of combinational circuits, the corresponding graph is no longer acyclic since nodes at different levels in the tree may be collapsed.

[6]A classical construction starts from a specification of a finite automaton and stores the current state of the automaton in binary registers while realizing the transition function by means of a combinational circuit (itself possibly optimized via BDD's).

These formulæ show that the function $\eta = \xi/3$ is causal and that the bit $y_{t+1} = f_{t+1}(x_0, \ldots, x_{t+1})$ depends only on the last input bit together with the "carry" r_t that is inherited from past history. The carry can only assume three values and accordingly the number of predictors is finite, to the effect that the SDT has only six nodes. Thus, f is on-line computable. Figure 5 shows the result of the construction. (In the diagram, a transition denoted by α/β is triggered by reading the bit α and results in producing the bit β.)

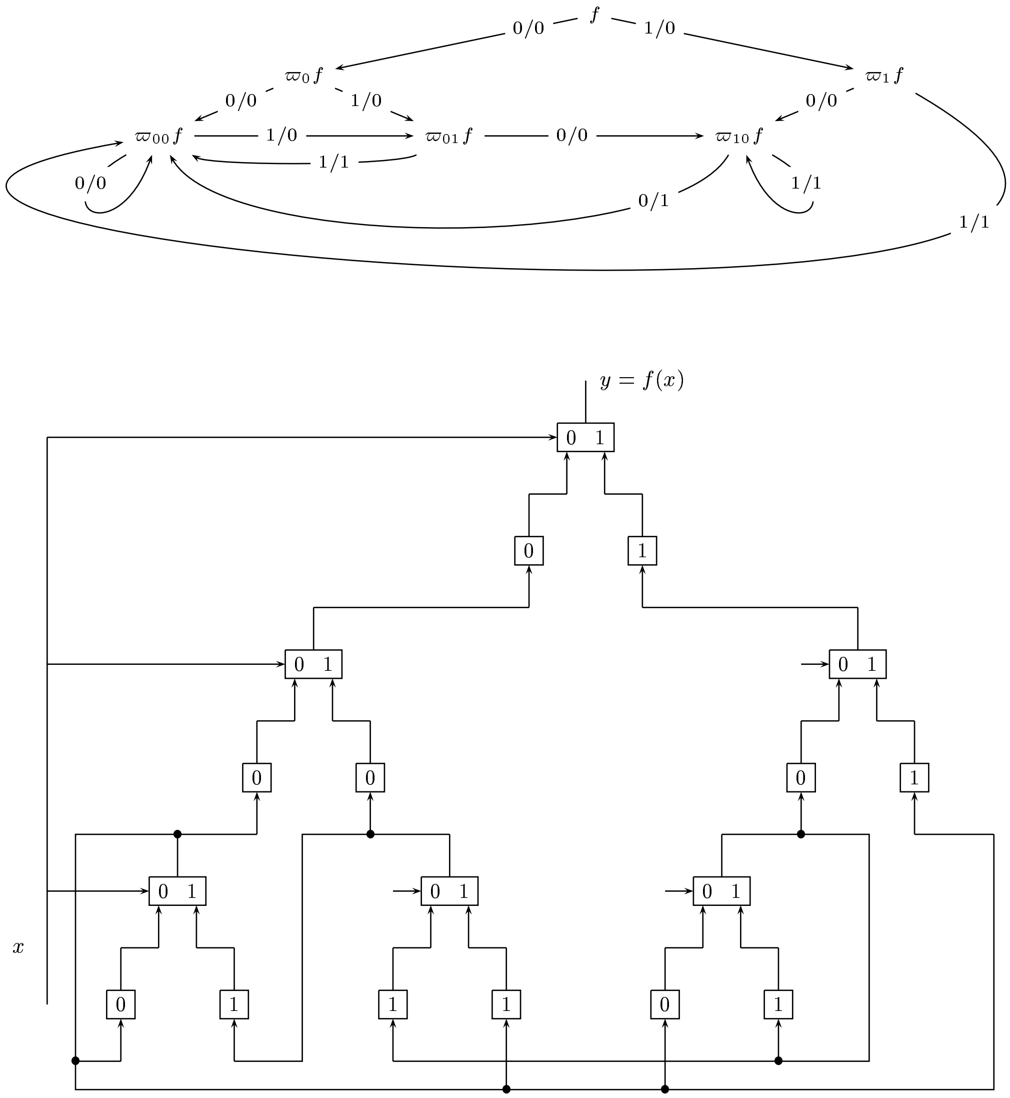


FIGURE 5. The '(1/3) ξ' function: its abstract SDD representation (top) and the circuit realization (bottom).

4. Formal Power Series Representations of SDD's

A causal function f is characterized by its truth table. This is the power series representation $\mathbf{T}f$, an element of $\mathbb{F}_2[[z]]$ defined as

$$\mathbf{T}f(z) := \sum_{t\geq 0} \sum_{x_0,\ldots,x_t\in\mathcal{B}} f_t(x_0,\ldots,x_t) z^{\beta(1x_0\ldots x_t)-2}.$$

(The correction of -2 in the exponent is a convenience chosen to ensure that exponents start at 0.) We shall refer to $\mathbf{T}f$ as the *truth-table representation* of f. This notion extends in a natural way the corresponding definition for combinational functions. Indeed, an alternative definition of $\mathbf{T}f$ for causal functions is as follows: take F_t as the truth table of f_t in "listed" form, and build the truth table of f in "listed" form by

$$F_0F_1F_2\ldots = \big[f_0(0)f_0(1)\big]\big[f_1(00)f_1(01)f_1(10)f_1(11)\big]\big[f_2(000)f_2(001)\ldots f_2(111)\big]\ldots;$$

then $F(z) = \mathbf{T}f(z)$ satisfies a sort of a "generating function relation,"

$$F(z) = \big[f_0(0) + zf_0(1)\big] + z^2\big[f_1(00) + zf_1(01) + z^2f_1(10) + z^3f_1(11)\big] \\ + z^6\big[f_2(000) + zf_2(001) + \cdots + z^7f_2(111)\big] + \cdots,$$

so that there is a simple relation between truth tables of combinational functions and of sequential functions:

$$F(z) = \sum_{t\geq 0} z^{2^{t+1}-2}\mathbf{T}f_t(z).$$

Equipped with these definitions, we observe the action of sections,

$$S_0F(z) = \big[f_0(0)\big] + z\big[f_1(00) + zf_1(10)\big] + z^3\big[f_2(000) + zf_2(010) + \cdots + z^3f_2(110)\big] + \cdots,$$
$$S_1F(z) = \big[f_0(1)\big] + z\big[f_1(01) + zf_1(11)\big] + z^3\big[f_2(001) + zf_2(011) + \cdots + z^3f_2(111)\big] + \cdots,$$

which entails

$$S_0F(z) = \sum_{t\geq 0} z^{2^t-1}S_0F_t(z) = f_0(0) + z\sum_{t\geq 0} z^{2^{t+1}-2}S_0F_{t+1}(z),$$
$$S_1F(z) = \sum_{t\geq 0} z^{2^t-1}S_1F_t(z) = f_0(1) + z\sum_{t\geq 0} z^{2^{t+1}-2}S_1F_{t+1}(z).$$

This provides a direct relation between the sections of the truth table of any causal f and the predictors of f, namely,

$$S_0(\mathbf{T}f)(z) = f_0(0) + z\mathbf{T}(\varpi_0 f)(z), \qquad S_1(\mathbf{T}f)(z) = f_0(1) + zT(\varpi_1 f)(z).$$

Now, by definition, f is on-line computable when its predictors lie in a finite set. The equation above shows that this is equivalent to the finiteness of vector space over $\mathbb{F}_2(z)$ of all the (iterated) sections of the truth table. The connection is thereby established with what is otherwise known as automatic series;[7] see the foundational paper by Christol *et al.* [3], Dumas's thesis [4], and several summaries in previous issues of the Algorithms Seminar Proceedings. We state:

Theorem 1. *The truth table $\mathbf{T}f$ of an online computable function f is a 2-automatic series. Consequently, it is an algebraic function over the field $\mathbb{F}_2(z)$.*

[7] A sequence is defined to be automatic if its nth element is produced by a finite transducer applied to the binary representation of n; a series is automatic if its sequence of coefficients is automatic. Equivalent characterizations of automatic series are as algebraic elements over $\mathbb{F}_2(z)$ or as solutions to Mahlerian equations; refer to [3, 4].

Each causal finite function f may thus be represented by a bivariate characteristic polynomial $P(z,y)$ so that the truth table $\mathbf{T}f$ is the only root $y \in \mathbb{F}_2[[z]]$ of $P(z,y) = 0$. Conceivably, this theorem opens an avenue to circuit design and verification by means of polynomial elimination algorithms—typically, Gröbner bases. Given the superexponential complexity of algebraic elimination, it seems however to the authors of the summary that a direct approach based on linear algebra (in accordance with standard techniques of 2-automatic series [4]) should yield decision procedures of lower complexity.

5. From Circuits to Functions

In this section, we show how to put to good use the formalism introduced above in order to analyse circuits: starting from a given circuits, the goal is to determine a mathematical specification of what it does. Note that the dual problem of synthesis has been already implicitly tackled on the occasion of the "one-third" function ($\xi \mapsto \xi/3$).

Let us first consider a circuit that takes as input a stream of bits (g_t) and produces the stream (h_t) which is the same stream delayed by 1 in time. In other words, we have $h_0 = z_0$ (the initialization value) and $h_t = g_{t-1}$ for $t \geq 1$. In the context of a finite circuit, the values h_t are described by their truth table and they depend on the global input sequence $x = (x_t)_{t\geq 0}$ of the circuit. Thus, in terms of the finite boolean functions $g_t(x_0, \ldots, x_t)$ and $h_t(x_0, \ldots, x_t)$, we have

$$h_0 = z_0, \quad h_t(x_0, \ldots, x_t) = g_{t-1}(x_0, \ldots, x_{t-1}) \quad \text{for } t \geq 1.$$

This relation translates into a relation between the truth tables of the input (G) and the output (H) of the register,

$$H(z) = (1+z)\big(z_0 + z^2 G(z^2)\big). \tag{1}$$

Thus, in the formal power series representation, a register operates by way of the "Mahlerian operator," $G(z) \mapsto G(z^2)$.

Consider next a multiplexer that takes as input two streams of bits $a(x)$ and $b(x)$ (themselves causal functions of the input stream x) and assume that control is achieved by the input stream x. The output $m(x)$ is a causal function defined by

$$m_t(x_0, \ldots, x_t) = \operatorname{mux}\big(x_t, a_t(x_0, \ldots, x_t), b_t(x_0, \ldots, x_t)\big),$$

which we abbreviate as

$$m(x) = \operatorname{mux}\big(x, a(x), b(x)\big).$$

A little reflection shows that the truth table of m is obtained by suitably merging the truth tables of a and b as follows

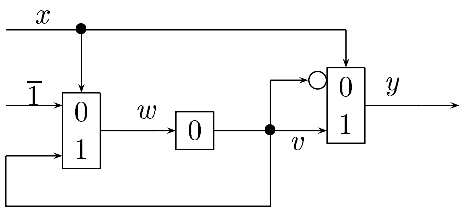
$$\begin{array}{c|cc|cccc|ccc}
A & a_0(0) & a_0(1) & a_1(00) & a_1(01) & a_1(10) & a_1(11) & a_2(000) & a_2(001) & \ldots \\
B & b_0(0) & b_0(1) & b_1(00) & b_1(01) & b_1(10) & b_1(11) & b_2(000) & b_2(001) & \ldots \\
M & b_0(0) & a_0(1) & b_1(00) & a_1(01) & b_1(10) & a_1(11) & b_2(000) & a_2(001) & \ldots .
\end{array}$$

This relation translates into

$$M(z) = (S_0 B)(z^2) + z(S_1 A)(z^2), \tag{2}$$

which now involves a blend of sectioning and Mahlerian operators.

Now, a finite circuit can be translated into a system of fixed-point equations: to each entity is associated its truth table; then relations (1) and (2) (used repeatedly) provide the system of equations. Here is an application to a circuit discussed in [8]. This circuit comprises one inverter (represented by a circle), two multiplexers, and one register that is initially set at 0. The upper entry of the leftmost multiplexer receives a continuous stream of 1's which is represented by $\overline{1}$.



What is required is to verify that the circuit computes the function

$$\xi \mapsto \eta = \xi + 1,$$

where the input and output streams are now interpreted as dyadic numbers, that is

$$\xi = \sum_{t \geq 0} x_t 2^t, \quad \eta = \sum_{t \geq 0} y_t 2^t, \quad x, y \in \mathbb{Z}_2.$$

To each of the flows, y, v, w, one associates its truth table, respectively $Y(z), V(z), W(z)$. Given the rules (1) and (2), the structural description of the circuit is translated (compiled!) into the system of equations:

$$\begin{aligned}
Y(z) &= \left(S_0\left(\frac{1}{1-z} + V\right)\right)(z^2) + z\,(S_1 V)\,(z^2) = \frac{1}{1-z^2} + (S_0 V)\,(z^2) + z\,(S_1 V)\,(z^2) \\
&= \frac{1}{1-z^2} + V(z), \\
V(z) &= (1+z)z^2 W(z^2), \\
W(z) &= \left(S_0 \frac{1}{1-z}\right)(z^2) + z\,(S_1 V)\,(z^2) = \frac{1}{1-z^2} + z\,(S_1 V)\,(z^2).
\end{aligned}$$

In order to understand the function computed by the circuit, we proceed to solve this system. The second equation provides $S_1 V(z) = zW(z)$, a relation that, when carried into the third equation, gives:

$$W(z) = \frac{1}{1-z^2} + z^3 W(z^2).$$

Such a functional equation is now easily solved by iteration,

$$W(z) = \sum_{k=0}^{+\infty} \frac{z^{3(2^k-1)}}{1 - z^{2^{k+1}}},$$

and this form entails in turn

$$\begin{aligned}
Y(z) = \sum_{k=0}^{+\infty} \frac{z^{3\cdot 2^k - 1}}{1 - z^{2^{k+1}}} + \sum_{k=0}^{+\infty} \frac{z^{3\cdot 2^k}}{1 - z^{2^{k+1}}} \\
= \left[1\right] + z^2\left[z + z^2\right] + z^6\left[z + z^3 + z^5 + z^6\right] + z^{14}\left[z + \cdots\right] + \cdots .
\end{aligned}$$

From there, it is an easy exercise (left to the reader) to check that the truth table $Y(z)$ is equal to the truth table corresponding to the dyadic function $\xi \mapsto \xi + 1$.

6. Conclusion

Due to constraints already evoked, we could only scratch the surface in this brief[8] seminar summary. The point of view developed in the talk bases itself further on the existence of isomorphisms between various domains. For instance, as we have seen, the boolean domain may be viewed as $\mathcal{B}$ or $\mathbb{F}_2$; boolean functions are representable as elements of $\mathbb{F}_2[z]$; on-line computable functions are equivalent to algebraic elements of $\mathbb{F}_2[[z]]$ and to 2-automatic series. There exist several other interesting connections, for instance, with the ring of dyadic integers $\mathbb{Z}_2$ that form one of the conceptual basis of the original paper [8]. Such isomorphisms do increase the expressive power of the SDD formalism that we have opted to develop here only over $\mathcal{B}$ while making use of representations in $\mathbb{F}_2[[z]]$.

There is also great practical potential in the algorithms associated with the SDD concept. Quoting from Vuillemin: *The SDD of f is a cyclic data structure, which represents the minimal finite state machine for f. In the worst case, its size is doubly exponential in the size of f. However, efficient algorithms exist to operate on the SDD representations with the following characteristics: constant time*[9] *for $f(\lambda x)$, $f(1+\lambda x)$; linear time for $\neg f(x)$, $\lambda f(x)$, the inverse $g\big(f(x)\big)=x$, and the fixed point $y=\lambda g(x,y)$; quadratic time for the composition $f\big(g(x)\big)$ and for boolean operations, $f(x)\wedge f(y)$, etc; cubic time for the more general composition $f\big(g(x),h(x)\big)$. The SDD opens an approach to sequential circuit synthesis and verification whose implementation is straightforward in a high-level language, and which can cope automatically with synchronous circuits of limited size.*

Bibliography

[1] Bryant (Randal E.). – Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, vol. C-35, n° 8, 1986, pp. 679–691.

[2] Bryant (Randal E.). – Symbolic boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys*, vol. 24, n° 3, 1992, pp. 293–318.

[3] Christol (G.), Kamae (T.), Mendès France (M.), and Rauzy (G.). – Suites algébriques, automates et substitutions. *Bulletin de la Société Mathématique de France*, vol. 108, n° 4, 1980, pp. 401–419.

[4] Dumas (Philippe). – *Récurrences mahlériennes, suites automatiques, études asymptotiques.* – Institut National de Recherche en Informatique et en Automatique, Rocquencourt, 1993, 241p. Thèse, Université de Bordeaux I, Talence, 1993.

[5] Flajolet (Philippe), Sipala (Paolo), and Steyaert (Jean-Marc). – Analytic variations on the common subexpression problem. In Paterson (M. S.) (editor), *Automata, languages and programming. Lecture Notes in Computer Science*, vol. 443, pp. 220–234. – Springer, New York, 1990. Proceedings of the 17th ICALP Conference, Warwick, July 1990.

[6] Liaw (Heh Tyan) and Lin (Chen Shang). – On the OBDD-representation of general boolean functions. *IEEE Transactions on Computers*, vol. 41, n° 6, 1992, pp. 661–664.

[7] van Leeuwen (Jan) (editor). – *Handbook of theoretical computer science. Vol. A.* – Elsevier Science Publishers, Amsterdam, 1990, x+996p. Algorithms and complexity.

[8] Vuillemin (Jean E.). – On circuits and numbers. *IEEE Transactions on Computers*, vol. 43, n° 8, 1994, pp. 868–879.

[8] *Editor's Note.* I wish to express my true gratitude to the authors of the summary for not exceeding three times the expected length of a typical summary.

[9] Dyadic interpretations ($\mathbb{Z}_2$) are understood in the first two examples.

Part IV

Computational Biology and Combinatorics of Words

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 111–113.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Bayesian Approach to DNA Segmentation into Regions with Different Average Nucleotide Composition

Vsevolod Makeev
Engel'hard Institute of Molecular Biology, Moscow

October 7, 1999

Summary by Mireille Régnier

1. Biological Motivation

Local nucleotide composition, that is, the distribution of nucleotides A, C, G, T along a chromosome, is important for many biological issues. Moreover, local nucleotide composition is accounted for in many algorithms developed to search for different patterns in DNA sequences. We present a method of segmentation of nucleotide sequences into regions with different average composition. The sequence is modelled as a series of segments; within each segment the sequence is considered as a random Bernoulli process. The partition algorithm proceeds in two stages. In the first stage the optimal partition is found, which maximizes the overall product of marginal likelihoods computed for each segment and prevents segmentation into short segments. In the next stage, optimal boundaries are filtered, and segments with close compositions are merged. This allows us to study segments with the chosen length-scale.

2. Optimal Segmentation

2.1. **Probabilistic formulation.** A symbolic sequence over an alphabet Ω of V letters is considered as a series of segments. Each segment is modelled as a Bernoulli random sequence. Bernoulli probabilities are estimated from the vector $\mathbf{n} = (n_1, \ldots, n_V)$ where n_j denotes the number of occurrences of the jth symbol in the segment. In the Bayesian approach [1] estimated parameters are random variables. The probability distribution of these random variables is estimated from the data by a bootstrapping approach. First, one assumes an initial probability distribution—the so-called prior distribution—that may be chosen rather arbitrarily. These probability distributions are re-estimated from the data using the Bayes formula. The results of Bayesian estimation are always some probability distributions of the estimated quantity. Bayesian and classical statistics, however, agree for large samples because Bayesian distributions converge to the maximal likelihood estimation for any reasonable prior distribution. Denote the set of letter probabilities (the segment composition) as $\sigma = (\theta_1, \ldots, \theta_V)$ with $\sum_{k=1}^{V} \theta_k = 1$. The likelihood of the individual sequence is $L(\sigma) = \prod_{k=1}^{V} \theta_k^{n_k}$. Given a composition $\sigma = (\theta_1, \ldots, \theta_V)$, one writes the probability density function $p(\sigma)$, with normalisation condition $\int p(\sigma)\, d\sigma = 1$.

One starts from some prior distribution $p(\sigma)$, say the uniform distribution on $\sum_k \theta_k = 1$. The composition σ of the Bernoulli random process is picked up according to this prior distribution, $p(\sigma)$. The estimated probability density function $p(\sigma/\mathbf{n})$ satisfies Bayes's theorem:

$$p(\sigma/\mathbf{n}) = \frac{L(n/\sigma)p(\sigma)}{P(\mathbf{n})}$$

where $P(\mathbf{n}) = \int L(\mathbf{n}/\sigma)p(\sigma)\,d\sigma$. The normalisation constant $P(\mathbf{n})$ is called marginal likelihood [3]. It reflects the overall probability of the given sequence in the two stage random process. For a uniform prior distribution, one has:

$$P(\mathbf{n}) = \frac{(V-1)!}{(N+V-1)!} n_1! \dots n_V!.$$

Surprisingly, this quantity is also obtained in a conceptually similar but different probabilistic model (G. Shaeffer, 1999). For a sequence of length N, the probability of this sequence in the shuffling procedure is computed. Numbers $(n_1, \dots n_V)$ are picked up according to uniform distribution. With the assumption that segments as independent, the complete likelihood of the sequence segmentation into k segments with known boundary location is:

$$P = \prod_k P_k(n_k).$$

This quantity is optimized over the set of all possible boundary configurations yielding the optimal segmentation.

2.2. **Dynamic programming.** The maximization algorithm is as follows. Consider a sequence $S = s_1 s_2 s_3 \dots s_N$ of length N, where $s_i \in \Omega$. For every segment $S(a,b) = s_a \dots s_b$, one introduces a weight $W(a,b)$: for example, $W(a,b)$ can be $\ln P\big(S(a,b)\big)$. A segmentation R in m blocks is determined as a set of boundaries $R = \{k_0 = 0, k_1, \dots, k_{m-1}, k_m = N\}$, where k_i separates s_k and s_{k+1}. Its weight is:

$$F(R) = \sum_{j=1}^{m} W(k_{j-1}+1, k_j).$$

For functions determined on the segmentations, one also uses another set of variables, the indicators of the boundary positions q_k, $1 \le k \le N$. By definition, $q_k = 1$ if there exists a segment boundary after the kth letter, otherwise it is 0. Below, we use the notations $F(R)$ and $F(q_1, \dots, q_k)$ indifferently. The segmentation R^* with maximal weight is computed in a recursive manner. Denote by $R^*(k)$ the optimal segmentation of the fragment $S(1,k)$, $1 \le k \le N$. $R^*(1)$ is trivial. When optimal segmentations $R^*(1)$, …, $R^*(k-1)$ are known, the optimal segmentation $R^*(k)$ is found using the following recurrence expression:

$$F\big(R^*(k)\big) = \max_{0 \le i \le k-1} \left[F\big(R^*(i)\big) + W(i+1,k)\right], \tag{1}$$

with $F\big(R^*(0)\big) = 0$. This equation yields the algorithm. Since the segmentation $R^*(k)$ is built in time $O(k)$, the total time can be estimated as $O(N^2)$.

2.3. **Fluctuations in local composition.** It appears that segments in optimal segmentation are usually very short. Even a random uniform Bernoulli sequence is divided into many segments. More generally, when the sequence consists of several random homogeneous domains, the optimal segmentation may include many borders located within the domains. This phenomenon is due to statistical fluctuations of the local nucleotide composition in random sequences. Thus it is advantageous to extract boundaries, which separate long regions with different compositions from those that reflect statistical fluctuations. This can be done by penalizing those segmentations that contain more boundaries. The correct penalty choice was initially chosen from computer simulations.

3. Filtration of Boundaries

3.1. Partition function. To study the relative significance of a boundary, one can calculate a score, that reflects how the addition of this particular boundary influences weights of segmentations. Given the probability $\Pi(\mathbf{q})$ of each segmentation $\mathbf{q} = (q_1, \ldots, q_N)$, one defines the partition function of the segmentations in a standard way [2] by summing the probabilities of all possible partitions:

$$Z(N) = \sum_{q_1,\ldots,q_{N-1}} \Pi(q_1, \ldots, q_{N-1}) \tag{2}$$

With the partition function at hand, one can compute the probability of a boundary to be located after a particular letter k. One computes two partition functions for the regions to the left and to the right of this border, Z_L and Z_R respectively:

$$\Pi(k) = \frac{Z_L(k) Z_R(N-k)}{Z(N)}. \tag{3}$$

3.2. Dynamic programming. The partition function in (2) rewrites as follows [2]:

$$Z(N) = \sum_{q_1,\ldots,q_{N-1}} e^{F(q_1,\ldots,q_{N-1})}. \tag{4}$$

To compute the probability of a boundary after the letter k, we also need the partition functions of the segments to the left and to the right of this boundary, and recursive formulæ to compute $Z_L(k)$ and $Z_R(k)$ are analogous to (1). They are obtained through the formal substitution of operations. Summation is used instead of taking the maximum, and multiplication is used instead of summation [2]. Equation (1) becomes:

$$Z_L(k) = \sum_{j=0}^{k-1} e^{W(j+1,k-1)} Z_L(j),$$

$$Z_R(k) = \sum_{j=k}^{N} e^{W(k,j)} Z_R(j),$$

with boundary conditions $Z_L(0) = Z_R(N+1) = 1$ and $W(k-1,k) = W(N,N+1) = 0$. An obvious modification of dynamic programming calculates the partition function in the case when only the given set of boundaries is allowed.

3.3. Filtration strategy. For the best result one should combine calculation of optimal segmentation with filtration. At the first stage, an optimal segmentation is found. Then a cut-off value is chosen and all the boundaries with probabilities (3) lower than that cut-off value are removed. The resulting set of boundaries usually is not optimal in the sense that some boundaries can also be removed, yielding a configuration with a higher probability P. So an additional round of optimisation is performed, removing some boundaries. Iterations converge rapidly to the stable set of boundaries all of which have the partition function probabilities greater than the cut-off value.

Bibliography

[1] Durbin (R.), Eddy (S.), Krogh (A.), and Mitchinson (G.). – *Biological sequences analysis:probabilistic models of protein and nucleic acids.* – Cambridge University Press, 1998.
[2] Finkelstein (A. V.) and Roytberg (M. A.). – Computation of biopolymers: A general approach to different problems. *Biosystems*, vol. 30, 1993, pp. 1–19.
[3] Liu (S. L.) and Lawrence (C. E.). – Bayesian inference of biopolymer models. *Bioinformatics*, vol. 15, 1999, pp. 38–52.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 115–118.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Enumeration of Autocorrelations and Computation of Their Populations

Éric Rivals
LIRMM, Université Montpellier II

November 22, 1999

Summary by Pierre Nicodème

Abstract

This talk presents in a first part Guibas and Oldlyzko's characterization of autocorrelations and in a second part algorithms developed by Éric Rivals with Sven Rahmann (TBI, DKFZ, Heidelberg) to enumerate the autocorrelations and to simultaneously compute their populations.

1. Introduction

An interesting statistics about a random text of size N is the number of different words of a given size n it contains, or, equivalently, how many words of size n are missing in the random text. These statistics are closely linked with the autocorrelations of the words, that are sets of periods of the words. We consider here the enumeration of autocorrelations and the populations of the autocorrelations, originally studied by Guibas and Odlyzko [3]. The original motivation of Rivals and Rahmann comes from searching genomic databases with q-grams [1].

2. Definitions

We consider a finite *alphabet* Σ. Let $w = w_1 w_2 \cdots w_n$ where $w_i \in \Sigma$. A *period* of w is an integer p such that for all i between 1 and $n-p$ we have $a_i = a_{i+p}$. As an example, the word `abracadabra` has for periods 0, 7, and 10. Its factor `abra` has for periods 0 and 1. The *autocorrelation vector* of a word w, denoted by $V(w)$, is the binary vector $V = (v_0, v_1, \ldots, v_{n-1})$ such that v_i is equal to one if i is a period of w and to zero otherwise. Alternatively, the autocorrelation will be denoted by the corresponding binary word $v_0 v_1 \cdots v_{n-1}$. We denote $\Pi(w)$ the set of autocorrelations of the word w.

We are interested in statistics about the whole set of words of size n and therefore denote $\Gamma(n)$ the set of autocorrelations of size n and $\kappa(n)$ its cardinality.

The periods have the following properties:

1. 0 is always a period;
2. if i is a period, then for all i in the range $\big(1, \lfloor n/p \rfloor\,\big]$ the integer ip is a period;
3. if p and q are periods of w, with $p < q$, then $q - p$ is a period of the prefix of length $n - p$ of w.

Theorem 1 (Fine and Wilf)**.** *Let p and q be periods of a word w, with $p < q$. If $p+q \leq |w| + \gcd(p,q)$ then $\gcd(p,q)$ is a period of w.*

See [2, 3] for this theorem.

3. Periods in Strings

This section follows the lines of Guibas and Odlyzko [3].

In order to give equivalent characterizations of autocorrelations vectors within binary vectors in Theorem 2 below, we now give the definitions of the forward and backward propagation rules and of the Ξ predicate that are used in this theorem.

If $p < q$ are periods of a word w, then $q + (q - p)$ is also period. This gives the following rule.

Definition 1 (Forward Propagation Rule). A binary vector $V = (v_0, v_1, \cdots, v_n)$ satisfies the forward propagation rule if, whenever we have $v_p = v_q = 1$ with $p < q$, we also have $v_t = 1$ for all t in $[p, n)$ such that $t = p + i(q - p)$ with $i = 0, 1, 2, \ldots$.

The backward propagation rule asserts that if p and q are periods with $p < q$ and if $p - (q - p)$ is not a period, then none of the positive integers $p - i(q - p)$ may be a period.

Definition 2 (Backward Propagation Rule). A binary vector $V = (v_0, v - 1, \cdots, v_{n-1})$ satisfies the backward propagation rule if the following condition holds. Consider every p and q such that $p < q \leq 2p$ with $v_p = v_q = 1$, but $v_{2p-q} = 0$; then for all t in the range $[0, 2p - q]$ such that $t = p - i(q - p)$ and i belongs to the interval $\left[1, \left\lfloor \frac{n-p}{q-p} \right\rfloor\right]$ we have $v_t = 0$.

We now introduce a recursive predicate on binary vectors that is equivalent to the condition that the binary vector is an autocorrelation vector. In the following, we note the shortest period of a word v by $\pi(v)$.

Definition 3 (Recursive Predicate Ξ). Let $V = (v_0, v_1, \ldots, v_{n-1})$ be a non-empty binary vector. Define $p = \pi(v_0 v_1 \cdots v_{n-1})$. The vector V satisfies the predicate Ξ if and only if V is such that $v_0 = 1$ and V satisfies one of the following two conditions:

- *Case (A),* $p \leq \lfloor \frac{n}{2} \rfloor$.
  Let $r = n \mod p$ and $q = p + r$ and let $w = w_1 \cdots w_q$ be the suffix of $v_0 v_1 \cdots v_{n-1}$ of length q. Then:
  1. for all j in the range $[1, n - q]$, $v_j = 1$ if $j = ip$ for some i, and $v_j = 0$ otherwise;
  2. $w_p = 1$ or $r = 0$;
  3. if $\pi(w) < p$ then $\pi(w) > (q - p) + \gcd(p, \pi(w))$;
  4. the vector $(w_1, \ldots, w_q)$ satisfies predicate Ξ.
- *Case (B),* $p > \lfloor \frac{n}{2} \rfloor$.
  Let $w = w_1 \cdots w_{n-p}$ be the suffix of $v_0 \cdots v_{n-1}$ of length $n - p$. Then for all j in the range $[1, n - p]$ we have $v_j = 0$ and the vector $(w_1, \ldots, w_{n-p})$ satisfies predicate Ξ.

The algorithmic check of the predicate Ξ requires $O(n)$ operations on a vector V of size n.

Theorem 2. *Let $V = (v_0, v_1, \cdots, v_n)$ be a non-empty binary vector. Then the following four statements are equivalents:*

1. *V is a correlation vector of a binary string;*
2. *V is a correlation vector of some string;*
3. *$v_0 = 1$ and V satisfies the forward and backward propagation rules;*
4. *V satisfies the predicate Ξ.*

Note that equivalence between statements 1 and 2 implies that the characterization of an autocorrelation vector is independent of the size of the alphabet.

```
Autocorrelations(n)
    if n = 1 then return {1}
    elif n = 2 then return {11, 10}
    else
        Γ(n) := {}
        # Case (A), p ≤ ⌊n/2⌋
        for p for ⌊n/3⌋ to ⌊n/2⌋ do
            r := n mod p
            q := p + r
            Γ(q) := Autocorrelations(q)
            j_0 := min { j | j + p > q + gcd(j, p) }
            for w in Γ(q) do
                if π(w) > j_0 and p mod π(w) ≠ 0 then
                    Γ(q) := Γ(q) ∪ { (10^{p-1})^{⌊n/p⌋-1} w }
                fi
            od
        od
        # Case (B), p > ⌊n/2⌋
        for p for ⌊n/2⌋ to n do
            Γ(n − p) := Autocorrelations(n − p)
            for w in Γ(n − p) do
                Γ(n) := Γ(n) ∪ {10^{p-1} w}
            od
        od
        return Γ(n)
    fi
end

u^p is the word u···u where u is repeated p times
```

FIGURE 1. Recursive algorithm **Autocorrelations**.

4. An Algorithm to Enumerate all Autocorrelations of Size n

We use the predicate Ξ to build a recursive bottom-up procedure that constructs autocorrelation vectors. To this end, note that the condition (2) of Case (A) of the predicate Ξ is equivalent to

$$\pi(w) \text{ does not divide } p \quad \text{and} \quad \pi(w) > j_0 = \min\{\, j \mid j + p > q + \gcd(j,p)\,\}.$$

Algorithm **Autocorrelation** to enumerate all autocorrelations until size n is given in Figure 1.

Implementation. The autocorrelations are stored as binary vectors. The implementation has been done as an iterative procedure, although the algorithm presented in Figure 1 is recursive. Note that in Case (A) of the algorithm the tests of conditions (a) and (b) of the Ξ predicate can be done in $O(1)$ operations. Moreover only the valid subset of $\Gamma(q)$ is computed.

Complexity and optimality. Each bit of an autocorrelation is computed only once. The complexity is unknown, no close formula for the number of autocorrelations of size n being known.

Asymptotic bounds. Guibas and Odlyzko [3] give the following bounds for the logarithm of the number $\kappa(n)$ of autocorrelations of size n:

$$b_l = \left(\frac{1}{2\log 2} + o(1)\right)\log^2 n \le \log\kappa(n) \le \left(\frac{1}{2\log(3/2)} + o(1)\right)\log^2 n.$$

For numerical computations up to $n = 200$, Rivals and Rahmann obtain $\kappa(n) < b_l$. They conjecture that the asymptotic value of $\kappa(n)$ is b_l, the lower bound of Guibas and Odlyzko.

5. Computation of the Populations of Autocorrelations

In this section, the size n of the autocorrelations vectors is fixed.

Definition 4. The population N of an autocorrelation vector V is defined as

$$N(V) = \operatorname{Card}\{\, w \mid V \text{ is the autocorrelation vector of } w\,\}.$$

We define a partial order $\preceq$ on the autocorrelation vectors by $V = v_0v_1\cdots v_{n-1} \preceq V' = v'_0v'_1\cdots v'_{n-1}$ if for all i in $[\,0, n-1\,]$, $v'_i = 1$ whenever $v_i = 1$. We also define the total order $\le$ by $V \le V'$ if the word $v_0v_1\cdots v_{n-1}$ precedes lexicographically the word $v'_0v'_1\cdots v'_{n-1}$. Then $V \preceq V'$ implies $V \le V'$. Autocorrelation vectors of size n are sorted along the total order $\le$ and numbered along this order from 1 to $\kappa(n)$. The notation V_k refers to the vector at rank k in this order.

Definition 5. The number ρ_k of free characters of the autocorrelation V_k is the number of characters that we can choose freely to build a word with the correlation V_k. The other characters are determined by the periods of the autocorrelation.

With an alphabet of size σ, for k from κ $(= \kappa(n))$ to 1, we get

$$N(V_k) = \sigma^{\rho_k} - \sum_{k<j<\kappa \text{ and } V_j \succ V_k} N(V_j).$$

The implementation is quadratic in $\kappa(n)$.

Bibliography

[1] Burkhardt (S.), Crauser (A.), Ferragina (P.), Lenhof (H.-P.), Rivals (E.), and Vingron (M.). – q-gram based database searching using a suffix array (QUASAR). In *Third International Conference on Computational Biology.* pp. 77–83. – ACM-Press, 1999. S. Istrail, P. Pevzner and M. Waterman, editors.

[2] Fine (N. J.) and Wilf (H. S.). – Uniqueness theorems for periodic functions. *Proceedings of the AMS*, vol. 16, 1965, pp. 109–114.

[3] Guibas (Leo J.) and Odlyzko (Andrew M.). – Periods in strings. *Journal of Combinatorial Theory. Series A*, vol. 30, n° 1, 1981, pp. 19–42.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), p. 119.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Classification by Trees: the Shape of the Inferred Tree Depends on the Algorithmic Scheme Selected

Olivier Gascuel
LIRMM, Université Montpellier II

November 22, 1999

Abstract

Two algorithmic schemes are broadly used to construct a tree distance based on a dissimilarity. The first one, initially used for hierarchies, consists in iteratively agglomerating pairs of leaves until only three leaves remain, which corresponds to a unique tree structure. The second one starts with a tree on three leaves and iteratively grafts the objects on the previously build tree. On top of those two construction schemes, the exchange of subtrees is used to iteratively improve the trees obtained by either of the schemes above. We show that, independently of the optimized criterion, these schemes generally induce trees of quite different shapes. The agglomerative scheme tends to produce compact trees with low diameter, whereas grafting and exchange tend to generated more outstretched trees with high diameter. This phenomenon is explained by the difference between prior probability distributions induced by each of these schemes. We illustrate this very distinct difference by the data of the mitochondrial Eve.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 121–124.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Factor Oracle, Suffix Oracle

Matthieu Raffinot
Institut Gaspard-Monge, Université de Marne-la-Vallée

October 4, 1999

Summary by Alain Denise and Matthieu Raffinot

Abstract

The aim of this work is to design efficient algorithms for string matching. For this purpose, we introduce a new kind of automaton: the *factor oracle*, associated with the string p to be recognized in a text. This leads to simple algorithms which are as efficient in time as already known ones, while using less memory. This is a joint work with Cyril Allauzen and Maxime Crochemore.

1. Introduction

The efficiency of string matching algorithms depends on the underlying automaton which represents the string p to be found in the text. Ideally, this automaton A should satisfy the following properties:

1. A is acyclic;
2. A recognizes at least the factors of p;
3. A has the fewer states as possible;
4. A has a linear number of transitions according to m, the length of p. (Such an automaton has at least $m+1$ states.)

The suffix or factor automaton [3, 5] satisfies 1., 2., and 4. but not 3. whereas the subsequence automaton [2] satisfies 1., 2., and 3. but not 4. We present in Section 2 an intermediate structure called *factor oracle*: an automaton with $m+1$ states that satisfies all the above requirements. Section 3 is devoted to the study of a string matching algorithm based on the factor oracle.

2. Construction of the Factor Oracle

The *factor oracle* of a word $p = p_1p_2\dots p_m$, denoted Oracle(p), is the automaton built by the algorithm *Build_Oracle* (Figure 1). All the states of the automaton are final. Figure 2 gives the factor oracle of the word $p = abbbaab$. On this example, the reader will notice that the word aba is recognized whereas it is not a factor of p.

Here are some notations which are used in the following. The set of all prefixes (resp. suffixes) of p is denoted by $\mathrm{Pref}(p)$ (resp. $\mathrm{Suff}(p)$). The word $\mathrm{pref}_p(i)$ is the prefix of length i of p for $0 \le i \le m$. For any $u \in \mathrm{Fact}(p)$, we define

$$\mathrm{poccur}(u,p) = \min\{\, |z| \;\big|\; z = wu \text{ and } p = wuv \,\},$$

the ending position of the first occurrence of u in p. For any $u \in \mathrm{Fact}(p)$, we define the set

$$\mathrm{endpos}_p(u) = \{\, i \mid p = wup_{i+1}\dots p_m \,\}.$$

Build_Oracle($p = p_1 p_2 \dots p_m$)
  For i from 0 to m
    Create a new state i
  For i from 0 to $m-1$
    Build a new transition from i to $i+1$ by p_{i+1}
  For i from 0 to $m-1$
    Let u be a minimal length word in state i
    For all $\sigma \in \Sigma, \sigma \neq p_{i+1}$
    If $u\sigma \in \mathrm{Fact}(p_{i-|u|+1} \cdots p_m)$
      then build a new transition from i to $i + \mathrm{poccur}(u\sigma, p_{i-|u|+1} \cdots p_m)$ by σ

FIGURE 1. High-level construction of the Oracle.

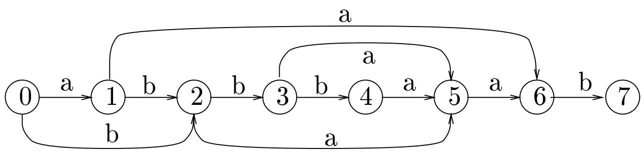


FIGURE 2. Factor oracle of *abbbaab*.

Given two factors u and v of p, we write $u \sim_p v$ if $\mathrm{endpos}_p(u) = \mathrm{endpos}_p(v)$.

The authors prove in [1] the following lemmas.

Lemma 1. *Given a state i of* $\mathrm{Oracle}(p)$*, let $u \in \Sigma^*$ be a minimal length word among the words recognized in i. Then $u \in \mathrm{Fact}(p)$ and $i = \mathrm{poccur}(u, p)$.*

Corollary 1. *For any state i of* $\mathrm{Oracle}(p)$*, there exists an unique minimal length word among the words recognized in state i.*

We denote $\min(i)$ the minimal length word of state i.

Corollary 2. *Let i and j be two states of* $\mathrm{Oracle}(p)$ *such that $j < i$. Then $\min(i)$ cannot be a suffix of $\min(j)$.*

Lemma 2. *Let i be a state of* $\mathrm{Oracle}(p)$*. Then $\min(i)$ is a suffix of any word $c \in \Sigma^*$ which is the label of a path leading from state 0 to state i.*

Lemma 3. *Any word $w \in \mathrm{Fact}(p)$ is recognized by* $\mathrm{Oracle}(p)$ *in a state $j \leq \mathrm{poccur}(w, p)$.*

Corollary 3. *Let $w \in \mathrm{Fact}(p)$. Every word $v \in \mathrm{Suff}(w)$ is recognized by* $\mathrm{Oracle}(p)$ *in a state $j \leq \mathrm{poccur}(w)$.*

Lemma 4. *Let i be a state of* $\mathrm{Oracle}(p)$*. Any path ending by $\min(i)$ leads to a state $j \geq i$.*

Lemma 5. *Let $w \in \Sigma^*$ be a word recognized by* $\mathrm{Oracle}(p)$ *in i. Any suffix of w is recognized in a state $j \leq i$.*

Lemma 6. *The number $T_{Or}(p)$ of transitions in* $\mathrm{Oracle}(p = p_1 p_2 \dots p_m)$ *satisfies $m \leq T_{Or}(p) \leq 2m - 1$.*

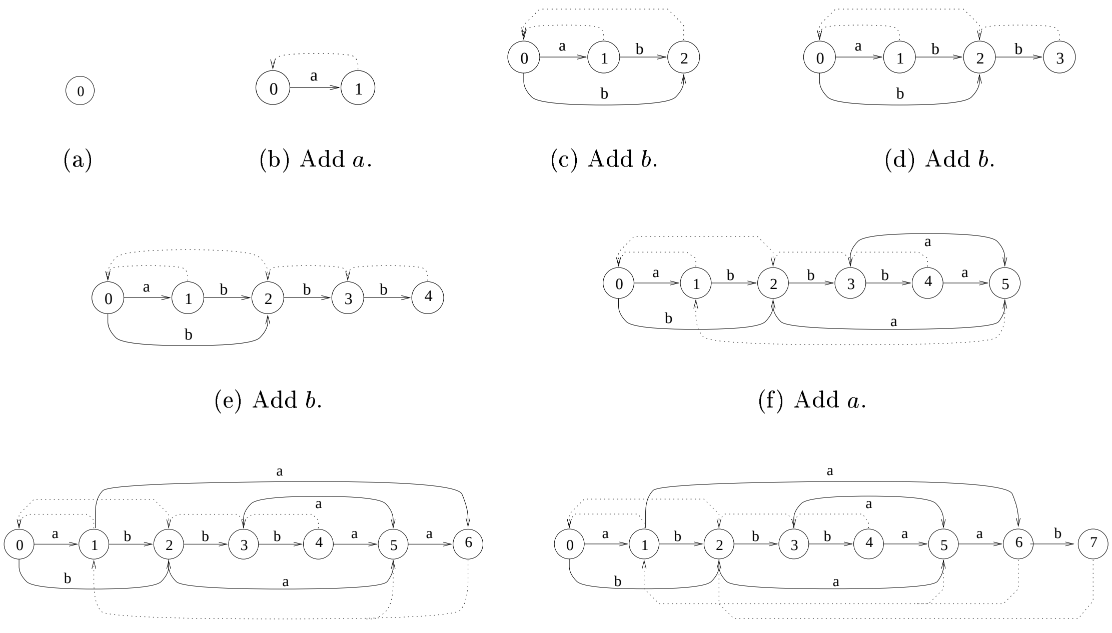
The high-level construction of the factor oracle is equivalent to the on-line algorithm given in Figure 3. An example of this construction is shown in Figure 4.

Exemple. The on-line construction of $\mathrm{Oracle}(abbbaab)$ is given Figure 4.

Fonction add_letter(Oracle($p = p_1 p_2 \dots p_m$), σ)
  Create a new state $m+1$
  Create a new transition from m to $m+1$ labeled by σ
  $k \leftarrow S_p(m)$
  While $k > -1$ and there is no transition from k by σ **Do**
    Create a new transition from k to $m+1$ by σ
    $k \leftarrow S_p(k)$
  End While
  If $(k = -1)$ **Then** $s \leftarrow 0$
  Else $s \leftarrow$ where leads the transition from k by σ.
  $S_{p\sigma}(m+1) \leftarrow s$
  Return Oracle($p = p_1 p_2 \dots p_m \sigma$)

Oracle-on-line($p = p_1 p_2 \dots p_m$)
  Create Oracle(ϵ) with:
    one single state 0
    $S_\epsilon(0) \leftarrow -1$
  For $i \leftarrow 1$ à m **Do**
    Oracle($p = p_1 p_2 \dots p_i$) $\leftarrow$ `add_letter`(Oracle($p = p_1 p_2 \dots p_{i-1}$), p_i)
  End For

FIGURE 3. On-line construction of Oracle($p = p_1 p_2 \dots p_m$).



(a) (b) Add a. (c) Add b. (d) Add b.

(e) Add b. (f) Add a.

(g) Add a. (h) Add b.

FIGURE 4. On-line construction of Oracle($abbaba$).

3. String Matching

The authors replace the suffix automaton with a factor oracle in the BDM (for *backward dawg matching*) [4, 6], obtaining the BOM (for *backward oracle matching*) algorithm.

The BOM algorithm consists in shifting a window of size m on the text. For each new position of this window, the factor oracle of the mirror image of p is used to search the suffix of the window from right to left. The basic idea is that if this backward search fails on a letter σ after the reading of a word u then σu is not a factor of p and the beginning of the window can be shifted just after σ. The worst-case complexity of BOM is $O(nm)$.

The average complexity of the original BDM is in $O\big(n \log_{|\Sigma|}(m)/m\big)$ under a uniform Bernoulli model. In view of the experimental results (see [1]), the authors claim that their new BOM algorithm is also optimal on average:

Conjecture 1. *Under a model of independence and equiprobability of letters, the BOM algorithm has an average complexity of* $O\big(n \log_{|\Sigma|}(m)/m\big)$.

The authors show in [1] how to obtain a linear (in n) worst case algorithm from the BOM.

Bibliography

[1] Allauzen (Cyril), Crochemore (Maxime), and Raffinot (Mathieu). – *Oracle des facteurs, oracle des suffixes.* – Technical Report n° 99-08, Institut Gaspard-Monge, Université de Marne-la-Vallée, 1999. Available from `http://www-igm.univ-mlv.fr/~raffinot/ftp/IGM99-08.ps.gz`.

[2] Baeza-Yates (Ricardo A.). – Searching subsequences. *Theoretical Computer Science*, vol. 78, n° 2 (Part A), 1991, pp. 363–376.

[3] Blumer (A.), Blumer (J.), Haussler (D.), Ehrenfeucht (A.), Chen (M. T.), and Seiferas (J.). – The smallest automaton recognizing the subwords of a text. *Theoretical Computer Science*, vol. 40, n° 1, 1985, pp. 31–55. – Special issue: Eleventh international colloquium on automata, languages and programming (Antwerp, 1984).

[4] Crochemore (M.), Lecroq (T.), Czumaj (A.), Gasieniec (L.), Jarominek (S.), Plandowski (W.), and Rytter (W.). – Speeding up two string-matching algorithms. *Algorithmica*, vol. 12, n° 4-5, 1994, pp. 247–267.

[5] Crochemore (Maxime). – Transducers and repetitions. *Theoretical Computer Science*, vol. 45, n° 1, 1986, pp. 63–86.

[6] Crochemore (Maxime) and Rytter (Wojciech). – *Text algorithms.* – The Clarendon Press Oxford University Press, New York, 1994, xiv+412p.

Part V

Miscellany

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), p. 127.

Available online at the URL
http://algo.inria.fr/seminars/.

On Random Graph Homomorphisms into $\mathbb{Z}$

Elchanan Mossel
Institute of Mathematics, Hebrew University of Jerusalem

November 15, 1999

Abstract

The study of Lipschitz functions on graphs and metric spaces is rather advanced. Uniform measure on graph homomorphisms into $\mathbb{Z}$ provides a model for looking at typical Lipschitz functions. Given a bipartite connected finite graph $G = (V, E)$ and a vertex $v_0 \in V$, we consider a uniform probability measure on the set of graph homomorphisms $f : V \to \mathbb{Z}$ satisfying $f(v_0) = 0$. This measure can be viewed as a G-indexed random walk on $\mathbb{Z}$, generalizing both the usual time-indexed random walk and tree-indexed random walk. We will present several general inequalities for G-indexed random walks, including an upper bound on fluctuations implying that the distance $d\bigl(f(u), f(v)\bigr)$ between $f(u)$ and $f(v)$, is stochastically dominated by the distance to 0 of a simple random walk on $\mathbb{Z}$ having run for $d(u, v)$ steps. We will also discuss various special cases, some conjectures and algorithmic aspects of these models.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 129–132.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Distributional Analysis of Recursive Algorithms by the Contraction Method

Ralph Neininger
University of Freiburg

November 22, 1999

Summary by Elchanan Mossel

1. Basic Algorithms

We consider *records* which belong to a k-dimensional region $D = D_1 \times \cdots \times D_k \subset \mathbb{R}^k$. A *file* is a finite subset F of D. Given a *query* $q \in (D_1 \cup \{*\}) \times \cdots \times (D_k \cup \{*\})$, the objective is to find all the records $r \in F$ such that $r_i = q_i$ when $q_i \neq *$. The probabilistic assumption is that all the coordinates of the records and the queries (which are not $*$) are independent uniform random variables. For the discussion below, it is easy to see that this assumption may be replaced by a weaker assumption that all variables are independent with the same continuous distribution. The *specification pattern* consists of the configuration in $\{*, S\}^k$ of specified and unspecified variables.

There exist several comparison-based trees:

- *Quadtrees.* Each record x has 2^k subtrees which correspond to all possible elements of $\{<, >\}^k$. Thus (y_0, y_1, y_2) will belong to the $(<, >, <)$ subtree of (x_0, x_1, x_2) if $y_0 < x_0, y_1 > x_1, y_2 < x_2$. See Figure 1.
- *kD trees.* Each record x at level l has two subtrees corresponding to $x_{l \bmod k} > y_{l \bmod k}$ and $x_{l \bmod k} < y_{l \bmod k}$, respectively.
- *Randomised kD trees.* Each record x at level l has two children corresponding to $x_{l(x)} > y_{l(x)}$ and $x_{l(x)} < y_{l(x)}$ where $l(x)$ are i.i.d. uniform variables in the range $0, \ldots, k-1$.

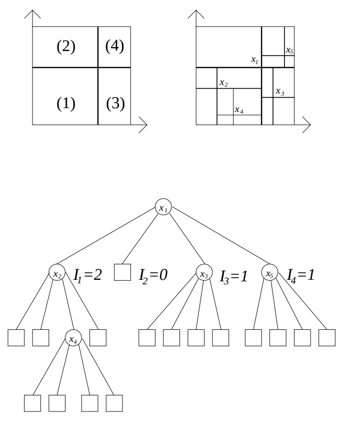


FIGURE 1. A quadtree: the data partition the unit-cube recursively into quadrants; the quadtree corresponds to this partitioning.

– *Squarish kD trees.* This is another version in which every node has two subtrees, but the coordinate with respect to which we split depends more strongly on the tree structure.

The basic quantity we are after is the limit law of C_n where C_n is the random number of nodes we traverse when finding all records which match the query. Here n is the size of F.

2. Quadtrees in Two Dimensions

We let $W = (U, V)$ be the first key to be inserted and $q = (Y, *)$ be the query. So U, V, and Y are uniform i.i.d. variables. We also let I^n be the vector of cardinalities for the subtrees of the root. We thus derive the following recursive distributional equation:

$$C_n = 1_{Y<U}\left(C^1_{I^n_1} + C^2_{I^n_2}\right) + 1_{Y>U}\left(C^3_{I^n_3} + C^4_{I^n_4}\right) + 1,$$

wherein the variables Y, U, V, and C^j_i are independent and all the C^j_i have the distribution of C_i. Given (U, V) the variable I^n is multi-monomial with parameters (U, V) and n.

By previous works [1, 2, 6] there are known constants α, β, and γ for which

$$\mathbf{E}[C_n] \sim \gamma n^{\alpha-1}, \qquad \mathbf{Var}[C_n] \sim \beta n^{2\alpha-2}.$$

Looking for a limit, we consider the variable: $X_n = (C_n - \mathbf{E}[C_n])/n^{\alpha-1}$.

In this way we obtain the equation:

$$\begin{aligned}(1)\quad X_n = {} & 1_{Y<U}\left(\left(\frac{I^n_1}{n}\right)^{\alpha-1}\left(X^1_{I^n_1} + \gamma\right)\right) + 1_{Y<U}\left(\left(\frac{I^n_2}{n}\right)^{\alpha-1}\left(X^2_{I^n_2} + \gamma\right)\right) \\ & + 1_{Y>U}\left(\left(\frac{I^n_3}{n}\right)^{\alpha-1}\left(X^3_{I^n_3} + \gamma\right)\right) + 1_{Y>U}\left(\left(\frac{I^n_4}{n}\right)^{\alpha-1}\left(X^4_{I^n_4} + \gamma\right)\right) - \gamma + o(1).\end{aligned}$$

By the law of large numbers we have

$$I^n/n \to W = \big(UV, U(1-V), (1-U)V, (1-U)(1-V)\big)$$

in probability. We thus obtain the following limiting equation:

$$\begin{aligned}(2)\quad X = {} & 1_{Y<U}W_1^{\alpha-1}(X^1 + \gamma) + 1_{Y<U}W_2^{\alpha-1}(X^2 + \gamma) \\ & + 1_{Y>U}W_3^{\alpha-1}(X^3 + \gamma) + 1_{Y>U}W_4^{\alpha-1}(X^4 + \gamma) - \gamma,\end{aligned}$$

where the X^i are independent copies of X.

This suggests that we should consider the following operator on random variables Z:

$$\begin{aligned}T(Z) = {} & 1_{Y<U}W_1^{\alpha-1}\left(Z^1 + \gamma\right) + 1_{Y<U}W_2^{\alpha-1}\left(Z^2 + \gamma\right) \\ & + 1_{Y>U}W_3^{\alpha-1}\left(Z^3 + \gamma\right) + 1_{Y>U}W_4^{\alpha-1}\left(Z^4 + \gamma\right) - \gamma,\end{aligned}$$

where the Z^i's are independent copies of Z.

We now work with the following metric (on the space of variables with zero mean and finite variance): $l_2(Z, Z') = \inf\big(\mathbf{E}[Z - Z']^2\big)^{1/2}$ where the infimum is taken over all couplings of Z and Z'. It turns out that this space equipped with this metric is a Banach space. Moreover, using the representation of T one can see that T is a contraction on this space. It therefore follows that there exists a unique random variable Z which satisfies $T(Z) = Z$.

The main technical part of the proof is showing that we obtain the same limit if we work with the exact equations (1) instead of the approximate equations (2). This essentially uses the known estimates that $\mathbf{E}[C_n] = \gamma n^{\alpha-1}\big(1 + o(1)\big)$. In this way we obtain the following theorem.

Theorem 1. *Let X_n be the normalised number of traversed nodes and X the variable such that $T(X) = X$, then $l_2(X_n, X) \to 0$.*

3. Other Trees

3.1. Multidimensional quadtree. In a similar manner one can prove the same kind of result for multidimensional quadtree. One of the differences is that in this case the variance $\mathbf{Var}[C_n]$ is not known beforehand. Instead, we guess that the right normalisation should be

$$X_n = \frac{C_n - \mathbf{E}[C_n]}{n^{\alpha-1}}.$$

In this way we obtain again a limit law similar to the above: the limit X depends only on the number of $*$'s in the query. Given this limit law we can now compute a constant which depends only on the number of $*$'s in the query such that $\mathbf{Var}[C_n] = \beta n^{2\alpha-2}$.

3.2. kD Trees. Vaguely speaking, the difference between quadtrees and kD trees, is that for kD trees different levels behave differently. Thus, in order to obtain a theorem similar to the above, a single recursion step should go k levels forward instead of just one. Doing that, we obtain a result similar to the above.

3.3. Randomised kD tree. The randomisation allows one to use one-level recursion, therefore obtaining a theorem and a proof similar to the case of quadtrees.

3.4. Squarish kD tree. It seems like the above methods do not work in this case. This is because the coordinate with respect to which we split depends on the structure of the tree and on the data stored in it.

4. Internal Path Length in Random Trees

In the previous sections we studied the cost of a query. In this section we consider the cost of building the tree which is nothing but the sum of depths of nodes in the tree. For the quadtrees we obtain the following recursive equation:

$$Y_n = \sum_{k=0}^{2^d-1} Y^k_{I^n_k} + n.$$

The article [3] gives the expectation $\mathbf{E}[Y_n] = (2/d)\, n \ln n + u_d n + o(n)$, but the variance was not derived there. We guess the normalisation: $X_n = \big(Y_n - \mathbf{E}[Y_n]\big)/n$. We therefore obtain the equation:

$$X_n = \sum_{i=0}^{2^d-1} \frac{I^n_k}{n} X^k_{I^n_k} + C_n(I^n)$$

where

$$C_n(i_0, \dots, i_{2^d-1}) = 1 + \frac{1}{n} \sum_{i=0}^{2^d-1} \mathbf{E}[Y_{i_k}] - \mathbf{E}[Y_n].$$

Using the expectation formula we obtain:

$$C_n(i) = 1 + \frac{2}{d} \sum_{i=0}^{2^d-1} \frac{i_k}{n} \ln \frac{i_k}{n} + o(1).$$

We now continue in the same route as before to obtain the l_2 limit and the asymptotic variance.

5. Find Algorithm

We consider the following version of quicksort. We want to sort the values $\{1, \ldots, n\}$ which are given in a random uniform permutation. In order to perform the sort we pick a pivotal element p and continue sorting the elements larger than this element, and the elements smaller than this element. The way to pick p is by taking three independent uniform keys k_1, k_2, k_3 and taking p to be their median. We thus obtain the following recursion equation:

$$C_n = 1_{Z_n > M_n} C'_{Z_{n-1}} + 1_{Z_n < M_n} C''_{n-Z_n} + n - 1$$

where M_n is uniform in $\{1, \ldots, n\}$ and Z_n is a median of three uniform variables in $\{1, \ldots, n\}$. We now continue in a similar way: it is known [4, 5] that $\mathbf{E}[C_n] = 5n/2 + O(\ln n)$, we guess that the normalisation is: $Y_n = \big(C_n - \mathbf{E}[C_n]\big)/n$ to obtain a limit law. This limit law enables us to give asymptotic form for all the moments: $\mathbf{E}[C_n^k] \sim m_k n^k$ where we have a closed formula for m_k.

Bibliography

[1] Flajolet (Philippe), Gonnet (Gaston), Puech (Claude), and Robson (J. M.). – The analysis of multidimensional searching in quad-trees. In *Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1991)*. pp. 100–109. – ACM, New York, 1991.

[2] Flajolet (Philippe), Gonnet (Gaston), Puech (Claude), and Robson (J. M.). – Analytic variations on quadtrees. *Algorithmica*, vol. 10, n° 6, 1993, pp. 473–500.

[3] Flajolet (Philippe), Labelle (Gilbert), Laforest (Louise), and Salvy (Bruno). – Hypergeometrics and the cost structure of quadtrees. *Random Structures & Algorithms*, vol. 7, n° 2, 1995, pp. 117–144.

[4] Kirschenhofer (P.), Prodinger (H.), and Martínez (C.). – Analysis of Hoare's FIND algorithm with median-of-three partition. *Random Structures & Algorithms*, vol. 10, n° 1-2, 1997, pp. 143–156. – Average-case analysis of algorithms (Dagstuhl, 1995).

[5] Kirschenhofer (Peter), Martínez (Conrado), and Prodinger (Helmut). – Analysis of an optimized search algorithm for skip lists. *Theoretical Computer Science*, vol. 144, n° 1-2, 1995, pp. 199–220. – Special volume on mathematical analysis of algorithms.

[6] Martínez (Conrado), Panholzer (Alois), and Prodinger (Helmut). – On the number of descendants and ascendants in random search trees. *Electronic Journal of Combinatorics*, vol. 5, n° 1, 1998, pp. Research Paper 20, 36 pp.

[7] Neininger (Ralph). – Asymptotic distributions for partial match queries in kD trees. – 1999. Preprint.

[8] Neininger (Ralph) and Rüschendorf (Ludger). – On the internal path length of d-dimensional quad trees. *Random Structures & Algorithms*, vol. 15, n° 1, 1999, pp. 25–41.

Algorithms Seminar 1999–2000,
F. Chyzak (ed.), INRIA, (2000), pp. 133–136.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Analytic Information Theory and the Redundancy Rate Problem

Wojciech Szpankowski
Department of Computer Sciences, Purdue University

February 13, 2000

Summary by Philippe Flajolet

1. Information, Entropy, and Codes

One of the most basic problems of information theory [1] is that of *source coding.* A source is by definition a mechanism that produces messages over a finite alphabet $\mathcal{A}$, a message of length n being conventionally denoted by $x_1^n = (x_1, \ldots, x_n)$. A code C is a translation mechanism (an injective function, an algorithm) that, for each n, takes as input a message from $\mathcal{A}^n$ and transforms it into a binary sequence. Such a translation is thus a fixed-length to variable-length encoding.

Messages have some structure. For the English language source, the sequence 'Rzqxwa gkvzzxq wzd aaaaaaa rxbleurp' is much less likely than the sequence 'It rained yesterday over England'. Indeed, some letters are more frequent than others, certain letter combinations are impossible, etc. It is then customary to try and capture the principal features of the source by some probability distribution of sorts over $\mathcal{A}^n$. The main models considered in the talk are the following.

M1. A *memoryless model* considers letters as independent identically distributed random variables, with letter $i \in \mathcal{A}$ having probability p_i. (This is sometimes called the Bernoulli model.)

M2. A *Markov model* assumes an underlying finite set of states with transition probability $p_{i,j}$ between states i and j and a mapping from states to letters.

As discovered by Shannon around 1949, information is measured by entropy. The entropy of a probability distribution $P = \{p_s\}_{s\in S}$ over any finite set S is defined as

$$H(P) := -\sum_{s\in S} p_s \lg p_s,$$

where $\lg x = \log_2 x$. (Roughly, the definition extends the fact that an element in a set of cardinality m needs to be encoded by about $\lg m$ bits in order to be distinguished from its companions elements.) Most "reasonable" source models have an *entropy rate* h; namely, if x_1^n is randomly drawn according to the source model P, then the following limit exists,

$$h = \lim_{n\to\infty} -\frac{1}{n} \sum_{x_1^n \in \mathcal{A}^n} P(x_1^n) \lg P(x_1^n).$$

For instance, the entropy rate of a sequence drawn according to the memoryless model equals the entropy of the distribution of individual characters. For a Markov chain with transition probabilities $p_{i,j}$, the entropy rate is

$$h = \sum_i \pi_i \sum_j p_{i,j} \lg p_{i,j},$$

with π_i the stationary probability distribution of the chain. The entropy rate of written English is estimated to be about 1.3 bits per character.

Sources produce messages which are not uniformly random and this lies at the basis of data compression—the fact that one may find codes that tend to be shorter than the original message. (E.g., the present summary is compressed by `gzip` at a rate of about 3.5 bits per character.) We cannot compress arbitrarily however. The most fundamental theorem of information is due to Shannon and asserts the following: *You cannot beat entropy. In other words, any code has an expected length per character that is at least as large as the entropy rate of the source.*

Another famous theorem of Shannon goes the other direction and asserts: *The entropy rate is asymptotically achievable.* This leaves plenty of room for algorithmic design. As a matter of fact, coding algorithms separate into two groups: *(i)* codes that are designed for a specific (known) probability distribution over the inputs; *(ii)* universal codes that do not assume such a probabilistic distribution to be known *a priori* and do their best to come close to the optimum over an entire class of models. Amongst the first group, we find Huffman codes [3, pp. 402–406] and Shannon–Fano[1] codes [1, pp. 101–103]. Amongst the second group, the best known algorithms are the ones due to Lempel and Ziv[2] in 1977 and 1978.

2. Redundancy of Classical Codes

The codes normally considered are at least near-optimal with respect to the entropy lower bound. Define first the pointwise redundancy of a code C with respect to a model P as

$$R_n(C,P;x_1^n) := L\big(C(x_1^n)\big) + \lg P(x_1^n),$$

where L is length. Two critical parameters are then the *average redundancy* and the *maximal redundancy* defined by

$$\bar{R}_n(C,P) = E\big[R_n(C,P;x_1^n)\big], \qquad R_n^*(C,P) = \max_{x_1^n}\big[R_n(C,P;x_1^n)\big]. \tag{1}$$

where both average and maximum are meant with respect to x_1^n. In other words, the question asked is: *How far are we from the information theoretic optimum, either on average or in the worst case?* There, we assume the source distribution to be known and the code to be fixed, and analyse the redundancy parameters of the given code.

In this perspective, the talk first reviews results relative to the classical Huffman code and to a version of Fano–Shannon codes, this in the case of a memoryless source. Redundancy is then $O(1)$ but with fluctuations that depend on the fine arithmetic structure of the parameters of the model under consideration; see Figure 1. The methods use Fourier analysis and *Gleichverteilung* mod 1.

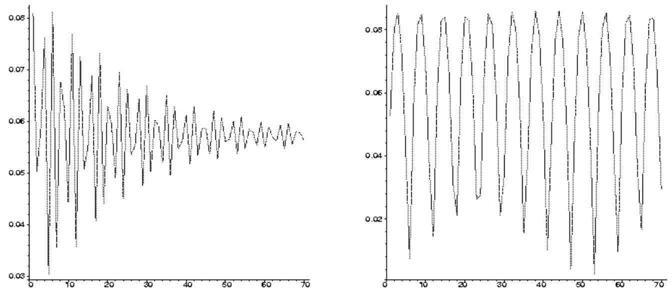
Louchard and Szpankowski (1997), Savari (1997), Wyner (1998), and Jacquet–Szpankowski (1995) proved that the Lempel–Ziv algorithms under either a memoryless or a Markov model have rates that are $\Theta(n/\log n)$ for LZ'78[3] and $\Theta(n \log\log n/\log n)$ for LZ'77. The proofs provide detailed asymptotic information on the redundancy. The results again involve subtle fluctuations. The analysis is close to that of digital tries, with Mellin transforms playing a prominent rôle.

---

[1]To design a Shannon–Fano code for the distribution P on S, partition S as $S = S_0 \cup S_1$ in such a way that the probabilities of S_0 and S_1 differ by as little as possible from $1/2$. All elements of S_j are assigned a code that starts with j. Proceed recursively.

[2]Roughly, the LZ algorithms recognize, as characters flow, the frequently repeated blocks of letter and avoid copying these over and over again, but instead output pointers to the location of the first occurrence of such a block.

[3]LZ'78 parses a sequence into "phrases" and outputs a pointer to the longest phrase already encountered; LZ'77 outputs a pointer to the longest factor already encountered.



(a) (b)

FIGURE 1. Huffman code redundancy for a memoryless source with control parameter $\alpha = \lg(1/p - 1)$: (a) irrational case ($p = 1/\pi$); (b) rational case ($p = 1/9$).

3. Minimax Redundancy for Classes of Source Models

The *strong redundancy-rate problem* asks what can be achieved when the source model ranges over a whole class of sources $\mathcal{S}$. Thus, the source model is a bit constrained but basically unknown and the question becomes information-theoretic rather than algorithmic (no coding algorithm is fixed any more). Consider redundancies in the sense of (1) and define the *minimax redundancies*,

$$\bar{R}_n(\mathcal{S}) = \min_C \max_{P\in\mathcal{S}} \bar{R}_n(C,P), \qquad R_n^*(\mathcal{S}) = \min_C \max_{P\in\mathcal{S}} \bar{R}_n^*(C,P), \tag{2}$$

corresponding to an average-case or a worst-case scenario, respectively. By their definitions, these quantities represent the additional cost on top of entropy incurred (at least) by any code (this is $\min_C$) in order to be able to cope with *all* sources (this is $\max_{P\in S}$).

It would seem that the minimax problem of estimating the quantities in (2) is intractable. However, Shtarkov proved in 1978 that the (worst-case) minimax redundancy is narrowly bounded by the (Shtarkov) inequalities

$$\lg \sum_{x_1^n\in\mathcal{A}^n} \sup_{P\in S} P(x_1^n) \le R_n^*(\mathcal{S}) \le 1 + \lg \sum_{x_1^n\in\mathcal{A}^n} \sup_{P\in S} P(x_1^n). \tag{3}$$

There the quantity $\sup P(x_1^n)$ could be termed a "maximum likelihood coefficient" since it describes the probability of any individual realization x_1^n under the model $P \in \mathcal{S}$ that assigns to it the highest probability. Take for instance a binary word $x_1^n \in \{0,1\}^n$ comprising k letters 0 and $n-k$ letters 1, and $\mathcal{S}$ the class of all memoryless models with $\mathbf{P}(0) = p$ and $\mathbf{P}(1) = 1-p$. Clearly, the maximum likelihood coefficient is given by the Bernoulli distribution whose parameter is $p = k/n$ (maximum likelihood probabilities equal frequencies), and its value is $\big(k/n\big)^k \big((n-k)/n\big)^{n-k}$. The sum appearing in (3) then evaluates to

$$A_n := \sum_{x_1^n\in\mathcal{A}^n} \sup_{P\in S} P(x_1^n) = \sum_{k=0}^{n} \binom{n}{k} \left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}.$$

This has the same flavour as Abel's identities. Indeed, we have

$$A_n = \frac{n!}{n^n}\,[z^n]\frac{1}{\big(1-T(z)\big)^2} \quad \text{where} \quad T(z) = ze^{T(z)}$$

is the tree function. It is then a simple matter, by singularity analysis of the tree function, to get

$$A_n \sim \frac{1}{2}\,\frac{n!e^n}{n^n} \sim \sqrt{\frac{\pi n}{2}} \quad \text{and} \quad \lg A_n = \frac{1}{2}\lg n + \lg\sqrt{\frac{\pi}{2}} + o(1).$$

The quantity $\lg A_n$ is at most 1 from the minimax redundancy as results from inequalities (3).

Renewal sources. Another topic of the talk is to analyse redundancy for the class of renewal sources defined as follows.

M3. A renewal model starts with a random sequence $(x_i)_{-\infty}^{+\infty}$ of '0's and '1's, infinite in both directions and such that the spacings between the '1's are independent identically distributed random variables. Then extract the window corresponding to $x_1^n = x_1 \dots x_n$. (You're sitting under a bus shelter and record every minute whether you're seeing a bus passing or not.)

This class of sources makes for an interesting study since minimax redundancy turns out to be $O(\sqrt{n})$; see [2] for a complete analysis.

The maximal likelihood approach leads to the consideration of the sum

$$r_n = \sum_k \sum_{\mathcal{P}(n,k)} \binom{k}{k_0, \dots, k_{n-1}} \left(\frac{k_0}{k}\right)^{k_0} \left(\frac{k_1}{k}\right)^{k_1} \cdots \left(\frac{k_{n-1}}{k}\right)^{k_{n-1}}.$$

There, the summation condition $\mathcal{P}(n,k)$ is $n = k_0 + 2k_1 + \cdots$, $k = k_0 + k_1 + \cdots$. The computation heavily involves the tree function $T(z)$ and proceeds in several steps.

First, one disposes of the normalizing factor of $k!/k^k \sim e^{-k}\sqrt{2\pi k}$ by introducing as an artefact a random variable K_n and relating r_n to $E\big[\sqrt{2\pi K_n}\,\big]$. Second, the distribution of K_n is described by the bivariate generating function

$$S(z,u) := \prod_{i=1}^{\infty} \beta(z^i u) \quad \text{where} \quad \beta(z) = \frac{1}{1 - T(ze^{-1})}.$$

This has roughly the character of (the square root of) a partition generating function with u marking the number of parts. Third, the saddle-point method is applied to extract coefficients. Fourth, the saddle-point analysis conduces to a local analysis near 1 that is solved by Mellin transform techniques. The eventual result is that

$$\lg r_n = \frac{2}{\log 2}\sqrt{\left(\frac{\pi^2}{6} - 1\right) n} - \frac{5}{8}\lg n + \frac{1}{2}\lg\log n + O(1),$$

and this quantity closely approximates the minimax redundancy of renewal sources by Shtarkov's inequalities. Note the asymptotic form $r_n \approx e^{\sqrt{n}}$ that is typical of partition estimates.

Conclusion. The redundancy problem is typical of situations where second-order asymptotics are essential. Such problems of information theory are thus candidates *par excellence* for the methods of *analytic information theory.* By this, it is meant the study of randomness in words and codes by means of the classical methods of analytic combinatorics. The reader interested in these questions will be well-advised to consult the forthcoming book by Szpankowski [4] and references therein.

Bibliography

[1] Cover (Thomas M.) and Thomas (Joy A.). – *Elements of information theory.* – John Wiley & Sons Inc., New York, 1991, xxiv+542p. A Wiley-Interscience Publication.

[2] Flajolet (Philippe) and Szpankowski (Wojtek). – *Analytic Variations on Redundancy Rates of Renewal Processes.* – Research Report n° 3553, Institut National de Recherche en Informatique et en Automatique, 1998. 10 pages. Submitted to IEEE Transactions on Information Theory.

[3] Knuth (Donald E.). – *The art of computer programming.* – Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, 1997, third edition, xx+650 pp.p. Volume 1: Fundamental algorithms.

[4] Szpankowski (Wojciech). – *Average-case analysis of algorithms on sequences.* – John Wiley & Sons Inc. To appear, preliminary version available from `http://www.cs.purdue.edu/people/spa`.