INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

Algorithms seminar, 1995-1996

Bruno SALVY, éditeur scientifique

N° 2992

Septembre 1996

THÈME 2

Rapport de recherche

ISSN 0249-6399

ALGORITHMS SEMINAR,

1995-1996

Bruno Salvy

(Editor)

Abstract

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

SÉMINAIRE ALGORITHMES,

1995-1996

Abstract

Ces notes de séminaires représentent les actes, en anglais, d'un séminaire consacré à l'analyse d'algorithmes et aux domaines connexes. Les thèmes abordés comprennent : combinatoire, calcul formel, analyse asymptotique et analyse en moyenne d'algorithmes et de structures de données.

ALGORITHMS SEMINAR
1995–1996

Bruno Salvy[1]
(Editor)

Abstract

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

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

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

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

Computer algebra plays an increasingly important rôle in this area. 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.

Asymptotic analysis is an essential ingredient in the interpretation of quantitative results supplied by the resolution of combinatorial models. Various asymptotic methods are found to be relevant to the analysis of particular algorithms.

The thirty-two 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

The enumeration of self-avoiding walks in dimension d is a very old open problem of combinatorics. In **[1]**, a related simpler problem is solved. A class of partitions of integers having nice and surprising generating functions is studied in **[2]**. An introduction to symmetric functions, together with work involving q-analogues of the Catalan numbers is given in **[3]**. Sums of powers of harmonic numbers divided by powers of the variable are related to special values of Riemann's ζ function. A uniform approach to the computation of these sums is given in **[4]**. A logics viewpoint on some combinatorial objects is taken in **[5]**. The last summary **[6]** takes a formal language approach to problems related with the study of DNA sequences.

[1] This work was supported in part by the Long Term Research Project Alcom-IT (#20244) of the European Union.

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

[1] Three-Dimensional Convex Polygons. *Mireille Bousquet-Mélou*

[2] Lecture Hall Partitions. *Mireille Bousquet-Mélou*

[3] Déterminants, nombres de Catalan et fonctions symétriques de Macdonald. *Dominique Gouyou-Beauchamps*

[4] Sommes d'Euler. *Philippe Flajolet*

[5] A 0-1 Law for Planar Maps. *Kevin Compton*

[6] Grammaires et séquences biologiques. *Fabrice Lefebvre*

PART II. SYMBOLIC COMPUTATION

This part starts with a survey [7] of numerous algorithms related to linear recurrences and linear differential equations, mostly in the univariate case. New algorithms for the multivariate case are described in [8] and [9]. The numerical resolution of systems of polynomials is studied from different viewpoints in [10] and [11]. An algorithm from computational number theory is developed in [12]. The next two summaries study specific problems: [13] answers the question of describing the functions satisfying all the differential equations satisfied by a given function; [14] describes polynomials analogous to the Chebyshev polynomials, but much harder to compute. This part ends with a short presentation of the computation of Padé approximants of various kinds [15].

[7] Linear Recurrences, Linear Differential Equations and Fast Computation. *Bruno Salvy*

[8] Creative Telescoping and Applications. *Frédéric Chyzak*

[9] ∂-Finite Functions. *Frédéric Chyzak*

[10] Computing the Distance of a Point to an Algebraic Hypersurface and Application to Exclusion Methods. *Xavier Gourdon*

[11] Méthodes matricielles pour la résolution de systèmes algébriques. *Ioannis Z. Emiris*

[12] Le calcul de grandes valeurs de la fonction $\pi(x)$. *Marc Deléglise*

[13] Sur un problème de Rubel. *John Shackell*

[14] Les polynômes de Tchébychev entiers. *Bruno Salvy*

[15] Algebraic Computation of Matrix-like Padé Approximants. *George Labahn*

PART III. ASYMPTOTIC ANALYSIS

The asymptotic analysis of a class of staircase polygons is studied in [16]. It involves a nonlinear q-equation for the generating function, and asymptotics where the Airy function arises. The relevance of this and similar problems to statistical mechanics is the topic of [17]. Partitions of integers give rise to very subtle asymptotic analyses. A historical survey of the litterature in that area is given in [18]; and [19] studies a specific problem. Asymptotic techniques from probability theory are used in [20] to study a network that models self-service electrical car pools.

[16] The Tricritical Scaling Function of Partially Directed Vesicles. *Thomas Prellberg*

[17] The Statistical Mechanics of Vesicles. *Thomas Prellberg*

[18] Partitions d'entiers et méthode de Meinardus. *Philippe Dumas*

[19] Measures of distinctness for partitions and compositions. *Hsien-Kuei Hwang*

[20] Étude asymptotique et charge critique pour les grands réseaux fermés à forme produit via le théorème de la limite centrale. *Jean-Marc Lasgouttes*

PART IV. ANALYSIS OF ALGORITHMS AND DATA STRUCTURES

The Quickselect algorithm uses the partitioning process of Quicksort to find the k-th element among n without sorting them. Its average-case analysis is described in [21], as well as analyses of variants of Quickselect. Next, [22] shows the relevance of basic hypergeometric series to the analysis of digital search trees and of an approximate counting algorithm. Tools from probability theory are used in the analysis of bin-packing **[23]**. In **[24]**, a problem from computational learning theory is attacked with urn models and involves modified Bessel functions. The last four papers **[25–28]** are related to pattern-matching and strings.

[21] Analysis of Quickselect. *Helmut Prodinger*
[22] Basic hypergeometric series, digital search trees, approximate counting. *Helmut Prodinger*
[23] Biased Random Walks, Lyapunov Functions, and Stochastic Analysis of Best Fit Bin Packing. *Claire Kenyon*
[24] Un modèle d'urnes pour l'apprentissage. *Danièle Gardy*
[25] Pattern Matching Image Compression: Theory, Algorithms and Experiments. *Wojciech Szpankowski*
[26] Fast Approximate Pattern Matching. *Ricardo Baeza-Yates*
[27] Rotation of Periodic Strings and Short Superstring. *Dany Breslauer*
[28] Recherche de motifs : combinatoire et probabilités. *Mireille Régnier*

PART V. MISCELLANY

Worst-case analyses of algorithms give rise to equations with a max operator. An algebraic framework for such equations is surveyed in **[29]**. Next, **[30]** gives an introduction to DNA computers and the biology involved in them. Applications of the Mellin transform in signal processing are listed in **[31]**. The last summary **[32]** is concerned with random boolean formulæ.

[29] Le semi-anneau (max,+) : une introduction. *Stéphane Gaubert*
[30] Computation with DNA. *Alain Hénaut and Didier Contamine*
[31] Utilisation de la transformée de Mellin en traitement de signaux fractals. *Jacques Lévy-Vehel*
[32] Évolution de la satisfiabilité et de la difficulté de formules booléennes aléatoires. Applications pour la résolution. *Olivier Dubois*

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, François Morain and Bruno Salvy) and the greater Paris area—especially École Polytechnique (Jean-Marc Steyaert), University of Paris Sud at Orsay (Dominique Gouyou-Beauchamps) and LITP (Michèle Soria).

The editor expresses his gratitude to the various persons who actively supported this joint enterprise, most notably: Philippe Dumas for his careful rereading, Eithne Murray for straightening out some English texts, and Frédéric Chyzak, Pierre Nicodème and Philippe Dumas for writing more than their share of summaries. 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
B. Salvy

Part 1

Combinatorics

Three-Dimensional Convex Polygons

Mireille Bousquet-Mélou
LaBRI, Université Bordeaux 1

February 26, 1996

[summary by Eithne Murray]

Abstract

A method to enumerate self-avoiding convex polygons, which in theory will work for all dimensions, is presented. The generating series for polygons of dimensions 2 (already known) and 3 are given. They are both the quotients of two D-finite series, and it appears that this property might hold for higher dimensions.

1. Introduction

A very old open problem is to enumerate self-avoiding walks (self-avoiding polygons) in dimension d. This talk answers a slightly more restricted problem by presenting a method of enumerating convex self-avoiding polygons. The 2-dimensional case has already been solved in [3] and [6], but this method works in higher dimensions, and provides a combinatorial interpretation of the 2-dimensional result.

Some basic definitions are required. An *(oriented) polygon* of perimeter $2n$ is a closed path $(s_1, s_2, \ldots, s_{2n})$ of vertices on $\mathbb{Z}^d$ such that s_i and s_{i+1} are neighbours for $1 \leq i \leq 2n$ and $s_{2n+1} = s_1$. It is defined up to cyclic permutations of its vertices. The *rooted* polygon $(s_1, s_1, \ldots, s_{2n})$ *represents* all the polygons formed by the cyclic permutations. A *self-avoiding polygon* is such that $s_i \neq s_j$ for $1 \leq i \neq j \leq 2n$; in other words, it never crosses itself except at the start/end point. A non-empty self-avoiding polygon is also called a *loop*. Note that the polygon (s_1, s_2) is a loop.

Polygons are often represented as words over an alphabet. This representation means the polygons are defined up to a translation in $\mathbb{Z}^d$, which is a requirement for counting them, and also gives a convenient method to define additional properties of the polygons. Thus a rooted polygon of perimeter $2n$ will often be regarded as a word $u = u_1 u_2 \cdots u_{2n}$ on the alphabet $\mathcal{A} = \{1, 2, \ldots, d\} \cup \{\bar{1}, \bar{2}, \ldots, \bar{d}\}$. Then if $(e_1, \ldots, e_d)$ is the canonical basis of $\mathbb{Z}^d$, and $u_i = k$ (resp. $\bar{k}$), then u_i is a unitary step from the vertex s_i to s_{i+1} along e_k (resp. $-e_k$). Note that for all $k \leq d$, the number of occurrences of k in u, denoted $|u|_k$, is equal to the number of occurrences of $\bar{k}$ in u. Conversely, any word u on $\mathcal{A}$ that satisfies $|u|_k = |u|_{\bar{k}}$ for $1 \leq k \leq d$ is a rooted polygon. For example, the polygon $12\bar{1}\bar{2}$ would be a unit square. More examples can be seen in figure 1.

This representation is used to define *dimension*, *unimodal polygon* and *convex polygon* (see below). These concepts are important since the method to count the convex polygons involves decomposing them into their unimodal parts, and counting their loops of each dimension.

The *dimension* of a polygon is the dimension of its convex hull, which is equal to the number of k such that $|u|_k > 0$. For example, the loop (s_1, s_2), represented by $u = k\bar{k}$, has dimension 1.

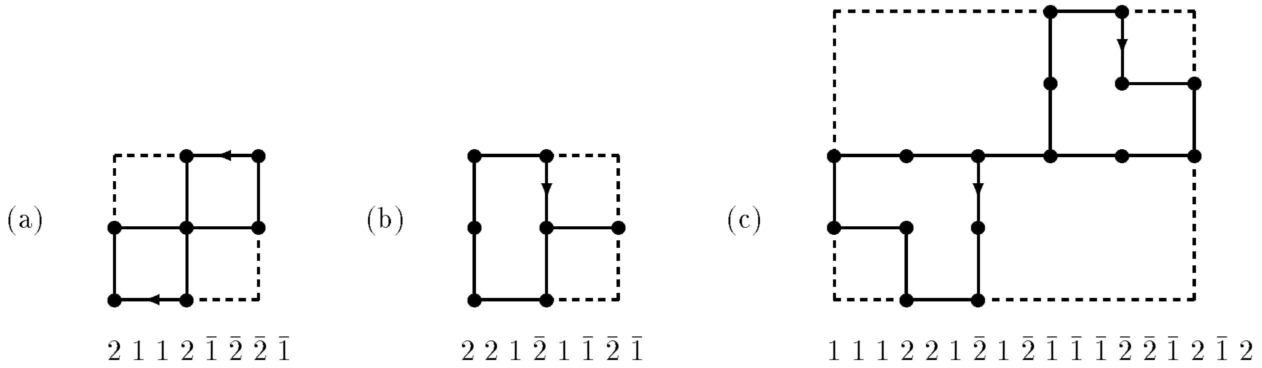

(a)
(b)
(c)
$2\ 1\ 1\ 2\ \bar{1}\ \bar{2}\ \bar{2}\ \bar{1}$
$2\ 2\ 1\ \bar{2}\ 1\ \bar{1}\ \bar{2}\ \bar{1}$
$1\ 1\ 1\ 2\ 2\ 1\ \bar{2}\ 1\ \bar{2}\ \bar{1}\ \bar{1}\ \bar{1}\ \bar{2}\ \bar{2}\ \bar{1}\ 2\ \bar{1}\ 2$

Figure 1. (a) staircase (b) unimodal (c) convex polygons

A polygon is *unimodal* if, for each direction k, the polygon can be written $u = vw$ with $|v|_{\bar{k}} = |w|_k = 0$. In other words, all the k's come before all the $\bar{k}$'s in its representative word u, and so all the steps taken in a given direction occur before all the steps taken to return from that direction.

A polygon is *convex* if for each k there is a cyclic permutation of the polygon such that all the k's come before all the $\bar{k}$'s. More intuitively, for each k, and each $a \in \mathbb{R}$, the intersection of a convex polygon with the half-space $\{(a_1, \ldots, a_d) : a_k \leq a\}$ is connected. Another characteristic is that the length of the perimeter of a convex polygon is equal to the length of the perimeter of the smallest bounding box of the polygon. A unimodal polygon is a convex polygon that contains the vertex of minimal coordinates of its smallest bounding box. See figure 1.

2. Enumeration Method

To count the self-avoiding convex polygons, the idea is to count all convex polygons and then remove those that are not self-avoiding. Let P represent the number of all convex polygons of dimension d, and P_k be the number of convex polygons of dimension d with a k-dimensional loop but no loops of dimension $< k$. Then

$$P = P_1 + P_2 + \cdots + P_d. \tag{1}$$

Polygons will be enumerated by using a generating function based on their perimeters. If $\mathcal{P}$ is a set of polygons, then the *perimeter generating function* for the elements of $\mathcal{P}$ is

$$\sum_{u \in \mathcal{P}} t^{|u|/2},$$

where $|u|$ stands for the number of letters of u; and the *multi-perimeter generating function* is

$$\sum_{u \in \mathcal{P}} x_1^{|u|_1} \cdots x_d^{|u|_d}.$$

A *staircase polygon* is a pair of directed paths having the same end-points, so all the steps taken in positive directions (words on $\{1, \ldots, d\}$) occur before all the steps taken in negative directions (words on $\{\bar{1}, \ldots, \bar{d}\}$). The multi-perimeter generating function for staircase polygons, where n_i is the number of steps taken in direction e_i in $\mathbb{Z}^d$, is

$$Z_d(x_1, \ldots, x_d) = \sum_{n_1, \ldots, n_d} \binom{n_1 + \cdots + n_d}{n_1, \ldots, n_d}^2 x_1^{n_1} \cdots x_d^{n_d}$$

(see [4]). This series is D-finite, that is, it satisfies a linear differential equation with polynomial coefficients [7]. Moreover,

$$(2) \qquad Z_2(x_1,x_2) = \sum_{n_1,n_2} \binom{n_1+n_2}{n_1,n_2}^2 x_1^{n_1} x_d^{n_2} = \frac{1}{\sqrt{1-2x_1-2x_2-2x_1x_2+x_1^2+x_2^2}}$$

is algebraic. This series has a generalization to Z_λ where λ is a partition [4].

Theorem 1. *The multi-perimeter generating function of the number of d-dimensional convex polygons that have no* 1*-dimensional loops is*

$$P - P_1 = E\left[\frac{(d-1)! x_1 \cdots x_d (1-x_1)^2 \cdots (1-x_d)^2}{(1-x_1-\cdots-x_d)^d}\right]$$

where if $f(x_1,\ldots,x_d) = \sum_{n_1,\ldots,n_d} a_{n_1,\ldots,n_d} x_1^{n_1}\cdots x_d^{n_d}$, then the even part *of f is*

$$E\left[f(x_1,\ldots,x_d)\right] = \sum_{n_1,\ldots,n_d} a_{2n_1,\ldots,2n_d} x_1^{2n_1}\cdots x_d^{2n_d}.$$

The proof of this theorem uses the inclusion/exclusion principle and a decomposition of the word-representations of the polygons.

The following gives the formula which will be applied to count convex loops. The idea is that for a convex polygon having loops of dimension d, two cases can occur: either it has only one loop (it itself is a d-dimensional loop), or it can have two loops. There are 2^d possible loop structures, and the loops are unimodal. If the polygon is represented by ul_1vl_2, where the l_i are loops, then uv is essentially a staircase polygon, and so counted by Z_d. Details are presented in [2].

Theorem 2. *In dimension d, let P_d and Z_d be defined as above, and let U_d be the multi-perimeter generating function for unimodal polygons having only loops of dimension d, and C_d be the generating function for convex polygons having only loops of dimension d. Then*

$$P_d = C_d + 2^{d-1} Z_d U_d^2.$$

Since a convex polygon of dimension d which has only loops of dimension d is self-avoiding, C_d counts the d-dimensional self-avoiding convex polygons. Now U_d can be calculated for all d by rewriting it in terms of Z_d using induction. An important element of the proof is that a loop of a rooted unimodal polygon is unimodal, and hence if a rooted unimodal polygon $u_0l_1u_1l_2u_2$ has loops l_i in $I_i \subset \{1,\ldots,d\}$, then $I_1 \cap I_2 = \emptyset$. Thus a unimodal polygon is made up of a sequence of unimodal loops separated by staircase polygons where the structure of the distribution of the loops can be described by a partition of d. The generating function for unimodal polygons having loops corresponding to this partition can be expressed in terms of Z_λ, λ the partition of d, and U_k, $k \leq d$. Then this result, together with equation (1) and theorem 2 gives a means of calculating the number of self-avoiding convex polygons.

3. 2-D Polygons

In dimension $d = 2$, $P - P_1 = P_2$, so combining theorems 1 and 2 gives

$$E\left[\frac{x_1x_2(1-x_1)^2(1-x_2)^2}{(1-x_1-x_2)^2}\right] = C_2 + 2Z_2U_2^2$$

Setting $\Delta = 1 - 2x_1 - 2x_2 - 2x_1x_2 + x_1^2 + x_2^2$, and solving for C_2 using $U_2 = 2\frac{x_1x_2}{\sqrt{\Delta}}$ and (2) gives

$$C_2 = \frac{2x_1x_2A}{\Delta^2} - \frac{8x_1^2x_2^2}{\Delta^{3/2}}$$

where

$$A = 1 - 3x_1 - 3x_2 + 3x_1^2 + 3x_2^2 + 5x_1x_2 - x_1^3 - x_2^3 - x_1^2x_2 - x_1x_2^2 - x_1x_2(x_1 - x_2)^2.$$

This was first proved by Lin and Chang [6], and is a refinement of a result by Delest and Viennot [3]. Alternate proofs are found in [1] and [5]. This work gives a nice combinatorial interpretation of each of the two parts of C_2 in terms of convex polygons having no one-dimensional loops, thereby solving an open problem due to Viennot.

4. 3-D Polygons

This time, the situation is more complicated. Given $P - P_1 = P_2 + P_3$, where $P - P_1$ is calculated using theorem 1, and $P_3 = C_3 + 4Z_3U_3^2$ by theorem 2, it remains to find a way to count P_2, the number of polygons in $\mathbb{Z}^3$ having 2-dimensional loops but no 1-dimension loops. This can be done by a case-by-case analysis of the 7 possible loop structures. The result is

$$C_3 = A(t) + \frac{B(t)}{Z_3}$$

where $A(t)$ and $B(t)$ are algebraic in t, and Z_3 is D-finite. $A(t)$ is of degree 16, and $B(t)$ has degree 8. (The exact value of C_3 would take up a quarter of the page.)

5. Conclusion

This method works because the loops of unimodal polygons are non-overlapping. In theory this method is extensible to higher dimensions, though of course in practice the calculation of the P_i's for $i < d$ would become difficult. Since for each d the series Z_d is D-finite and the series U_d can be written in terms of Z_d, is seems reasonable from the formula to believe that the result will continue to be a quotient of two D-finite series. There may be generalizations to polygons that are convex along $d - 1$ directions, and 3-choice polygons.

Bibliography

[1] Bousquet-Mélou (M.). – Codage des polyominos convexes et équations pour l'énumération suivant l'aire. *Discrete Applied Mathematics*, vol. 48, 1994, pp. 21–43.

[2] Bousquet-Mélou (Mireille) and Guttmann (Anthony J.). – *Enumeration of three-dimensional convex polygons.* – Technical Report n° 1132, Laboratoire Bordelais de Recherche en Informatique, Bordeaux, France, July 1996.

[3] Delest (M.-P.) and Viennot (G.). – Algebraic languages and polyominoes enumeration. *Theoretical Computer Science*, vol. 34, 1984, pp. 169–206.

[4] Guttmann (A. J.) and Prellberg (T.). – Staircase polygons, elliptic integrals, Heun functions and lattice Green functions. *Phys. Rev. E*, vol. 47, 1993, pp. R2233–R2236.

[5] Kim (D.). – The number of convex polyominoes with given perimeter. *Discrete Mathematics*, vol. 70, 1988, pp. 47–51.

[6] Lin (K. Y.) and Chang (S. J.). – Rigorous results for the number of convex polygons on the square and honeycomb lattices. *Journal of Physics Series A: Math. Gen.*, vol. 21, 1988, pp. 2635–2642.

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

Lecture Hall Partitions

Mireille Bousquet-Mélou
LaBRI, Unversité de Bordeaux 1

February 26, 1996

[summary by Dominique Gouyou-Beauchamps]

Abstract

A well-known theorem of Euler [2, Chap. 16] says that the number of partitions of an integer N into distinct parts is equal to the number of partitions of N into odd parts. The talk gives a finite version of this theorem that says that the number of "lecture hall partitions of length n" of N equals the number of partitions of N into small odd parts: $1, 3, 5, \ldots, 2n-1$. This work is a common work with Kimmo Eriksson [1].

1. Lecture hall partitions

Let $\mathcal{D}$ be the set of integer partitions with distinct parts. For $n \geq 1$, let $\mathcal{L}_n$ be the following set of partitions (having possibly some empty parts):

$$\mathcal{L}_n = \{(\lambda_1, \ldots, \lambda_n) : 0 \leq \lambda_1/1 \leq \lambda_2/2 \leq \cdots \leq \lambda_n/n\}.$$

We call the members of $\mathcal{L}_n$ *lecture hall partitions of length n*, since they describe all possible ways of designing a lecture hall with space for up to n rows of seats placed on integer heights, such that at every seat there is a clear view of the speaker without obstruction from the seats in front (Figure 1).

Removing the empty parts puts $\mathcal{L}_n$ in one-to-one correspondence with the following subset of $\mathcal{D}$:

$$\mathcal{D}_n = \left\{(\mu_1, \mu_2, \ldots, \mu_m) : m \leq n \quad \text{and} \quad 0 < \frac{\mu_1}{n-m+1} \leq \frac{\mu_2}{n-m+2} \leq \cdots \leq \frac{\mu_m}{n}\right\}.$$

We will prove the following remarkable theorem.

Theorem 1 (Lecture Hall Theorem). *The generating function for lecture hall partitions of length n is*

$$L_n(q) = \sum_{\lambda \in \mathcal{L}_n} q^{|\lambda|} = \prod_{i=0}^{n-1} \frac{1}{1-q^{2i+1}},$$

where the weight $|\lambda|$ of a partition $\lambda = (\lambda_1, \ldots, \lambda_m)$ is $\lambda_1 + \cdots + \lambda_m$.

Equivalently, the generating function for the partitions of $\mathcal{D}_n$ is $L_n(q)$. Observe that $\mathcal{D}_n \subset \mathcal{D}_{n+1}$ and $\mathcal{D} = \lim_{n\to\infty} \mathcal{D}_n$, so in the limit this theorem yields the familiar Euler identity [2, Chap. 16]: the generating function for the elements of $\mathcal{D}$ is equal to the generating function for the elements of $\mathcal{O}$, the set of integer partitions with odd parts:

$$\sum_{\mu \in \mathcal{D}} q^{|\mu|} = \prod_{i \geq 1} (1+q^i) = \prod_{i \geq 1} \frac{1-q^{2i}}{1-q^i} = \prod_{i \geq 0} \frac{1}{1-q^{2i+1}} = \sum_{\mu \in \mathcal{O}} q^{|\mu|}.$$

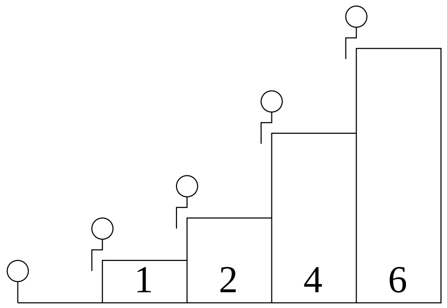


FIGURE 1. The design of a lecture hall of four rows corresponding to the lecture hall partition (1,2,4,6).

We will prove a refinement of the Lecture Hall Theorem. We define the *even* and *odd* weights $|\lambda|_e$ and $|\lambda|_o$ of a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$ by

$$|\lambda|_e = \sum_{0\le k\le \lfloor (n-1)/2\rfloor} \lambda_{n-2k} \qquad \text{and} \qquad |\lambda|_o = \sum_{0\le k\le \lfloor n/2\rfloor -1} \lambda_{n-2k-1}.$$

Of course, $|\lambda| = |\lambda|_e + |\lambda|_o$. We will prove the bivariate identity

$$\sum_{\lambda\in\mathcal{L}_n} x^{|\lambda|_e} y^{|\lambda|_o} = \prod_{i=0}^{n-1} \frac{1}{1-x^{i+1}y^i}.$$

This identity is a corollary of Theorem 3 in section 4, taking $k = l = 2$.

We will in fact discuss a generalization to other sets of partitions of the form $\{(\lambda_1, \lambda_2, \ldots, \lambda_n) : 0 \le \lambda_1/a_1 \le \lambda_2/a_2 \le \cdots \le \lambda_n/a_n\}$ where $(a_1, a_2, \ldots, a_n)$ is a given non-decreasing sequence of integers. We define now $\mathcal{L}_n$ and $S_{(a_1,a_2,\ldots,a_n)}$ as:

$$\mathcal{L}_n = \{(\lambda_1, \ldots, \lambda_n) : 0 \le \lambda_1/a_1 \le \lambda_2/a_2 \le \cdots \le \lambda_n/a_n\} \qquad \text{and} \qquad S_{(a_1,a_2,\ldots,a_n)} = \sum_{\lambda\in\mathcal{L}_n} q^{|\lambda|}.$$

Here are surprisingly simple values of $S_{(a_1,a_2,\ldots,a_n)}$:

$$\begin{aligned}
S_{1,2,5,8} &= \frac{1}{(1-q)(1-q^3)(1-q^8)(1-q^{13})},\\
S_{1,2,5,8,19} &= \frac{1}{(1-q)(1-q^4)(1-q^7)(1-q^{11})(1-q^{27})},\\
S_{1,2,5,8,19,30} &= \frac{1}{(1-q)(1-q^3)(1-q^8)(1-q^{13})(1-q^{31})(1-q^{49})},\\
S_{1,2,7,12,41} &= \frac{1}{(1-q)(1-q^5)(1-q^9)(1-q^{31})(1-q^{53})}.
\end{aligned}$$

2. Reduction of lecture hall partitions

Fix a non-decreasing sequence $a = (a_i)_{i\ge 1}$ of positive integers, and fix a positive integer n. An n-tuple $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{N}^n$ is a lecture Hall partition if and only if $\lambda_i \ge \lceil \lambda_{i-1} a_i / a_{i-1} \rceil$ for $2 \le i \le n$. For $1 \le i \le n$, let $\lambda^{(i)} = (0, \ldots, 0a_i, a_{i+1}, \ldots, a_n) \in \mathbb{N}^n$. If λ belongs to $\mathcal{L}_n$, then the sum $\lambda + \lambda^{(i)}$ also belongs to $\mathcal{L}_n$.

LEMMA 1. *Let λ be a lecture hall partition belonging to $\mathcal{L}_n$. Then $\lambda - \lambda^{(i)}$ belongs to $\mathcal{L}_n$ if and only if $\lambda_i - \lceil \lambda_{i-1} a_i / a_{i-1} \rceil \ge a_i$ for $1 \le i \le n$.*

Definition 1. A lecture hall partition of length n is said to be *reduced* if $0 \leq \lambda_i - \lceil \lambda_{i-1} a_i / a_{i-1} \rceil < a_i$ for $1 \leq i \leq n$. The set of reduced partitions of $\mathcal{L}_n$ will be denoted by $\mathcal{R}_n$.

Lemma 2. *Let λ be a lecture hall partition of length n. Then there exists a unique reduced lecture hall partition μ and a unique sequence of integers $(k_i)_{1 \leq i \leq n}$ such that $\lambda = \mu + \sum_{i=1}^n k_i \lambda^{(i)}$.*

Consequently, the generating function for lecture hall partitions of length n is

$$S_n = \sum_{\lambda \in \mathcal{L}_n} x^{|\lambda|_e} y^{|\lambda|_o} = \frac{P_n(x,y)}{\prod_{i=1}^n (1 - x^{|\lambda^{(i)}|_e} y^{|\lambda^{(i)}|_o})}$$

where the polynomial $P_n(x,y) = \sum_{\mu \in \mathcal{R}_n} x^{|\mu|_e} y^{|\mu|_o}$ enumerates reduced lecture hall partitions.

3. An involution on $\mathcal{R}_n$

For $\mu \in \mathcal{R}_n$, let $\mu^* = (\mu_1^*, \ldots, \mu_n^*)$ be the unique n-tuple such that

$$\begin{cases} \mu^*_{n-2k} = \mu_{n-2k} & \text{for } n-2k \geq 1 \\ \mu^*_{n-2k-1} - \left\lceil \frac{a_{n-2k-1}}{a_{n-2k-2}} \mu^*_{n-2k-2} \right\rceil = \left\lfloor \frac{a_{n-2k-1}}{a_{n-2k}} \mu_{n-2k} \right\rfloor - \mu_{n-2k-1} & \text{for } n-2k-1 \geq 1. \end{cases}$$

Theorem 2. *The correspondence $\mu \mapsto \mu^*$ defines an involution on the set $\mathcal{R}_n$.*

We can extend the involution $\mu \mapsto \mu^*$ into a bijection f from $\mathcal{R}_n \times [0, a_{n+1}]$ onto $\mathcal{R}_{n+1}$, by defining

$$f(\mu_1, \ldots, \mu_n; i) = \left(\mu_1^*, \ldots, \mu_n^*, \left\lceil \frac{a_{n+1}}{a_n} \mu_n^* \right\rceil + i \right).$$

It is clear that:

$$\begin{aligned} |f(\mu,i)|_o &= |\mu^*|_e = |\mu|_e, \\ |f(\mu,i)|_e &= i - |\mu|_o + \sum_k \left(\left\lceil \frac{a_{n-2k+1}}{a_{n-2k}} \mu_{n-2k} \right\rceil + \left\lfloor \frac{a_{n-2k-1}}{a_{n-2k}} \mu_{n-2k} \right\rfloor \right). \end{aligned}$$

4. The $(k-l)$-sequences

By a $(k-l)$-sequence we shall mean a sequence a defined by the initial values $a_1 = 1$ and $a_2 = l$ and the following recurrence relations:

$$\begin{cases} a_{2n} = l a_{2n-1} - a_{2n-2} & \text{for } n \geq 2 \\ a_{2n+1} = k a_{2n} - a_{2n-1} & \text{for } n \geq 1 \end{cases}$$

where $k, l \geq 2$ are two integers. We obtain

$$\begin{aligned} |f(\mu,i)|_o &= |\mu|_e, \\ |f(\mu,i)|_e &= i - |\mu|_o + \begin{cases} k|\mu|_e & \text{if } n \text{ is even,} \\ l|\mu|_e & \text{if } n \text{ is odd.} \end{cases} \end{aligned}$$

This implies that the generating functions $P_n(x,y) = \sum_{\mu \in \mathcal{R}_n} x^{|\mu|_e} y^{|\mu|_o}$ for reduced lecture hall partitions can be computed inductively via the following recurrence relations:

$$P_{2n+1}(x,y) = \frac{1 - x^{a_{2n+1}}}{1-x} P_{2n}(x^k y, x^{-1}) \qquad \text{and} \qquad P_{2n}(x,y) = \frac{1 - x^{a_{2n}}}{1-x} P_{2n-1}(x^l y, x^{-1})$$

with the initial condition $P_0 = 1$.

The sequence a^* is defined by $a_0^* = 0$, $a_1^* = 1$ and the recurrence relations:

$$\begin{cases} a_{2n} = la_{2n-1} - a_{2n-2} & \text{for } n \geq 2 \\ a_{2n+1} = ka_{2n} - a_{2n-1} & \text{for } n \geq 1. \end{cases}$$

Theorem 3. *Given a (k,l)-sequence a, the generating functions $S_n = \sum_{\lambda \in \mathcal{L}_n} x^{|\lambda|_e} y^{|\lambda|_o}$ for lecture hall partitions of even and odd length are given by:*

$$S_{2n} = \prod_{i=1}^{2n} \frac{1}{1 - x^{a_i} y^{a_i^*}} \qquad \textit{and} \qquad S_{2n+1} = \prod_{i=1}^{2n+1} \frac{1}{1 - x^{a_{i+1}^*} y^{a_{i-1}}}.$$

5. Limit theorems

Taking the limit $n \to \infty$ in Theorem 3 leads to the following results:

Theorem 4. *For $k \in \mathbb{N}$ and $k \geq 2$, the bivariate generating function of partitions $(\mu_1, \ldots, \mu_n)$ such that $\frac{\mu_{i+1}}{\mu_i} > \frac{k+\sqrt{k^2-4}}{2}$ is:*

$$\sum_{\mu} x^{|\mu|_e} y^{|\mu|_o} = \prod_{i \geq 1} \frac{1}{1 - x^{a_i} y^{a_{i-1}}}$$

with $a_0 = 0$, $a_1 = 1$ and $a_{i+1} = ka_i - a_{i-1}$.

Theorem 5. *For $k \in \mathbb{N}$ and $k \geq 2$, the generating function of partitions $(\mu_1, \ldots, \mu_n)$ such that $\frac{\mu_{i+1}}{\mu_i} > \frac{k+\sqrt{k^2-4}}{2}$ is:*

$$\sum_{\mu} q^{|\mu|} = \prod_{i \geq 1} \frac{1}{1 - q^{e_i}}$$

with $e_1 = 1$, $e_2 = k+1$ and $e_{i+1} = ke_i - e_{i-1}$.

Example. $k = 2$. In that case $\mu_{i+1} > \mu_i$ and we obtain the Euler identity [2, Chap. 16]:

$$\sum_{\mu \in \mathcal{D}} q^{|\mu|} = \prod_{i \geq 0} \frac{1}{1 - q^{2i+1}}.$$

Example. $k = 3$. In that case $\mu_{i+1} > \frac{3+\sqrt{5}}{2}\mu_i$ and:

$$\sum_{\mu \in \mathcal{L}_n} q^{|\mu|} = \frac{1}{(1-q)(1-q^4)(1-q^{11})(1-q^{29})(1-q^{76})\cdots} = \prod_{i \geq 1} \frac{1}{(1-q^{e_i})},$$

with $e_1 = 1$, $e_2 = 4$ and $e_{i+1} = 3e_i - e_{i-1}$. In fact $e_i = F_{2i-3} + F_{2i-1}$ where F_i is the ith Fibonnaci number.

6. Questions

(1) Give a characterization of the sequences $(a_1, \ldots, a_n)$ that have a simple expression for the corresponding generating functions.

(2) Find finite version of other theorems like the Rogers-Ramanujan theorem for instance.

Bibliography

[1] Bousquet-Mélou (M.) and Eriksson (K.). – *Lecture Hall Partitions.* – Rapport LaBRI n° 1107-96, Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux I, 1996.

[2] Euler (L.). – *Introductio in analysin infinitorum.* – Marcum-Michælem Bousquet, Lausannæ, 1748.

Determinants, Catalan numbers and Macdonald's symmetric functions

Dominique Gouyou-Beauchamps
LRI, Orsay

March 25, 1996

[summary by Bruno Salvy]

Abstract

A famous conjecture in the theory of symmetric functions states that the coefficients of Macdonald's polynomials in the basis of Schur's symmetric functions are positive. F. Bergeron, A. M. Garsia and M. Haiman have introduced a linear operator ∇ whose eigenvalues are related to Macdonald's polynomials. Properties of this operator in a special case are related to combinatorial determinants which can be evaluated by the Gessel-Viennot technique relating them to non-intersecting paths.

1. Introduction to symmetric functions

This section and the following one are based on [4].

Partitions and symmetric functions are strongly related. A *partition* is an infinite decreasing sequence of positive integers $\lambda = (\lambda_1, \lambda_2, \dots)$, with finitely many non-zero elements. The index of the last non-zero element in the partition is called its *length* and is denoted $\ell(\lambda)$; the sum of the λ_i's is called the *weight* of the partition and is denoted $|\lambda|$. For $n \geq \ell(\lambda)$, λ is identified with the n-tuple of its first elements. Then if $x = (x_1, \dots, x_n)$ is a n-tuple of indeterminates, x^λ denotes the monomial $x_1^{\lambda_1} \cdots x_n^{\lambda_n}$ and S_n^λ denotes a maximal set of distinct permutations of λ.

A fundamental basis of symmetric functions is constituted by the *monomial* symmetric functions, indexed by the partitions: for $n \geq \ell(\lambda)$,

$$m_\lambda(x_1, \dots, x_n) = \sum_{\sigma \in S_n^\lambda} x^{\sigma(\lambda)}.$$

Clearly, the set of m_λ's, when λ runs through all partitions of length at most n is a basis of the symmetric polynomials in n variables. The set Λ of symmetric functions is *defined* as the vector space generated by the m_λ's.

Three important sets of symmetric functions, $e_r = m_{(1^r)}$ (elementary), $h_r = \sum_{|\lambda|=r} m_\lambda$ (complete) and $p_r = m_{(r)}$ (power sum), have simple generating functions:

$$\begin{aligned}
E(t) &= \sum_{r \geq 0} e_r t^r = 1 + t \sum_i x_i + t^2 \sum_{i<j} x_i x_j + \cdots = \prod_{i>0} (1 + x_i t),\\
H(t) &= \sum_{r \geq 0} h_r t^r = 1 + t \sum_i x_i + t^2 \sum_{i \leq j} x_i x_j + \cdots = \prod_{i>0} \frac{1}{1 - x_i t},\\
P(t) &= \sum_{r \geq 0} p_r t^r = \sum_i x_i + t \sum_i x_i^2 + \cdots \qquad = \sum_{i>0} \frac{x_i}{1 - x_i t}.
\end{aligned}$$

Each of these three sets of symmetric functions generates Λ as a ring. In all three cases, defining for a partition λ a function $f_\lambda = f_{\lambda_1} f_{\lambda_2} \cdots$, where f is e, h or p yields a basis of Λ as a vector space, when λ runs through the set of partitions.

Formulæ giving the coordinates of one of these functions in terms of the other families are obtained by extracting the coefficient of t^n in the following straightforward relations between the generating functions:

$$(1) \qquad E(t)H(-t) = 1, \qquad P(t) = \frac{H'(t)}{H(t)}, \qquad P(-t) = \frac{E'(t)}{E(t)}.$$

The last two equations yield the classical Newton formulæ between power sums and elementary symmetric functions. Integrating these equation also yields

$$H(t) = \exp \sum_{r>0} p_r \frac{t^r}{r} = \sum_\lambda p_\lambda \frac{t^{|\lambda|}}{z_\lambda}, \qquad E(t) = \sum_\lambda (-1)^{|\lambda| - \ell(\lambda)} p_\lambda \frac{t^{|\lambda|}}{z_\lambda}, \qquad \text{with} \qquad z_\lambda = \prod_{i>0} i^{m_i} m_i!,$$

where m_i is the number of occurrences of the part i in λ.

Another family of symmetric functions, the *Schur functions*, is defined for $n \geq \ell(\lambda)$ by

$$s_\lambda(x_1, \ldots, x_n) = \frac{\det(x_i^{\lambda_j+n-j})_{1\leq i,j\leq n}}{\det(x_i^{n-j})_{1\leq i,j\leq n}}.$$

The s_λ's are indeed polynomials, since the numerator is a polynomial in the x_i's which vanishes whenever $x_i = x_j$ with $i \neq j$, and thus is a multiple of the Vandermonde determinant in the denominator. The s_λ form another basis of Λ. They are related to the complete and elementary symmetric functions by the Jacobi-Trudi identities:

$$(2) \qquad s_\lambda = \det(h_{\lambda_i - i + j})_{1\leq i,j\leq n}, \qquad s_\lambda = \det(e_{\lambda'_i - i + j})_{1\leq i,j\leq m},$$

where λ' is the *conjugate* of λ, i.e. the partition whose Ferrers diagram is the reflexion of that of λ with respect to the diagonal.

Recall that a *Young tableau* of shape λ is a Ferrers diagram of shape λ with squares numbered by consecutive positive integers $1, 2, \ldots, r$, the numbers increasing strictly in each column and weakly along each row. The *weight* $w(T)$ of a tableau T is the r-tuple $(m_1, \ldots, m_r)$, m_i counting the number of occurrences of i. The tableau is called *standard* when it contains each number $1, 2, \ldots, |\lambda|$ exactly once, i.e. its weight is $(1^{|\lambda|})$. The Schur functions are related to tableaux by

$$s_\lambda = \sum_T x^{w(T)},$$

summed over all tableaux T of shape λ. From this follows that the coordinates $K_{\lambda\mu}$ of s_λ in the basis m_μ are positive integers counting the number of tableaux of shape λ and weight μ and thus are positive integers. Macdonald's conjecture is a generalization of this property.

All these symmetric functions can also be related by expanding in several ways the doubly infinite product $P(x, y) = \prod (1 - x_i y_j)^{-1}$. Thus one gets

$$(3) \qquad \prod_{i,j} \frac{1}{1 - x_i y_j} = \sum_\lambda z_\lambda^{-1} p_\lambda(x) p_\lambda(y) = \sum_\lambda h_\lambda(x) m_\lambda(y) = \sum_\lambda m_\lambda(x) h_\lambda(y) = \sum_\lambda s_\lambda(x) s_\lambda(y).$$

This motivates the definition of a scalar product by $\langle h_\lambda, m_\mu \rangle = \delta_{\lambda\mu}$ for all partitions λ, μ, where $\delta_{\lambda\mu}$ is the Kronecker delta. The relations (3) show that the p_λ's form an orthogonal basis, while the s_λ's form an orthonormal basis of Λ. This property characterizes the Schur functions.

The next step is to consider the Hall-Littlewood symmetric functions with one parameter

$$P_\lambda(x_1,\ldots,x_n;t) = \sum_{\sigma\in S_n^\lambda} \sigma\left(x^\lambda \prod_{i<j} \frac{x_i - tx_j}{x_i - x_j}\right).$$

These functions interpolate between the monomial symmetric functions—obtained when $t=1$—and the Schur symmetric functions—obtained when $t=0$. They form a $\mathbb{Z}[t]$-basis of $\Lambda[t]$. Therefore, one may consider the polynomials $K_{\lambda\mu}(t)$ defined by

$$s_\lambda(x) = \sum_\mu K_{\lambda\mu}(t) P_\mu(x;t).$$

The polynomials $K_{\lambda\mu}(t)$ turn out to have positive coefficients, and this has been proved by Lascoux and Schützenberger who gave an expression of the form

$$K_{\lambda\mu}(t) = \sum_T t^{c(T)},$$

summed over all tableaux T of shape λ and weight μ, where $c(T)$ is a certain combinatorial function of the tableau (its *charge*). Several expansions of the product $P(x,y;t) = \prod_{i,j}(1-tx_iy_j)/(1-x_iy_j)$ lead to results very similar to those obtained above and to the definition of a scalar product on $\Lambda[t]$ with values in $\mathbb{Q}(t)$ with respect to which the $P_\lambda(x;t)$ are orthogonal. Also $\langle P_\lambda, m_\mu\rangle = 0$ when $\mu \not\leq \lambda$ (the Ferrers diagram of μ is not included in that of λ), and together with their orthogonality this characterizes the P_λ. The basis which is dual to the Schur functions $s_\lambda(x)$ with respect to this scalar product is denoted $S_\lambda(x;t)$, i.e., $\langle S_\lambda(x;t), s_\mu(x)\rangle = \delta_{\lambda\mu}$.

2. Macdonald's conjecture

Macdonald's conjecture concerns the Macdonald symmetric functions, which have two parameters. The doubly infinite product

$$\Pi(x,y;q,t) = \prod_{i,j} \frac{(tx_iy_j;q)_\infty}{(x_iy_j;q)_\infty}, \qquad \text{where} \qquad (a;q)_\infty = \prod_{r=0}^{\infty}(1-aq^r),$$

can be expanded as

$$\Pi(x,y;q,t) = \sum_\lambda \frac{1}{z_\lambda(q,t)} p_\lambda(x)p_\lambda(y), \qquad \text{with} \qquad z_\lambda(q,t) = z_\lambda \prod_{i=1}^{\ell(\lambda)} \frac{1-q^{\lambda_i}}{1-t^{\lambda_i}}.$$

This motivates the definition of a scalar product by

$$\langle p_\lambda, p_\mu\rangle_{q,t} = \delta_{\lambda\mu} z_\lambda(q,t).$$

The Macdonald symmetric functions are defined uniquely by two properties: they are orthogonal with respect to this scalar product and they decompose in the basis of the monomial symmetric functions as

$$P_\lambda(x;q,t) = m_\lambda + \sum_{\mu<\lambda} u_{\lambda\mu} m_\mu.$$

When $q=t$, they reduce to the Schur functions s_λ, and when $q=0$ to the Hall-Littlewood functions $P_\lambda(x;t)$.

For a partition λ and a cell $c=(i,j)$ of its Ferrers diagram, one defines the *arm* of c to be $a(c) = \lambda_i - j$ and its *leg* to be $l(c) = \lambda'_j - i$. Now we can state Macdonald's conjecture.

Conjecture 1 (Macdonald). *The coefficients $K_{\lambda\mu}(q,t)$ of the following decomposition are polynomials with positive coefficients:*

$$(4)\ \tilde{H}_\lambda(x;t) := c_\lambda(q,t)P_\lambda(x;q,t) = \sum_\lambda K_{\lambda\mu}(q,t)S_\lambda(x;t), \quad \textit{where} \quad c_\lambda(q,t) = \prod_{c\in\lambda}(1-q^{a(c)}t^{l(c)+1}).$$

These coefficients possess a lot of structure. For instance, for $\lambda = (3,1)$, Eq. (4) becomes

$$\tilde{H}_{(3,1)} = S_{(4)} + (q^2+t+q)S_{(3,1)} + (t+q)qS_{(2,2)} + (tq+q^2+t)qS_{(2,1,1)} + tq^3S_{(1,1,1,1)}.$$

Only special cases of Macdonald's conjecture have been proved.

3. Combinatorial properties of ∇ when $t = 1$

In order to study the polynomials $\tilde{H}_\lambda$, Bergeron, Garsia and Haiman have introduced a linear operator ∇ which is diagonal in the basis $\tilde{H}_\lambda$, with eigenvalues $T_\lambda(q,t) = q^{n(\lambda')}t^{n(\lambda)}$, where $n(\lambda) = \sum(i-1)\lambda_i$. The matrix of ∇ in the Schur basis turns out to have a fascinating structure of which much is still only conjectured [2].

The aim of [1] is to study this operator in more detail in the special case $t = 1$. Then the basis $\tilde{H}_\lambda(x;q) := \tilde{H}_\lambda(x;q,1)$ becomes multiplicative: $\tilde{H}_\lambda(x;q) = \tilde{H}_{(\lambda_1)}(x;q)\tilde{H}_{(\lambda_2)}(x;q)\cdots$ and ∇ becomes multiplicative too. Thus any identity involving symmetric functions gives rise to a similar identity for its image by ∇. In particular, from (2) follows $\nabla(s_\lambda) = \det(\nabla e_{\lambda'_i+j-i})_{1\le i,j\le m}$. Moreover, still when $t = 1$, the coordinate $\nabla(e_n)|_{e_n}$ of $\nabla(e_n)$ on e_n is a q-Catalan number C_n, with generating function $C(x)$ defined by $C(x) = 1 + xC(x)C(xq)$. Hence $D(\lambda) := \nabla(s_\lambda)|_{e_n} = \det(C_{\lambda'_i+j-i})_{1\le i,j\le m}$, and the idea of [1] is to use the Gessel-Viennot technique [3] to evaluate determinants of this type for various classes of partitions λ. Typical results are summarised in the following theorem.

Theorem 1.

$$D((k^k)) = (-1)^{\binom{k}{2}}q^{\frac{k(k-1)(4k+1)}{6}}, \qquad D((k^{k+1})) = (-1)^{\binom{k+1}{2}}q^{\frac{k(k+1)(4k-1)}{6}},$$

$$D((k^{k+2})) = (-1)^{\binom{k+2}{2}+1}q^{\frac{k(k+1)(4k-1)}{6}+k^2}[k+1], \quad D((k+1)^k) = (-1)^{\binom{k}{2}}q^{\frac{k(k-1)(4k+7)}{6}}[k+1],$$

$$D((k+2)^k) = (-1)^{\binom{k}{2}}q^{\frac{k(k-1)(4k+13)}{6}}\frac{[k+1][k+2]([k+1]+q[k+2])}{[2][3]},$$

where $[k] = 1 + q + q^2 + \cdots + q^{k-1}$.

Another linear operator diagonal in the basis $\tilde{H}_\lambda$ is also studied in [1]. Similar techniques apply and results of a similar kind are obtained.

Bibliography

[1] Bergeron (François), Bousquet-Mélou (Mireille), and Gouyou-Beauchamps (Dominique). – Preprint, 1996.

[2] Bergeron (François), Garsia (Adriano), and Haiman (Mark). – New identities and conjectures for Macdonald's $\tilde{H}_\mu[x;q,t]$ polynomials. – Preprint, 1995.

[3] Gessel (I.) and Viennot (G.). – Binomial determinants, paths and hook length formulae. *Advances in Mathematics*, vol. 58, 1985, pp. 300–321.

[4] Macdonald (Ian Grant). – *Symmetric functions and Hall polynomials.* – Oxford University Press, 1995, 2nd edition, *Oxford Mathematical Monographs.*

Euler sums

Philippe Flajolet
INRIA Rocquencourt

January 29, 1996

[summary by Jean-Paul Allouche]

In 1742 Goldbach wrote a letter to Euler proposing the study of the sums

$$S_{p,q} := \sum_{n=1}^{\infty} \left(\frac{1}{1^p} + \frac{1}{2^p} + \cdots + \frac{1}{n^p} \right) \frac{1}{n^q} = \sum_{n=1}^{\infty} \frac{H_n^{(p)}}{n^q},$$

where $H_n^{(r)}$ and $H_n = H_n^{(1)}$ are the *harmonic numbers* defined by

$$H_n^{(r)} := \sum_{j=1}^{n} \frac{1}{j^r}.$$

Euler was able to compute all the sums $S_{p,q}$ for $p + q \leq 13$, for example

$$\sum_{n=1}^{\infty} \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} \right) \frac{1}{n^2} = 2\zeta(3).$$

Then, in 1906, Nielsen gave relations linking the sums $S_{p,q}$ having the same weight $w = p + q$. Hence the $S_{p,q}$ of odd weight are polynomials in the values of zeta, for example

$$S_{2,5} = \sum_{n=1}^{\infty} \frac{H_n^{(2)}}{n^5} = 5\zeta(2)\zeta(5) + 2\zeta(3)\zeta(4) - 10\zeta(7).$$

Many similar identities have then been found or conjectured. Some of them involve multiple zeta functions; see the papers of Bayley, D. and J. Borwein, De Doelder, Don Zagier, Girgensohn, Hoffman, Markett.

The authors [1] propose a simple and unifying method that gives most of the known results about these identities. Furthermore they are able to prove some conjectures. The key idea is to use a contour integral *with a well-chosen kernel.*

1. The idea of the authors: a simple case

Let us denote by $I(p,q)$ the integral

$$I(p,q) = \frac{1}{2i\pi} \int_C (\psi(-s) + \gamma)^2 \frac{ds}{s^q},$$

where C is a circle whose radius goes to infinity, and where ψ is the logarithmic derivative of the Γ function. Denoting by γ the Euler constant, we have

$$\psi(z) = \frac{d}{dz} \log \Gamma(z) = -\gamma - \frac{1}{z} + \sum_{n=1}^{\infty} \left(\frac{1}{n} - \frac{1}{n+z} \right).$$

Hence, when s tends to a positive integer m, then

$$(\psi(-s)+\gamma)^2 \underset{s\to m}{=} \frac{1}{(s-m)^2} + 2H_m \frac{1}{s-m} + \cdots .$$

If s tends to 0, we use the relation $\psi(s)+\gamma = -1/s + \zeta(2)s - \zeta(3)s^2 + \cdots$. Hence, by residue computation the Euler sum $S_{1,q}$ can be expressed as an explicit quantity which is "homogeneous" of degree 2 in the zeta values.

In the general case the authors consider integrals

$$\frac{1}{2i\pi}\int_C r(s)\xi(s)\,ds,$$

where r is a rational function, and ξ a *suitable* kernel. They then obtain numerous results: some of them were already known, but some of them were only conjectures.

2. A zoo of beautiful identities

The authors obtain the following results.

Theorem 1 (Euler). *Let q be an integer ≥ 2. Then*

$$S_{1,q} = \sum_{n=1}^{\infty} \frac{H_n}{n^q} = \left(1+\frac{1}{2}q\right)\zeta(q+1) - \frac{1}{2}\sum_{k=1}^{q-2}\zeta(k+1)\zeta(q-k).$$

For example

$$\sum_{n=1}^{\infty}\frac{H_n}{n^2} = 2\zeta(3), \quad \sum_{n=1}^{\infty}\frac{H_n}{n^3} = \frac{5}{4}\zeta(4), \quad \sum_{n=1}^{\infty}\frac{H_n}{n^4} = 3\zeta(5)-\zeta(2)\zeta(3).$$

Theorem 2 (Euler, Borwein et al.). *If the weight $m = p+q$ is odd, then*

$$\sum_{n=1}^{\infty}\frac{H^{(p)}(n)}{n^q} = \zeta(m)\left[\frac{1}{2} - \frac{(-1)^p}{2}\binom{m-1}{p} - \frac{(-1)^p}{2}\binom{m-1}{q}\right] + \frac{1-(-1)^p}{2}\zeta(p)\zeta(q)$$
$$+ (-1)^p \sum_{k=1}^{\lfloor\frac{p}{2}\rfloor}\binom{m-2k-1}{q-1}\zeta(2k)\zeta(m-2k) + (-1)^p \sum_{k=1}^{\lfloor\frac{q}{2}\rfloor}\binom{m-2k-1}{p-1}\zeta(2k)\zeta(m-2k),$$

where any occurrence of $\zeta(1)$ has to be replaced by 0.

If we then use the symmetry $S_{p,q}+S_{q,p} = \zeta(p)\zeta(q)+\zeta(p+q)$, we obtain

$$5\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^6} + 2\sum_{n=1}^{\infty}\frac{H_n^{(3)}}{n^5} = -\frac{21}{2}\zeta(8) + 10\zeta(3)\zeta(5) + \frac{9}{2}\zeta(4)^2,$$
$$7\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^8} + 2\sum_{n=1}^{\infty}\frac{H_n^{(3)}}{n^7} = -33\zeta(10) + 14\zeta(3)\zeta(7) + 15\zeta(4)\zeta(6) + 8\zeta(5)^2,$$
$$7\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^8} - 2\sum_{n=1}^{\infty}\frac{H_n^{(4)}}{n^6} = -\frac{229}{5}\zeta(10) + 14\zeta(3)\zeta(7) + 21\zeta(4)\zeta(6) + 10\zeta(5)^2,$$
$$9\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^{10}} + 2\sum_{n=1}^{\infty}\frac{H_n^{(3)}}{n^9} = -\frac{143}{2}\zeta(12) + 18\zeta(3)\zeta(9) + 21\zeta(4)\zeta(8) + 24\zeta(5)\zeta(7) + \frac{25}{2}\zeta(6)^2,$$

$$8\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^{10}}-\sum_{n=1}^{\infty}\frac{H_n^{(4)}}{n^{8}}=-\frac{575}{7}\zeta(12)+16\zeta(3)\zeta(9)+24\zeta(4)\zeta(8)+28\zeta(5)\zeta(7)+\frac{295}{21}\zeta(6)^2,$$
$$7\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^{10}}+\sum_{n=1}^{\infty}\frac{H_n^{(5)}}{n^{7}}=-73\zeta(12)+28\zeta(5)\zeta(7)+21\zeta(4)\zeta(8)+14\zeta(3)\zeta(9)+\frac{35}{3}\zeta(6)^2.$$

Then

$$\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^2}=\frac{7}{4}\zeta(4),\quad \sum_{n=1}^{\infty}\frac{H_n^{(3)}}{n^3}=\frac{1}{2}\zeta(3)^2+\frac{1}{2}\zeta(6),\quad \sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^4}=\zeta(3)^2-\frac{1}{3}\zeta(6).$$

Theorem 3 (Borwein et al.). *The following relations hold.*

$$\sum_{n=1}^{\infty}\frac{(H_n)^2}{n^q}-S_{2,q}=qS_{1,q+1}-\frac{q(q+1)}{6}\zeta(q+2)+\zeta(2)\zeta(q).$$

For example

$$\sum_{n=1}^{\infty}\frac{(H_n)^2}{n^3}=\frac{7}{2}\zeta(5)-\zeta(2)\zeta(3),$$
$$\sum_{n=1}^{\infty}\frac{(H_n)^2}{n^5}=6\zeta(7)-\zeta(2)\zeta(5)-\frac{5}{2}\zeta(3)\zeta(4),$$
$$\sum_{n=1}^{\infty}\frac{(H_n)^2}{n^7}=\frac{55}{6}\zeta(9)-\zeta(2)\zeta(7)-\frac{7}{2}\zeta(3)\zeta(6)-\frac{5}{2}\zeta(4)\zeta(5)+\frac{1}{3}\zeta(3)^3,$$

and

$$\sum_{n=1}^{\infty}\frac{H_n^2}{n^6}-\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^6}=\frac{91}{12}\zeta(8)-8\zeta(3)\zeta(5)+\zeta(2)\zeta(3)^2,$$
$$\sum_{n=1}^{\infty}\frac{H_n^2}{n^8}-\sum_{n=1}^{\infty}\frac{H_n^{(2)}}{n^8}=\frac{473}{40}\zeta(10)-10\zeta(3)\zeta(7)-5\zeta(5)^2+\zeta(4)\zeta(3)^2+2\zeta(2)\zeta(3)\zeta(5),$$
$$\sum_{n=1}^{\infty}\frac{(H_n)^2}{n^2}=\frac{17}{4}\zeta(4),$$
$$\sum_{n=1}^{\infty}\frac{(H_n)^2}{n^4}=\frac{307}{24}\zeta(6)-5\zeta(2)\zeta(4)-2\zeta(3)^2.$$

Theorem 4. *If $i+j+k$ is odd, with $i>1$, $j>1$, $k>1$, then*

$$[(-1)^k+(-1)^{i+j}]\sum_{n\geq 1}\frac{H_n^{(i)}H_n^{(j)}}{n^k}+A+B+C+D+E+F=0,$$

where

$$A=(-1)^{i+j+k}\zeta(i)\zeta(j)\zeta(k)+(-1)^{i+k}\zeta(i)S_{j,k}+(-1)^{j+k}\zeta(j)S_{i,k},$$
$$B=-2(-1)^k\sum_{q+2r+t=i}\binom{j+q-1}{q}\binom{k+t-1}{k-1}[(-1)^qS_{j+q,k+t}+(-1)^j\zeta(j+q)\zeta(k+t)]\zeta(2r),$$
$$C=-2(-1)^k\sum_{p+2r+t=j}\binom{i+p-1}{p}\binom{k+t-1}{k-1}[(-1)^pS_{i+p,k+t}+(-1)^i\zeta(i+p)\zeta(k+t)]\zeta(2r),$$

$$\begin{aligned}
D &= -2(-1)^k \sum_{2r+t=i+j} \binom{k+t-1}{k-1} \zeta(2r)\zeta(k+t),\\
E &= (-1)^{i+j}\,[-S_{i,j+k} - S_{j,i+k} - \zeta(j)S_{i,k} - \zeta(i)S_{j,k}\\
&\qquad + \zeta(i+j+k) + \zeta(i+k)\zeta(j) + \zeta(j+k)\zeta(i) + \zeta(i)\zeta(j)\zeta(k)],\\
F &= \sum_{p+q+r=i+j+k} \zeta(2r)\lambda_p^{(i)}\lambda_q^{(j)},
\end{aligned}$$

and

$$\lambda_0^{(i)} = 1, \quad \lambda_1^{(i)} = \lambda_2^{(i)} = \cdots = \lambda_{i-1}^{(i)} = 0, \quad \lambda_{i+t}^{(i)} = (-1)^i \zeta(t) \binom{t+i-1}{i-1}.$$

The summations are over the indices ≥ 0. One has to replace $\zeta(0)$ by $-\frac{1}{2}$, and $\zeta(1)$ by 0.

Corollary 1 (Borwein and Girgensohn). *Let $c > 1$. If the weight $a+b+c$ is even, the triple zeta function $\zeta(a,b,c) = \sum_{0<n_1<n_2<n_3} \frac{1}{n_1^a n_2^b n_3^c}$ can be reduced to linear Euler sums.*

Theorem 5. *(i) The cubic expression $\sum_{n=1}^{\infty} \frac{(H_n)^3}{n^q} - 3\sum_{n\geq 1} \frac{H_n H_n^{(2)}}{n^q}$ can be expressed in terms of the zeta values, for any weight.*

(ii) For even weights, $\sum_{n=1}^{\infty} \frac{(H_n)^3}{n^q}$ can be computed in terms of $S_{2,q+1}$ and polynomials in the zeta values.

As a consequence, this gives a proof of conjectures of Bailey, Borwein and Girgensohn:

Corollary 2. *We have*

$$\begin{aligned}
\sum_{n=1}^{\infty} \frac{(H_n)^3}{(n+1)^2} &= \frac{15}{2}\zeta(5) + \zeta(2)\zeta(3)\\
\sum_{n=1}^{\infty} \frac{(H_n)^3}{(n+1)^3} &= -\frac{33}{16}\zeta(6) + 2\zeta(3)^2\\
\sum_{n=1}^{\infty} \frac{(H_n)^3}{(n+1)^4} &= \frac{119}{16}\zeta(7) - \frac{33}{4}\zeta(3)\zeta(4) + 2\zeta(2)\zeta(5)\\
\sum_{n=1}^{\infty} \frac{(H_n)^3}{(n+1)^6} &= \frac{197}{24}\zeta(9) - \frac{33}{4}\zeta(4)\zeta(5) - \frac{37}{8}\zeta(3)\zeta(6) + \zeta(3)^3 + 3\zeta(2)\zeta(7).
\end{aligned}$$

3. Other relations?

If the reader wants to discover other relations, including relations on alternating Euler sums, read the details of the proofs, check that he was able to discover tricky integration contours, or know where some of these relations naturally occur in theoretical computer science, he should read this very nice paper. He will certainly enjoy it.

Bibliography

[1] Flajolet (Philippe) and Salvy (Bruno). – *Euler Sums and Contour Integral Representations.* – Research Report n° 2917, Institut National de Recherche en Informatique et en Automatique, June 1996.

A Zero-One Law for Maps

Kevin Compton
University of Michigan, Ann Arbor, U.S.A.

June 10, 1996

[summary by Frédéric Chyzak]

Abstract

A class of structures has a 0–1 law when any property expressible in a certain logic has limiting probability 0 or 1 as the size of the structures tends to infinity. We prove 0–1 laws for classes of maps of a given genus. This is a joint work with E. Bender and B. Richmond [1].

1. Definition of the problem

Let S be a set of primitive elements called *sorts.* A *vocabulary* Σ consists of a collection of constant and relation symbols, together with a mapping from each constant symbol to a sort, and a mapping from each relation symbol to a sequence of sorts, the *arity* of the relation (see [4] for an introduction to model theory). A *multi-sorted structure* $\mathcal{A}$ over Σ then consists of

- a collection of disjoints sets (or *universes*) A_s, one for each sort s;
- elements $c^{\mathcal{A}} \in A_s$, one for each constant symbol c of sort s;
- relations $R^{\mathcal{A}} \subset A_{s_1} \times \cdots \times A_{s_p}$, one for each relation symbol R of arity $(s_1, \ldots, s_p)$.

A *class* of structures is a set of structures defined on the same vocabulary. In the study of random structures, one says that a class of finite structures *has a 0–1 law* when any property expressible in a certain logic has limiting probability 0 or 1 as the size of the structures tends to infinity. The *relational signature* of a class of structures over Σ is the common set of relation symbols in the vocabulary Σ, together with their arities. A famous theorem by Glebskiĭ, Kogan, Liogon'kiĭ and Talanov [9], and proved independently by Fagin [7], states that if $\mathcal{C}$ is the class of all structures for a given relational signature, then $\mathcal{C}$ has a first-order 0–1 law. However, deciding the limiting probability of a given property is a difficult problem, as formalized by a theorem by Grandjean: when a class $\mathcal{C}$ has a 0–1 law, the set of first-order sentences of limiting probability 1 is PSPACE-complete.

A *map* $\mathcal{M}$ is an embedding of a connected graph $\mathcal{G}$ into a closed surface $\mathcal{S}$ such that all connected components of $\mathcal{S} \setminus \mathcal{G}$, the *faces* of $\mathcal{M}$, are homeomorphic to a disc. Let $t = 1 - (v - e + f)/2$ be the *genus* of $\mathcal{M}$, with v, e and f its number of vertices, edges and faces respectively. When t is an integer, the map is called *orientable.* The *size* $|\mathcal{M}|$ of a map is e. The purpose of this exposition is to provide similar results to the theorems mentioned above for maps, even in the non-orientable case. Our main result is the following theorem [1].

THEOREM 1. *The class of all maps on surfaces of fixed genus has a 0–1 law. The set of first-order sentences of limiting probability 1 for this class has lower bound complexity of* $\mathrm{DTIME}(\exp_\infty(cn))$, *for some* $c > 0$.

(Recall that $\exp_\infty(n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$, with n nested exponentiations.)

The 0–1 law theorem for structures cannot be applied to maps, since the latter do not form a full class of structures of a given relational since. Besides, we have to explain how maps can be represented as structures.

2. Representation of maps as structures

Any naive attempt of representing a map $\mathcal{M}$ on a surface $\mathcal{S}$ by its graph, i.e., by its set of edges, is bound to fail. Indeed, this representation would not encapsulate any information about the embedding of $\mathcal{M}$ into $\mathcal{S}$: easy examples show that isomorphic graphs need not correspond to homeomorphic maps, and that the order of edges around a vertex has to be taken into account. However, on a non-orientable surface there is no consistent way to choose an edge order around each vertex.

A solution stems from an idea of Edmonds [5], later elaborated by Tutte [10] as a basis for a combinatorial theory of maps: to each edge, one associates a pair of *darts*, pointing in opposite directions. On orientable surfaces, a possible representation of maps is then given by an involution α on the set of darts, mapping a dart to its opposite dart, together with a permutation β whose cycles consist of all darts out of a vertex, listed clockwise. Then, $\alpha\beta$ is a permutation whose cycles consist of all darts around a face, listed counter-clockwise. One is thus able to determine the embedding using α and β. In the context of possibly non-orientable surfaces, a map is analogously described as a structure by the sets U_v, U_d and U_f of its vertices, darts and faces, together with incidence relations $I(x_v, x_d)$ and $J(x_f, x_d)$ of darts with vertices and faces, a co-dart relation $C(x_d, x_{d'})$ and a dart adjacency relation $A(x_d, x_{d'}, x_f)$. The co-dart relation is an analogue for α, while the dart adjacency relation encapsulates the information formerly supplied by β, specifying a face to supply the orientable information.

3. Ehrenfeucht-Fraïssé games

The 0–1 law theorem for structures still does not apply to maps: not all structures of signature (I, J, C, A) are maps. We overcome this difficulty in the case of the class of all maps on surfaces of a fixed genus by determining subclasses of limiting probability 1.

The sentences of first-order logic under consideration for our 0–1 laws can all be written in the form $S = \theta_1 x_1 \ldots \theta_r x_r \phi(x_1, \ldots, x_r)$, where the θ_i's are quantifiers, either $\forall$ or $\exists$, the x_i's are variables and ϕ is a boolean expression free from quantifiers built on the x_i's using conjunctions and disjunctions. The *rank* of the sentence S is the integer r. Let $\mathcal{A}$ and $\mathcal{B}$ be two structures with same relational signature. We write $\mathcal{A} \equiv_m \mathcal{B}$ when both structures satisfy exactly the same sentences of rank m. This defines an equivalence relation between structures. The next paragraph describes this equivalence relation by a game-theoretic approach.

The Ehrenfeucht-Fraïssé game is an m-round game between two players called *Spoiler* and *Duplicator* and played on a pair of structures $\mathcal{A}$ and $\mathcal{B}$ of same relational signature. In each round, Spoiler picks any element from either structure and Duplicator responds by picking any element from the other structure. This yields two substructures $\mathcal{A}' = \{a_1, \ldots, a_m\} \subset \mathcal{A}$ and $\mathcal{B}' = \{b_1, \ldots, b_m\} \subset \mathcal{B}$, with relations induced in a natural way. Duplicator wins if he is able to choose his responses so as to make $\mathcal{A}'$ and $\mathcal{B}'$ isomorphic; if not, Spoiler wins. Duplicator *has a winning strategy* if and only if he is capable of winning for any choices made by Spoiler. A fundamental result used in the sequel is the Ehrenfeucht-Fraïssé theorem [6, 8] which states that Duplicator has a winning strategy in the m-round first-order game played on two structures $\mathcal{A}$ and $\mathcal{B}$ if and only if $\mathcal{A} \equiv_m \mathcal{B}$.

Now, the relation $\equiv_m$ defines a finite number of (possibly infinite) equivalence classes on the ambient class. It can be proved that one of these classes has limiting probability 1, and this suffices to prove our theorem. For the sake of clarity, we present the idea of the proof on a simplified example only.

4. A 0–1 law by a 3^{r-k} strategy for a simplified problem

For this example, the class of structures under consideration is the set of square toroidal grids with a unary relation (we simply tag some vertices). We play r-round Ehrenfeucht-Fraïssé games on pairs of grids. The crucial fact we use is that any fixed square plane grid with vertices tagged at random appears in a toroidal grid with limiting probability 1.

It follows that Duplicator has a strategy to win *almost surely*, i.e., with limiting probability 1. Define a *distance* between two vertices of a grid by the minimum number of edges in a connecting path. The *ball* $N(c_1, \ldots, c_p; d)$ is the set of vertices at distance at most d from any c_i. Let $\mathcal{A}$ and $\mathcal{B}$ be two structures. Assume we are in round $k+1$ and that $a_1, \ldots, a_k$ have already been picked out of $\mathcal{A}$, $b_1, \ldots, b_k$ out of $\mathcal{B}$ in a way such that $N(a_1, \ldots, a_k; 3^{r-k})$ and $N(b_1, \ldots, b_k; 3^{r-k})$ are isomorphic, when viewed as substructures with naturally induced relations. Now, Spoiler picks an element out of either structure, say a_{k+1} out of $\mathcal{A}$—the case b_{k+1} out of $\mathcal{B}$ is symmetric. If $N(a_1, \ldots, a_{k+1}; 3^{r-k-1}) \subset N(a_1, \ldots, a_k; 3^{r-k})$, then Duplicator can trivially choose b_{k+1} in $N(b_1, \ldots, b_k; 3^{r-k})$ so that $N(a_1, \ldots, a_{k+1}; 3^{r-k-1})$ and $N(b_1, \ldots, b_{k+1}; 3^{r-k-1})$ are isomorphic. Otherwise, there is almost surely a ball in the complement of $N(b_1, \ldots, b_k; 3^{r-k-1})$ in $\mathcal{B}$ which is isomorphic to $N(a_{k+1}; 3^{r-k-1})$. Duplicator then chooses b_{k+1} to be its center. After r rounds, the balls $N(a_1, \ldots, a_r; 1)$ and $N(b_1, \ldots, b_r; 1)$ are almost surely isomorphic. Thus, Duplicator wins almost surely by following the strategy that we have just described. By the Ehrenfeucht-Fraïssé theorem, $\mathcal{A} \equiv_r \mathcal{B}$ almost surely. Therefore, one of the (finitely many) equivalence classes of $\equiv_r$ has limiting probability 1. Call it $\mathcal{E}_r$.

Consider now a first-order sentence S of rank r on toroidal grids. By the Ehrenfeucht-Fraïssé theorem, the set of all grids satisfying S is either contained in $\mathcal{E}_r$, or disjoint from $\mathcal{E}_r$. In the former case S has limiting probability 1, in the latter 0. We have thus proved a 0–1 law for the class of toroidal grids with a unary relation.

5. A 0–1 law for maps of a given genus

We first recall two difficult results on maps.

The first result [2, Sec. 5] plays the rôle of the crucial fact we used in the previous section, namely the limiting probability 1 of the appearance of a fixed plane grid in a toroidal grid. It states that for a class $\mathcal{C}$ of maps of fixed genus, there is a $c > 0$ such that for any given planar map $\mathcal{P}$, the property for maps in $\mathcal{C}$ to contain more than cn disjoint copies of $\mathcal{P}$ has limiting probability 1.

The second result [3] is about *representativity* of maps. The representativity of a map $\mathcal{M}$ on a surface $\mathcal{S}$ is the smallest number of intersections a non-contractible curve in $\mathcal{S}$ has with $\mathcal{M}$. The result is that for a class of maps of fixed genus, there is a $c > 0$ such that the property for maps to have representativity more than $c \ln n$ has limiting probability 1. This result is used in the proof of Theorem 1 to ensure the planarity of certain submaps built on balls playing a rôle similar to the $N(a_1, \ldots, a_k; 3^{r-k})$ of the previous section.

Next, the proof of Theorem 1 runs as for the example of the previous section: we prove a first-order 0–1 law for the class of all maps of a given genus by showing that for each r, Duplicator has an almost surely winning strategy in r-round Ehrenfeucht-Fraïssé games. More specifically, this strategy is a 3^{r-k} strategy using balls around elements picked by Spoiler and Duplicator. However, the notion of distance used is not that of the previous section. The proper distance to prove the result is by means of *quadrangulations* of maps. For a given map $\mathcal{M}$ on a surface, add a new point on each edge and a point in each face. Next add new edges from the new points on the edges to the new points in the faces. The quadrangulation of $\mathcal{M}$ is then the new map on the same surface built in this way. This construction induces a natural mapping from a map $\mathcal{M}$ to its quadrangulation $\mathcal{Q}$. We extend this map to the dart representation of $\mathcal{M}$ by mapping both co-darts defined by an edge to the image of this edge in $\mathcal{Q}$. A distance is then defined on the set $U_v \cup U_d \cup U_f$ of all vertices, darts and faces of the dart representation, as the distance between the images in $\mathcal{Q}$. This distance is not a metric, since two co-darts are at distance 0 for each other. However, the concept of balls it induces is sufficient for the proof of Theorem 1.

6. Conclusions

Theorem 1 has been refined for several classes of maps on a surface of fixed genus [1] (see this reference for missing definitions): the class of all maps; the class of smooth maps; the class of k-connected maps where k is 2 or 3; the class of k-connected triangulations where k is 1, 2 or 3. However, the question of a 0–1 law for planar graphs remains open, though we believe it should be true.

As for complexity results, we proved an $\exp_\infty(cn)$ *lower* bound for the complexity of the set of first-order sentences of limiting probability 1 in the case of the dart representation. Another result holds for an *extended* dart representation (see [1] for the definition): in this extended representation, we proved undecidability. What we have not been able to prove is an *upper* bound in the case of the dart representation, though we feel $\exp_\infty(dn)$ is a good candidate for such an upper bound.

Finally, all results presented here concern sentences of first-order logic. An extension to other logics seems reasonable, in particular to MSO (monadic second-order) logic, with application to the theory of databases.

Bibliography

[1] Bender (Edward A.), Compton (Kevin J.), and Richmond (L. Bruce). – Zero-one laws for maps. – In preparation, 1996.

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

[3] Bender (Edward A.), Gao (Zhi-Cheng), and Richmond (L. Bruce). – Almost all rooted maps have large representativity. *Journal of Graph Theory*, vol. 18, 1994, pp. 545–555.

[4] Chang (Chen Chung) and Keisler (H. Jerome). – *Model theory.* – North-Holland, Amsterdam, 1990, third edition, *Studies in Logic and the Foundations of Mathematics*, vol. 73.

[5] Edmonds (Jack R.). – A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society*, vol. 7, 1960, p. 646.

[6] Ehrenfeucht (A.). – An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, vol. 49, 1961, pp. 129–141.

[7] Fagin (Ronald). – Probabilities on finite models. *Journal of Symbolic Logic*, vol. 41, 1976, pp. 50–58.

[8] Fraïssé (Roland). – *Sur quelques classifications des systèmes de relations.* – Technical report, Université d'Alger, 1954. English summary.

[9] Glebskiĭ (Y. V.), Kogan (D. I.), Liogon'kiĭ (M. I.), and Talanov (V. A.). – Range and degree of realizability of formulas in the restricted predicate calculus. *Cybernetics*, vol. 5, 1969, pp. 142–154. – English translation.

[10] Tutte (William T.). – Combinatorial oriented maps. *Canadian Journal of Mathematics*, vol. 31, 1979, pp. 986–1004.

A grammar-based unification of several alignment and folding algorithms

Fabrice Lefebvre
LIX - École Polytechnique

June 24, 1996

[summary by Pierre Nicodème]

Abstract

We show that many popular models of folding and/or alignment may be described by a new formalism: multi-tape S-attribute grammars (MTSAGs). This formalism relieves the designer of biological models of implementation details. We present also a tool which, given a MTSAG, will output an efficient parser for this grammar and show that MTSAGs offer a new, efficient and useful way to handle stochastic context-free grammars. This summary is an extended abstract of [7].

1. Introduction

We shall see here that most popular models of alignment and/or folding of DNAs, RNAs or proteins, HMMs (Hidden Markov Models) [5], SCFGs (Stochastic Context-Free Grammars) [8] and CMs (Covariance Models) [3] share a common representation in terms of a new formalism: Multi-Tape S-Attribute Grammars (MTSAGs). This formalism is not only a help for the description of old or new methods. We designed and implemented a tool which, from the high-level description given by a MTSAG, will automatically generate the C source of an efficient C parser which is able to compute alignments and foldings. The speed and memory requirements of such generated parsers stand the comparison with programs manually written from dynamic programming relations. As a consequence, we show how to automatically build SCFGs from sets of unaligned, unfolded RNAs.

2. Definitions

We define a special "m-tape" alphabet which will handle sequence alignments, and then a "m-tape" extension of context-free grammars which will handle structures of alignments.

Definition 1. A *m-tape alphabet* Σ is a product of m alphabets $\Sigma^{(i)}$ augmented with the empty string: $\Sigma = \bigotimes_{i=1\ldots m}(\Sigma^{(i)} \cup \{\epsilon\})$. An element $a_1 \cdots a_l$ of the free monoid Σ^*, generated by formal concatenation of m-tape elements of Σ, is called a *m-tape alignment* of length l. The empty alignment of Σ^* is denoted by ϵ.

Example. $(abba, dcd)$ is a 2-tape input string on $\Sigma^{(1)} = \{a, b\}$ and $\Sigma^{(2)} = \{c, d\}$. We shall also write this 2-tape input string as $\substack{abba\\dcd}$, which is a somewhat more natural notation in the context of alignments. This 2-tape input string has a 2-tape input substring $\substack{bb\\dc}$.

Definition 2. Given any m-tape alignment $a_1 \cdots a_l$, we get a m-tape input string by concatenation, or *ϵ-deletion*, of symbols of the projection of $a_1 \cdots a_l$ on every tape.

$$\Sigma^* \longrightarrow \langle \Sigma^* \rangle, a_1 \cdots a_l \longrightarrow \langle a_1 \cdots a_l \rangle.$$

$$
\begin{array}{rll}
start \longrightarrow & frame0 & (0)\\
frame0 \longrightarrow & frame0\begin{bmatrix} X \\ X \end{bmatrix} \mid \begin{bmatrix} - \\ - \end{bmatrix} & (0)\\
\mid & frame0\begin{bmatrix} X \\ Y \end{bmatrix} & (2)\\
\mid & frame1\begin{bmatrix} - \\ X \end{bmatrix} \mid frame2\begin{bmatrix} X \\ - \end{bmatrix} & (1)\\
frame1 \longrightarrow & frame1\begin{bmatrix} X \\ X \end{bmatrix} & (1)\\
\mid & frame1\begin{bmatrix} X \\ Y \end{bmatrix} & (3)\\
\mid & frame0\begin{bmatrix} X \\ - \end{bmatrix} \mid frame2\begin{bmatrix} - \\ X \end{bmatrix} & (3)\\
frame2 \longrightarrow & frame2\begin{bmatrix} X \\ X \end{bmatrix} & (1)\\
\mid & frame2\begin{bmatrix} X \\ Y \end{bmatrix} & (3)\\
\mid & frame0\begin{bmatrix} - \\ X \end{bmatrix} \mid frame1\begin{bmatrix} X \\ - \end{bmatrix} & (3)
\end{array}
$$

FIGURE 1. In this weighted left-regular grammar, weights are written in parentheses after each group of productions having the same weight. Later on, weights will be turned into attribute evaluation functions.

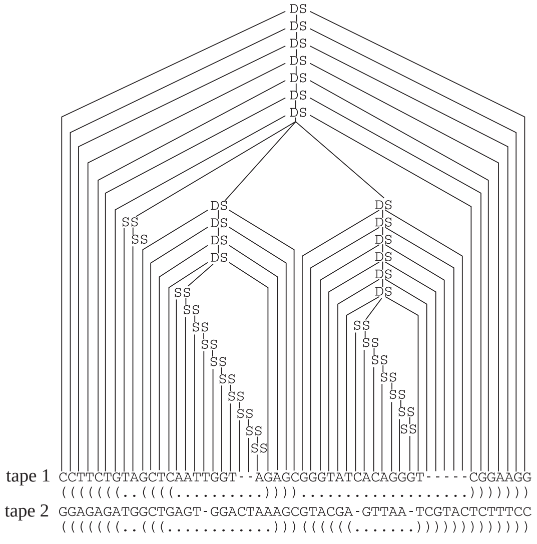

DS
SS
tape 1 CCTTCTGTAGCTCAATTGGT--AGAGCGGGTATCACAGGGT-----CGGAAGG
(((((((..((((..........))))...................)))))))
tape 2 GGAGAGATGGCTGAGT-GGACTAAAGCGTACGA-GTTAA-TCGTACTCTTTCC
(((((((..(((............)))((((((.......)))))))))))))

FIGURE 2. Derivation tree of an alignment of two RNAs.

EXAMPLE. Our 2-tape input string $\genfrac{}{}{0pt}{}{abba}{dcd}$ may be defined as an ϵ-deletion of the alignments $\langle \left[\begin{smallmatrix}\epsilon\\ d\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\ \epsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\ \epsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\ c\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\ d\end{smallmatrix}\right]\rangle$ or $\langle \left[\begin{smallmatrix}a\\ \epsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\ d\end{smallmatrix}\right]\left[\begin{smallmatrix}\epsilon\\ c\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\ \epsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\ d\end{smallmatrix}\right]\rangle$.

Searls did show that the alignment of two strings according to some edit-distance may be carried out by some simple 2-tape nondeterministic finite automaton (NFA) with weighted transitions [9]. The sought alignment has a minimal total weight. The set of alignments recognized by a Searls' NFA is a regular language over our 2-tape terminal alphabet, and may be described by a regular grammar with weighted productions (see figure 1).

As regular grammars are a proper subset of context-free grammars, we found natural to generalize this idea of alignment to m-tape (i.e. the terminal alphabet is a subset of a m-tape alphabet) context-free grammars (MTCFGs) and their recognizing devices, namely m-tape nondeterministic pushdown automata (NPDA). Weighted transitions of NFA are easily translated into weighted pop-transitions of NPDA. The sought alignment is obtained from a sequence of pop-transitions of the NPDA which has an optimal (minimal for some problems, maximal for others, etc...) total weight.

Figure 2 shows how alignments and structures may be deduced from a single m-tape derivation. The underlying grammar may be easily recovered. Base pairings are inferred from derivations of DS (Double-Strand) and they are given below each tape. Notice that a double-strand has been defined as a substructure whose ends must be paired on at least one tape, whereas a single-strand (SS) may only have unpaired bases on both tapes.

DEFINITION 3. A ***m*-tape context-free grammar** $G = (V_T, V_N, P, S)$ consists of a finite set of terminals V_T such that V_T *is a subset of a **m**-tape alphabet*, a finite set of nonterminals V_N such that $V_N \cap V_T = \emptyset$, a finite set of productions (rewriting rules) P and a start symbol $S \in V_N$. Let $V = V_T \cup V_N$ denote the vocabulary of the grammar. Each production in P has the form $A \rightarrow \alpha$, where $A \in V_N$ and $\alpha \in V^*$. A is the left-hand side of the production and α its right-hand side.

A *derivation tree* is a planar representation of a sequence of derivations (replacements of a nonterminals A in a string of V^* by strings α such that $A \rightarrow \alpha$) and it is a result of *parsing.* The language $L(G)$ is the set of m-tape input strings generated by G: $L(G) = \{\langle u\rangle \in \langle V_T^*\rangle \mid S \rightarrow^* u\}$.

Example. The following toy MTCFG will align two properly parenthesized strings interspersed with a:

$$S \rightarrow \left[\begin{smallmatrix}(\\(\end{smallmatrix}\right] S \left[\begin{smallmatrix})\\)\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}a\\a\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}a\\-\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}-\\a\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}-\\-\end{smallmatrix}\right] \mid SS$$

In this MTSAG, the structure defined by parentheses must be the same on both tapes, but substrings of a may be aligned with gaps (denoted by $-$ in terminals instead of ϵ, because a dash is appropriate, and even expected, in the context of alignments).

Definition 4. Let $G = (V_T, V_N, P, S)$ be a proper m-tape context-free grammar. For every tape $i\,(1 \leq i \leq m)$, define the *projected grammar* $G^{(i)}$ as the conversion of the grammar $(V_T^{(i)}, V_N, P^{(i)}, S)$ into a proper grammar, where $V_T^{(i)}$ and $P^{(i)}$ are the sets of values on tape i of all the elements of V_T and P respectively.

Example. The MTCFG of the preceding example has the same projected grammar on both tapes :

$$S \rightarrow (S) \mid () \mid a \mid SS$$

Projected grammars are useful for the study of the complexity of our parsing algorithm as a function of the ambiguity of MTCFGs.

We said earlier that we could assign a cost to each alignment or folding produced by a NPDA, thanks to weights on pop-transitions. This cost-evaluation step is essential for the determination of an optimal cost alignment or folding.

We use the general mechanism of synthesized attributes, or S-attributes which, together with MTCFGs, give us m-tape S-attribute context-free grammars, or MTSAGs. S-attributes are attributes which are assigned to every vertex of a derivation tree and which are computed from the bottom of a derivation tree (i.e. every terminal has a known S-attribute) to its root by means of attribute evaluation functions associated to grammar productions. Thanks to these functions, the computation of the final attribute of the derivation tree does not have to rely on a fixed, predetermined, operation (summation, multiplication, ...), as it would have been the case with weighted productions. In our implementation, attribute evaluation functions are C functions. We have already shown the effectiveness of S-attributes with our adaptation of the thermodynamic model of folding to context-free grammars [6]. This algorithm uses a parse table to store the shared forest of derivation trees of a m-tape input string.

Definition 5. A *m-tape S-attribute grammar* is denoted by $G = (V_T, V_N, P, S, \mathcal{A}, S_{\mathcal{A}}, F_P)$. It is an extension of a m-tape context-free grammar $G = (V_T, V_N, P, S)$, where an attribute $x \in \mathcal{A}$ is attached to each symbol $X \in V$ and a string of attributes $\lambda \in \mathcal{A}^*$ to each string $\alpha \in V^*$. $S_{\mathcal{A}}$ is a function from V_T to $\mathcal{A}$ assigning attributes to terminals. F_P is a set of functions from $\mathcal{A}^*$ to $\mathcal{A}$. A function $f_{A \rightarrow \alpha}$ is in F_P iff $A \rightarrow \alpha$ is in P.

The attribute λ of a string α is the concatenation of the attributes of the symbols in α. When a function $f_{A \rightarrow \alpha}$ is applied to the attribute λ of a string α derived from A, it returns the attribute x of A (hence the bottom-up computation of attributes).

3. Syntax analysis for MTSAGs

A generalization of Cocke-Younger-Kasami's algorithm (CYK) would be an easy algorithm to parse m-tape input strings. This algorithm has a time complexity of $O(n^3)$ and a space complexity of $O(n^2)$ when only one tape of size n is considered [1]. A generalization to m tapes, each of size n, would lead to an algorithm having a complexity of $O(n^{3m})$ in time and $O(n^{2m})$ in space.

To overcome the limitations of CYK's algorithm, we generalized our parsing algorithm for 1-tape MTSAGs [6].

When constructing the parse table, a minimum condition of usefulness is applied. This condition means that an item is never add to an entry if it has no chance of being used in a derivation tree, up to the already parsed part of the m-tape input string. This condition is akin to a condition verified by Earley's parsing algorithm and it is the key to the lower parsing complexities of our algorithm when some projections of the underlying MTSAG are unambiguous.

In fact, out algorithm may be considered as an improvement of Earley's algorithm, where Earley's items $[\Delta \rightarrow \alpha \cdot \beta, i], (\alpha, \beta \in V^*)$ which share the same α and i are factorized into a single item $[\Delta \rightharpoonup \alpha, i]$. Also, non-kernel items of Earley's algorithm are replaced by much smaller sets of expected non-terminals.

PROPOSITION 1 (1-TAPE COMPLEXITY). *Let G be a proper 1-tape MTSAG and let $r \geq 1$ be the maximum number of nonterminals appearing at the right-hand side of a production of G. For a tape of length n, the time and space complexities of the previous parsing algorithm are, in order of decreasing constraints on G:*

- *Equal and at most $O(n)$ if G is LR(k) and not right-recursive (this encompasses left-regular grammars);*
- *equal and at most $O(n^2)$ if G is unambiguous;*
- *$O(n^{r+1})$ and $O(n^r)$ for a generic proper MTSAG.*

PROPOSITION 2 (**m**-TAPE COMPLEXITY). *Let G be a proper m-tape MTSAG. The time complexity of our parsing algorithm on G is equal to the product of the parsing complexities of the same algorithm applied on each tape i with each projected grammar $G^{(i)}$. The same kind of result holds for space complexities. Hence the time complexity is at most $O(n^{m(r+1)})$, and the space complexity is at most $O(n^{mr})$, for m-tapes of size n*

In practice, MTSAGs that we used verified $r \leq 2$ and thus the time and space complexities of our parsers for those grammars were respectively $O(n^{3m})$ and $O(n^{2m})$ at most, but were sometimes much better.

4. Stochastic Context-Free Grammars

An essential aspect of MTSAGs is the ability to easily generate efficient parsers from grammars. On the basis of the tool we had already written to generate parsers from 1-tape S-attribute grammars, we designed a new tool, MTSAG2C, which automatically generates the C source of a parser from a given MTSAG. The generated parser is able to read tapes (thanks to a lexical analyzer provided by the user), parse tapes, and then output a single derivation tree which satisfies constraints given in the MTSAG.

When using 2-tape MTSAGs for SCFGs, we transfer on the first tape the high-level description of a family of RNAs usually used with SCFGs, and on the second tape the RNA to be folded and aligned. All rules used by the traditional SCFG generating tool to generate a SCFG from its high level description are then written down as a single, fixed, MTSAG. This has the additional benefit of shortening the development cycle (see figure 3b).

We compared the parser generated from a 1-tape version of a 97 nonterminals SCFG (this parser already proved to be quite fast [6]) to the parser generated from the 2-tape version of this SCFG (figure 4(a)).

Tests done on an Alpha 2100-500MP give the results:

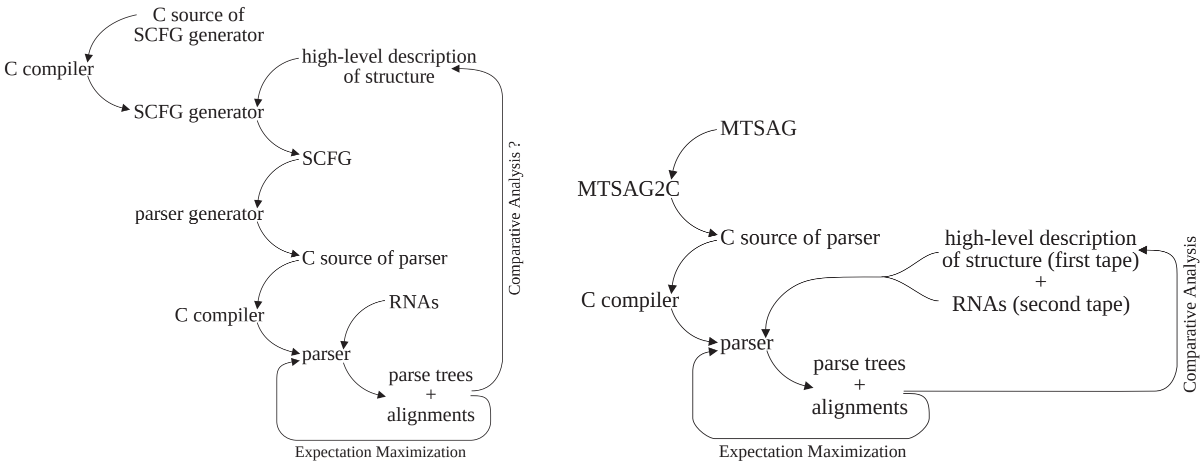


FIGURE 3. (left) 1-tape MTSAG; (right) 2-tape MTSAG. Development cycle of a MTSAG implementation of SCFGs. It has been suggested [4] that a comparative analysis of alignments resulting from parsing may be used to build a new SCFG or a new high-level description of it. With 2-tape MTSAGs, this kind of feedback is as easy to implement as the feedback designed for CMs by Eddy and Durbin [3].

	1-tape	2-tape
83 bases tRNA		
time in seconds:	0.45	0.33
space in Mbytes:	1.8	0.9

With MTSAGs you do not have to generate and compile another parser every time you modify the high-level description of your family (figure 3 (left)). Instead, we may use the following adaptation of the procedure of Eddy and Durbin to learn their CMs from initially unaligned and unfolded RNAs:

(1) Use a MTSAG adaptation of any folding algorithm (Sankoff, Zuker) to get a rough (and even wrong) initial folding. Convert this folding to a suitable first tape (by replacing all single strands by '*' for instance);
(2) Use Dirichlet priors to estimate probabilities;
(3) Align and fold all RNAs with a 2-tape MTSAG;
(4) Optimize probabilities from results of the previous step and repeat the previous step until probabilities converge;
(5) Use a comparative analysis algorithm on alignments of step 3 to get a new approximation of the common structural features of all RNAs. Then convert this approximation to a suitable first tape;
(6) Repeat steps 2 to 6 until the first tape converges.

5. Conclusion

We introduce a new way to describe SCGFs in the form of 2-tape MTSAGs and special first tapes. This new way alleviates the need for specialized SCFG building tools and for recompilations of parsers every time the model is changed (only the first tape has to be changed).

MTSAGs may also be applied to most useful sequence analysis methods which were usually expressed with dynamic programming relations (Smith-Waterman alignment model, global alignment, HMMs, simultaneous alignment and folding of RNAs). We believe that MTSAGs should be

```
(((((((..((((********)))).(((((.......)))))********(((((.......)))))))))))).
CCUUCUGUAGCUCAAUUGGUAGAGCAUGUGACUGUAGAGUAUGCGGGUAUCACAGGGUCGCUGGUUCGAUUCCGGCCGGAAGG
```

(a) unaligned 2-tape input string.

```
(((((((..((((********)))).(((((.......)))))*******----------*(((((.......)))))))))))).
CCUUCUGUAGCUCAAUUGGUAGAGCAUGUGACUGUAGAGUAUGC--GG-GUAUCACAGGGUCGCUGGUUCGAUUCCGGCCGGAAGG
```

(b) 2-tape alignment of the previous 2-tape input string.

FIGURE 4. Unaligned and aligned version of a 2-tape input string. The first tape of this 2-tape input-string has a cloverleaf-like structure. This structure has two single strands which may have a variable length around 8 bases. The second tape is the RNA DY6050 extracted from a well known freely available compilation of tRNAs [10].

used instead of dynamic programming relations because these relations hinder the inventiveness of designers of new sequence analysis models.

We also gave a sketch of a method to build stochastic models from unaligned, unfolded RNAs. However, divide and conquer methods may be a prerequisite for long RNAs [2, 4]. We will try to apply MTSAGs to these methods.

Bibliography

[1] Aho (Alfred V.) and Ullman (Jeffrey D.). – *The Theory of Parsing, Translation, and Compiling.* – Prentice-Hall, 1972, vol. 1.

[2] Corpet (Florence) and Michot (Bernard). – RNAlign program: alignment of RNA sequences using both primary and secondary structures. *Computing Applications in the Biosciences*, vol. 10, n° 4, 1994, pp. 389–399.

[3] Eddy (Sean R.) and Durbin (Richard). – RNA sequence analysis using covariance models. *Nucleic Acids Research*, vol. 22, n° 11, 1994, pp. 2079–2088.

[4] Grate (Leslie). – Automatic RNA secondary structure determination with stochastic context-free grammars. In *Third International Conference on Intelligent Systems for Molecular Biology.* pp. 136–144. – AAAI Press, 1995.

[5] Krogh (A.), Brown (M.), Mian (I. S.), Sjölander (K.), and Haussler (D.). – Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, vol. 235, 1994, pp. 1501–1531.

[6] Lefebvre (Fabrice). – An optimized parsing algorithm well suited to RNA folding. In *Third International Conference on Intelligent Systems for Molecular Biology.* pp. 222–230. – AAAI Press, 1995.

[7] Lefebvre (Fabrice). – A grammar-based unification of several alignment and folding algorithms. In *Fourth International Conference on Intelligent Systems for Molecular Biology.* pp. 143–154. – AAAI Press, 1996.

[8] Sakakibara (Yasubumi), Brown (Michael), Hughey (Richard), Mian (I. Saira), Sjölander (Kimmen), Underwood (Rebecca C.), and Haussler (David). – Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research*, vol. 22, 1994, pp. 5112–5120.

[9] Searls (David B.) and Murphy (Kevin P.). – Automata – theoretic models of mutation and alignment. In *Third International Conference on Intelligent Systems for Molecular Biology.* pp. 341–349. – AAAI Press, 1995.

[10] Steinberg (S.), Misch (A.), and Sprinzl (M.). – Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Research*, vol. 21, 1993, pp. 3011–3015.

Part 2

Symbolic Computation

Linear recurrences, linear differential equations, and fast computation

Bruno Salvy
INRIA Rocquencourt

November 13, 1995

[summary by Philippe Dumas]

Linear recurrences and linear differential equations with polynomial coefficients provide a finite representation of special functions or special sequences. Many algorithms are at our disposal; some give a way to automate the computation of recurrences or differential equations; some provide solutions to recurrences or differential equations; and some give the asymptotic behaviour of these solutions, directly from the recurrence or differential equation. All of this provides a method to efficiently compute special functions and special sequences.

1. Classical algorithms concerning formal power series

In the sequel, we use the ring $\mathbb{A}[[x]]$ of formal power series

$$F(x) = \sum_{n=0}^{+\infty} f_n x^n$$

with coefficients f_n in a commutative ring $\mathbb{A}$; this ring is assumed to contain the field $\mathbb{Q}$ of rational numbers, even though it is possible to consider a more general situation. Practically, one deals with truncated series

$$F(x) = \sum_{n=0}^{N} f_n x^n + O(x^{N+1}),$$

that is to say essentially polynomials. It must be noted that there exist lazy algorithms to deal with truncated series of arbitrary order, but their cost is generally excessive. We indicate how to deal with basic operations [7, Chap. 4].

Product of polynomials. The naive method to obtain the product of two polynomials of degree N has complexity $O(N^2)$ arithmetic operations. A better way is Karatsuba's algorithm, which has complexity $O(N^{\log_2 3}) = O(N^{1.59})$. The idea behind the algorithm resides in writing

$$P(x) = P_0(x) + x^k P_1(x), \qquad Q(x) = Q_0(x) + x^k Q_1(x),$$

$$P(x)Q(x) = P_0(x)Q_0(x) + R(x)x^k + P_1(x)Q_1(x)x^{2k},$$

$$R(x) = (P_0(x) + P_1(x))(Q_0(x) + Q_1(x)) - (P_0(x)Q_0(x) + P_1(x)Q_1(x)),$$

with $k \simeq N/2$; this formula needs only three multiplications of polynomials of degree less than k instead of four multiplications, and this leads to an efficient recursive computation.

From a practical standpoint, Karatsuba's method becomes efficient in Maple for an N greater than about a hundred. The fast Fourier transform algorithm needs a much larger value of N to be useful.

Composition. Here the goal is the computation of the first N coefficients of the series $F(G(x))$, where $g_0 = 0$. The naive method leads to a computation with $O(N)$ series multiplications. Brent and Kung's algorithm [2] has a better behaviour. It consists of three steps; first write $F(x)$ as

$$F(x) = F_0(x) + F_1(x)x^k + F_2(x)x^{2k} \cdots + F_{k-1}(x)x^{k(k-1)},$$

where $F_0(x)$, $F_1(x)$, ..., $F_{k-1}(x)$ are the series obtained by factoring out the powers of x^k, where $k = \lceil (N+1)^{1/2} \rceil$; next compute the powers $G^i(x)$ for i from 2 to $k-1$, and the series $F_i(G(x))$; finally, compute $T(x) = G^k(x)$ and $F(G(x))$ using a Horner scheme.

The algorithm uses $3k$ series multiplications and $O(N)$ coefficient multiplications, hence it has cost $O(N^{1/2})$ if the unit cost is series multiplication. Via Karatsuba's algorithm, this gives a cost of $O(N^{2.09})$ expressed in terms of coefficient multiplications, while the naive method has cost $O(N^3)$.

Powering and simple functions. Powering and simple functions are a particular case of composition, but in this case it is possible to be more efficient. We show the idea for the case of powering. If $H(x) = F^\alpha(x)$, then $H(x)$ satisfies the equation

$$H'(x)F(x) = \alpha F'(x)H(x),$$

therefore the coefficients of $H(x)$ are provided by the following recurrence

$$\sum_{k=0}^{n} k h_k f_{n-k} = \alpha \sum_{k=0}^{n} (n-k) h_k f_{n-k}.$$

This makes it possible to compute the first N coefficients at a cost of $O(N^2)$ coefficient multiplications, instead of $O(N^{2.09})$.

Newton iteration. An ever better way to compute elementary functions is Newton's method. If we search for a series $y(x)$ such that $\Phi(x, y(x)) = 0$, we use the recurrence

$$y_{k+1}(x) = y_k(x) - \frac{\Phi(x, y_k(x))}{\partial\Phi/\partial y(x, y_k(x))} \mod x^{2k+2}.$$

We start from $y_0(x) = 0$ and the formula is iterated until $2k + 2 > N$. The number of correct coefficients is doubled at each step. For example, the reciprocal $y(x) = 1/F(x)$ satisfies the equation $\Phi(x, y) = 1/y - F(x) = 0$ and the recursion is $y_{k+1} = 2y_k - F(x)y_k^2$. In the same way one can compute the logarithm $\ln(F(x))$, the exponential $\exp(F(x))$, and solutions of simple differential equations. In all these cases the complexity of the computation is the complexity of the multiplication, that is $O(N^{1.59})$.

For the reversion of series the same method can be used. Given $F(x)$ with $f_0 = 0$, $f_1 \neq 0$, one looks for a series $y(x)$ such that $F(y(x)) = x$. This is carried out by Newton's method applied to the equation $F(y) - x = 0$; hence the recurrence is

$$y_{k+1}(x) = y_k(x) - \frac{F(y_k(x)) - x}{F'(y_k(x))} \mod x^{2k+2}.$$

The cost is of the same order as the composition cost, because of the terms $F(y_k(x))$ and $F'(y_k(x))$.

Linear differential equations. Assume that the power series $F(x)$ satisfies a linear differential equation

$$a_0(x)y^{(k)} + a_1(x)y^{(k-1)} + \cdots + a_k(x)y = 0,$$

whose coefficients are polynomials. If 0 is an ordinary point, this differential equation translates into a linear recurrence for the coefficients of $F(x)$. This leads to an algorithm whose cost is $O(N)$, while the preceding ones use at best $O(N^{1.59})$ basic operations.

Obviously, the complexity $O(N)$ is optimal, therefore for large N there is great interest in finding a linear differential equation with $F(x)$ as a solution, if possible. In the sequel, we will focus our attention on such power series.

2. Univariate holonomic series

A power series is said to be holonomic if it is a solution of a linear differential equation with polynomial coefficients. In the same manner, a sequence is said to be holonomic if it is a solution of a linear recurrence with polynomial coefficients.

It is easy to see that rational series, $\exp(x)$, $\sin x$, $\cos x$, $\log(1+x)$, and the Bessel functions $J_\nu(x)$ are all holonomic. Rational, factorial, Fibonacci, and hypergeometric sequences are all holonomic sequences. Recall that a sequence is hypergeometric if the sequence of quotients of consecutive terms is a rational sequence.

Both definitions are related by the following property: a sequence is holonomic if and only if its generating series is holonomic. The proof is easy and uses the simple but basic correspondences which may be summarized as follows,

$$\begin{array}{rcl} F(x) & \longleftrightarrow & f_n, \\ x^k F(x) & \longleftrightarrow & f_{n-k}, \\ xF'(x) & \longleftrightarrow & nf_n. \end{array}$$

Closure properties. The set of holonomic series is closed with respect to sum, Cauchy product, Hadamard product, Borel transform, and Laplace transform [10]. We give a sketch of the proof for the Cauchy product. If $F(x)$ satisfies a differential equation of order s and $G(x)$ satisfies a differential equation of order t, we formally compute the derivatives of $H(x) = F(x)G(x)$ and, using the equations satisfied by $F(x)$ and $G(x)$, we express them as linear combinations of the products $F^{(i)}(x)G^{(j)}(x)$ where the indices i and j vary from 0 to $s-1$ and from 0 to $t-1$ respectively. The space of such combinations has a finite dimension, hence the derivatives of $H(x)$ satisfy a dependance relation, that is a linear differential equation.

There is a similar result for holonomic sequences: the sum, product, and convolution of two holonomic sequences are holonomic; the indefinite summation of a holonomic sequence is holonomic. Both types of closure properties are interrelated, and the proofs use whichever is easier.

Identity proving. An application of holonomy, widely exemplified by D. Zeilberger, is identity proving [12]. The idea is the following: to prove $F(x) = G(x)$, build the equation satisfied by $F(x) - G(x)$, and compute sufficiently many initial conditions to ensure $F(x) - G(x) = 0$.

Here is a simple example. Suppose we want to prove the identity

$$\sqrt{x}J_{1/2}(x) = \sqrt{\frac{2}{\pi}}\sin x,$$

where $J_{1/2}(x)$ is a Bessel function of index 1/2. This function satisfies a second order differential equation, while the square root satisfies a first order equation; hence the product is a solution of an equation of order not greater than 2. On the other hand, sine satisfies a second order equation; therefore the difference of the two sides of the formula satisfies an equation of order not greater than 4. It suffices to verify that the power series of the difference is $O(x^4)$, using the differential equations and the initial conditions defining the components. The alert reader may think we were lucky, because $\sqrt{x}J_{1/2}(x)$ has a power series expansion at 0, while $J_{1/2}(x)$ has not. But, if this had not been the case, we would have used use another point than 0.

Algebraic functions. Algebraic functions are holonomic. Comtet [4] gave an algorithm to compute the differential equation satisfied by a function $F(x)$ solution of $P(x,y) = 0$, where P is an irreducible polynomial. The idea is to find a Bezout relation $UP + VP_y = 1$ by the extended Euclidean algorithm and use $P_x + P_y F' = 0$ to express the successive derivatives of $F(x)$ as polynomials in $F(x)$ of degree less than $d = \deg_y P$. The family of powers 1, $F(x)$, …, $F^{d-1}(x)$ is a basis of the space generated by the derivatives of $F(x)$, and there is a dependance relation between $F(x)$, $F'(x)$, …, $F^{(d)}(x)$.

Algebraic substitution. If $F(x)$ is holonomic and $G(x)$ is algebraic, then $F(G(x))$ is holonomic by the same kind of technique as above. An immediate application of this result is the following: if f_n is a holonomic sequence, then its Euler transform

$$h_n = \sum_{k=0}^{n} (-1)^k \binom{n}{k} f_k$$

is holonomic too. This is obvious because the two generating functions are connected by $H(x) = F(-x/(1-x))/(1-x)$.

3. Search for solutions

Recurrence relations and differential equations almost never have explicit solutions, but if an explicit solution exists it might be important to recognize it, and find the solution. Above all an explicit solution gives a global information about the equation.

Rational solutions to recurrences. Abramov [1] gives a method to obtain the rational solutions $u_n = P(n)/Q(n)$ of a recurrence

$$a_0(n)u_{n+k} + \cdots + a_k(n)u_n = b(n),$$

where a_0, …, a_k, and b are polynomials. The principle which guides the algorithm is: the zeros of the coefficients must match the poles of u_n and its shifts $u_{n+\ell}$. As a consequence, Q must be a multiple of $\gcd(a_0(n-k), \ldots, a_k(n))$ if the roots of Q do not differ by an integer. The last condition is not necessarily fulfilled; to avoid this problem one considers a recurrence satisfied by the sequence $v_n = u_{nh}$, where h is the maximal difference between two roots of Q. It must be noted that the number h is not greater than the maximal difference between the roots of $a_k(n)$ and $a_0(n-k)$.

Indefinite hypergeometric summation. The indefinite sum [6] of f_k is equivalent to finding a closed formula for $F_n = \sum^n f_k$ where f_k is a given sequence. This relation means

$$F_n - F_{n-1} = f_n$$

for all n. If f_n is assumed to be hypergeometric, and we look for a hypergeometric F_n, the relation $1 - F_{n-1}/F_n = f_n/F_n$ shows that the sequence $u_n = F_n/f_n$ must be rational. Hence we are led to search for a rational solution of the equation

$$u_n - \frac{f_{n-1}}{f_n} u_{n-1} = 1.$$

Hypergeometric solutions to recurrences. Petkovšek's algorithm [8] provides the hypergeometric solutions of a linear recurrence

$$a_0(n)u_{n+k} + \cdots + a_k(n)u_n = 0,$$

where a_0, …, a_k, and b are polynomials. Writing $u_{n+1}/u_n = P(n)/Q(n)$ and substituting leads to a non-linear equation, which is not tractable. There exists a decomposition

$$\frac{u_{n+1}}{u_n} = \frac{P(n)}{Q(n)} \frac{A(n+1)}{A(n)}$$

in which all pairs $(A(n), P(n))$, $(A(n), P(n))$, $(P(n), Q(n))$, $(P(n), Q(n+1))$, …, $(P(n), Q(n+k))$ are relatively prime. With this decomposition a substitution gives

$$a_0(n)A(n+k)P(n+k)\cdots P(n) + a_1(n)A(n+k-1)P(n+k-1)\cdots P(n)Q(n+k) \\ + \cdots + a_k(n)A(n)Q(n+k)\cdots Q(n) = 0.$$

This equation is still non-linear, but it shows that $P(n)$ divides $a_k(n)$, and $Q(n+k)$ divides $a_0(n)$. Finally it suffices to test all pairs of factors of $a_0(n-k)$ and $a_k(n)$.

Note that this algorithm is a powerful tool; it is equivalent to finding factors of order 1 on the right of the recurrence.

Symbolic solutions to differential equations. Searching for generalized hypergeometric solutions is a first approach to a linear differential equation: the recurrence satisfied by the coefficients of the series is computed; the hypergeometric solutions to this recurrence are found; finally the result is translated from sequences to generating functions.

The more general class of Liouvillian functions may be used. Liouvillian functions are obtained from rational functions with rational coefficients by repeated use of the four elementary operations, taking exponentials and logarithms, integration, and algebraic extensions. Singer gives a purely theoretic algorithm to obtain Liouvillian solutions of linear differential equations of arbitrary order. Kovacic's algorithm for equations of order 2 is partially implemented in most computer algebra systems. The theory behind all these algorithms is differential Galois theory. It is difficult to use, because for each order it is necessary to classify the Galois groups which come into play [11].

4. Asymptotic analysis

Even when no explicit solution of a differential equation is known, it is possible to perform an asymptotic analysis. The theory of linear differential equations prescribes the asymptotic behaviour of a solution near a singularity and this asymptotic behaviour is strongly related to the asymptotic behaviour of the Taylor coefficients of the solution.

Singular points. The solutions of a linear differential equation

$$a_0(x)y^{(k)}(x) + \cdots + a_k(x)y(x) = 0$$

may only have singularities at the roots of the dominant coefficient $a_0(x)$, and possibly at infinity. In addition all formal solutions to the equation are known. A logarithmic sum is a formal series

$$\lambda(z) = z^\alpha \sum_{j=1}^{J} \sum_{i \geq 0} c_{ij} z^i \log^j z,$$

and a formal solution in the neighbourhood of the root a of $a_0(x)$ is a finite combination of logarithmic sums

$$y(x) = \sum_{k=0}^{K} \lambda_k(z) \exp(P_k(z)), \qquad z = \frac{1}{(1-x/a)^{1/r}},$$

which formally satisfies the differential equation. All quantities involved in these formulae can be explicitly computed. In the case where the point a is a regular singular point, that is to say $a_\ell(x) = (x-a)^{k-\ell} A(x)$ for $\ell = 0, \ldots, k$, and $A_\ell(x)$ is analytic in the neighbourhood of a, the formal solutions are logarithmic sums and locally define actual solutions, with a possible ramification point at a. Conversely, in the case of an irregular singular point the formal solutions are generally divergent series, but provide asymptotic expansions for actual solutions in a sector with origin a.

The preceding classification demonstrates that the composition of two holonomic functions is not necessarily holonomic. For instance $1/\sin x$, which is the composition of the two holonomic functions $\sin x$ and $1/x$, is not holonomic because it has an infinite number of singularities. The sequence of Bell numbers is not holonomic because its exponential generating function $\exp(e^x - 1)$ does not have the right form, given by the formula above (after changing x into $1/x$).

Singularity analysis. The smallest singularity ρ of a function analytic in a neighbourhood of zero prescribes the behaviour of the Taylor coefficients of the function. This rough correspondence may be strongly refined [5]; indeed an asymptotic expansion in some sufficiently large neighborhood of the singularity a of smallest modulus

$$f(x) \underset{x \to a}{=} c_0 (1-x/a)^{\alpha_0} \log^{\beta_0} \frac{1}{1-x/a} + c_1 (1-x/a)^{\alpha_1} \log^{\beta_1} \frac{1}{1-x/a} + \cdots$$

translates into an asymptotic expansion for the coefficient of the Taylor expansion of $f(x)$ at 0

$$f_n \underset{n \to \infty}{=} \rho^{-n} \frac{n^{-\alpha_0 - 1}}{\Gamma(-\alpha_0)} \log^{\beta_0} n \left(c_0 + \frac{d_1}{\log n} + \cdots \right).$$

This result leads to the following idea: to study the asymptotic behaviour of a sequence which satisfies a linear recurrence it suffices to translate the recurrence into a differential equation for the generating function; next a singularity analysis of this function gives the asymptotic behaviour of the sequence. This simple method presents a difficulty. The function is determined as a solution of a differential equation and some initial conditions, which are specified at the point 0. The study of the differential equation provides a basis of formal solutions near the smallest singularity, but there is no direct way to express the generating function with respect to this basis. Obviously if a closed form of the function is available it is possible to realize the connection between the data at 0 and the behaviour at the smallest singularity; but in that case more direct procedures may be used. Generally, it is necessary to use analytic continuation and a resummation method [9]. Note that such a method needs to know about the singularities of the Borel transform of the function; and we have seen that it is possible to compute the differential equation satisfied by the Borel transform of a holonomic function.

5. Multivariate holonomy

The machinery of holonomic sequences or functions is so powerful that it is tempting to generalize holonomy for sequences or functions with more than one variable.

Weyl algebra. The Weyl algebra $A_N(\mathbb{K})$ is an algebra of linear operators which is defined over the space of polynomials $\mathbb{K}[\mathbf{x}] = \mathbb{K}[x_1, \ldots, x_N]$. These operators are the partial derivatives ∂_j, the multiplications by the variable x_i's, and all their combinations. The generators $\partial_1, \ldots, \partial_N, x_1, \ldots, x_N$ satisfy the following commutation rules:

$$\partial_i \partial_j = \partial_j \partial_i, \qquad x_i x_j = x_j x_i$$

$$\partial_i x_j = x_j \partial_i \quad \text{for } i \neq j, \qquad \partial_i x_i = x_i \partial_i + 1.$$

Then, an element f of a module over the Weyl algebra is D-finite if the submodule spanned by f and all its derivatives $\boldsymbol{\partial}^\alpha f$ has a finite dimension over the field of rational functions $\mathbb{K}(\mathbf{x})$. An equivalent definition is obtained as follows: for f from an $A_N(\mathbb{K})$-module, consider the set of all equations $P(\mathbf{x}, \boldsymbol{\partial})f = 0$ satisfied by f; the polynomials $P(\mathbf{x}, \boldsymbol{\partial})$ are elements of the left ideal $\operatorname{Ann}(f)$ in the Weyl algebra; then f is D-finite if the quotient $A_N(\mathbb{K})/\operatorname{Ann}(f)$ of the Weyl algebra by the annihilator ideal $\operatorname{Ann}(f)$ has a finite dimension over $\mathbb{K}(\mathbf{x})$ as a vector space.

A more effective definition uses the idea of a rectangular system. A set of N polynomials $P_k(\mathbf{x}, \boldsymbol{\partial})$ from the Weyl algebra is a rectangular system if each polynomial involves only one partial derivative ∂_i, and each partial derivative appears in exactly one of these polynomials $P_k(\mathbf{x}, \boldsymbol{\partial})$. One proves that f is D-finite if and only if there exists a rectangular system contained in the annihilator ideal $\operatorname{Ann}(f)$. As a consequence a D-finite element f satisfies a special set of equations of the form

$$P_1(\mathbf{x}, \partial_1)f = 0, \qquad P_2(\mathbf{x}, \partial_2)f = 0, \qquad \ldots, \qquad P_N(\mathbf{x}, \partial_N)f = 0.$$

In addition, Bernstein worked out the concept of multivariate holonomy. The Weyl algebra is naturally graded by the degree: the degree of the monomial $\mathbf{x}^\alpha \boldsymbol{\partial}^\beta$ is $|\alpha| + |\beta|$, and the component F_d of the natural filtration is composed of the polynomials of degree not greater than d. For f from a module over the Weyl algebra, this induces a filtration of the submodule $A_N(\mathbb{K})f$; the component Γ_d is merely $F_d f$. It turns out that the dimension of Γ_d over $\mathbb{K}$ is expressed as a polynomial in d for all sufficiently large d. The degree of this polynomial is the Bernstein dimension of the module $A_N(\mathbb{K})f$. Moreover it is shown that the Bernstein dimension of $A_N(\mathbb{K})f$ is greater or equal to N. Now, f is *holonomic* if the Bernstein dimension of $A_N(\mathbb{K})f$ is exactly N.

Kashiwara's theorem proves that D-finiteness and holonomy are the same concept. But each one has its own merits. The D-finiteness property makes it easy to show that sums and products of holonomic functions are holonomic too. On the other hand, definite integration with respect to one of the x_i's preserves holonomy, and this is more easily shown using the definition of holonomy.

The link between sequences and generating functions is not as nice in the multivariate case as in the univariate case. A sequence u_ν, where the index ν is an N-tuple $(n_1, \ldots, n_N)$ is P-finite if the sequence u_ν and all its shifts $u_{\nu+\tau}$ span a finite dimensional space over $\mathbb{K}(\tau_1, \ldots, \tau_N)$. An equivalent formulation of the P-finiteness can be written as follows: there exists a rectangular system

$$P_1(\nu, S_1)u = 0, \qquad P_2(\nu, S_2)u = 0, \qquad \ldots, \qquad P_N(\nu, S_N)u = 0,$$

where S_i is the shift operator defined by $S_i u_\nu = u_{n_1, \ldots, n_i+1, \ldots, n_N}$. One proves that a sequence is P-finite if its multivariate generating function is D-finite. The reciprocal assertion is false.

The study of P-finite sequences shows it is interesting to consider a more general concept than Weyl algebras. This leads to Ore algebras, which are defined as polynomial algebras with some commutation rules for the variables [3]. For instance, the finite difference calculus in one variable is formalized by the algebra $\mathbb{K}\langle n, \Delta\rangle$ with $\Delta n = (n+1)\Delta + 1$.

Creative telescoping. We search for a recurrence relation for the definite sum $U_n = \sum_k u_{n,k}$, where the double sequence $u_{n,k}$ is P-finite. The idea is to find an equation $P(n, S_n, \Delta_k)u = 0$, where the variable k does not occur, S_n is the shift operator with respect to n, and Δ_k is the difference operator with respect to k; then, U satisfies $P(n, S_n, \Delta_k)U = 0$. Contrary to the case of holonomic functions such an equation does not exist a priori; but if it exists, it is possible to find it by a Gröbner basis technique. As an example we want to rederive the Franel relation on the sum

$$U_n = \sum_{k=0}^{n} \binom{n}{k}^3.$$

First we give a rectangular system for the double sequence $u_{n,k} = \binom{n}{k}^3$,

$$[(n-k+1)^3 S_n - (n+1)^3]u = 0, \qquad [(k+1)^3 S_k - (n-k)^3]u = 0.$$

Here the analogue to the Bernstein dimension is 2, hence elimination provides a relation $P(n, S_k, S_n)u = 0$. Next the summation with respect to k, and the substitution $S_k = 1$, or equivalently $\Delta_k = 0$, give the desired formula:

$$\begin{aligned}[(n+3)^3(3n+4)S_n^3 - (18n^3 + 114n^2 + 232n + 148)S_n^2 \\ -(3n+5)(15n^2+55n+48)S_n - 8(n+1)^2(3n+7)]\, U = 0.\end{aligned}$$

Bibliography

[1] Abramov (S. A.). – Rational solutions of linear differential and difference equations with polynomial coefficients. *USSR Computational Mathematics and Mathematical Physics*, vol. 29, n° 11, 1989, pp. 1611–1620. – Translation of the Zhurnal vychislitel'noi matematiki i matematicheskoi fiziki.

[2] Brent (R. P.) and Kung (H. T.). – Fast algorithms for manipulating formal power series. *Journal of the ACM*, vol. 25, 1978, pp. 581–595.

[3] Chyzak (Frédéric) and Salvy (Bruno). – *Non-commutative Elimination in Ore Algebras Proves Multivariate Holonomic Identities.* – Research Report n° 2799, Institut National de Recherche en Informatique et en Automatique, February 1996.

[4] Comtet (L.). – Calcul pratique des coefficients de Taylor d'une fonction algébrique. *L'Enseignement Mathématique*, vol. 10, 1964, pp. 267–270.

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

[6] Gosper (R. William). – Decision procedure for indefinite hypergeometric summation. *Proceedings of the National Academy of Sciences USA*, vol. 75, n° 1, January 1978, pp. 40–42.

[7] Knuth (Donald E.). – *The Art of Computer Programming.* – Addison-Wesley, 1981, 2nd edition, vol. 2: Seminumerical Algorithms.

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

[9] Ramis (Jean-Pierre). – *Séries divergentes et théories asymptotiques.* – Société Mathématique de France, 1993, *Panoramas et Synthèses*, vol. 121.

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

[11] Tournier (Évelyne) (editor). – *Computer Algebra and Differential Equations.* – Academic Press, 1990. Proceedings of CADE 89.

[12] Zeilberger (Doron). – A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, vol. 32, n° 3, 1990, pp. 321–368.

Creative Telescoping and Applications

Frédéric Chyzak
INRIA Rocquencourt

January 15, 1996

[summary by Bruno Salvy]

Abstract

Creative telescoping is a method to compute definite sums and integrals. Numerous examples are given, together with an introduction to algorithmic techniques based on Gröbner bases of linear operators.

Creative telescoping applies to solutions of systems of linear recurrences and linear differential equations. It yields a linear recurrence or differential equation satisfied by the definite sum or integral of the solutions. It can be used to "compute" generating functions, to extract their coefficients, and to prove identities.

1. Examples

A typical example is the sum $S_n = \sum_{k=0}^{n} \binom{n}{k}$. One starts with a system of equations defining the summand:

$$Au := (n+1-k)u_{n+1,k} - (n+1)u_{n,k} = 0, \qquad Bu := (k+1)u_{n,k+1} - (n-k)u_{n,k} = 0.$$

The aim is to derive a recurrence satisfied by S_n from these equations. This is done by first finding an equation satisfied by $u_{n,k}$ where k does not appear in the coefficients. Such an equation is given by Pascal's triangle rule $u_{n+1,k+1} = u_{n,k+1} + u_{n,k}$ which can be deduced from the above equations as $(S_k+1)A + S_nB$, where S_k (resp. S_n) denotes the shift with respect to k (resp. n). This equation is then rewritten in a form suitable for summation with respect to k:

$$(u_{n+1,k+1} - u_{n+1,k}) - (u_{n,k+1} - u_{n,k}) + u_{n+1,k} - 2u_{n,k} = 0.$$

Since the binomial coefficient $\binom{n}{k}$ is 0 when $k < 0$ or $k > n$, summing over k simply yields the desired result $S_{n+1} - 2S_n = 0$ (this is where telescoping takes place). Using the initial condition $S_0 = 1$, any solver of recurrence equations would then produce $S_n = 2^n$.

A similar example is provided by $U_n = \sum_{k=0}^{n} \binom{n}{k}^2$. The system of equations is a simple modification of the former one. Finding an equation which does not involve k in the coefficients is slightly harder. One finds

$$(n+1)u_{n+2,k+2} - (2n+3)u_{n+1,k+2} + (n+1)u_{n,k+2} - (2n+3)u_{n+1,k+1} - 2(n+1)u_{n,k+1} + u_{n,k} = 0.$$

Again, this is rewritten in a form where telescoping will take place by repeatedly expressing $v_{k+1} = (v_{k+1} - v_k) + v_k$. Summing then yields

$$(n+1)U_{n+1} - 2(2n+1)U_n = 0.$$

Again, with the initial condition $U_0 = 1$, it is easy to conclude that $U_n = \binom{2n}{n}$.

Exactly the same computation applies to definite integrals. For instance, to compute $F(x) = \int_{-\infty}^{+\infty} \exp(-xy^2)\, dy$, one starts from a system satisfied by the integrand

$$D_x + y^2 = 0, \qquad D_y + 2xy = 0,$$

where D_x denotes differentiation with respect to x (and similarly for D_y). Then we look for an equation satisfied by f without y in the coefficients. It is not difficult to find that such an equation is $(D_y^2 + 4x^2 D_x + 2x)f = 0$. Since for any value of x, $\exp(-xy^2)$ and its derivatives with respect to y tend to 0 at $\pm\infty$, integrating this equation over y yields $4x^2F'(x) + 2xF(x) = 0$. The initial condition $F(1) = \sqrt{\pi}$ leads to $F(x) = \sqrt{\pi/x}$.

2. Ore algebras

A very natural framework to describe creative telescoping is provided by a special case of skew polynomial rings called Ore algebras. These are algebras of linear operators which generalize the difference and differential operators.

Definition 1. Let $\mathbb{K}$ be a (possibly skew) field. Let $\partial_1, \ldots, \partial_r$ be defined by the following commutation rules with all the elements P in $A = \mathbb{K}(x_1, \ldots, x_p)[y_1, \ldots, y_q]$:

$$\partial_i P = \sigma_i(P)\partial_i + \delta_i(P),$$

where σ_i is a ring endomorphism of A and δ_i is an additive endomorphism which satisfies the following Leibniz rule:

$$\delta_i(ab) = \sigma_i(a)\delta_i(b) + \delta_i(a)b, \qquad \forall a, b \in A.$$

Then $\mathbb{K}(x_1, \ldots, x_p)[y_1, \ldots, y_q]\langle \partial_1, \ldots, \partial_r \rangle$ is called an *Ore algebra.*

Examples of Ore operators are given in Table 1. These can be combined in an algebra where each operator acts on a different variable. For instance, the Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$ can be described in $\mathbb{Q}(\alpha, \beta, x, n)\langle S_n, D_x \rangle$ by a linear differential equation and a linear recurrence.

More complicated examples arise when one of the ∂_i has a special commutation rule with several of the commutative variables. For instance, in $\mathbb{Q}(n, q, q^n)\langle S_n^{(q)} \rangle$, the q-shift operator satisfies the following commutation rule:

$$S_n^{(q)} n^i (q^n)^j = q^j (n+1)^i (q^n)^j S_n^{(q)}.$$

In this framework, creative telescoping becomes an elimination process. Given a set of operators generating an ideal of operators which vanish on the function we want to sum or integrate, the main

Operator ∂	$\sigma(a)$	$\delta(a)$	Commutation	Action of ∂
Differentiation	$a(x)$	$a'(x)$	$\partial x = x\partial + 1$	$f(x) \mapsto f'(x)$
Shift	$a(x+1)$	0	$\partial x = (x+1)\partial$	$f(x) \mapsto f(x+1)$
Difference	$a(x+1)$	$a(x+1) - a(x)$	$\partial x = (x+1)\partial + 1$	$f(x) \mapsto f(x+1) - f(x)$
q-Dilation	$a(qx)$	0	$\partial x = qx\partial$	$f(x) \mapsto f(qx)$
q-Difference	$a(qx)$	$a(qx) - a(x)$	$\partial x = qx\partial + (q-1)x$	$f(x) \mapsto f(qx) - f(x)$
q-Differentiation	$a(qx)$	$\frac{a(qx)-a(x)}{(q-1)x}$	$\partial x = qx\partial + 1$	$f(x) \mapsto \frac{f(qx)-f(x)}{(q-1)x}$
Eulerian operator	$a(x)$	$xa(x)$	$\partial x = x\partial + x$	$f(x) \mapsto xf'(x)$
e^t-Differentiation	$a(x)$	$xa(x)$	$\partial x = x\partial + x$	$f(t) \mapsto f'(t) \qquad (x = e^t)$
Mahlerian operator	$a(x^p)$	0	$\partial x = x^p\partial$	$f(x) \mapsto f(x^p) \qquad (p \geq 2)$

Table 1. Ore operators

step of creative telescoping asks for an operator in the ideal that does not involve the variable with respect to which we want to integrate or sum. It turns out that under mild conditions on the σ_i's and δ_i's, Ore algebras are Noetherian and an extension of Buchberger's algorithm can be used to compute Gröbner bases. The elimination necessary for creative telescoping can thus be performed automatically provided we have a good description of the ideal.

Given an ideal $\mathcal{I}$ and an operator ∂ of the Ore algebra $\mathcal{O} = \mathbb{K}[x_1, \dots, x_n]\langle\partial_1, \dots, \partial_k\rangle$, let $\mathbf{x}$ be those elements of $\{x_1, \dots, x_n\}$ which commute with ∂. The first step of creative telescoping is therefore to find a basis of the ideal $\mathcal{J} = \mathcal{I} \cap \mathbb{K}[\mathbf{x}]\langle\partial_1, \dots, \partial_k\rangle$ by elimination. The elements of $\mathcal{J}$ can be written

$$\partial A + B, \tag{1}$$

where B does not involve ∂. Since this is an element of $\mathcal{I}$, it cancels whatever function f the ideal $\mathcal{I}$ was cancelling. Now assuming Af to be 0 on the "borders" of the domain, multiplying by ∂^{-1} shows that B is the result we are after (see [2] for a more rigourous description and the application to indefinite operations).

3. More examples

The computation of Gröbner bases of Ore algebras has been implemented by F. Chyzak in his *Mgfun* Maple package available at the URL `http://www-rocq.inria.fr/algo/`. We now illustrate some uses of this package.

3.1. Generating Function of the Jacobi Polynomials. The idea is first to define operators annihilating $P_n^{(\alpha,\beta)}(x)y^n$ and then to compute the sum over n by creative telescoping.

We start with two operators in D_x and S_n annihilating $P_n^{(\alpha,\beta)}(x)$ (omitted here for space reasons):

```
G:=[...,...]:
```

We then load the package and define the Ore algebra in which this computation will take place.

```
with(Mgfun):
A:=orealg(diff=[Dx,x],diff=[Dy,y],shift=[Sn,n],comm=[alpha,beta]):
```

This expresses that there are two variables with a differentiation-like commutation rule, one variable with a shift-like commutation rule and two commutative variables. From the operators annihilating $P_n^{(\alpha,\beta)}(x)$, it is easy to derive operators annihilating $P_n^{(\alpha,\beta)}(x)y^n$:

```
G:=map(primpart,map(numer,[op(subs(Sn=Sn/y,G)),y*Dy-n]),[Sn,Dx,Dy]):
```

Then we are ready for elimination: we create an appropriate term order and then compute a Gröbner basis with respect to it:

```
T:=termorder(A,lexdeg=[[n],[Sn,Dx,Dy]]):
GB:=gbasis(G,T,ratpoly(rational,[x,y,alpha,beta])):
```

We finally select those operators in this basis which do not involve n, and sum over n, which is equivalent to taking the remainder of the division by Δ_n:

```
subs(Sn=1,remove(has,GB,n)):
```

The computation has taken 17 seconds (on a Dec Alpha). After a further fast Gröbner basis computation, the result is reduced to a system of two equations, a large one of order 2 in D_y and another one linear in D_x and D_y. It is then possible to interact with a differential equation solver and, using the initial conditions, obtain the closed-form formula

$$F(x, y) = \frac{1}{R(1 - y + R)^a(1 + y + R)^b}, \qquad R = \sqrt{1 - 2xy + y^2}.$$

3.2. q-Dixon identity. The aim is to show that

$$\sum_k (-1)^k q^{\frac{k(3k+1)}{2}} \binom{a+b}{a+k}_q \binom{b+c}{b+k}_q \binom{a+c}{c+k}_q = \binom{a+b+c}{a,b,c}_q. \tag{2}$$

The algebra is $\mathbb{Q}(q, q^a, q^b, q^c, q^k)\langle S_a, S_b, S_c, S_k\rangle$ which has only q-shift operators:

```
A:=orealg(comm=[q],qshift=[Sa,qa,q],qshift=[Sb,qb,q],
          qshift=[Sc,qc,q],qshift=[Sk,qk,q]):
```

The operators defining the summand are all of order 1 and can be obtained in *Mgfun* by

```
G:=subs([q^a=qa,q^b=qb,q^c=qc,q^k=qk], hypergeomtoholon((-1)^k*q^(k*(3*k+1)/2)
      *qbinomial(a+b,a+k)*qbinomial(a+c,c+k)*qbinomial(b+c,b+k),A)):
```

Then we eliminate q^k and proceed with the telescoping:

```
T:=termorder(A,lexdeg=[[qk],[Sa,Sb,Sc,Sk]]):
GB:=gbasis(G,T,ratpoly(rational,[q,a,b,c,qa,qb,qc])):
CT:=subs(Sk=1,remove(has,GB,[k,qk])):
```

This yields a system of operators symmetrical in a, b, c. Using one more Gröbner basis computation, one obtains an operator involving only S_a. By symmetry similar operators in S_b and S_c can be found. Then checking that the right-hand side of (2) satisfies these equations and that sufficiently many initial condition coincide proves the identity. It is also possible to use Abramov and Petkovšek's q-version of Petkovšek's algorithm to find the right-hand side.

4. Takayama's algorithm

The computation of A and B in (1) is slightly more than what is strictly necessary. Actually we only need to compute B. N. Takayama gave an algorithm for doing so in the Weyl algebra, and this algorithm generalizes to Ore algebras.

The idea is that it is possible to throw away all the right multiples of ∂ during the computation as long as we know they will only be multiplied by polynomials which commute with ∂ during later computations (so that they will remain right multiples of ∂). This is done by working in increasingly large modules where multiplication by the x_i's which do not commute with ∂ is forbidden. The operator ∂ can then easily be eliminated in a preprocessing phase.

This results in an algorithm which is generally faster than the general one, but which is only guaranteed to terminate when there is an element free of the undesirable variables in the ideal.

Bibliography

[1] Abramov (Sergei A.) and Petkovšek (Marko). – Finding all q-hypergeometric solutions of q-difference equations. In Leclerc (B.) and Thibon (J. Y.) (editors), *Formal power series and algebraic combinatorics.* pp. 1–10. – Université de Marne-la-Vallée, 1995. Proceedings SFCA'95.

[2] Chyzak (Frédéric) and Salvy (Bruno). – *Non-commutative Elimination in Ore Algebras Proves Multivariate Holonomic Identities.* – Research Report n° 2799, Institut National de Recherche en Informatique et en Automatique, February 1996.

[3] Ore (Oystein). – Theory of non-commutative polynomials. *Annals of Mathematics*, vol. 34, 1933, pp. 480–508.

[4] Petkovšek (Marko), Wilf (Herbert), and Zeilberger (Doron). – *A=B.* – A. K. Peters, Wellesley, Mass., 1996.

[5] Takayama (Nobuki). – An algorithm of constructing the integral of a module — an infinite dimensional analog of Gröbner basis. In *Symbolic and algebraic computation.* pp. 206–211. – ACM, 1990. Proceedings of ISSAC'90, Kyoto.

∂-finite functions

Frédéric Chyzak
INRIA Rocquencourt

January 15, 1996

[summary by Bruno Salvy]

Abstract

The algebra of ∂-finite functions and sequences enjoys several closure properties useful when computing a description suitable for creative telescoping. A simple description of ∂-finiteness can be given in the context of Ore algebras. In the special case of the Weyl algebra, a special property called holonomy plays a crucial role.

We consider an Ore algebra $\mathcal{A} = \mathbb{K}(x_1, \ldots, x_n)\langle\partial_1, \ldots, \partial_k\rangle$ (see previous summary). A function is ∂-finite with respect to $\mathcal{A}$ when its pseudo-derivatives $\boldsymbol{\partial}^{\boldsymbol{\alpha}} f = \partial_1^{\alpha_1} \cdots \partial_k^{\alpha_k} f$ with $\alpha_i \in \mathbb{N}$ for $i = 1, \ldots, k$ span a finite-dimensional vector space over $\mathbb{K}(x_1, \ldots, x_n)$. Examples of ∂-finite functions in the univariate case are: hypergeometric power series and sequences, solutions of linear recurrences and solutions of linear differential equations. In several variables, it becomes necessary to specify with respect to which operators one considers ∂-finiteness; for instance, all sequences of orthogonal polynomials are ∂-finite with respect to shift of the index and differentiation in the argument.

An equivalent definition is that f is ∂-finite when the module $\mathcal{M} = \mathcal{A} \cdot f$ is finitely generated: $\mathcal{M} = \oplus_{\alpha \in A} \mathbb{K}(\mathbf{x})\boldsymbol{\partial}^{\boldsymbol{\alpha}} f$, for a finite set of indices A. If $\operatorname{Ann} f$ denotes the ideal of the elements of $\mathcal{A}$ vanishing on f, then $\mathcal{A}/\operatorname{Ann} f$ is isomorphic to $\mathcal{A} \cdot f$, and this yields a purely ideal-theoretic definition of ∂-finiteness which avoids the introduction of functions. An ideal $\mathcal{I}$ of $\mathcal{A}$ is thus called ∂-finite when $\mathcal{A}/\mathcal{I}$ is finitely generated as a $\mathbb{K}(\mathbf{x})$-module.

1. Closure properties

What makes ∂-finite functions so useful is that it is possible to compute with these functions without reference to any sort of "closed-form". Many computations can be performed directly on sets of generators of their annihilating ideal. In particular, sum and product of ∂-finite functions can be obtained this way.

1.1. Rectangular systems. Before giving the algorithms for sum and product we note that a ∂-finite function f is always annihilated by a *rectangular* system of polynomials, which is such that each ∂_i of the algebra is involved in exactly one of the polynomials. Consequently, each of the polynomials involves only one ∂_i. That this is so follows from the finite dimension of $\sum_n \mathbb{K}(\mathbf{x})\partial_i^n f$, which implies the existence of a linear relation between a finite number of $\partial_i^n f$. Rectangular systems are useful to prove ∂-finiteness of various constructions, or in the case where Gröbner bases are not available. In other cases, they generally describe an ideal which is smaller than the one we would like to work with, and this leads to slower computations.

Example. In $\mathcal{A} = \mathbb{Q}(x, y)\langle\partial_x, \partial_y\rangle$, the sum of the Bessel functions $J_\mu(x)$ and $J_\nu(y)$ is annihilated by the rectangular system $\mathcal{S} = \{\partial_x(x^2\partial_x^2 + x\partial_x + x^2 - \mu^2), \partial_y(y^2\partial_y^2 + y\partial_y + y^2 - \nu^2)\}$. If $\phi(x, y)$

is a solution of $\mathcal{S}$, and $\mathcal{I}$ is the ideal generated by $\mathcal{S}$ in $\mathcal{A}$, then it is easily checked that $\mathcal{A}/\mathcal{I}$ is generated by $\{\phi, \partial_x\phi, \partial_x^2\phi, \partial_y\phi, \partial_x\partial_y\phi, \partial_x^2\partial_y\phi, \partial_y^2\phi, \partial_x\partial_y^2\phi, \partial_x^2\partial_y^2\phi\}$ and thus is of dimension 9. However, the annihilating ideal of $f = J_\mu(x) + J_\nu(y)$ also contains $\partial_x\partial_y$. The ideal generated by the adjunction of this polynomial to the rectangular system above is $\operatorname{Ann} f$ and $\mathcal{A}/\operatorname{Ann} f$ is generated by $\{f, \partial_x f, \partial_x^2 f, \partial_y f, \partial_y^2 f\}$ and is only of dimension 5.

1.2. Sum. If f and g are two ∂-finite functions, then by linearity $\boldsymbol{\partial}^{\boldsymbol{\alpha}}(f+g) \in \mathcal{A}f + \mathcal{A}g$ which is finite-dimensional. Hence a sum of ∂-finite functions is ∂-finite.

Given a rectangular system for f and a rectangular system for g a rectangular system for $f+g$ is obtained by reducing $h_n = \partial_i^n f + \partial_i^n g$ for increasing values of n. These reductions use the initial rectangular systems and right Euclidean division, which works in any Ore algebra. All the h_n's are thus rewritten in a finite basis $\{f, \partial f, \dots, \partial^J f, g, \partial g, \dots, \partial^K g\}$. The value of n is increased until a linear relation between the h_n's is found by Gaussian elimination.

1.3. Product. We assume that for each ∂_i in the algebra, the morphisms σ_i and δ_i defined by the commutation rule

$$\partial_i p = \sigma_i(p)\partial_i + \delta_i(p)$$

are polynomials in ∂_i over $\mathbb{K}(x_1, \dots, x_n)$. This is not a severe restriction. Then by the same kind of argument as above, the product of two ∂-finite functions is ∂-finite. The algorithm to produce a rectangular system for the product out of two rectangular systems for the functions being multiplied is exactly the same as above.

1.4. Generalizations. Actually, the same algorithm extends to the direct computation of a rectangular system for any polynomial h in some $\boldsymbol{\partial}^{\alpha_{i,j}} f_i$'s given the rectangular systems defining the f_i's.

The FGLM algorithm [3] provides another generalization: given rectangular systems defining the f_i's and a term order T on the $\boldsymbol{\partial}^{\boldsymbol{\alpha}}$'s, this algorithm returns a Gröbner basis for T. Roughly speaking, this algorithm considers all the monomials $\boldsymbol{\partial}^{\boldsymbol{\alpha}} h$ in the order T and stops when it has found sufficiently many relations. More precisely we start with $F = \{h\}$, the resulting basis is set to $L = \{\}$ and the basis of $\mathcal{A}.h$ is set to $R = \{\}$. At each step the smallest element t of F with respect to T is selected and reduced by the rectangular systems defining the $f_i's$. Gaussian elimination is then performed to detect a linear dependency between t and the elements of R. If no linear dependency is found, t is added to R, removed from F, and all the $\partial_i t$ are added to F. Otherwise, the dependency is added to L. The algorithm stops when F is empty, and returns L.

Note that the Gröbner basis returned by this method is not necessarily a basis of $\operatorname{Ann} f$ since, as we have already seen, the rectangular systems do not necessarily generate a sufficiently large ideal.

Yet another extension consists in using any Gröbner basis for the f_i's instead of a rectangular system. In the reduction step, the Euclidean division is replaced by a reduction using the Gröbner bases.

Once again, when it is available, the advantage of this approach over manipulating only rectangular systems is that it results in modules of a smaller dimension, and therefore lessens the complexity of further computations.

1.5. Example. The following identity between Apéry numbers and Franel numbers was proved by V. Strehl:

$$\sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 = \sum_{k=0}^{n} \binom{n}{k} \binom{n+k}{k} \sum_{j=0}^{k} \binom{k}{j}^3. \tag{1}$$

A system is easily found for $\binom{k}{j}^3$ which is hypergeometric:

$$(k+1-j)^3 S_k - (k+1)^3, \qquad (k-j)^3 S_j - (j+1)^3.$$

Then using creative telescoping (see previous summary), one gets an equation for the sum over j:

$$(k+2)^2 S_k^2 - (7k^2+21k+16) S_k - 8(k+1)^2.$$

Again, a system is easily found for $\binom{n}{k}\binom{n+k}{k}$ which is hypergeometric:

$$(n+1-k)S_n - (n+1+k), \qquad (k+1)^2 S_k - (n(n+1)-k(k+1)).$$

The product of this with the previous equation yields a system for the summand of the right-hand side of (1) whose first equation is the first one above (obviously!) and whose second equation is:

$$(k+2)^4 S_k^2 + (n-k-1)S_k + 8(n+k+2)(n+k+1)(n-k) - (7k^2+21k+6)(n+k+2).$$

Now, creative telescoping yields an equation for the right-hand side of (1):

$$(n+2)^3 S_n^2 - (2n+3)(17n^2+51n+39)S_n + (n+1)^3. \tag{2}$$

The same process is then applied to the left-hand side. First, $\binom{n}{k}^2\binom{n+k}{k}^2$ is hypergeometric and satisfies

$$(n+1-k)^2 S_n - (n+1+k)^2, \qquad (k+1)^4 S_k - (n(n+1)-k(k+1))^2.$$

Creative telescoping then yields (2) again. The identity is then proved by checking that two initial conditions coincide, which they do. The whole computation takes less that 10 seconds on a Dec Alpha.

2. Holonomy

The algorithms for creative telescoping which we have described in the previous summary depend on the existence of a polynomial free of one of several variables in the ideal we are working in. It is thus very important to be working in the proper ideal and to be able to check whether such a polynomial exists or not. In the Weyl algebra case, holonomy theory provides such a guarantee. We describe elements of this theory, and give some hints on what remains valid in the more general Ore algebra case.

2.1. Hilbert dimension. Let $\mathcal{A}$ be an Ore algebra: $\mathcal{A} = \mathbb{K}[\mathbf{x}]\langle\boldsymbol{\partial}\rangle$. Let deg denote the total degree with respect to $\mathbf{x}$ and $\boldsymbol{\partial}$. We consider the graduation F_n of $\mathcal{A}$ where F_n contains the elements of $\mathcal{A}$ of degree at most n. Finally, let $h_n = \dim_{\mathbb{K}}(F_n \cdot f)$.

Example. For $f=1$ in the algebra $K[x_1,\dots,x_p]\langle\partial_1,\dots,\partial_p\rangle$, one has $h_n = \binom{n+p}{p} \sim n^p/p!$.

For $f=\exp(x^2)$ in $\mathbb{K}[x]\langle\partial_x\rangle$, it is easy to compute the first few values and be convinced that $h_n = n+1$.

For $f=(s^3-s^2+sx)^{-1/2}$ in $\mathbb{K}[s,x]\langle\partial_x,\partial_y\rangle$, the first values indicate that $h_n = 3n^2+2$.

For $f=\exp(\sin(x))$ in $\mathbb{K}[x]\langle\partial_x\rangle$, one gets $h_n = n^2/2+3n/2+1$.

Finally, for $f=\binom{n}{k}$ in $\mathbb{K}[n,k]\langle S_n,S_k\rangle$, $h_n = 2n+1$.

A general theorem of Hilbert implies that asymptotically, $h_n \sim cn^d$ with d an integer which is called the Hilbert dimension of the ideal. The relevance of this notion to creative telescoping is of a combinatorial nature: if $\mathcal{B}$ is obtained by forming all the monomials in q of the variables $(\mathbf{x},\boldsymbol{\partial})$, then $F_n\cap\mathcal{B}$ contains $\binom{n+q}{q}$ monomials. As soon as this number grows faster than n^d where d is the Hilbert dimension of the annihilating ideal of some f, then a linear combination of elements of $\mathcal{B}$ has to vanish on f, which means that the ideal contains elements of $\mathcal{B}$.

2.2. Weyl algebra. The Weyl algebra is a special case of a polynomial Ore algebra $\mathcal{A}_p = \mathbb{K}[x_1, \ldots, x_p]\langle\partial_1, \ldots, \partial_p\rangle$ where ∂_i is the differentiation operator with respect to the corresponding x_i, for $i = 1, \ldots, p$. A fundamental theorem of Bernstein states that in this case, the Hilbert dimension of an ideal is always larger than p. Those ideals for which the Hilbert dimension is exactly p are called *holonomic*. By extension, a function whose annihilating ideal in a Weyl algebra is holonomic will be called holonomic too. In the examples above, $\exp(x^2)$ and $(s^3 - s^2 + sx)^{-1/2}$ are holonomic functions, while $\exp(\sin x)$ is not.

Holonomy of functions is preserved under sum and product, algebraic functions are holonomic, algebraic substitution preserves holonomy, the diagonal of a holonomic function is holonomic [4, 5]. In addition, a result due to Kashiwara states that when an ideal $\mathcal{I}$ in the *rational* Ore algebra $\mathbb{K}(x_1, \ldots, x_p)\langle\partial_1, \ldots, \partial_p\rangle$ is ∂-finite, then $\mathcal{I} \cap \mathcal{A}$ is a holonomic ideal. This means that all ∂-finite functions with respect to differentiation are also holonomic. Finally, creative telescoping always works in holonomic ideals.

3. Conclusions

The algorithms we have given work in a very general context of Ore algebras. However, creative telescoping is never guaranteed *a priori* to give an answer in the general case, unless the existence of the result is ensured, for instance by holonomy. An advantage of our approach is that it may well return results in non-holonomic cases.

An important difficulty will be the subject of future work. Even in the Weyl algebra case, the ideals $\mathcal{I}$ we are dealing with have a natural description in *rational* Ore algebras $\mathbb{K}(\mathbf{x})\langle\boldsymbol{\partial}\rangle$. However, for creative telescoping what we need is a basis of $\mathcal{I} \cap \mathbb{K}[\mathbf{x}]\langle\boldsymbol{\partial}\rangle$. At the moment, we do not have any algorithm to produce this basis. However, algorithms exist to deal with the same problem in the commutative case, and they might extend to this framework.

This problem is illustrated by the computation of the diagonal of $1/(1 - x - y)$. This can be obtained via a residue computation as the definite integral of $f = (s^2 - s + x)^{-1}$ which is holonomic. Thus creative telescoping applies and there exists an operator free of s in the ideal. The annihilating ideal $\operatorname{Ann} f$ of f in $\mathbb{K}(s, x)\langle\partial_s, \partial_x\rangle$ is generated by $\mathcal{S} = \{(s^2 - s + x)\partial_s + 2s - 1, (s^2 - s + x)\partial_x + 1\}$. However, the ideal generated by $\mathcal{S}$ in the Weyl algebra $\mathcal{A} = \mathbb{K}[s, x]\langle\partial_s, \partial_x\rangle$ is smaller than $\operatorname{Ann} f \cap \mathcal{A}$ and does not contain any polynomial free of s. To get such a polynomial, it is necessary to augment $\mathcal{S}$, for instance with $(s^2 - s + x)\partial_s\partial_x + 2\partial_s$.

Bibliography

[1] Cartier (Pierre). – Démonstration 'automatique' d'identités et fonctions hypergéométriques. *Astérisque*, vol. 206, 1992, pp. 41–91. – Séminaire Bourbaki.

[2] Chyzak (Frédéric) and Salvy (Bruno). – *Non-commutative Elimination in Ore Algebras Proves Multivariate Holonomic Identities.* – Research Report n° 2799, Institut National de Recherche en Informatique et en Automatique, February 1996.

[3] Faugère (J. C.), Gianni (P.), Lazard (D.), and Mora (T.). – Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, vol. 16, 1993, pp. 329–344.

[4] Lipshitz (L.). – The diagonal of a D-finite power series is D-finite. *Journal of Algebra*, vol. 113, 1988, pp. 373–378.

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

[6] Petkovšek (Marko), Wilf (Herbert), and Zeilberger (Doron). – *A=B.* – A. K. Peters, Wellesley, Mass., 1996.

Computing the Distance of a Point to an Algebraic Hypersurface and Application to Exclusion Methods

Xavier Gourdon
Algorithms Project, INRIA Rocquencourt

February 12, 1996

[summary by Pierre Nicodème]

Abstract

We compute lower bounds for the distance in $\mathbb{C}^n$ from a point u to an algebraic surface $\mathcal{Z}$. Such lower bounds or proximity tests give an approximation of $\mathcal{Z}$. We present tests based on both Taylor's formula and a generalization of the Dandelin-Graeffe process to the multivariate case, and their application to the exclusion method [2].

1. Introduction

Given a point a in $\mathbb{C}^n$, and an algebraic hypersurface

$$\mathcal{Z}(P) = \{(z_1, \ldots, z_n) \in \mathbb{C}^n \,|P(z_1, \ldots, z_n) = 0\},$$

with $P \in \mathbb{C}[z_1, \ldots, z_n]$, we want to evaluate the distance $d(a, \mathcal{Z})$ corresponding to the norm

$$\|z\| = \max_{1 \le k \le n} |z_i|.$$

By shifting the variable z, we can restrict to the case $a = 0$.

2. Univariate Polynomials

Let $P(z) = \sum_{i=0}^{d} a_i z^i \in \mathbb{C}[z]\, , a_d \neq 0$, and $\mathcal{Z}(P) = \{U_1, \ldots, U_d\}$. We want to evaluate $d(0, \mathcal{Z}) = \min_i |U_i|$. In Henrici [4, vol. 1], Theorems 6.4.d and 6.4.i give the following classical bound for $\mathcal{Z}(P)$:

Proposition 1. *If $\rho(P)$ is the nonnegative root of the equation $|a_0| = \sum_{j=1}^{d} |a_j| \rho^j$, then*

$$\rho(P) \le d(0, \mathcal{Z}) \le \frac{1}{2^{1/d} - 1} \rho(P) \approx \frac{d}{\log 2} \rho(P).$$

Graeffe Iteration. With $P(z) = a_d \prod_{i=1}^{d} (z - U_i)$, we consider

$$P(z)P(-z) = (-1)^d a_d^2 \prod_{i=1}^{d} (z^2 - U_i^2) = P^{\langle 1 \rangle}(z^2).$$

We note $P^{\langle 1 \rangle}$ the classical Graeffe iterate; the roots of $P^{\langle 1 \rangle}$ are the squares of those of P, and $d(0, \mathcal{Z}(P^{\langle 1 \rangle})) = d(0, \mathcal{Z}(P))^2$; we have

$$\rho(P^{\langle 1 \rangle}) \le d(0, \mathcal{Z}(P^{\langle 1 \rangle})) \le \frac{\rho(P^{\langle 1 \rangle})}{2^{1/d} - 1};$$

so with $\rho_1 = \sqrt{\rho(P^{\langle 1\rangle})}$, we get

$$\rho_1 \le d(0, \mathcal{Z}(P)) \le \frac{\rho_1}{(2^{1/d}-1)^{1/2}}.$$

Generally, we define $P^{\langle k\rangle} = \mathrm{Graeffe}(P^{\langle k-1\rangle})$; then, we get $d(0, \mathcal{Z}(P^{\langle k\rangle})) = d(0, \mathcal{Z}(P))^{2^k}$; with $\rho_k = \rho(P^{\langle k\rangle})^{1/2^k}$, we have

$$\rho_k \le d(0, \mathcal{Z}(P)) \le \frac{\rho_k}{(2^{1/d}-1)^{1/2^k}}.$$

The upper bound tends rapidly to the lower bound as k increases, thus we have obtained an effective process to compute $d(0, \mathcal{Z})$.

Computing the $P^{\langle k\rangle}$. With $A(z) = \sum_{i\equiv 0 \bmod 2} a_i z^{i/2}$ and $B(z) = \sum_{i\equiv 1 \bmod 2} a_i z^{(i-1)/2}$, we have

$$P(z)P(-z) = A(z^2)^2 - z^2 B(z^2)^2,$$

and therefore,

$$\mathrm{Graeffe}(P) = A(z)^2 - zB(z)^2.$$

A practical problem is that the coefficient size doubles at each Graeffe iteration.

3. Multivariate Polynomials

In the multivariate case, the polynomial $P(z)P(-z)$ can not be written as $Q(z^2)$ where $Q(z)$ is a polynomial, thus we need to modify the definition. We generalize the Graeffe process to the multivariate case as follows:

Definition 1. We call the N-th Graeffe iterate of $P(z) \in \mathbb{C}[z_1, \ldots, z_n]$ the polynomial $P^{[N]}(z)$ defined by

$$P^{[N]}(z) = \prod_{j=0}^{2^N-1} P(\omega^j z), \qquad \omega = \exp\left(\frac{2i\pi}{2^N}\right), \quad i^2 = -1,$$

where $\omega^j z$ denotes the point $(\omega^j z_1, \ldots, \omega^j z_n)$.

Proposition 2. *For all non negative integer N, the N-th Graeffe iterate of $P(z)$ writes as*

$$P^{[N]}(z) = \sum_{j\ge 0} B_j^{[N]}(z),$$

where the $B_j^{[N]}$'s are homogeneous polynomials of degree $2^N j$. The $(N+1)$-st Graeffe iterate can be computed from the N-th thanks to the formula

$$P^{[N+1]}(z) = P_0^{[N]}(z)^2 - P_1^{[N]}(z)^2, \qquad P_k^{[N]}(z) = \sum_{j\equiv k \bmod 2} B_j^{[N]}(z).$$

With the multivariate Graeffe process, we easily generalize the univariate algorithm to compute $d(0, \mathcal{Z})$ in the multivariate case.

Theorem 1. *Let $P(z)$ be a polynomial in $\mathbb{C}[z_1, \ldots, z_n]$ of total degree d. Let $P^{[N]}(z) = \sum_{j\ge 0} B_j^{[N]}(z)$ be its N-th Graeffe iterate and R_N the non-negative solution of the equation in R*

$$|P^{[N]}(0)| = \sum_{j\ge 1} \|B_j^{[N]}\|_\infty R^j, \tag{1}$$

d	r_0/d	r_1/d	r_2/d	r_3/d	r_4/d
2	0.7673	0.9725	0.9996	1.0000	1.0000
5	0.6525	0.9479	0.9973	1.0000	1.0000
7	0.6325	0.9400	0.9960	0.9999	1.0000
15	0.6067	0.9271	0.9938	0.9999	1.0000

TABLE 1. Some values of $r_N/d(0,\mathcal{Z}_{n,d})$ for $n=2$.

d	r_0/d	r_1/d	r_2/d	r_3/d
2	0.5832	0.6338	0.8108	0.8224
3	0.4802	0.5108	0.6478	0.7561

TABLE 2. Some values of $r_N/d(0,\mathcal{Z}_{n,d})$ for $n=7$.

where $\|B_j^{[N]}\|_\infty = \sup_{\|z\|=1}\|B_j(z)\|$. *Then we have*

$$r_N \le d(0,\mathcal{Z}) \le \left(\frac{1}{2^{1/d}-1}\right)^{2^{-N}} r_N, \qquad r_N = R_N^{2^{-N}}. \tag{2}$$

Computing $\|B_j^{[N]}\|_\infty$ raises a difficult practical problem; therefore, we make use of the norm $\|\sum_\alpha a_\alpha z^\alpha\| = \sum |a_\alpha|$, easy to compute. Our main result is stated using this norm; one demonstrates the equivalence of the norms $\|\cdot\|_\infty$ and $\|\cdot\|$ by combination of the Parseval identity and of the Cauchy-Schwarz inequality.

THEOREM 2. *Let ρ_N be the unique nonnegative solution of*

$$|P^{[N]}(0)| = \sum_{j=1}^{d} \|B_j^{[N]}\|\rho^j \tag{3}$$

The distance from 0 to $\mathcal{Z}$ satisfies

$$r_N \le d(0,\mathcal{Z}) \le \kappa_N r_N, \tag{4}$$

where

$$r_N = \rho_N^{2^{-N}} \quad \text{and} \quad \kappa_N = \left(\frac{1}{2^{1/d}-1}\sqrt{\binom{2^N+n-1}{n-1}}\right)^{1/2^N}.$$

Moreover $\lim_{N\to\infty}\kappa_N = 1$, *which implies* $\lim_{N\to\infty} r_N = d(0,\mathcal{Z})$.

4. Examples

We take a polynomial of degree d in n variables: $P_{n,d} = \sum_{j=1}^n (1-z_j)^d - 1$. With $\mathcal{Z}_{n,d} = \mathcal{Z}(P_{n,d})$, we have $d(0,\mathcal{Z}_{n,d}) = 1 - \frac{1}{n^{1/d}}$.

Tables 1 and 2 give the value of the ratio $r_N/d(0,\mathcal{Z}_{n,d})$ of Theorem 3 for several values of n, d and N. The computations were performed in Maple. These examples show that the bound is quite good for a small value N of Graeffe iterates.

5. Exclusion methods

We give the principle of the method for a polynomial of one variable $P(z) \in \mathbb{C}[z]$.

- Let the *exclusion function* be: $z_0 \mapsto \rho(z_0)$, with ρ given by theorem 2 after a proper shift of the variable, and
  (1) $\rho(z_0) = 0 \iff P(z_0) = 0$,
  (2) P has no zero in $|z-z_0| < \rho(z_0)$, which is equivalent to $\rho(z_0) \le d(z_0,\mathcal{Z})$;
- then, the *exclusion test* is: let C be a square of centre z_0 and half-side $a > 0$. If $\rho(z_0) \ge \sqrt{2}a$, C contains no zero of P.

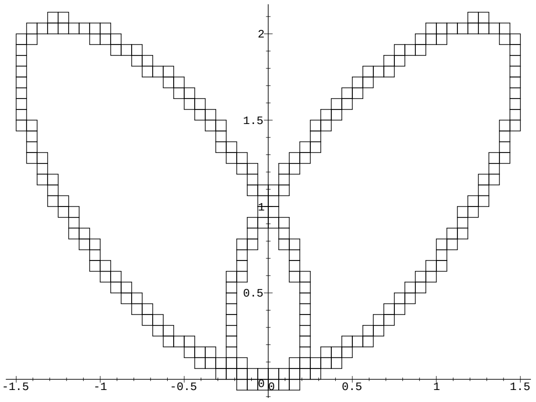


FIGURE 1. Representing by exclusion the curve $y^4-2y^3+y^2-3x^2y+2x^4=0$ (petal).

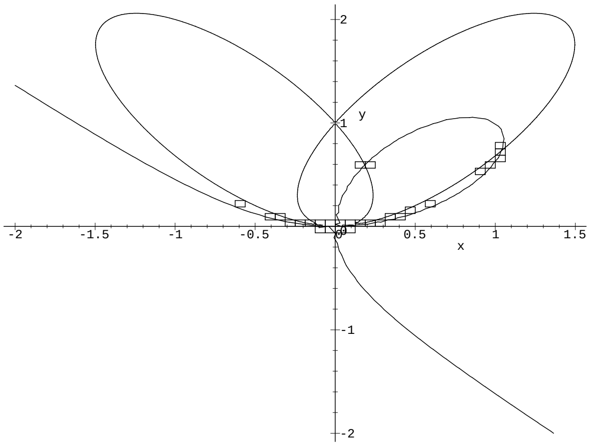


FIGURE 2. Intersection of the curves $x^3+y^3-2xy=0$ (Descartes folium) and $y^4-2y^3+y^2-3x^2y+2x^4=0$ (petal).

Exclusion algorithm.

- Consider the reciprocal polynomial $R(z)$ of $P(z)$; compute by Graeffe a lower bound of the smallest root of $R(z)$, which gives an upper bound b_u of the largest root of $P(z)$;
- Start from a big square centred at the origin, with side $2b_u$, which contains all the roots of $P(z)$;
- Recursively split the square in four squares of equal size, discarding by the exclusion test squares containing no zeros;
- Stop the recursion when the desired precision is reached (the surface of the area covering the zeros decreases exponentially fast to zero).

Figure 1 shows an application of the exclusion method to localize an algebraic curve in $\mathbb{R}^2$.

For an algebraic variety $\mathcal{Z}_i = \mathcal{Z}(P_i)$ and $\mathcal{Z} = \bigcap_i \mathcal{Z}(P_i)$, with $P_1, \ldots, P_m \in \mathbb{C}[z_1, \ldots, z_n]$, let $\rho_i(z_0)$ be an exclusion function defined by theorem 2 for $P_i, (1 \leq i \leq m)$; we can define an exclusion function for the variety as $\rho(z_0) = \sup_{1 \leq i \leq m} \rho_i(z_0)$.

An application of exclusion method to localize the intersection of two curves in $\mathbb{R}^2$ is given in Figure 2.

Bibliography

[1] Bareiss (Erwin H.). – Resultant procedure and the mechanization of the Graeffe process. *Journal of the ACM*, vol. 7, 1960, pp. 346–386.

[2] Dedieu (Jean-Pierre), Gourdon (Xavier), and Yakoubsohn (Jean-Claude). – Computing the distance from a point to an algebraic hypersurface. – July 1996. Seminar of the American Mathematical Society. Park City. 8 pages. In press.

[3] Dedieu (Jean-Pierre) and Yakoubsohn (Jean-Claude). – Localization of an algebraic hypersurface by the exclusion algorithm. *Applicable Algebra in Engineering, Communication and Computing*, vol. 2, 1992, pp. 239–256.

[4] Henrici (Peter). – *Applied and Computational Complex Analysis.* – John Wiley, New York, 1977. 3 volumes.

[5] Pan (V.). – Solving a polynomial equation: some story and recent progress. – 1995. Preprint.

Matrix-based methods for solving polynomial systems

Ioannis Emiris
Projet SAFIR, Inria Sophia-Antipolis

11 mars, 1996

[summary by Frédéric Chyzak]

Abstract

We present a uniform approach to the elimination of variables between polynomials and the construction of matrices that express resultants. Building a matrix whose determinant is a multiple of the resultant reduces the solving of a polynomial system to a generalized eigenvalues/eigenvectors problem for a square matrix. Several such matrices are of interest, in particular the Newton and Bézout/Dixon matrices, which lead to efficient calculations.

1. Classical resultants versus sparse resultants

Classically, the resultant is a single polynomial which characterizes the solvability of a system of dense polynomials [7]. We introduce another concept of resultant which takes the structure of the coefficients into account.

Let $f_1(\mathbf{c}, x), \dots, f_{n+1}(\mathbf{c}, x)$ be $n+1$ polynomials in the n indeterminates $x_1, \dots, x_n$ and with coefficients that are polynomial in $c_1, \dots, c_N$ over a field $\mathbb{K}$. A *sparse resultant* $R(\mathbf{c})$ with respect to a subfield $\mathbb{L}$ of the algebraic closure $\overline{\mathbb{K}}$ is an irreducible polynomial of $\mathbb{K}[c_1, \dots, c_N]$ that vanishes at a specialization γ of the c_i's if and only if the corresponding specializations of the f_i's have a common zero. In other words, the resultant satisfies

$$\forall \gamma \in \mathbb{L}^N \quad (R(\gamma) = 0 \Longleftrightarrow \exists \xi \in \mathbb{L}^n \quad \forall i = 1, \dots, n \quad f_i(\gamma, \xi) = 0).$$

For some applications, one requires that the coefficients of the f_i's be generic, i.e., that one c_i be introduced for each coefficient. Special cases are of particular interest. In the case of dense homogenized polynomials

$$f_i(x_0, x_1, \dots, x_n) = \sum_{a_0 + \cdots + a_n = d_i} c_{a_0, \dots, a_n} x_0^{a_0} \dots x_n^{a_n},$$

we recover the classical *homogeneous resultant* [7]. In the case of two (dense) univariate polynomials, we recover Sylvester's classical notion of the *univariate resultant* [6], whose expression as a determinant is recalled in the next section. In the case of (possibly sparse) polynomials with generic coefficients, i.e., when

$$f_i(x_1, \dots, x_n) = \sum_{j=1}^{r_i} c_{i,j} x_1^{a_{i,j,1}} \dots x_n^{a_{i,j,n}}$$

for non-zero undetermined coefficients $c_{i,j}$ that are transcendental over the field $\mathbb{K}$, the resultant $R(\mathbf{c})$ is called the *sparse resultant* of the f_i's.

A major difference between the classical and the sparse resultants is that the former express simultaneous solvability in a *projective* space $\mathbb{P}^n\left(\overline{\mathbb{K}}\right)$ whereas the latter express simultaneous solvability in the *torus* $\left(\overline{\mathbb{K}}^*\right)^n$ which is a proper subset of $\mathbb{P}^n\left(\overline{\mathbb{K}}\right)$.

2. Expression of the resultant as a determinant

Two important examples of classical resultants are given as the determinant of a matrix. First, in the case of dense linear polynomials $f_i = c_{i,0} + c_{i,1}x_1 + \cdots + c_{i,n}x_n$, the corresponding homogeneous resultant [7] is

$$R(\mathbf{c}) = \det \begin{bmatrix} c_{1,0} & \cdots & c_{1,n} \\ \vdots & & \vdots \\ c_{n+1,0} & \cdots & c_{n+1,n} \end{bmatrix}.$$

Second, in the case of dense univariate polynomials $f(a,x) = a_n x^n + \cdots + a_0$ and $g(b,x) = b_m x^m + \cdots + b_0$, the univariate resultant [6] is the following determinant

$$R(a,b) = \det \begin{bmatrix} a_n & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 & a_0 & 0 & \cdots & 0 \\ 0 & a_n & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 & a_0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_n & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 & a_0 \\ b_m & b_{m-1} & b_{m-2} & \cdots & b_2 & b_1 & b_0 & 0 & \cdots & 0 \\ 0 & b_m & b_{m-1} & b_{m-2} & \cdots & b_2 & b_1 & b_0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_m & b_{m-1} & b_{m-2} & \cdots & b_2 & b_1 & b_0 \end{bmatrix},$$

where the matrix has constant values on diagonals and each row corresponds to the product of either polynomial times a power of x, written in the basis $(x^{\max(n,m)}, \ldots, x, 1)$. Sparse resultants can be expressed as the determinant of a matrix. More precisely, we proceed to give an expression of a multiple of the resultant in the case of sparse polynomials with generic undetermined coefficients.

To give this expression, define the *support* of a polynomial $f = \sum_{a_1,\ldots,a_n} c_{a_1,\ldots,a_n} x_1^{a_1} \ldots x_n^{a_n}$ as the set $\mathrm{Supp}(f) \subset \mathbb{N}^n$ of those $(a_1, \ldots, a_n)$ such that $c_{a_1,\ldots,a_n} \neq 0$. Note that

$$\mathrm{Supp}(fg) \subset \mathrm{Supp}(f) + \mathrm{Supp}(g) \qquad \text{and} \qquad \mathrm{Supp}(f+g) \subset \mathrm{Supp}(f) \cup \mathrm{Supp}(g).$$

With this definition, we now construct matrices that represent the specialization application of polynomials $f_i(\mathbf{c}, x)$ on a point $\xi \in \mathbb{K}^n$. For $i = 1, \ldots, n$, let S_i be a subset of $\mathbb{N}^n$. Next define S_0 to be $\bigcup_{i=1}^n (S_i + \mathrm{Supp}(f_i))$. For $i = 0, \ldots, n$, call P_i the set of polynomials $f \in \mathbb{K}[\mathbf{c}, x]$ such that $\mathrm{Supp}(f) \subset S_i$. Then, the application $\mathcal{M}$ from $P_1 \times \cdots \times P_n$ to P_0 given by $\mathcal{M}(l_1, \ldots, l_n) = \sum_{i=1}^n l_i f_i$ is a well-defined linear application. For $i = 0, \ldots, n$, write $S_i = \{s_{i,1}, \ldots, s_{i,N_i}\} \subset \mathbb{N}^n$. Then $\mathcal{M}$ has a matrix representation, $M = [m_{(i,i'),j}(\mathbf{c})]$, where, for convenience, we number the rows of M by (i, i') and the columns by j. This matrix is given by

$$x^{s_{i,i'}} f_i(\mathbf{c}, x) = \sum_{j=1}^{N_0} m_{(i,i'),j}(\mathbf{c}) x^{s_{0,j}}, \qquad \text{for } i = 1, \ldots, n \text{ and } i' = 1, \ldots, N_i.$$

Under this representation, the evaluation of $\mathcal{M}$ at the tuple $\left(\sum_{j=1}^{N_1} l_{1,j}(\mathbf{c}) x^{s_{1,j}}, \ldots, \sum_{j=1}^{N_n} l_{n,j}(\mathbf{c}) x^{s_{n,j}}\right)$ of $P_1 \times \cdots \times P_n$ is given by the product:

$$\begin{bmatrix} l_{(1,1)}(\mathbf{c}) & \cdots & l_{(n,N_n)}(\mathbf{c}) \end{bmatrix} \begin{bmatrix} m_{(1,1),1}(\mathbf{c}) & \cdots & m_{(1,1),N_0}(\mathbf{c}) \\ \vdots & \ddots & \vdots \\ m_{(n,N_n),1}(\mathbf{c}) & \cdots & m_{(n,N_n),N_0}(\mathbf{c}) \end{bmatrix} \begin{bmatrix} x^{s_{1,1}} \\ \vdots \\ x^{s_{n,N_n}} \end{bmatrix}.$$

On the other hand, the product of M by a column vector yields the simultaneous specialization of multiples of the f_i's at a point $\xi \in \mathbb{K}^n$:

$$\begin{bmatrix} m_{(1,1),1}(\mathbf{c}) & \dots & m_{(1,1),N_0}(\mathbf{c}) \\ \vdots & \ddots & \vdots \\ m_{(n,N_n),1}(\mathbf{c}) & \dots & m_{(n,N_n),N_0}(\mathbf{c}) \end{bmatrix} \begin{bmatrix} \xi^{s_{0,1}} \\ \vdots \\ \xi^{s_{0,N_0}} \end{bmatrix} = \begin{bmatrix} \xi^{s_{1,1}} f_1(\mathbf{c},\xi) \\ \vdots \\ \xi^{s_{n,N_n}} f_n(\mathbf{c},\xi) \end{bmatrix}.$$

From this second fact, it follows that if $\xi \in \left(\overline{\mathbb{K}}^*\right)^n$ is a common zero of the specializations of the $f_i(\mathbf{c},x)$ at $\mathbf{c} = \gamma$, there exists $v_\gamma = [\xi^{s_{1,1}}, \dots, \xi^{s_{n,N_n}}]^T \neq 0$ such that $M(\gamma)v_\gamma = 0$. Moreover, when M is a square matrix, we have that $\det M(\gamma)$ is zero. More is true: in the case when such a v_γ exists, $R(\mathbf{c})$ divides $\det M(\mathbf{c})$, and the matrix M is called a *matrix of the resultant*. One thus computes a multiple of the resultant as the determinant of the matrix M above. It only remains to determine suitable sets S_i, for which possible constructions are alluded to in Section 4.

3. Numerically solving polynomial systems

In this section, we assume that $f_1, \dots, f_n \in \mathbb{K}[x_1, \dots, x_n]$ is a well-determined system of polynomials with *determined* coefficients, whose variety is zero-dimensional, i.e., the roots are isolated. We assume further that the ideal $(f_1, \dots, f_n)$ is radical, i.e., that the roots are simple. Then, when the matrix M above is a matrix of the resultant, it can be used to numerically solve the system.

To do so, we look at an over-determined system in place of the well-determined system, so as to introduce genericness in the coefficients. Two such over-determined systems are available:

(1) either we add $f_{n+1} = r_1x_1 + \dots + r_nx_n + u$ for r_i in $\mathbb{K}$, and view the f_i's as elements of $\mathbb{K}[u][x_1, \dots, x_n]$, and we look for their sparse resultant in $\mathbb{K}[u]$;

(2) or we conceal one variable, say x_n, and view the f_i's as elements of $\mathbb{K}[x_n][x_1, \dots, x_{n-1}]$, and we look for their sparse resultant in $\mathbb{K}[x_n]$.

If the second system is chosen, we change n into $n-1$, then x_{n+1} into u, so that in both cases, we look for the sparse resultant $R(u) \in \mathbb{K}[u]$ of polynomials $f_i(u,x) \in \mathbb{K}[u][x_1, \dots, x_n]$. In either case, let us assume that the matrix $M(u)$ is a matrix of the resultant.

Again, let $\mathbb{L}$ be an algebraic field extension of $\mathbb{K}$ in $\overline{\mathbb{K}}$ and $(\xi, \eta) \in \mathbb{L}^n \times \mathbb{L}$ be a solution in (x,u) of the over-determined system. Then $\det M(\eta) = 0$ and $M(\eta)v_\xi = 0$. If case (1) above was chosen, we only need to determine ξ. If case (2) above was chosen, we need to determine both ξ and η. In both cases, we look for (ξ, η), or equivalently for (v_ξ, η). This reduces the initial problem of solving a polynomial system to a generalized eigenvalues/eigenvectors problem, for which optimized numerical algorithms are available. More specifically, this problem takes several possible forms, amongst which both following extreme cases:

- if the matrix $M(u)$ is linear in u, $M(u) = M_1u + M_0$, with M_1 invertible, the problem is a (simple) eigenvalues/eigenvectors problem:

$$M(\eta)v_\xi = 0 \iff \left(-M_1^{-1}M_0 - \eta \mathrm{Id}\right) v_\xi = 0;$$

- if the matrix $M(u)$ is non-linear in u, $M(u) = M_du^d + \dots + M_0$, with M_d non-invertible, the problem is a generalized eigenvalues/eigenvectors problem:

$$M(\eta)v_\xi = 0 \iff \left(\begin{bmatrix} 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & 1 \\ -M_0 & -M_1 & \dots & M_{d-1} \end{bmatrix} - \eta \begin{bmatrix} 1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \\ 0 & \dots & 0 & M_d \end{bmatrix} \right) \begin{bmatrix} v_\xi \\ \eta v_\xi \\ \vdots \\ \eta^{d-1} v_\xi \end{bmatrix} = 0.$$

To reduce the size of the matrices and achieve more efficiency, we perform operations on rows and permutations on columns of M beforehand, rewriting M and v_ξ in the form

$$\tilde{M}(u) = \begin{bmatrix} \tilde{M}_{1,1} & \tilde{M}_{1,2}(u) \\ \tilde{M}_{2,1}(u) & \tilde{M}_{2,2}(u) \end{bmatrix} \qquad \text{and} \qquad \tilde{v}_\xi = \begin{bmatrix} w_\xi \\ w'_\xi \end{bmatrix}, \qquad \text{respectively.}$$

It follows that

$$M(\eta)v_\xi = 0 \Longleftrightarrow \tilde{M}(\eta)\tilde{v}_\xi = 0 \Longleftrightarrow \begin{bmatrix} \tilde{M}_{1,1} & \tilde{M}_{1,2}(u) \\ 0 & \tilde{M}_{2,2}(x) - \tilde{M}_{2,1}(u)\tilde{M}_{1,1}^{-1}\tilde{M}_{1,2}(u) \end{bmatrix} \begin{bmatrix} w_\xi \\ w'_\xi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

whence $M'(x) = \tilde{M}_{2,2}(x) - \tilde{M}_{2,1}(u)\tilde{M}_{1,1}^{-1}\tilde{M}_{1,2}(u)$ satisfies $M'(x)w'_\xi = 0$. Solving this smaller problem yields possible roots of the initial problem.

4. Mixed volume and various matrices of resultants

The *mixed volume* of convex polyhedra $Q_1, \dots, Q_n \subset \mathbb{R}^n$ is classically defined by the single mapping VM to $\mathbb{R}$ which is multilinear with respect to the addition of polyhedra and such that $\mathrm{VM}(Q, \dots, Q) = n!\,\mathrm{Vol}(Q)$, where Vol is the Euclidean volume. We next define the *Newton polytope* of a polynomial f as the convex hull of its support. A famous theorem by Bernstein [1] states the number of isolated roots of a polynomial system counted with multiplicity is bounded by the mixed volume of the Newton polytopes of the polynomials, a bound which is much better in case of sparse polynomials than the older Bézout's bound for dense polynomials. An efficient algorithm is given in [2, 5], where the construction of the Newton matrix of a resultant is derived.

Another matrix of a resultant is of interest, the Bézout-Dixon matrix [3], which is defined by introducing new indeterminates a_i as

$$\begin{bmatrix} f_1(x) & \cdots & \frac{f_1(a_1,\dots,a_i,x_{i+1},\dots,x_n) - f_1(a_1,\dots,a_{i-1},x_i,\dots,x_n)}{a_i - x_i} & \cdots & \frac{f_1(a) - f_1(a_1,\dots,a_{n-1},x_n)}{a_n - x_n} \\ \vdots & & \vdots & & \vdots \\ f_{n+1}(x) & \cdots & \frac{f_{n+1}(a_1,\dots,a_i,x_{i+1},\dots,x_n) - f_{n+1}(a_1,\dots,a_{i-1},x_i,\dots,x_n)}{a_i - x_i} & \cdots & \frac{f_{n+1}(a) - f_{n+1}(a_1,\dots,a_{n-1},x_n)}{a_n - x_n} \end{bmatrix}.$$

Bibliography

[1] Bernstein (D. N.). – The number of roots of a system of equations. *Functional Analysis and Applications*, vol. 9, n° 2, 1975, pp. 183–185.

[2] Canny (J.) and Emiris (I.). – An efficient algorithm for the sparse mixed resultant. In Cohen (G.), Mora (T.), and Moreno (O.) (editors), *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Lecture Notes in Computer Science*, pp. 89–104. – Springer Verlag, 1993. Proceedings AAECC'93, May, Puerto Rico.

[3] Dixon (A. L.). – The eliminant of three quantics in two independent variables. *Proceedings of the London Mathematical Society*, vol. 6, 1908, pp. 49–69 and 209–236.

[4] Emiris (I. Z.). – *Sparse Elimination and Applications in Kinematics.* – PhD thesis, Computer Science Division, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, December 1994.

[5] Emiris (I. Z.) and Canny (J.F.). – Efficient incremental algorithms for the sparse resultant and the mixed volume. *Journal of Symbolic Computation*, vol. 20, n° 2, August 1995, pp. 117–149.

[6] Sylvester (J. J.). – On a theory of syzygetic relations of two rational integral functions, comprising an application to the theory of Sturm's functions, and that of the greatest algebraic common measure. *Philosophical Transactions*, vol. 143, 1853, pp. 407–548.

[7] van der Waerden (B. L.). – *Modern Algebra.* – Frederick Ungar Publishing Co., New-York, 1950, third edition.

Computation of large values of $\pi(x)$

Marc Deléglise
Université Lyon-1

February 12, 1996

[summary by Philippe Dumas and François Morain]

Every textbook about number theory explains the sieve of Eratosthenes [3], which is one of the oldest known algorithms. This algorithm enables us to compute the prime numbers less than a fixed number x. It consists in successively striking out the multiples of the already known prime numbers, the first one being 2. The cost of the algorithm is $O(x^{1+\varepsilon})$ for all $\varepsilon > 0$. Pritchard has given a lot of theoretical algorithms that perform in sublinear time (see [8] for new results and a bibliography on this topic). From a practical point of view, many tricks can be used to find all primes less than 10^{12} in a fast way, as explained for example in [1].

Clearly the enumeration of all the primes less than x cannot have a lower cost than $\pi(x)$. Besides the computation of $\pi(x)$, the number of primes less or equal to x, does not need to find all the primes less than x. This fact is set up by the formula of Legendre, which uses the prime numbers less or equal to $\sqrt{x}$. Next, the works of Meissel and Lehmer provides more subtle formulæ, which reduce the amount of computation. As an example Meissel computed the value of $\pi(10^8)$. Nevertheless, these methods all have a cost of $O(x^{1+\varepsilon})$. Lagarias, Miller, and Odlyzko gave a method which for the first time had a complexity $O(x^{\alpha})$ with $\alpha < 1$. More precisely the time complexity is $O(x^{2/3+\varepsilon})$ and the space complexity is $O(x^{1/3+\varepsilon})$. This permits them to compute the value of $\pi(10^{16})$. Deléglise and Rivat [2] lessen the time complexity by a logarithmic factor using a slight modification of the previous method, hence they obtained the value of $\pi(10^{18})$.

All these methods use the idea of sieve, but Lagarias and Odlyzko [5] proposed an entirely different way to compute $\pi(x)$. The method is based on an analytic formula, and its expected cost is $O(x^{1/2+\varepsilon})$. It has never been implemented.

1. Sieve function

Let us assume that we use the sieve of Eratosthenes. We write all the integers between 1 and x, and we strike out successively the multiples of $p_1 = 2$, $p_2 = 3$, and so on. We stop when we have used the a-th prime number p_a. The number of integers which remain is $\phi(x, a)$. The function $\phi(x, a)$ is the partial sieve function. As a convention, we set $\phi(x, 0) = \lfloor x \rfloor$. A mere combinatorial argument gives the following recursion rule,

$$\phi(x, a) = \phi(x, a-1) - \phi(x/p_a, a-1).$$

A raw application of this rule gives the formula

$$\phi(x, a) = \sum_{\substack{m \le x \\ P(m) \le p_a}} \mu(m) \lfloor x/m \rfloor,$$

where $\mu(m)$ is the Möbius function and $P(m)$ is the largest prime factor of m.

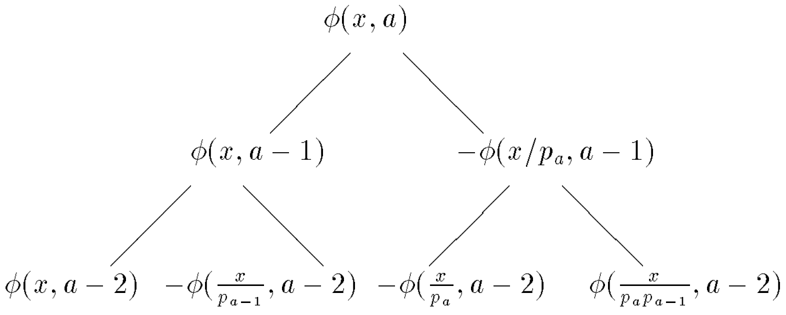

$\phi(x,a)$
$\phi(x,a-1)$
$-\phi(x/p_a,a-1)$
$\phi(x,a-2)$
$-\phi(\frac{x}{p_{a-1}},a-2)$
$-\phi(\frac{x}{p_a},a-2)$
$\phi(\frac{x}{p_a p_{a-1}},a-2)$

FIGURE 1. A computation tree for $\phi(x,a)$. The sum of the leaves is $\phi(x,a)$.

In the sequel, an important point will be a clever refinement in the use of the recursion rule. Indeed the last formula contains too many terms. The recursion rule may be viewed as an expansion rule, which provides a computation tree for $\phi(x,a)$ (see Fig. 1). The problem is to give a stopping criterion in order to avoid an excessive growth of the number of leaves.

The partial sieve function $\phi(x,a)$ is used in the following manner. Let us denote by $P_k(x,a)$ the number of integers less or equal to x with exactly k equal or distinct prime factors, those prime factors being all greater than p_a. With the equality $P_0(x,a)=1$, we have immediately

$$\phi(x,a)=P_0(x,a)+P_1(x,a)+P_2(x,a)+P_3(x,a)+\cdots .$$

But it is manifest that

$$P_1(x,a)=\pi(x)-a,$$

hence the following basic formula

$$\pi(x)=\phi(x,a)-1+a+P_2(x,a)+P_3(x,a)+\cdots . \tag{1}$$

With $a=\pi(\sqrt{x})$, the quantities $P_k(x,a)$ are zero for $k>2$ because any composite number with three prime factors larger than $\sqrt{x}$ is larger than x. Hence, we obtain Legendre's formula [9]

$$\pi(x)=\phi(x,a)+a-1, \qquad a=\pi(\sqrt{x}).$$

An expanded form of this formula is

$$\pi(x)=\pi(\sqrt{x})-1+\sum_H(-1)^{\#H}\lfloor x/p_H\rfloor,$$

where H runs through the subsets of $\{1,2,\ldots,\pi(\sqrt{x})\}$ and $p_H=\prod_{h\in H}p_h$. The computation of $\pi(x)$ based on this formula has cost $O(x)$.

2. Meissel and Lehmer

Meissel chose the value $a=\pi(x^{1/3})$ in the basic formula (1), hence the formula reduces to

$$\pi(x)=\phi(x,a)+a-1+P_2(x,a), \qquad a=\pi(x^{1/3}). \tag{2}$$

The most time consuming part of the formula is the term $\phi(x,a)$ and Lehmer proposed the following truncation rule for the computation tree of Figure 1:

Do not split a node labelled $\pm\phi(x/n,b)$ if either of the following holds:
(i) $x/n<p_b$,
(ii) $b=5$.

Lehmer used $a = \pi(x^{1/4})$ and the tree has leaves labelled by $\pm\phi(x/n, b)$ for n a product of four prime numbers between $p_6 = 13$ and p_a; this leads to a number of leaves essentially of order x. For a detailed description of the implementation, see the original article of Lehmer [6] or the problem [7, Problème 5].

3. Lagarias, Miller, and Odlyzko

In [4], Lagarias, Miller, and Odlyzko use a sharper truncation rule, namely

> Do not split a node labelled $\pm\phi(x/n, b)$ if either of the following holds:
> (i) $b = 0$ and $n \leq x^{1/3}$,
> (ii) $n > x^{1/3}$.

They use $a = \pi(x^{1/3})$ and for this value the number of leaves of the computation tree is no more than $O(x^{2/3})$. The leaves associated with the case (i) are the *ordinary leaves*, and the leaves associated with the case (ii) are the *special leaves.*

According to (2) there are two terms to compute: $\phi(x, a)$ and $P_2(x, a)$. The computation has four steps; first a preparatory step; next the computation of $P_2(x, a)$; then the computation of the contribution of the ordinary leaves; finally the computation of the special leaves. The sum which correspond to $\phi(x, a)$ is the sum of these last two quantities.

Preparatory step. Using an ordinary Eratosthenes sieve, one finds all the primes p_1, $p_2, \ldots, p_a$ below $x^{1/3}$. During the sieving, several quantities are also computed and stored for a later use. When sieving with p_i, the values of the Möbius function $\mu(n)$ for $n \leq x^{1/3}$ can be updated. The values of the function f which gives the least prime factor of an integer n in the interval is computed too. Having sieved with the i-th prime, the value of $\phi(x^{1/3}, i)$ is known and stored.

Finally, the value $\pi(x^{1/4})$ is computed. All this has a cost $O(x^{1/3+\varepsilon})$ arithmetic operations and space cost $O(x^{1/3})$.

Computation of $P_2(x, a)$. The quantity $P_2(x, a)$ is computed according to the formula

$$P_2(x, a) = \binom{a}{2} - \binom{a'}{2} + \sum_{x^{1/3} < p \leq x^{1/2}} \pi(x/p), \qquad a = \pi(x^{1/3}), \quad a' = \pi(x^{1/2}).$$

The computation of the Meissel sum

$$\sum_{x^{1/3} < p \leq x^{1/2}} \pi(x/p)$$

needs to count the prime numbers in the interval $[x^{1/3}, x^{2/3}]$. This interval is sieved slice by slice, where the slices are intervals of width $x^{1/3}$. The computation uses for each slice an auxiliary sieve, in order to determine the prime numbers p such that x/p falls in the current slice. The value of π is updated during the handling of the slice. The value of $\pi(x^{1/2})$ is stored when the suitable slice is processed.

Estimating the contribution of ordinary leaves. During the preceding step the sum associated to the ordinary leaves

$$\sum_{1 \leq n \leq x^{1/3}} \mu(n) \lfloor x/n \rfloor$$

is also computed.

Estimating the contribution of special leaves. This is the most intricate part of the method. We have to evaluate

$$S = \sum_{(n,b)} \mu(n)\phi(x/n, b)$$

for all special leaves (n, b), i.e., $n = p_{a_1} \cdots p_{a_r}$ with $a \geq a_1 > a_2 > \cdots > a_r = b + 1$ and $n \geq x^{1/3} \geq n/p_{b+1}$.

We will evaluate this sum by sieving the interval $[x^{1/3}, x^{2/3}]$ by subintervals of length $x^{1/3}$. Let $N = \lfloor x^{1/3} \rfloor$. Suppose the number x/n is in the k-th subinterval $[(k-1)N + 1, kN]$. Then (n, b) is a special leaf if and only if $n = n^* p_{b+1}$, $f(n^*) > p_{b+1}$ and

$$\frac{x}{(kN+1)p_{b+1}} < n^* \leq \frac{x}{((k-1)N+1)p_{b+1}}.$$

In other words, n^* belongs to an interval $[L, M]$ and the contribution of $(x/n, b)$ to the sum S is non-zero if and only if $\mu(n^*) \neq 0$. This shows the process: we loop through those numbers m in $[L, M]$ such that $f(m) > p_{b+1}$ and for which $\mu(m) \neq 0$. This is easy using the tables precomputed in phase 1. In order to complete the evaluation, one must set up the computations in a clever way, described in the original paper (see also [2]). This crude description yields an algorithm with time $O(x^{2/3})$ which can be lowered to $O(x^{2/3}/\log x)$ using a trick due to Miller and described in the paper.

At the end, the values of a, $P_2(x, a)$ and $\phi(x, a)$ are combined and $\pi(x)$ is obtained. The total time for computing $\pi(x)$ is thus $O(x^{2/3}/\log x)$ operations and $O(x^{1/3}\log^2 x \log\log x)$ space.

4. Deléglise and Rivat

In [2], the authors describe a variant of the above approach that uses $O(x^{2/3}/\log^2 x)$ operations and $O(x^{1/3}\log^3 x \log\log x)$ space. They have computed all values of $\pi(x)$ for $x \geq 10^{15}$ up to 10^{18} for which $\pi(10^{18}) = 24739954287740860$.

Bibliography

[1] Brent (R. P.). – The first occurrence of large gaps between successive primes. *Mathematics of Computation*, vol. 27, n° 124, October 1973, pp. 959–963.

[2] Deléglise (M.) and Rivat (J.). – Computing $\pi(x)$: The Meissel, Lehmer, Lagarias, Miller, Odlyzko method. *Mathematics of Computation*, vol. 65, n° 213, January 1996, pp. 235–245.

[3] Hardy (G. H.) and Wright (E. M.). – *An Introduction to the Theory of Numbers.* – Oxford University Press, 1979, fifth edition.

[4] Lagarias (J. C.), Miller (V. S.), and Odlyzko (A. M.). – Computing $\pi(x)$: The Meissel-Lehmer method. *Mathematics of Computation*, vol. 44, n° 170, April 1985, pp. 537–560.

[5] Lagarias (J. C.) and Odlyzko (A. M.). – Computing $\pi(x)$: an analytic method. *Journal of Algorithms*, vol. 8, 1987, pp. 173–191.

[6] Lehmer (D. H.). – On the exact number of primes less than a given limit. *Illinois Journal of Mathematics*, vol. 3, 1959, pp. 381–388.

[7] Morain (F.) and Nicolas (J.-L.). – *Mathématiques / Informatique – 14 problèmes corrigés.* – Vuibert, 1995, *Enseignement Supérieur et Informatique.*

[8] Pritchard (P.). – Improved incremental prime number sieves. In Adleman (L.) and Huang (M.-D.) (editors), *ANTS-I. Lecture Notes in Computer Science*, vol. 877, pp. 280–288. – Springer-Verlag, 1994. First Algorithmic Number Theory Symposium - Cornell University, May 6–9, 1994.

[9] Riesel (Hans). – *Prime Numbers and Computer Methods for Factorization.* – Birkhäuser, 1985, *Progress in Mathematics*, vol. 57.

On a problem of Rubel

John Shackell
University of Kent at Canterbury, U.K.

April 22, 1996

[summary by Frédéric Chyzak]

Abstract

For a given function f, we study all the functions that satisfy every algebraic differential equation with constant coefficients which is satisfied by f. This question was suggested by Lee Rubel in [3, Problem 22]. Here the author characterizes this set of functions, first when f is a linear combination of exponential functions, next when f is a Liouvillian function. Finally, he applies these results to the computation of a series expansion of solutions of algebraic differential equations.

1. Exponential functions

For two functions f and g, define $g \ll f$ to mean that g satisfies every algebraic differential equation with constant coefficients which is satisfied by f. Let f be the following $\mathbb{C}$-linear combination of exponential functions

$$\sum_{k=1}^{n} a_k e^{\lambda_k x}.$$

Trivially, $g \ll f$ implies that $g = \sum_{k=1}^{n} A_k e^{\lambda_k x}$ with $A_k \in \mathbb{C}$, since the differential polynomial $L(y)$ defined by the linear operator $\prod_{k=1}^{n} \left(\frac{d}{dx} - \lambda_k\right)$ vanishes at f. (We refer the reader to [2] for an introduction to differential algebra.) This necessary condition for $g \ll f$ is not always sufficient. Two cases occur, according to the dimension d of the $\mathbb{Q}$-vector space generated by the λ_k. Note that this dimension is also the transcendence degree of $\mathbb{C}(e^{\lambda_1 x}, \ldots, e^{\lambda_n x})$ over $\mathbb{C}$.

Transcendence degree $d = n$. In this case, no equation of order less than d is satisfied by f. If $P(y)$ is another differential polynomial of order d that vanishes at f, L must divide P. Otherwise, using L to rewrite $f^{(d)}$ as a polynomial in the derivatives of f of lower orders yields a differential polynomial of order less than d. This polynomial must then be zero, which gives a contradiction. Therefore, g satisfies any equation of order d satisfied by f. Next, let $Q(y)$ be a differential polynomial satisfied by f. Differentiating L sufficiently many times makes it possible to rewrite all the derivatives of f of order greater or equal to d that occur in Q as polynomials in derivatives of order less than d. Once again, L divides Q so that $Q(g) = Q(f) = 0$. Hence, $g \ll f$.

Transcendence degree $d \leq n$. In this case, assume that $\lambda_1, \ldots, \lambda_d$ are linearly independent over $\mathbb{Q}$, whereas

$$\lambda_i = \sum_{j=1}^{d} c_{i,j} \lambda_j \qquad \text{for } c_{i,j} \in \mathbb{Q}, \text{ when } i = d+1, \ldots, n. \tag{1}$$

Taking $n-1$ derivatives of the equation $f = \sum_{k=1}^{n} a_k e^{\lambda_k x}$ yields a linear system relating the $a_k e^{\lambda_k x}$'s and the derivatives of f. This system has a Vandermonde determinant, hence we obtain linear expressions

$$a_k e^{\lambda_k x} = R_k(f, \ldots, f^{(n-1)}) = R_k, \qquad \text{for } k = 1, \ldots, n. \tag{2}$$

Combining equations (1–2) so as to eliminate the λ_k's yields the equations

$$a_i^{b_i} \prod_{j=1}^{d} a_j^{-\gamma_{j,i}} = R_i^{b_i} \prod_{j=1}^{d} R_j^{-\gamma_{j,i}} = S_j(f, \ldots, f^{(n-1)}), \qquad i = d+1, \ldots, n, \tag{3}$$

where b_i is a least common multiple for the denominators of the $c_{i,j}$'s and each $\gamma_{i,j} = b_j c_{i,j}$ is an integer. Now, if $g \ll f$, the function g also satisfies the second equality in (3). In addition, it is of the form $g = \sum_{k=1}^{n} A_k e^{\lambda_k x}$ and therefore,

$$a_i^{b_i} \prod_{j=1}^{d} a_j^{-\gamma_{j,i}} = A_i^{b_i} \prod_{j=1}^{d} A_j^{-\gamma_{j,i}}, \qquad i = d+1, \ldots, n. \tag{4}$$

We have obtained necessary and sufficient conditions for $g \ll f$ when f is of the form $\sum_{k=1}^{n} a_k e^{\lambda_k x}$.

Another approach based on differential ring homomorphisms. We now give another derivation of these conditions. This second approach follows methods similar to methods of differential Galois theory and will prove very fruitful when generalizing to Liouvillian functions.

We have a tower of function rings

$$\Phi_0 = \mathbb{C} \subset \cdots \subset \Phi_k = \mathbb{C}[e^{\lambda_1 x}, \ldots, e^{\lambda_k x}] \subset \cdots \subset \Phi_n = \mathbb{C}[e^{\lambda_1 x}, \ldots, e^{\lambda_n x}].$$

Write $\hat{\Phi}_k$ for the quotient field of Φ_k. It follows from (1) that the field extensions $\hat{\Phi}_k : \hat{\Phi}_{k-1}$ are transcendental for $k = 1, \ldots, d$ and algebraic for $k = d+1, \ldots, n$, with minimal polynomials

$$m_k\left(e^{\lambda_k x}\right) = \left(e^{\lambda_k x}\right)^{b_k} - \prod_{i=1}^{d} \left(e^{\lambda_i x}\right)^{\gamma_{k,i}}. \tag{5}$$

For complex constants C_k, consider the ring homomorphism $T : \Phi_n \to \Phi_n$ given by $T\left(e^{\lambda_k x}\right) = C_k e^{\lambda_k x}$ for $k = 1, \ldots, n$. We want to constrain the C_k's so that T is also a differential ring homomorphism that maps $f = \sum_{k=1}^{n} a_k e^{\lambda_k x}$ to $g = \sum_{k=1}^{n} A_k e^{\lambda_k x}$. Necessarily, $A_k = C_k a_k$ and the minimal polynomials (5) are mapped to themselves, modulo non-zero multiplicative constants $\eta_k \in \mathbb{C}$, so that

$$T\left(m_k\left(e^{\lambda_k x}\right)\right) = \left(C_k e^{\lambda_k x}\right)^{b_k} - \prod_{i=1}^{d} \left(C_i e^{\lambda_i x}\right)^{\gamma_{k,i}} = \eta_k\left(m_k\left(e^{\lambda_k x}\right)\right) = \eta_k \left(\left(e^{\lambda_k x}\right)^{b_k} - \prod_{i=1}^{d} \left(e^{\lambda_i x}\right)^{\gamma_{k,i}} \right).$$

It follows that $\eta_k = C_k^{b_k} = \prod_{i=1}^{d} C_i^{\gamma_{k,i}}$, so that condition (4) is also a necessary and sufficient condition for T to be a differential ring isomorphism.

In the next section, we construct a set of differential ring homomorphisms and investigate its connection to the set $\{g \mid g \ll f\}$ when f is a Liouvillian function.

2. Liouvillian functions

We now turn to differential extension towers of the form

$$\Phi_0 = \mathbb{C} \subset \cdots \subset \Phi_k = \Phi_{k-1}[z_k] \subset \cdots \subset \Phi_n = \Phi_{n-1}[z_n], \tag{6}$$

where the extension $\Phi_k = \Phi_{k-1}[z_k]$ is either

(*i*) an algebraic extension given by the minimal polynomial $m_k(z_k) = 0$ with coefficients in Φ_{k-1};
(*ii*) an exponential extension given by $z_k = \exp(w_{k-1})$, for $w_{k-1} \in \hat{\Phi}_{k-1}$;
(*iii*) an integral extension given by $z_k = \int w_{k-1}$, for $w_{k-1} \in \hat{\Phi}_{k-1}$.

In cases (*ii*) and (*iii*), write $w_{k-1} = \zeta_{k-1}/\eta_{k-1}$ for coprime $\zeta_{k-1}, \eta_{k-1} \in \Phi_{k-1}$. An element of a field $\hat{\Phi}_k$ corresponding to a tower (6) is called a Liouvillian function.

We now proceed to define sets $\mathbf{G}_k$ of differential ring homomorphisms from Φ_k to rings of Liouvillian functions. This construction generalizes that of T in the previous section. We take $\mathbf{G}_0$ to be the singleton of the identity on $\mathbb{C}$ and define the $\mathbf{G}_k$'s by induction on k, considering the three cases above separately. For any differential polynomial $P \in \Phi_k\{y\}$ and any $\rho \in \mathbf{G}_k$, let $\tilde{\rho}(P)$ denote the differential polynomial in $\rho(\Phi_k)\{y\}$ obtained by applying ρ to each coefficient of P.

Algebraic extensions. For any $\rho \in \mathbf{G}_{k-1}$ and any choice of root s of $\tilde{\rho}(m_k)$, ρ extends to Φ_k as a differential ring homomorphism by mapping z_k to s. We define $\mathbf{G}_k$ to be the set of all these extensions.

Exponential extensions. For any $\rho \in \mathbf{G}_{k-1}$ such that $\rho(\eta_{k-1}) \neq 0$, $\rho(w_{k-1})$ is well-defined and ρ extends to Φ_k as a differential ring homomorphism by mapping z_k to $K\exp(\rho(w_{k-1}))$. We define $\mathbf{G}_k$ to be the set of all these extensions.

Integral extensions. For any $\rho \in \mathbf{G}_{k-1}$ such that $\rho(\eta_{k-1}) \neq 0$, $\rho(w_{k-1})$ is well-defined and ρ extends to Φ_k as a differential ring homomorphism by mapping z_k to $K \int \rho(w_{k-1})$. We define $\mathbf{G}_k$ to be the set of all these extensions.

The main theorem. The previous construction yields the following theorem. A proof is given in [5]. Similar results are also presented in [4, Proposition 2].

Theorem 1. *Let the Liouvillian extension tower (6) and $\mathbf{G}_n$ be as above. Let $f = f_1/f_2 \in \hat{\Phi}_n$, with coprime $f_1, f_2 \in \Phi_n$. Then $g \ll f$ if and only if there exists an open dense subset W of $\mathbb{C}$ such that g belongs to the closure of the set*

$$\left\{\rho(f) \middle| \rho \in \mathbf{G}_n, \rho(f_2) \neq 0\right\}$$

in the topology of uniform $\mathcal{C}^\infty$ convergence on compact subsets of W.

3. An example

As an example, we compute the set of functions g such that $g \ll f$ with $f = (\exp(e^x) - 1)/e^x$. An algebraic differential equation satisfied by f is

$$ff'' - f'^2 - ff' - f' + f - f^2 = 0. \tag{7}$$

We have the tower of Liouvillian extensions $\mathbb{C} \subset \mathbb{C}[x] \subset \mathbb{C}[x, e^x] \subset \mathbb{C}[x, e^x, e^{e^x}] \ni f$. The first extension is given by $x = \int 1$; the latter two are exponential extensions. The differential ring homomorphisms T are defined such that:

(i) they are the identity on $\mathbb{C}$

$$T|_{\mathbb{C}} = T_0 : 1 \mapsto 1;$$

(ii) they extend to the integral extension $\mathbb{C}[x]$ by introducing a constant K_0

$$T|_{\mathbb{C}[x]} = T_1 : x \mapsto \int T_0(1) = x + K_0;$$

(iii) they extend to the first exponential extension $\mathbb{C}[x, e^x]$ by introducing a constant K_1

$$T|_{\mathbb{C}[x,e^x]} = T_2 : e^x \mapsto Ke^{T_1(x)} = K_1 e^x;$$

(iv) they extend to the second exponential extension $\mathbb{C}[x, e^x, e^{e^x}]$ by introducing a constant K_2

$$T = T_3 : e^{e^x} \mapsto K' e^{T_2(e^x)} = K_2 e^{K_1 e^x}.$$

Finally, the set of functions g such that $g \ll f$ is the closure of the set

$$\left\{ \frac{K_2 e^{K_1 e^x} - 1}{K_1 e^x} \middle| K_1, K_2 \in \mathbb{C}, K_1 \neq 0 \right\}.$$

Making $K_2 = 1$, next K_1 tend to 0 yields the function 1, which is indeed a solution of (7). We have thus proved that $1 \ll (\exp(e^x) - 1)/e^x$.

4. Series expansion

Theorem 1 can be used to help compute a series expansion for a solution of an algebraic differential equation belonging to a Hardy field [1]. It can be proved that the number of possible nested (asymptotic) forms f_0 for a solution is finite. This number grows exponentially with the order of the equation. Writing f in the form $f_0(1 + \epsilon)$, and substituting it into the equation yields an equation for the rest ϵ, of possibly doubled order. It follows that the exponential complexity of this first, naive method makes it impracticable.

Assume f can be written in the form $F + g$, where F is the sum of a finite number of first terms in an asymptotic expansion and g is the rest, of smaller asymptotic growth. If f does not belong to the closure under consideration in Theorem 1 applied to the Liouvillian function F, then there is a differential polynomial $P(y)$ that vanishes on F but not on f. From the equation defining f, the finitely many possible orders of growth of $P(f)$ can be computed. Next, each term in $P(f) = P(F + g)$ contains g or one of its derivatives. This yields a number of possible orders of growth for g, hopefully smaller than the one obtained by the general method.

Bibliography

[1] Bourbaki (N.). – *Éléments de Mathématiques*, Chapter V: Fonctions d'une variable réelle (appendice), pp. 36–55. – Hermann, Paris, 1961, second edition.

[2] Ritt (Joseph Fels). – *Differential Algebra*. – A.M.S., 1950, *A.M.S. Colloquium*, vol. XXXIII.

[3] Rubel (Lee A.). – Some research problems about algebraic differential equations. *Transactions of the American Mathematical Society*, vol. 280, 1983, pp. 43–52.

[4] Shackell (John R.). – Growth orders occuring in expansions of Hardy field solutions of algebraic differential equations. *Annales de l'Institut Fourier*, vol. 45, 1995, pp. 183–221.

[5] Shackell (John R.). – On a problem of Rubel concerning the set of functions satisfying all the algebraic differential equations satisfied by a given function. – Preprint, 1995.

On integer Chebyshev Polynomials

Bruno Salvy
INRIA Rocquencourt

January 29, 1996

[summary by Xavier Gourdon]

Abstract

We deal with the problem of minimizing the supremum norm on $[0, 1]$ of non zero polynomials of degree at most n with integer coefficients.

1. Introduction

We consider the supremum norm on polynomials $\|P\|_\infty = \max_{[0,1]} |P(t)|$. We denote by $\mathbb{Z}_k[x]$ the set of polynomials with integer coefficients of degree $\le k$. We consider the polynomials P_k in $\mathbb{Z}_k[x]$ and the quantities C_k such that

$$\|P_k\|_\infty = \min_{P \in \mathbb{Z}_k[x] \setminus \{0\}} \|P\|_\infty, \qquad \text{and} \qquad C_k = -\frac{1}{k} \log \|P_k\|_\infty. \tag{1}$$

According to [1], the polynomials P_k are called integer Chebyshev polynomials in $[0, 1]$. These polynomials appeared in the literature because as we discuss below, it was thought that they could be used to obtain an elementary proof of prime number theorem. Aparicio showed that in fact, one cannot prove the prime number theorem in this way. However, the problem of finding the polynomials P_k is interesting in itself. According to Borwein and Erdélyi, "Even computing low-degree examples is difficult".

2. The prime number theorem

Let d_n denote the lowest common multiple of $1, 2, \ldots, n$. Proving the prime number theorem can be elementary reduced to proving the inequality

$$\liminf_{n \to \infty} \frac{\log d_n}{n} \ge 1.$$

An idea to obtain this result is to use the fact that $P \in \mathbb{Z}_m[x]$ implies $\int_0^1 P(x)\, dx \in \mathbb{Z}/d_{m+1}$. Applying this to the polynomial P_k^{2n} leads to

$$\|P_k\|_\infty^{2n} \ge \int_0^1 P_k^{2n}(x)\, dx \ge \frac{1}{d_{2kn+1}}, \qquad \text{thus} \qquad \liminf_{n \to \infty} \frac{\log d_n}{n} \ge -\frac{\log \|P_k\|_\infty}{k} = C_k.$$

Therefore, if we had $\limsup_{k \to \infty} C_k = 1$, one could prove the prime number theorem in this way. Indeed, it appears that this is not the case. The sequence (C_k) converges to a limit C, and Borwein and Erdélyi [1] showed that $C \in (0.8586616, 0.8657719)$. Thanks to our new results, we improve the lower bound on C.

3. Related problems

3.1. Integer transfinite diameter. Our problem can be stated in terms of *integer transfinite diameter*. The transfinite diameter of a set S of complex numbers is defined by

$$t(S) := \lim_{n\to\infty} \sup_{z_1,\ldots,z_n\in S} \prod_{i<j} |z_i - z_j|^{1/\binom{n}{2}}.$$

A theorem of Fekete states that

$$t(S) = \inf_{P\in\mathbb{C}[x],P \text{ monic}} \max_{x\in S} |P(x)|^{1/\deg(P)}.$$

The *integer transfinite diameter* of a subset S of $\mathbb{R}$ is defined by

$$t_{\mathbb{Z}}(S) = \inf_{P\in\mathbb{Z}[x],\deg(P)>0} \max_{x\in S} |P(x)|^{1/\deg(P)}.$$

Thus, our problem can be rephrased as: finding the integer transfinite diameter of the interval $[0,1]$. If I is the interval $[a,b]$ with $a<b$, it is known that $t(I) = |I|/4$, with $|I| = b-a$. If $|I|\geq 4$, we have the equality $t_{\mathbb{Z}}(I) = t(I)$. For $|I|<4$, the best known result is due to Fekete and states that $t(S) \leq t_{\mathbb{Z}}(S) \leq \sqrt{t(S)}$.

3.2. Trace of totally positive algebraic integers. Let α_1 be an algebraic integer of d, $\alpha_2,\ldots,\alpha_d$ its conjugates. We say that α_1 is *totally positive* if all the α_i are real and positive. Siegel has proved in 1945 that except for finitely many exceptions, we have the following lower bound on totally positive algebraic integers

$$\frac{\alpha_1 + \cdots + \alpha_d}{d} \geq 1.733.$$

A general result states that this problem is related to the integer transfinite diameter:

Theorem 1 (Borwein, Erdélyi). *Let m be a positive integer.*

$$\textit{If} \qquad t_{\mathbb{Z}}\left(\left[0,\frac{1}{m}\right]\right) < \frac{1}{m+\delta} \qquad \textit{then} \qquad \frac{\alpha_1+\cdots+\alpha_d}{d} \geq \delta$$

for totally positive algebraic integers, with finitely many exceptions.

4. Structure of the polynomials

The set $E_k = \{P \in \mathbb{Z}_k[x] :\ P(1-x) = (-1)^k P(x)\}$ is related to our problem by the following lemma [2].

Lemma 1. *For any nonnegative integer k, we have*

$$E_{2k} = \mathbb{Z}_k[x(1-x)] \qquad \textit{and} \qquad E_{2k+1} = (1-2x)\mathbb{Z}_k[x(1-x)],$$

and there exists an element F of degree k in E_k for which

$$C_k = -\frac{1}{k}\log\|F\|_\infty.$$

5. Computation of minimal polynomials

The previously known integer Chebyshev polynomials had small degrees. We now briefly describe the techniques used to compute a polynomial P_k of degree k satisfying (1) for k up to 75. The outline of the algorithm goes as follows:

(1) Find a good upper bound for $\|P_k\|_\infty$;
(2) Repeat
 – use this bound to determine factors of P_k,
 – use these factors to improve the bound,
 until no more factors are found;
(3) Perform an exhaustive search for the missing factors.

5.1. First upper bound. A good bound is given by $c_k = \min_{0<\ell<k} \|P_\ell P_{k-\ell}\|_\infty$.

5.2. Bounds and factors. We use the following facts to find factors of $G \in \mathbb{Z}_g[x]$.

- If $q^g|G(p/q)| < 1$ then $(qx-p)$ is a factor of G.
- This technique extends to multiple factors via Markov's inequality:

$$\max_{a\le x\le b} |G^{(r)}(x)| \le \frac{2^r}{(b-a)^r} \frac{n^2(n^2-1^2)\cdots(n^2-(r-1)^2)}{1\cdot 3\cdots(2r-1)} \max_{a\le x\le b} |G(x)|.$$

- At $x=0$, we have a better bound due to Borwein and Erdélyi:

$$G(x) = x^{g-p}Q(x) \qquad \Longrightarrow \qquad |Q(0)| \le \sqrt{2p+1}\binom{g+p+1}{g-p}\|G\|_\infty.$$

- More generally, we can find higher degree factors. Let $F = a_0x^n + \cdots + a_n \in \mathbb{Z}[x]$ be irreducible, $\alpha_1, \ldots, \alpha_n$ its roots. The expression $R = a_0^g G(\alpha_1)\cdots G(\alpha_n)$ is an integer (it is a resultant). If $|R| < 1$, then F is a factor of G.

Once factors have been obtained in this way, we have $P_k(x) = F(x)G(x(1-x))$, where F is known and G unknown. Bounds on $G(x)$ at a given x can be obtained using the fact that $|F(u(x))G(x)| \le \|P_k\|_\infty \le c_k$ with $u(x) = \frac{1}{2}(1-\sqrt{1-4x})$. This enables to find other factors. This technique provides all the integer Chebyshev polynomials of degree ≤ 12.

To get tighter bounds on the value of G at a given x, we then turn to *Lagrange interpolation.* If $x_0, \ldots, x_g$ are $g+1$ distinct points in $[0, 1/4]$ then

$$G(x) = \sum_{i=0}^{g} G(x_i) \prod_{j\ne i} \frac{x-x_j}{x_i-x_j} \quad \text{thus} \quad |G(x)| \le c_k \sum_{i=0}^{g} \frac{1}{|F(u(x_i))|} \prod_{j\ne i} \left|\frac{x-x_j}{x_i-x_j}\right|.$$

This gives a bound on $|G(x)|$, which can be further improved by finding a set $\{x_0, \ldots, x_g\}$ which minimizes the right-hand side of the inequality. By this technique, all Chebyshev of degree ≤ 30 are found.

5.3. Exhaustive search. By plugging values of x in the inequality $|F(u(x))| \cdot |G(x)| \le c_k$, we get linear inequalities satisfied by the coefficients of the factor G. These inequalities define a polyhedron whose interior integer points we have to determine. We solve this problem by using a simplex method to compute bounds on each coordinate. Then if the size of the bounding polyrectangle is not too large, we check each of its points to see whether it belongs to the polyhedron. For larger polyrectangles, we select the variable with least variation and apply recursively the same technique. In this way, we test a finite set of polynomials. This technique is reasonable for $n \le 13$ (i.e., degree 24).

5.4. A detailed example: P_{37}. We show how to find P_{37} using our algorithm.

A first upper bound is obtained from the previous polynomials

$$||P_{37}||_\infty \le c_{37} = \min_\ell ||P_\ell P_{37-\ell}||_\infty = 0.283\,10^{-13}.$$

We then look for factors of P_{37}. At each stage, we have $P_{37}(x) = F(x)G(x(1-x))$ with F known and G unknown, $g = \deg(G)$.

- Since 37 is odd, a factor is $F = 1 - 2x$ by lemma 1 ($g = 18$).
- We have $5^{18}c_{37} < |F(u(1/5))|$ thus $5^{18}|G(1/5)| < 1$, and a factor is $F := F \cdot (5x^2 - 5x + 1)$ ($g = 17$).
- Using the Borwein-Erdélyi bound, we find the factor $F := F \cdot x^9(1-x)^9$ ($g = 8$).
- Using Lagrange interpolation, we find $|G(0)| < 1$, thus a factor is $F := F \cdot x(1-x)$ ($g = 7$).
- The same technique applied with the new factor F gives $|G(0)| < 1$, thus a factor is $F := F \cdot x(1-x)$ ($g = 6$).
- The same technique gives $4^6|G(1/4)| < 1$, thus $F := F \cdot (4x^2 - 4x + 1)$ ($g = 5$).
- The same technique gives
$$29^5 \left| G\left(\frac{11+\sqrt{5}}{58}\right) G\left(\frac{11-\sqrt{5}}{58}\right) \right| < 1$$
thus $F := F \cdot (29x^4 - 58x^3 + 40x^2 - 11x + 1)$ ($g = 3$).
- The same technique gives $|G(0)| < 1$ thus $F = F \cdot x(1-x)$ ($g = 2$).
- The same technique gives $4^2|G(1/4)| < 1$, thus $F := F \cdot (4x^2 - 4x + 1)$ ($g = 1$).

The step of exhaustive search finally yields 6 solutions, and only one has the right $||\cdot||_\infty$. Eventually, we find

$$P_{37}(x) = x^{12}(1-x)^{12}(1-2x)^5(5x^2 - 5x + 1)^2(29x^4 - 58x^3 + 40x^2 - 11x + 1).$$

6. A new factor

The only factors of all the 75 first polynomials are the following, expressed in the variable $u = x(1-x)$,

$$A_1 = u, \quad A_2 = 4u - 1, \quad A_3 = 5u - 1, \quad A_4 = 6u - 1, \quad A_5 = 29u^2 - 11u + 1,$$
$$A_6 = 169u^3 - 94u^2 + 17u - 1, \quad A_7 = 961u^4 - 712u^3 + 194u^2 - 23u + 1,$$
$$A_8 = 4921u^5 - 4594u^4 + 1697u^3 - 310u^2 + 28u - 1.$$

The factor A_8 is a new one, and it has four non real root, which gives a negative answer to an open problem from [1]: Do all the integer Chebyshev polynomials on $[0,1]$ have all their zeros in $[0,1]$ Γ

Thanks to this new factor we can improve the bound on C. Following the lines of [1], we use a simplex method to compute a polynomial $Q = A_1^{\beta_1} A_2^{\beta_2} \cdots$ of degree $d = 10^{10} - 9$ such that $-\frac{1}{d}\log ||Q||_\infty = 0.8591978$, thus $C > 0.8591978$.

Bibliography

[1] Borwein (Peter) and Erdélyi (Tamás). – The integer Chebyshev problem. *Mathematics of Computation*, 1995. – To appear.

[2] Habsieger (Laurent) and Salvy (Bruno). – On integer Chebyshev polynomials. *Mathematics of Computation*, 1996. – To appear.

Algebraic Computation of Matrix-like Padé Approximants

George Labahn
University of Waterloo, Canada

June 10, 1996

[summary by Bruno Salvy]

Abstract

Padé approximants are rational approximants to functions represented as power series. There are many classes and generalizations of Padé approximants, with various kinds of applications. After reviewing some of these approximants and their use, this work presents a unified way of computing them.

1. A gallery of Padé approximants

Given a formal power series $A(z)$, a *Padé approximant* of type (m, n) is a pair of polynomials $(u(z), v(z))$ of degrees at most m and n respectively, such that $A(z) - u(z)/v(z) = O(z^{m+n+1})$.

Hermite-Padé approximants constitute a natural generalization of Padé approximants. Instead of *one* power series, the input consists in ℓ power series $A_1(z), \dots, A_\ell(z)$ and ℓ integers $n_1, \dots, n_\ell$. The approximant is then an ℓ-tuple of polynomials $(p_1(z), \dots, p_\ell(z))$, with $p_i(z)$ of degree at most $n_i - 1$, such that

$$p_1(z)A_1(z) + \cdots + p_\ell(z)A_\ell(z) = O(z^{N-1}),$$

where $N = \sum n_i$.

The extended Euclidean algorithm can be seen as the calculation of a Hermite-Padé approximant. Given two polynomials $P(z)$ and $Q(z)$, the extended Euclidean algorithm computes three polynomials $U(z)$, $V(z)$ and $G(z)$, such that $G(z)$ is the gcd of $P(z)$ and $Q(z)$, and the Bézout identity holds

$$U(z)P(z) + V(z)Q(z) = G(z).$$

This is the same as computing a Hermite-Padé approximant for the reciprocal polynomials of $P(z)$ and $Q(z)$.

Hermite-Padé approximants are used in `gfun` [3] to guess linear differential equations or algebraic equations satisfied by a formal power series $A(z)$. In this context, one starts with $A_i(z) = A^{(i-1)}(z)$ or $A_i(z) = A^{i-1}(z)$.

A generalization of these approximants is obtained by considering *vectors* or *matrices* of power series, leading to vector and matrix Hermite-Padé approximants. Vector Hermite-Padé approximants are used in algorithms factoring linear differential operators [4].

Another kind of generalization called *simultaneous Padé approximants* was introduced by Hermite in 1873 in order to prove the transcendence of e. As in the case of Hermite-Padé approximants one starts with ℓ power series $A_1(z), \dots, A_\ell(z)$. Given $\ell + 1$ integers $(n_0, n_1, \dots, n_\ell)$, the aim is to find $\ell + 1$ polynomials $q(z), p_1(z), \dots, p_\ell(z)$ such that $A_j(z) = p_j(z)/q(z) + O(z^K)$.

Again, vector and matrix versions are of interest.

2. Computation

All these approximants can be computed by linear algebra algorithms, since they correspond to solving an equation of the type $AX = B$, where X is a vector of the unknown coefficients of the approximants, A encodes the product modz^N by the initial data in the basis $1, z, z^2, \ldots$ and B represents the desired right-hand side modz^N. Thus efficient algorithms for Gaussian elimination and fraction free versions of these can be used. The solution set has the structure of a module. In many cases, this module has dimension one, so that any approximant generates all of them. In other cases, it might be useful to compute a basis of this module.

Example. This example helped discover a nice generating function [3]. The coefficients of the series

$$\begin{aligned} y(z) = 3 + 19z + 193z^2 + 2721z^3 + 49171z^4 + 1084483z^5 \\ + 28245729z^6 + 848456353z^7 + 28875761731z^8 + O(z^9) \end{aligned}$$

are the numerators of convergents to $e = \exp(1)$ of index $3k+1$. We are looking for a Hermite-Padé approximant of $(1, y, y')$ with degree constraints $(1, 2, 2)$. The matrix version of this problem is

$$\begin{bmatrix} 1 & 0 & 3 & 0 & 0 & 19 & 0 & 0 \\ 0 & 1 & 19 & 3 & 0 & 386 & 19 & 0 \\ 0 & 0 & 193 & 19 & 3 & 8163 & 386 & 19 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \cdot X = 0.$$

A basis of the kernel is readily found to be ${}^t(-3, -1, 1, -6, -1, 0, 0, -4)$, so that $y(z)$ satisfies the following differential equation up to $O(z^8)$:

$$4z^2 y'(z) - (1 - 6z - z^2)y + 3 + z = 0.$$

Another way of viewing the same computation, which preserves sparseness, is as a standard basis computation. For instance, in the case of Hermite-Padé approximants, one introduces new variables $t, a_1, \ldots, a_\ell$ and computes a standard basis for the set of series

$$a_1 - tA_1(z), \ldots, a_\ell - tA_\ell(z), z^{N-1},$$

with respect to any ordering such that $t > z$ and $z > z^2 > \cdots$ are smaller than the a_i's. The polynomials of the basis are linear in the a_i's, those which do not contain t generate the module of approximants.

Bibliography

[1] Beckermann (B.) and Labahn (G.). – A uniform approach for Hermite Padé and simultaneous Padé approximants and their matrix-type generalizations. *Numerical Algorithms*, vol. 3, 1992, pp. 45–54.

[2] Beckermann (Bernhard) and Labahn (George). – A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM Journal on Matrix Analysis and Applications*, vol. 15, n° 3, July 1994, pp. 804–823.

[3] Salvy (Bruno) and Zimmermann (Paul). – Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, vol. 20, n° 2, 1994, pp. 163–177.

[4] Van Hoeij (Mark). – *Formal Solutions and Factorization of Differential Operators with Power Series Coefficients.* – Report n° 9528, University of Nijmegen, July 1995.

Part 3

Asymptotic Analysis

The tricritical scaling function of partially directed vesicles

Thomas Prellberg
University of Oslo

October 9, 1995

[summary by Helmut Prodinger]

This talk is largely based on [4]; some other "Prellbergs" are cited therein[1]. The author considers *staircase polygons.* They are defined as the set of all polygons on the square lattice whose perimeter consists of two fully directed walks with common start and end points.

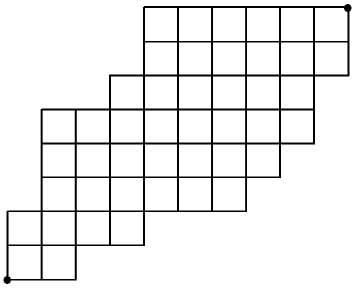


FIGURE 1. A staircase polygon with width 10, height 8, and area 45

If $c_m^{n_x,n_y}$ denotes the number of all staircase polygons with $2n_x$ horizontal and $2n_y$ vertical steps which enclose an area of size m, then the generating function

$$G(x,y,q)=\sum c_m^{n_x,n_y}x^{n_x}y^{n_y} \tag{1}$$

fulfills the functional equation

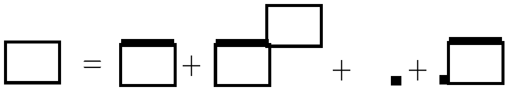
$$G(x,y,q)=\Big(G(qx,y,g)+qx\Big)\Big(G(x,y,g)+y\Big). \tag{2}$$

From this, an explicit expression is available;

$$G(x,y,q)=y\biggl(\frac{H(q^2x,qy,q)}{H(qx,qy,q)}-1\biggr)\quad\text{with}\quad H(x,y,q)=\sum_{n\geq 0}\frac{(-x)^n q^{\binom{n}{2}}}{(q;q)_n(y;q)_n}, \tag{3}$$

where $(y;q)_n := (1-y)(1-yq)(1-yq^2)\cdots(1-yq^{n-1})$.

[1] One might wonder why, then, the titles of talk and paper are so drastically different: "Vesicle" is a "closed fluctuating membrane", but combinatorialists think about polygons. And "tricritical" means that the generating function of interest has three ranges with a somehow different behaviour. The whole study is devoted to asymptotics of the generating function of interest, if the argument approaches the "tricritical" point.



$$G(x) \quad = G(qx) y + G(qx)G(x) + \quad qxy + qxG(x)$$

Prellberg derives this functional equation by setting up a *symbolic equation* which he translates into a functional equation for the generating function — very much in the tradition of the Algorithm seminar.

If we forget about the area, then we obtain the *perimeter generating function*

$$G(x,y,1) = \frac{1-x-y}{2} - \sqrt{\left(\frac{1-x-y}{2}\right)^2 - xy}. \tag{4}$$

The author concentrates in getting the following theorem.

Theorem 1. *Set $\epsilon = -\log q$. Then, as $q \to 1$,*

$$G(x,y,q) \sim \frac{1-x-y}{2} + \sqrt{\left(\frac{1-x-y}{2}\right)^2 - xy}\,\left(\frac{\mathrm{Ai}'(\alpha\epsilon^{-2/3})}{\alpha^{1/2}\epsilon^{-1/3}\,\mathrm{Ai}(\alpha\epsilon^{-2/3})}\right). \tag{5}$$

Here, α is some complicated function of x and y which simplifies to

$$\alpha(x,y) \sim \left(\frac{4}{1-(x-y)^2}\right)^{4/3}\left(\left(\frac{1-x-y}{2}\right)^2 - xy\right) \tag{6}$$

for $(1-x-y)^2 \approx 4xy$. $\mathrm{Ai}(x)$ is the Airy function *(see* [5]*).*

Everything boils down to a study of the function $H(x,y,q)$, and the author comes up with a lemma.

Lemma 1. *For $x \in \mathbb{C}$, $|\arg(x)| < \pi$, $y \in \mathbb{C}$, $y \neq q^{-n}$ for non-negative integers n and $0 < q < 1$, we have*

$$H(x,y,q) = \frac{(q;q)_\infty}{(y;q)_\infty}\frac{1}{2\pi i}\int_{\rho-i\infty}^{\rho+i\infty}\frac{(y/z;q)_\infty}{(z;q)_\infty} z^{-\log x/\log q}\,dz, \qquad 0<\rho<1. \tag{7}$$

Such a representation is no surprise at all; check out the wonderful survey papers [2] and [3]. The basic idea is to use the formula

$$\sum_{n\geq 0}(-x)^n c_n = \frac{1}{2\pi i}\int_{\mathcal{C}} x^s\, c(s)\frac{\pi}{\sin \pi s}\,ds \tag{8}$$

where $\mathcal{C}$ encloses the points $0, 1, \ldots$ in the counter-clock direction. The function $c(s)$ is an analytic continuation of the sequence c_n. Ramanujan was very fond of this formula, and it is also related with the names of Abel, Plana, and Lindelöf.

To do asymptotics, the author needs a better understanding of the 'ingredients' in his function $H(x,y,q)$ (a q-Bessel function), as $q \to 1$.

Interchanging sums,

$$\log(t;q)_\infty = -\sum_{m\geq 1}\frac{1}{m}\frac{t^m}{1-q^m}. \tag{9}$$

From here, *Euler's summation formula* gives for $|\arg(1-t)| < \pi$

$$\log(t;q)_\infty = \frac{1}{\log q}\operatorname{Li}_2(t) + \frac{1}{2}\log(1-t) + O(\log q), \tag{10}$$

with *Euler's dilogarithm*

$$\operatorname{Li}_2(t) = \sum_{m\geq 1}\frac{t^m}{m^2} = -\int_0^\infty \frac{\log(1-u)}{u}\,du. \tag{11}$$

For $(q;q)_\infty$ the author uses a *modular transformation*, viz. (see [1])

$$\log(q;q)_\infty = (r;q)^{1/24}\sqrt{\frac{2\pi}{-\log q}}\frac{1}{(r;r)_\infty} \tag{12}$$

to get

$$\log(q;q)_\infty = \frac{\pi^2}{6\log q} + \frac{1}{2}\log_{1/q}(2\pi) + O(\log q). \tag{13}$$

(The Mellin transform would also give this result.)

Continuing with approximations, the author notes the following.

Lemma 2.

$$\begin{aligned} H(x,y,q) = \frac{1}{2\pi i}\int_{\rho-i\infty}^{\rho+i\infty} &\exp\left(\tfrac{1}{\epsilon}\left[\log(z)\log(x) + \operatorname{Li}_2(z) - \operatorname{Li}_2(y/z)\right]\right)\sqrt{\frac{1-y/z}{1-z}}dz \\ &\times\exp\left(\tfrac{1}{\epsilon}(\operatorname{Li}_2(y) - \tfrac{\pi^2}{6})\right)\sqrt{\frac{2\pi}{\epsilon(1-y)}}\,(1+O(\epsilon)). \end{aligned} \tag{14}$$

The asymptotic evaluation of this integral will be done with the *saddle-point method.* There are two saddle points, and the whole thing becomes complicated when they coalesce (see [6] for an introduction to this problem).

A change of variable brings the function

$$V(\lambda) = \frac{1}{2\pi i}\int_{\mathcal{C}'} e^{u^3/3-\lambda u}\,du \tag{15}$$

into the picture ($\mathcal{C}'$ a certain contour). It is expressible by the Airy function $\operatorname{Ai}(\lambda)$.

Prellberg then presents his main lemma.

Lemma 3. *Let* $0 < x, y < 1$ *and* $q = e^{-\epsilon}$. *Then*

$$\begin{aligned} H(x,y,q) = &\Big(p_0\epsilon^{1/3}\operatorname{Ai}(\alpha\epsilon^{-2/3}) + q_0\epsilon^{2/3}\operatorname{Ai}'(\alpha\epsilon^{-2/3})\Big) \\ &\qquad\times\exp\left(\tfrac{1}{\epsilon}(\operatorname{Li}_2(y) - \tfrac{\pi^2}{6} + \log(x)\log(y)/2)\right)\sqrt{\frac{2\pi}{\epsilon(1-y)}}\,(1+O(\epsilon)), \end{aligned} \tag{16}$$

where

$$\frac{4}{3}\alpha^{3/2} = \log(x)\log\frac{z_m-\sqrt{d}}{z_m+\sqrt{d}} + 2\operatorname{Li}_2(z_m-\sqrt{d}) - 2\operatorname{Li}_2(z_m+\sqrt{d}) \tag{17}$$

with

$$z_{1,2} = z_m \pm \sqrt{d} \qquad z_m = \frac{1+y-x}{2} \quad \textit{and} \quad d = z_m^2 - y \tag{18}$$

and

$$p_0 = \left(\frac{\alpha}{d}\right)^{1/4}(1-x-y), \qquad q_0 = \left(\frac{d}{\alpha}\right)^{1/4}. \tag{19}$$

Bibliography

[1] Andrews (George E.). – *The Theory of Partitions.* – Addison-Wesley, 1976, *Encyclopedia of Mathematics and its Applications*, vol. 2.

[2] Flajolet (Philippe), Gourdon (Xavier), and Dumas (Philippe). – Mellin transforms and asymptotics: harmonic sums. *Theoretical Computer Science, Series A*, vol. 144, n° 1-2, June 1995, pp. 3–58. – Special Volume on Mathematical Analysis of Algorithms.

[3] Flajolet (Philippe) and Sedgewick (Robert). – Mellin transforms and asymptotics: finite differences and Rice's integrals. *Theoretical Computer Science*, vol. 144, n° 1-2, June 1995, pp. 101–124.

[4] Prellberg (Thomas). – Uniform q–series asymptotics for staircase polygons. *Journal of Physics Series A: Math. Gen.*, vol. 28, 1995, pp. 1289–1304.

[5] Whittaker (E. T.) and Watson (G. N.). – *A Course of Modern Analysis.* – Cambridge University Press, 1927, fourth edition. Reprinted 1973.

[6] Wong (Roderick). – *Asymptotic Approximations of Integrals.* – Academic Press, 1989.

The statistical mechanics of vesicles

Thomas Prellberg
Department of Mathematics, University of Oslo

October 16, 1995

[summary by Dominique Gouyou-Beauchamps]

1. Polygons as vesicle models

Biological membranes consist of lipid bilayers and, when closed, form vesicles as blood cells or bi-lipid layer membranes. These 3-dimensional vesicles form a variety of shapes depending on the surface tension, osmotic pressure, etc (see Fig. 1).

A convenient model for the boundary of the two-dimensional vesicle is a polygon either in the continuum or on a lattice. The polygon is taken to be self-avoiding and one asks, in the lattice version, for the number of polygons with $2n$ edges enclosing area m. Here, we consider polygons on the square lattice (see Fig. 2).

We denote $c_{n,m}$ the number of all polygons with $2n$ steps which enclose an area of size m, and define the polygon-generating function $G(x,q)$ to be

$$G(x,q) = \sum_{n,m} c_{n,m} x^n q^m.$$

Each class of polygons (staircase polygons, bar-graph polygons, column-convex polygons) defines a model of vesicles. We want to give an explicit formula for $G(x,q)$ and information on its singularity structure for all the models.

2. Statistical mechanics, some rigorous results

Mathematically, the model requires the calculation of the same object, the generalized partition function $G(x,q)$, where

$$G(x,q) = \sum_{m=1}^{\infty} q^m Z_m(x) \qquad \text{with} \qquad Z_m(x) = \sum_{n=2}^{\infty} c_{n,m} x^n.$$

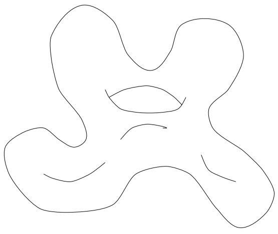


FIGURE 1. A vesicle.

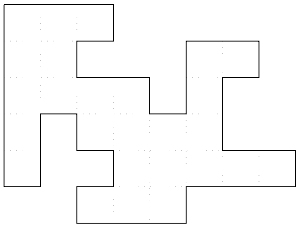


FIGURE 2. A polygon with area $m = 26$ and perimeter $2n = 42$.

Physically it is of interest to understand the behavior of the partition function $Z_m(x)$ of vesicles of fixed area m as the perimeter fugacity x is varied [6, 7, 4]. The behavior of the partition function for large vesicles is determined by the mathematical behavior of the generating function near its radius of convergence.

For a fixed area m, the free energy $H(\varphi)$ of a vesicle φ is related to the energy E and the perimeter $2n(\varphi)$ of φ through the relation $H(\varphi) = -E.n(\varphi)$. The partition function $Z_m(x)$ is

$$Z_m(x) = \sum_{|\varphi|=m} e^{-\beta H(\varphi)} = \sum_{n\geq 2} c_{n,m} e^{\beta E n}$$

with $x = e^{\beta E}$.

The total free energy is

$$-\beta f_m(x) = \frac{1}{m} \log Z_m(x)$$

and assuming the thermodynamic limit exists, we have for the thermodynamic free energy per step

$$f_\infty(x) = \lim_{m\to\infty} \frac{1}{m} \log(Z_m(x)).$$

We can also consider the internal energy

$$\frac{1}{E} u_m(x) = x \frac{d}{dx} \left(\frac{1}{m} \log Z_m(x) \right)$$

or the specific heat

$$\frac{1}{\beta E^2} c_m(x) = \left(x \frac{d}{dx} \right)^2 \left(\frac{1}{m} \log Z_m(x) \right).$$

Let $q_c(x)$ be the radius of convergence of the generating function $G(x, q)$ for fixed x:

$$q_c(x) = \lim_{m\to\infty} (Z_m(x))^{-\frac{1}{m}}.$$

For vesicles this is related to the free energy per unit length of vesicles of fixed area in the limit of large areas through the relation

$$q_c(x) = e^{\beta f_\infty(x)}, \qquad \text{where} \qquad -\beta f_\infty(x) = \lim_{m\to\infty} \frac{1}{m} \log(Z_m(x)).$$

3. Proof of the existence of the thermodynamic limit

We give here a sketch of the proof. For more details, see [9]. We use the following lemma:

LEMMA 1. *Let* $\{a_n\}_{n\geq 0}$ *be a sequence in* $\mathbb{R}$*. If the sequence is sub-additive* ($a_{n+m} \leq a_n + a_m$) *then* $\lim_{n\to\infty} \frac{1}{n} a_n = \inf_{n\to\infty} \frac{1}{n} a_n$ *exits (may be* $-\infty$*).*

By a standard concatenation construction in which two vesicles are joined by a 'neck' consisting of a single square, we obtain a larger vesicle and thereby find:

$$Z_{n+m}(q) \geq qZ_n(q)Z_m(q)$$

where $Z_n(q) = \sum_m c_{n,m}q^m$. Moreover, if we define

$$a_n = -\log(qZ_n(q))$$

then $\{a_n\}$ verifies $a_{n+m} \leq a_n + a_m$ and $\lim_{n\to\infty} (Z_n(q))^{\frac{1}{n}}$ exists.

Now, we examine bounds on $x_c(q) = \lim_{n\to\infty} (Z_n(q))^{\frac{1}{n}}$

Case $q \leq 1$. The minimum area for perimeter $2n$ is $m_{\min} = n - 1$ and hence $Z_n(q) \leq Z_n(1)q^{n-1}$ and $x_c(q) \geq \mu_{SAW}^{-2} q^{-1}$, where we write SAW for self-avoiding walk model.

The number of polygons with perimeter $2n$ and area $m_{\min}(n)$ is the number of site trees on dual lattice with $n-1$ vertices, say d_n, and hence $Z_n(q) \geq d_n q^{n-1}$ and $x_c(q) \leq \tilde{\mu} q^{-1}$ (see Fig. 3).

Since $Z_n(q)$ is monotone increasing in x, $x_c(q)$ is monotone non-decreasing. Therefore to prove that $x_c(q)$ is log-convex it suffices to show that:

$$\frac{x_c(p) + x_c(q)}{2} \geq x_c(\sqrt{pq}).$$

This follows immediately from

$$\begin{aligned} Z_n(q)Z_m(q) &= \sum_{m_1} c_{n,m_1} q^{m_1} \sum_{m_2} c_{n,m_2} q^{m_2} \\ &\geq \left(\sum_m c_{n,m} (pq)^{\frac{m}{2}} \right)^2 = (Z_n(\sqrt{pq}))^2 . \end{aligned}$$

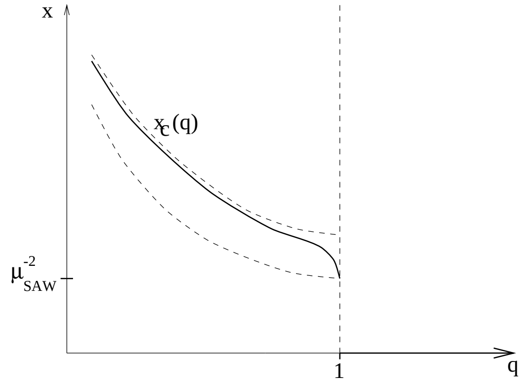


FIGURE 3. Schematic plot of the radius of convergence of the generating function showing the tricritical point.

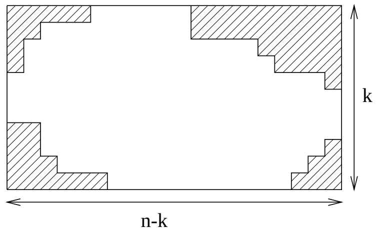


FIGURE 4. Interpretation of $Z_n^{(as)}(q)$.

Case $q \geq 1$. In that case, we have $q^{m_{\max}(n)} \leq Z_n(q) \leq q^{m_{\max}(n)} Z_n(1)$ with $m_{\max}(n) \sim \frac{n^2}{4}$ and $Z_n(1) \sim \mu_{SAW}^{2n}$. Thus $Z_n(q) \sim q^{\frac{n^2}{4}}$ and

$$x_c(q) \equiv 0.$$

In fact, the 'blown-up' configurations completely dominate the asymptotics.

THEOREM 1 (PRELLBERG, OWCZAREK, 1995).

$$Z_n(q) \sim Z_n^{(as)}(q) = \left(\frac{1}{q};\frac{1}{q}\right)_\infty^{-4} \sum_{k=-\infty}^{+\infty} q^{k(n-k)}$$

in the sense that for all $q > 1$ *there are* $C > 0$ *and* $0 < \rho < 1$ *such that for all* n

$$\left| Z_n(q)/Z_n^{(as)}(q) - 1 \right| < C\rho^n$$

We can interpret $Z_n^{(as)}(q)$ as the generating functions of $k \times (n-k)$ rectangles ($\sum_{k=1}^{n-1} q^{k(n-k)}$) where 4 corners (4 Ferrers diagrams: $\left(\frac{1}{q};\frac{1}{q}\right)_\infty^{-4}$) are removed, which are in fact convex polygons (see Fig. 4).

4. Tricritical phase diagram

We show that, for $q < 1$, $G(x,q)$ converges for $x < x_c(q)$. For $q > 1$, $G(x,q)$ converges only for $x = 0$. These results can be expressed in terms of a phase diagram in the space of the two fugacities x and q. The form of this phase diagram is shown in figure 3. For $x < x_c(q)$ and $q < 1$ the polygons are ramified objects, closely resembling branched polymers. As q approaches unity less ramified configurations predominate; at $q = 1$ one has standard self-avoiding polygons. This region, $\{x < x_c(q), y \leq 1\}$ might be referred to as the 'droplet' or 'compact' phase. For $q > 1$ the polygons become 'expanded' or 'inflated' and approximate squares, their average areas scaling as the square of their perimeters. For $q < 1$ and $x > x_c(q)$, we expect that this phase can be described as a single convoluted polygon that 'fills' the whole lattice rather like a closed Hamiltonian path: one might describe it a a 'seaweed phase' [9].

Here we give main results about the singularity diagram (see Fig. 5):

- $q_c(x)$ is singular in $x = x_t$ thus we have a phase transition.
- $G(x,q)$ diverges at $q_c(x)$ for $x > x_t$.
- $G(x,q)$ is singular at $q_c(x) = 1$ for $x < x_t$.
- $G(x,1)$ is finite with singularity exponent γ_u as $x \to x_t$.
- $G(x_t,q)$ has a singularity with exponent γ_t as $q \to 1$.
- $(x_t,1)$ is a *tricritical point* with crossover exponent $\phi = \frac{\gamma_t}{\gamma_u}$.

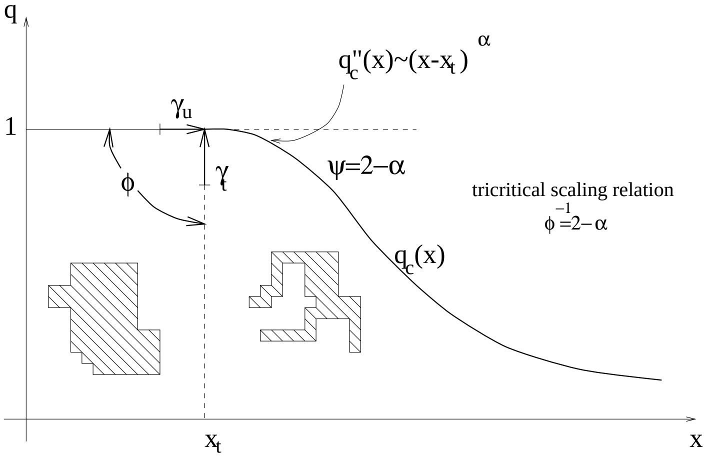


Figure 5. The singularity diagram.

- The scaling function f is:

$$G^{Sing}(x,q) \sim (1-q)^{-\gamma_t} f\left(\{1-q\}^{-\phi}\{x_t - x\}\right)$$

  with $f(z) \sim z^{-\gamma_u}$ as $z \to \infty$ and $f(z) \sim 1$ as $z \to 0$.
- The shape exponent is $\psi = \frac{1}{\phi}$ and $q_c(x) \sim 1 - a(x - x_t)^{\frac{1}{\psi}}$.

5. Partially convex polygons: a solvable model

The analysis of partially convex subsets of self-avoiding polygons confirms results of the previous section. These partially convex polygons form a universality class with the same crossover exponent as expected in the unrestricted problem. The particular models we consider are subsets of column-convex polygons: staircase polygons, directed column-convex polygons and column-convex polygons (see Fig. 6).

These models have been studied by a variety of methods:

- mapping to a q-extension of an algebraic language [8],
- recurrence relations [12, 5],
- linear functional equations [3, 2],
- transfer matrix techniques [1].

All these models possess the characteristic feature that their single-variable generating functions are algebraic, while the two-variable generating functions are expressed in term of quotients of q-series.

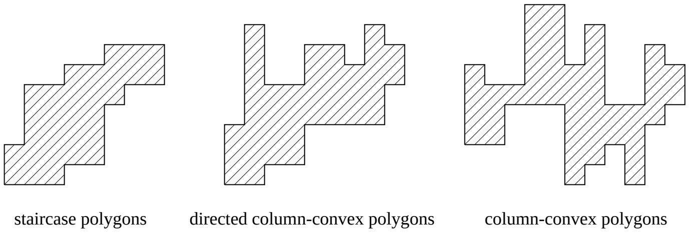
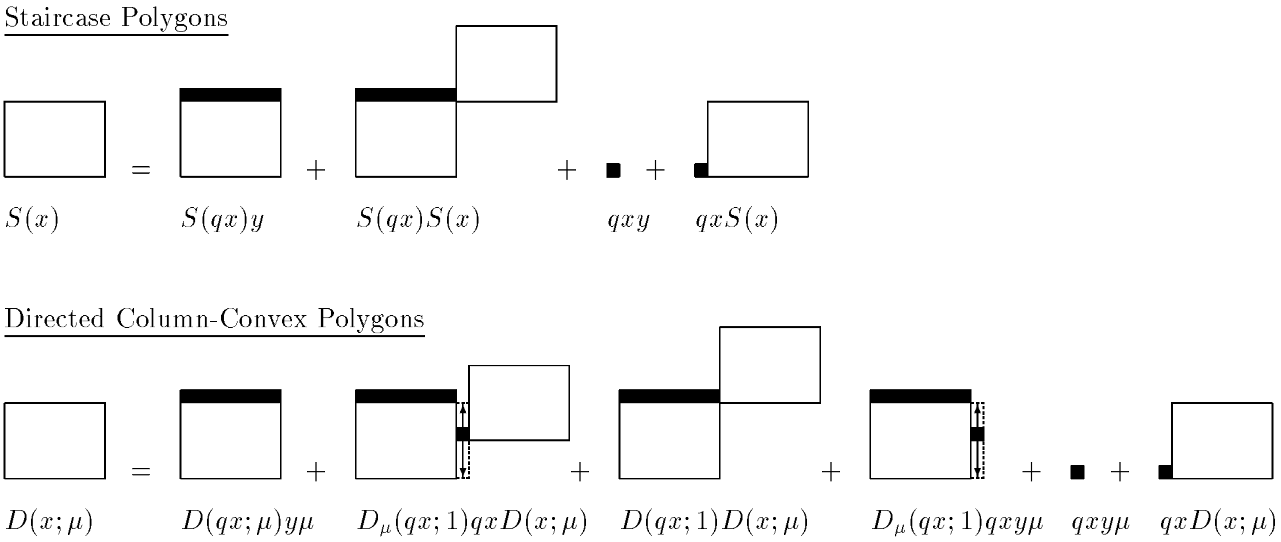

staircase polygons
directed column-convex polygons
column-convex polygons

Figure 6. Partially convex polygons.



Staircase Polygons

Directed Column-Convex Polygons

FIGURE 7. The diagrammatic form of the functional equations for staircase polygons and directed column-convex polygons.

We define the polygon generating function $G(x,y,q)$ to be

$$G(x,y,q) = \sum_{n_x,n_y,m} c_{n_x,n_y,m} x^{n_x} y^{n_y} q^m .$$

We derive the generating function for each models by using an inflation process [10, 11, 3]: the height of the polygon is increased by one lattice spacing and concatenated with rows of height one (see Fig. 7).

Denoting the generating function for the staircase polygons by $S(x,y,q)$, we therefore get immediately

$$S(x,y,q) = (S(qx,y,q) + qx)(y + S(x,y,q)).$$

In order to write down a functional equation for the column-convex polygons, we need to keep track of the height r of the rightmost column of these polygons. We define the generating function $D(x,y,q;\mu)$ to be

$$D(x,y,q;\mu) = \sum_{n_x,n_y,m,r} c_{n_x,n_y,m,r} x^{n_x} y^{n_y} q^m \mu^r .$$

If we denote $\frac{\partial}{\partial\mu} D(qx,y,q;\mu)\Big|_{\mu=1}$ by $D_\mu(qx,y,q;1)$, we get the following functional-differential equation:

$$\begin{aligned} D(x,y,q;\mu) = (1 + D_\mu(qx,y,q;1))\,qx\,(y\mu + D(x,y,q;\mu)) \\ + D(qx,y,q;\mu)y\mu + D(qx,y,q;1)D(x,y,q;\mu). \end{aligned}$$

We can transform this equation to one functional equation in $D(x) = D(x,y,q;1)$ by partially differentiating with respect to μ and setting $\mu = 1$. This leads to

$$\begin{aligned} 0 = D(q^2x)D(qx)D(x) + yD(q^2x)D(qx) + yD(q^2x)D(x) - (1+q)D(qx)D(x) + y^2D(q^2x) \\ - y(1+q)D(qx) + q(1 + qx(y-1))D(x) + yq^2x(y-1). \end{aligned}$$

Setting $q = 1$ gives the perimeter generating function which satisfies a cubic equation and has a square-root singularity at

$$y_c = \frac{\sqrt[3]{100} - 4}{3} \qquad \text{for} \qquad x = y$$

implying that $\gamma_u = -\frac{1}{2}$.

First we note that the functional equation for staircase polygons is of the form

$$G(x)G(qx) + a(x)G(x) + b(x)G(qx) + c(x) = 0$$

which can be linearized by the use of the transformation

$$G(x) = \alpha \frac{H(qx)}{H(x)} - b(x)$$

where α has to be chosen to match the initial condition. This leads to a linear functional equation in $H(x)$,

$$\alpha^2 H(q^2 x) + \alpha[a(x) - b(qx)]H(qx) + [c(x) - a(x)b(x)]H(x) = 0.$$

Lemma 2. *The solution of*

$$0 = xH(qx) + \sum_{k=0}^{N} \alpha_k H(q^k x) \qquad \text{with} \qquad \sum_{k=0}^{N} \alpha_k = 0$$

regular at $x = 0$ is given by

$$H(x) = \sum_{n=0}^{\infty} \frac{(-x)^n q^{\binom{n}{2}}}{\prod_{m=1}^{n} \Lambda(q^m)} \qquad \text{with} \qquad \Lambda(t) = \sum_{k=0}^{N} \alpha_k t^k.$$

We apply lemma 2 to staircase polygons, we choose $\alpha = y$ and we get the solution

$$S(x) = y\left(\frac{T(qx)}{T(x)} - 1\right) \qquad \text{with} \qquad T(x) = \sum_{n=0}^{\infty} \frac{(-qx)^n q^{\binom{n}{2}}}{(q, qy; q)_n}.$$

Surprisingly, this works also for directed column-convex polygons:

$$D(x) = y\left(\frac{E(qx)}{E(x)} - 1\right) \qquad \text{with} \qquad E(x) = \sum_{n=0}^{\infty} \frac{((y-1)qx)^n q^{\binom{n}{2}}}{(q, qy, y; q)_n}.$$

M. Bousquet-Mélou [3] found by other means that for column-convex polygons

$$G(x, y, q) = y \frac{(1-y)A}{1 + B + yA}$$

where

$$A = \frac{xq}{(1-y)(1-yq)} + \sum_{n=2}^{\infty} \frac{(-1)^{n+1} x^n (1-y)^{2n-4} q^{\binom{n+1}{2}} (y^2 q; q)_{2n-2}}{(q;q)_{n-1}(yq;q)_{n-2}(yq;q)_{n-1}^2 (yq;q)_n (y^2 q;q)_{n-1}}$$

and

$$B = \sum_{n=1}^{\infty} \frac{(-1)^n x^n (1-y)^{2n-3} q^{\binom{n+1}{2}} (y^2 q; q)_{2n-1}}{(q;q)_n (yq;q)_{n-1}^3 (yq;q)_n (y^2 q;q)_{n-1}}.$$

In [11] we consider simpler models of partially convex polygons as stacks and Ferrers diagrams (see Fig. 8).

For stacks ($s = 2$) and Ferrers diagram ($s = 1$), we obtain a non-alternating q-series for the generating function

$$G_s(x, y, q) = \sum_{n=1}^{\infty} \frac{x(yq)^n}{(xq;q)_{n-1}^s (1 - xq^n)}$$

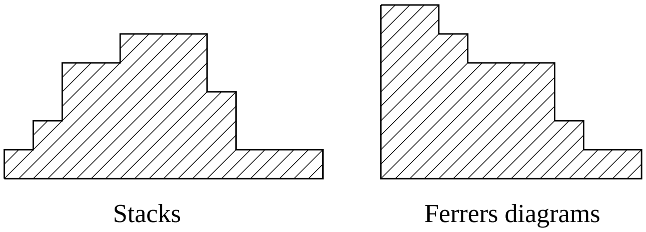


FIGURE 8. Typical configurations of stacks and Ferrers diagrams.

and a rational function for the perimeter-generating function

$$G_s(x,y,1) = \frac{xy(1-x)^{s-1}}{(1-x)^s - y}.$$

These models are interesting, as they show "pathological behavior". We have seen that considered as a function of x, the radius of convergence is a continuous function, while considered as a function of q, it has a jump discontinuity at $q = 1$ in the generic case for the vesicle models. But in the generic case we have left continuity at $x_c(1)$ whereas for stacks ($x_c(q) = 1/q$) there is an isolated point $x_c(1)$ at $q = 1$ ($x_c(1^-) = 1 > x_c(1) > x_c(1^+) = 0$). Thus stacks and Ferrers diagram are too simplified to give a reasonable physical model.

Bibliography

[1] Binder (P. M.), Owczarek (A. L.), Veal (A. R.), and Yeomans (J. M.). – Collapse transition in a simple polymer model : exact results. *Journal of Physics Series A*, vol. 23, 1990, p. L975.

[2] Bousquet-Mélou (Mireille). – Codage des polyominos convexes et équations pour l'énumération suivant l'aire. *Discrete Applied Mathematics*, vol. 48, 1994, pp. 21–43.

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

[4] Brak (R.), Enting (I. G.), and Guttmann (A. J.). – Exact solution of the row-convex polygon perimeter generating function. *Journal of Physics Series A*, vol. 23, 1990, pp. 2319–2326.

[5] Brak (R.) and Guttmann (A. J.). – Exact solution of staircase and row-convex polygon perimeter and area generating function. *Journal of Physics Series A*, vol. 23, 1990, pp. 4581–4588.

[6] Brak (R.), Owczarek (A. L.), and Prellberg (T.). – A scaling theory of the collapse transition in geometric cluster models of polymers and vesicles. *Journal of Physics Series A*, vol. 26, 1993, pp. 4565–4579.

[7] Brak (R.), Owczarek (A. L.), and Prellberg (T.). – Exact scaling behavior of partially convex vesicles. *Journal of Statistical Physics*, vol. 76, 1994, pp. 1101–1128.

[8] Delest (M. P.). – Generating functions for column-convex polyominoes. *Journal of Physics Series A*, vol. 48, 1988, pp. 12–31.

[9] Fisher (Michael E.), Guttmann (Anthony J.), and Whittington (Stuart G.). – Two-dimensional lattice vesicles and polygons. *Journal of Physics Series A*, vol. 24, 1991, pp. 3095–3106.

[10] Prellberg (T.) and Brak (R.). – Critical exponents from nonlinear functional equations for partially directed cluster models. *Journal of Statistical Physics*, vol. 78, 1995, pp. 701–730.

[11] Prellberg (Thomas) and Owczarek (Aleksander L.). – Staking models of vesicles and compact clusters. *Journal of Statistical Physics*, vol. 80, n° 3/4, 1995, pp. 755–779.

[12] Temperley (H. N. V.). – Combinatorial problems suggested by the statistical mechanics of domains and rubber-like molecules. *Physical Review*, vol. 103, n° 1, 1956, pp. 1–16.

Partitions of Integers: Asymptotics

Philippe Dumas
Projet Algorithmes, Inria Rocquencourt

December 11, 1995

[summary by Philippe Dumas and Bruno Salvy]

Abstract

The study of the asymptotics of the number of partitions of integers under various constraints is a very rich area initiated by two papers of Hardy and Ramanujan. Some of this literature is surveyed here.

If $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_\nu$ are positive integers, their sum $n = \lambda_1 + \lambda_2 + \cdots + \lambda_\nu$ is called a *partition* of n into ν summands (or parts). The number of partitions of n is denoted $p(n)$ or p_n. When there is no constraint on the λ_i, it is easy to see that the generating function of the p_n's satisfies the following identity due to Euler:

$$\mathcal{P}(q) = \sum_{n \geq 0} p_n q^n = \prod_{k>0} \frac{1}{1-q^k}. \tag{1}$$

Euler's pentagonal theorem also gives a formula for the reciprocal of this generating function:

$$\prod_{k>0}(1-q^k) = \sum_{m=-\infty}^{\infty} (-1)^m q^{m(3m-1)/2}.$$

This last relation yields a simple way to compute the number p_n by recurrence. Numerous other relations on partitions and their congruence properties can be derived from identities on generating functions. See in particular [1].

1. Origins

The asymptotic analysis of the generating function $\mathcal{P}(q)$ is very difficult. There are singularities at all roots of unity, which implies that the circle of convergence is a natural boundary. It can be proved that a saddle-point method applies. The coefficient p_n is given by the contour integral

$$p_n = \frac{1}{2i\pi} \int_\gamma \frac{\mathcal{P}(q)}{q^{n+1}}\, dq,$$

and the main contribution comes from the neighbourhood of 1, which yields

$$p_n \sim \frac{1}{4n\sqrt{3}} \exp\left(\pi\sqrt{\frac{2n}{3}}\right). \tag{2}$$

Then the next contribution comes from the neighbourhood of -1, then from the neighbourhood of $\exp(\pm 2i\pi/3)$, etc. Thus the contour of integration has to go through an infinity of saddle-points, whose contribution to the integral have to be estimated. It turns out that these contributions are related by a modular transform. For, the generating function $\mathcal{P}(q)$ is related to Dedekind's η function:

$$\eta(\tau) = e^{i\pi\tau/12} \prod_{m=1}^{\infty} (1 - e^{2i\pi m\tau}) = \frac{e^{i\pi\tau/12}}{\mathcal{P}(e^{i\pi\tau})}.$$

The final result is the following theorem [9].

Theorem 1. *The number $p(n)$ of partitions satisfies*

$$p(n) = \sum_{q=1}^{\nu} A_q \psi_q + O(n^{-1/4}),$$

where

$$\psi_q = \frac{\sqrt{q}}{2\pi\sqrt{2}} \frac{d}{dn} \left(\frac{\exp\left(\frac{\pi}{q}\sqrt{\frac{2}{3}}\sqrt{n - \frac{1}{24}}\right)}{\sqrt{n - \frac{1}{24}}} \right), \qquad A_q = \sum_{p \wedge q = 1, p \leq q} \omega_{p,q} e^{-2npi\pi/q},$$

and $\omega_{p,q}$ is a certain $24q$th root of unity.

This result is very precise: since the $O()$ term tends to 0 and the number $p(n)$ is an integer, it is sufficient to consider finitely many terms of this asymptotic expansion to compute the exact value of $p(n)$. In practice, the number of necessary terms is quite small. Theorem 1 has been refined by H. Rademacher [15] to obtain a full asymptotic expansion which is convergent. Other special types of partitions have been treated the same way. All these works rely on the theory of modular functions.

Wright followed the way opened by Hardy et Ramanujan in several works [20, 21, 22]. For instance, he studied the asymptotics of plane partitions, with generating function

$$\sum_{n \geq 0} p_{\text{plane}}(n)\, q^n = \prod_{\ell \geq 1} \frac{1}{(1 - q^\ell)^\ell}.$$

The result has the following form

$$p_{\text{plane}}(n) \sim \frac{K}{n^{25/36}} \exp\left(Cn^{2/3}\right),$$

which should be compared to (2) for ordinary partitions.

All these results are obtained by a saddle-point method combined with a Mellin transform.

2. Mahler's partition problem

In [12] Mahler studies the partitions whose summands are constrained to be powers of some integer $r \geq 2$. In that case, the generating function becomes

$$\prod_{k>0} \frac{1}{1 - q^{r^k}} = \sum p_r(n) q^n = \mathcal{P}_r(q).$$

Mahler computes an expansion of $\log p_r(n)$, whose error term is a $O(1)$. This expansion shows that $p_r(n)$ is essentially of order $\exp(\log^2 n / 2 \log r)$. The basic tool is a functional equation

$$\frac{f(z + \omega) - f(z)}{\omega} = f(qz), \qquad \text{with } q = 1/r.$$

The result was improved by de Bruijn [5], using a Mellin transform approach to the logarithm, followed with a saddle-point method. Besides, in de Bruijn's work, $r > 1$ can be any real number and Mahler's error term is expressed as the sum of an oscillating series. This oscillating behaviour is studied in more detail by Erdös and Richmond in [7, 16].

3. Saddle-point method

It it quite lucky that in the case of unrestricted ordinary partitions the whole computation provides an asymptotic convergent series. If one adds constraints on the summands of the partitions it is in general not possible anymore to derive a convergent asymptotic estimate of this form. In these cases, only the saddle-point close to 1 is considered and its contribution to the integral is often itself an infinite sum.

Meinardus [1, 13] gives some general conditions which ensure that the saddle-point method works. He considers a generating function

$$\prod_{k\geq 1}\frac{1}{(1-q^k)^{a_k}},$$

where the numbers a_n are real nonnegative, and the conditions concern the Dirichlet series $D(s) = \sum_{k\geq 1} a_k/k^s$, which extends as a meromorphic function to the left of its abscissa of convergence.

Roth and Szekeres [18] study a generating function

$$\prod_{k\geq 1}(1+q^{\lambda_k}).$$

They assume the limit $s = \lim_{k\to\infty}\log\lambda_k/\log k$ exists, and use some arithmetical conditions on the summands λ_k. Their result was extended by Richmond [17], who gives several sets of conditions. As an example, Roth and Szekeres give the following expansion for the number of partitions into distinct prime summands,

$$\log q_{\mathrm{prime}}(n) = \pi\sqrt{\frac{2}{3}}\sqrt{\frac{n}{\log n}}\left(1+O\left(\frac{\log\log n}{\log n}\right)\right).$$

The works of Meinardus and of Roth and Szekeres use the saddle-point method. The differences between them is rather a matter of style. Meinardus studies the behaviour of the generating function in the neighborhood of 1 using a Mellin transform; this gives an approximate saddle-point equation and an approximate saddle-point; next the Cauchy integral is studied. Roth and Szekeres directly use the saddle-point method and their result is expressed in an implicit manner; every application needs an auxiliary computation, in some cases with the Euler-McLaurin formula or with the Mellin transform, to obtain an explicit expansion.

4. Tauberian method

In [10], Ingham asks for a set of conditions *not highly extravagant* which leads to a result about the asymptotic behaviour of the number of partitions. He considers a sequence of real numbers $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_k < \cdots$ and its count function $\Lambda(u) = |\{\lambda_k; \lambda_k \leq u\}|$. The use of this function is natural because the generating function

$$\mathcal{P}(e^{-s}) = \prod_{k\geq 1}\frac{1}{1-e^{\lambda_k s}} = \sum_{\ell} p(\ell)e^{-\ell s}$$

and the count function are related by

$$\log\mathcal{P}(e^{-s}) = \int_0^{+\infty}\log\frac{1}{1-e^{-su}}\,d\Lambda(u).$$

Under the hypothesis

$$\Lambda(u) = Bu^\beta + R(u), \qquad \int_0^u \frac{R(v)}{v}\,dv \underset{u\to\infty}{=} b\log u + c + o(1),$$

he proves that

$$P_h(u) \sim Ku^{(a-1/2)(1-\alpha)-1/2}\exp(Cu^\alpha), \qquad \text{with } \alpha = \beta/(\beta+1),$$

for some explicit constants K and C. Here the function $P_h(u)$ generalises the function $p(n)$ we used previously; precisely, if $P(u)$ is the number of solutions in nonnegative integers of the inequation $n_1\lambda_1 + n_2\lambda_2 + \cdots + n_r\lambda_r + \cdots < u$ then for positive h, $P_h(u) = [P(u+h) - P(h)]/h$. Hence if $h = 1$ and the summands λ_k are integers, $P_h(n)$ is simply $p(n)$. The function $P(u)$ already appears in the work of Mahler, because it satisfies the equations $p(rm) = P(m+0)$ and $P(u) - P(u-1) = P(u/r)$ in that case.

The proof relies on a special Tauberian theorem. Indeed, the generating function appears to be a Laplace transform,

$$\mathcal{P}(e^{-s}) = \int_0^\infty e^{-su}\,dP(u).$$

The Tauberian theorem of Ingham provides an estimate of $P(u)$ in terms of $\phi(s) = \log\mathcal{P}(e^{-s})$ and the solution σ_u of the equation $\phi'(\sigma_u) + u = 0$ (which can be seen as a saddle-point equation).

The proof of Ingham works for $P(u)$ without any further condition, but for $P_h(u)$ one of the hypotheses is the monotonicity of this function. Auluck and Haselgrove [2] have extended the result of Ingham, and removed some of his hypotheses. Bateman and Erdös [3] have shown that for integer summands λ_k the function $p(n) = P_1(n)$ is monotonic if and only the set of summands has the following property: there are at least two λ's and if one removes any λ the remaining λ's have greatest common divisor unity.

5. Weak results

Hardy and Ramanujan [8] study the number $Q(x)$ of solutions of the inequation

$$2^{a_2}3^{a_3}5^{a_5}\cdots p^{a_p}\cdots \le x$$

into integers satisfying $a_2 \ge a_3 \ge \cdots \ge a_p \ge \cdots$. The numbers 2, 3, …, p, … are the prime numbers. If λ_k is the sum of the logarithms of the k first prime numbers, $Q(x)$ is essentially $P(\log x)$. They prove that

$$\log Q(x) \underset{x\to\infty}{=} \frac{2\pi}{\sqrt{3}}\sqrt{\frac{\log x}{\log\log x}} + o(1).$$

Such a result, which gives an equivalent of $\log P(u)$, is called a weak result.

The tools used by Hardy and Ramanujan is a Tauberian theorem; under some simple conditions this theorem says that

$$\log A_n \underset{n\to\infty}{=} B\ell_n^{\alpha/(1+\alpha)}/\log^{\beta/(1+\alpha)}\ell_n$$

if the behaviour near 0 of the logarithm of the Laplace transform

$$f(s) = \sum_{n\ge1} a_n e^{-\ell_n s} = \int_0^\infty e^{-su}\,dA(s)$$

is known, namely

$$\log f(s) \underset{s\to0}{=} \frac{A}{s^\alpha\log^\beta(1/s)}.$$

In these formulæ A_n is the summatory function

$$A_n = a_1 + a_2 + \cdots + a_n, \qquad A(x) = A_n \quad \text{for } n \le x < n+1.$$

The result is applied to the generating function

$$\mathcal{P}(e^{-s}) = \prod_{k\ge 1} \frac{1}{1 - e^{-\lambda_k s}},$$

which satisfies

$$\log \mathcal{P}(e^{-s}) \underset{s\to 0}{=} \frac{\pi^2}{6s\log(1/s)},$$

with $\ell_n = \log n$.

Brigham [4] extends the work of Hardy and Ramanujan, by considering the generating function

$$\mathcal{P}(e^{-s}) = \prod_{k\ge 1} \frac{1}{(1 - e^{-ks})^{\gamma_k}},$$

and the following hypothesis about the count function

$$\Lambda(u) = \sum_{k\le u} \gamma_k \underset{u\to\infty}{\sim} K u^\alpha \log^\beta u, \qquad \alpha > 0.$$

Two students of Bateman, Kohlbecker [11] first, and Parameswaran [14] next, consider the functional relation between the count function $\Lambda(u)$ and the summatory function $P(u)$,

$$\log \int_0^\infty e^{-su}\, dP(u) = \int_0^\infty \log \frac{1}{1 - e^{-su}}\, d\Lambda(u).$$

Kohlbecker shows the following behaviours are equivalent

$$\Lambda(u) \sim u^\alpha L(u), \qquad \log P(u) \sim u^{\alpha/(1+\alpha)} L^*(u), \qquad (\alpha > 0).$$

The function $L(u)$ and $L^*(u)$ are slowly varying, that is $L(cu) \sim L(u)$ for every $c > 0$. Moreover (L, L^*) is a dual pair; in every concrete case, $L^*(u)$ is explicitely computable from $L(u)$. The way from $P(u)$ to $\Lambda(u)$ is an Abelian theorem, and the way form $\Lambda(u)$ to $P(u)$ is a Tauberian theorem, like in the work of Hardy and Ramanujan.

Schwarz [19] gives a result which is surprising by its simplicity. The count function $\Lambda(u)$ tends to infinity (as we assumed in all preceding assertions) and satisfies $\Lambda(2u) = O(\Lambda(u))$ as $u \to \infty$. Under this hypothesis the behaviour of $\log P(u)$ is given by

$$\log P(u) \underset{u\to\infty}{=} \phi(\sigma_u) + u\sigma_u + O\left(u\sigma_u \sqrt{\psi(\sigma_u) \log \frac{1}{\psi(\sigma_u)}} \right),$$

where σ_u is the solution of the equation $\phi(\sigma) + u = 0$ for u large, and

$$\phi(\sigma) = \sum_{k\ge 1} \log \frac{1}{1 - e^{\lambda_k \sigma}}, \qquad \psi(\sigma) = \frac{\phi''(\sigma)}{|\phi'(\sigma)|^2}.$$

Schwarz gives a host of examples: ordinary partitions, $\lambda_k = k$, $\Lambda(u) \sim u$; partitions into prime numbers, $\lambda_k = p_k$, $\Lambda(u) \sim u/\log u$; partitions into rth powers, $\lambda_k = k^r$, $\Lambda(u) \sim u^{1/r}$; Mahler partitions, $\lambda_k = r^k$, $\Lambda(u) \sim \log_r u$; partitions whose summands are $\lambda_k = k^k$ or $k!$, $\Lambda(u) \sim \log u/\log\log u$, for example.

Conclusion

There is a wealth of papers on this subject. Parameters of partitions such as the number of summands can also be treated by the same kind of subject, although the computations are generally more technical. This is the problem that started Ph. Dumas in this domain, see [6] for details.

Bibliography

[1] Andrews (George E.). – *The Theory of Partitions.* – Addison-Wesley, 1976, *Encyclopedia of Mathematics and its Applications*, vol. 2.

[2] Auluck (F. C.) and Haselgrove (C. B.). – On Ingham's Tauberian theorem for partitions. *Proceedings of the Cambridge Philosophical Society*, vol. 48, 1952, pp. 566–570.

[3] Bateman (P. T.) and Erdös (P.). – Monotonicity of partition functions. *Mathematika*, vol. 3, 1956, pp. 1–14.

[4] Brigham (Nelson A.). – A general asymptotic formula for partition functions. *Proceedings of the American Mathematical Society*, vol. 1, 1950, pp. 182–191.

[5] De Bruijn (N. G.). – On Mahler's partition problem. *Indagationes Math.*, vol. 10, 1948, pp. 210–220. – Reprinted from Koninklijke Nederlandsche Akademie van Wetenschappen, Ser. A.

[6] Dumas (Ph.). – The number of summands in a binary partition. – To appear.

[7] Erdös (P.) and Richmond (B.). – Concerning periodicity in the asymptotic behaviour of partition functions. *Journal of the Australian Mathematical Society*, vol. 21, 1976, pp. 447–456.

[8] Hardy (G. H.) and Ramanujan (S.). – Asymptotic formulæ for the distribution of integers of various types. *Proceedings of the London Mathematical Society, Series 2*, vol. 16, 1918, pp. 112–132.

[9] Hardy (G. H.) and Ramanujan (S.). – Asymptotic formulæ in combinatory analysis. *Proceedings of the London Mathematical Society, Series 2*, vol. 17, 1918, pp. 75–115.

[10] Ingham (A. E.). – A Tauberian theorem for partitions. *Annals of Mathematics*, vol. 42, n° 5, 1941, pp. 1075–1090.

[11] Kohlbecker (Eugene E.). – Weak asymptotic properties of partitions. *Transactions of the American Mathematical Society*, vol. 88, 1958, pp. 346–365.

[12] Mahler (K.). – On a special functional equation. *Journal of the London Mathematical Society*, vol. 15, 1940, pp. 115–123.

[13] Meinardus (Günter). – Asymptotische Aussagen über Partitionen. *Mathematische Zeitschrift*, vol. 59, 1954, pp. 388–398.

[14] Parameswaran (S.). – Partition functions whose logarithms are slowly oscillating. *Transactions of the American Mathematical Society*, vol. 100, 1961, pp. 217–240.

[15] Rademacher (Hans). – On the partition function $p(n)$. *Proceedings of the London Mathematical Society, Series 2*, vol. 43, 1937, pp. 241–254.

[16] Richmond (Bruce). – Mahler's partition problem. *Ars Combinatoria*, vol. 2, 1976, pp. 169–189.

[17] Richmond (L. B.). – Asymptotic relations for partitions. *Journal of Number Theory*, vol. 7, 1975, pp. 389–405.

[18] Roth (K. F.) and Szekeres (G.). – Some asymptotic formulæ in the theory of partitions. *Quarterly Journal of Mathematics, Oxford Series*, vol. 5, 1954, pp. 241–259.

[19] Schwarz (Wolfgang). – Schwache asymptotische Eigenschaften von Partitionen. *Journal für die reine und angewandte Mathematik*, vol. 232, 1968, pp. 1–16.

[20] Wright (E. Maitland). – Asymptotic partition formulæ I. Plane partitions. *Quarterly Journal of Mathematics, Oxford Series*, vol. II, 1931, pp. 177–189.

[21] Wright (E. Maitland). – Asymptotic partition formulæ: (II) Weighted partitions. *Proceedings of the London Mathematical Society, Series 2*, vol. 36, 1934, pp. 117–141.

[22] Wright (E. Maitland). – Asymptotic partition formulæ: (III) Partitions into k-th powers. *Acta Arithmetica*, vol. 63, n° 141–191, 1934.

Measures of distinctness for summands in partitions and compositions

Hsien-Kuei Hwang
Academia Sinica, Taiwan

May 6, 1996

[summary by Philippe Dumas]

Abstract

Statistical properties of integer partitions and compositions are studied. The approach is based on generating functions and complex analysis, and uses Mellin transform.

The problem under treatment is mainly based on a work by Richmond and Knopfmacher [4], who considered compositions with distinct summands. It is also based on a work by Knopfmacher and Mays [2], who studied the number and the sum of distinct summands in compositions by elementary means. The approach of Hwang and Yeh [1] is different. It is based on generating functions and complex analysis, which allows them to consider a general scheme: the summands are taken form an infinite positive integer sequence (λ_j), and various types of partitions or compositions, inspired from combinatorial data structures, are studied.

There are different ways to estimate the degree of distinctness between the summands of a partition or of a composition. In this summary we content ourselves with the number of summands which occur h times or more in a partition or composition, though Hwang and Yeh consider many other criteria. This number may be viewed as a random variable $X_n^{[h]}$ indexed by the sum n of the partition or composition. In the case of compositions, the formula

$$\sum_{n\geq 1} c_n \operatorname{E}(X_n^{[h]}) = \sum_{j\geq 1} \frac{z^{h\lambda_j}}{(1-\Lambda(z))(1-\Lambda(z)+z^{\lambda_j})^h},$$

where c_n is the number of compositions of n and $\Lambda(z)=\sum_j z^{\lambda_j}$, provides a way to determine the asymptotic behavior of the mean $\operatorname{E}(X_n^{[h]})$.

Let us consider the simple case $\lambda_j = j$; so that $c_n = 2^{n-1}$. We have

$$\operatorname{E}(X_n^{[1]}) = \log_2 n - \frac{3}{2} + \frac{\gamma}{\log 2} - \frac{1}{2}\sum_{k\neq 0}\Gamma(\chi_k)n^{-\chi_k} + O\left(\frac{\log n}{n}\right),$$

with $\chi_k = 2ik\pi/n$. The proof is in four steps and relies on the formula

$$\operatorname{E}(X_n^{[1]}) = \sum_{j=1}^{n}\left(1 - \frac{1}{2^{n-1}}[z^n]\frac{1-z}{1-2z+z^j(1-z)}\right).$$

First, Rouché's theorem implies that the polynomial $1-2z+z^j(1-z)$ has only one root $(1+\varepsilon_j)/2$ inside the unit circle. Next the Lagrange inversion theorem gives an explicit expression of ε_j, namely

$$\varepsilon_j = \sum_{\ell\geq 1}\frac{1}{2^{(j+1)\ell}\ell}\sum_{i=0}^{\ell-1}\binom{\ell}{i+1}(-1)^{\ell-i-1}\binom{k\ell}{i}.$$

The singularities of the generating function being known, the next stage is an application of Cauchy's formula. One obtains

$$\mathrm{E}(X_n^{[1]}) = \sum_{j=1}^{n} \left(1 - \frac{1}{1+\varepsilon_j^n}\right) + O\left(\frac{\log n}{n}\right).$$

This new sum is a harmonic sum which can be expressed as an inverse Mellin transform, hence

$$\mathrm{E}(X_n^{[1]}) = \frac{-1}{2i\pi} \int_{-1/2-i\infty}^{-1/2+i\infty} \Gamma(s) n^{-s} U(s)\, ds + O\left(\frac{\log n}{n}\right),$$

with

$$U(s) = \begin{cases} \sum_{j\geq 1} \log(1+\varepsilon_j)^{-s}, & \Re(s) < 0, \\ 4^s/(1-2^s) + V(s), & \Re(s) < 1. \end{cases}$$

This provides the announced formula. The analysis differs from Knuth's one [3] and gives a better error term. The big oh term may be replaced by a sum

$$\sum_{k\geq 1} \frac{1}{n^k} \sum_{\ell=0}^{k} \varpi_{k,\ell}(\log_2 n) \log^\ell n,$$

where the $\varpi_{k,\ell}$s are periodic functions. More generally one obtains

$$\mathrm{E}(X_n^{[h]}) = \mathrm{E}(X_n^{[1]}) - \sum_{j=1}^{h-1} \frac{1}{j! \log 2} \left(1 + \sum_{k\neq 0} \Gamma(j+\chi_k) n^{-\chi_k}\right) + O\left(\frac{\log n}{n}\right).$$

All this is relative to the case $\lambda_j = j$.

In the general case $\Lambda(z) = \sum_j z^{\lambda_j}$ cannot be written as $z^a \Lambda_1(z^d)$ with $d \geq 2$ and the count function

$$A(x) = \sum_{\lambda_j \leq x} 1$$

tends to infinity with x. Under these conditions, one obtains

$$\mathrm{E}(X_n^{[h]}) = A(\log_r(cn)) + O(1),$$

where r and c are defined by $\Lambda(\rho) = 1$, $r = 1/\rho$ end $c = 1/\rho/\Lambda'(\rho)$. One may say there is a logarithmic transition from the behavior of A to the behavior of $\mathrm{E}(X_n^{[h]})$.

Hwang and Yeh consider others compositions like cyclic compositions where compositions are considered up to circular permutation, or branching compositions where the summands label the nodes of a binary tree.

Bibliography

[1] Hwang (H.-K.) and Yeh (Y.-N.). – Measures of distinctness for partitions and compositions of integers. – in preparation.

[2] Knopfmacher (A.) and Mayes (M. E.). – Compositions with m distinct parts. *Ars Combinatorica*, 1996. – To appear.

[3] Knuth (D. E.). – The average time for carry propagation. *Indagationes Mathematicae*, vol. 40, 1978, pp. 238–242.

[4] Richmond (L. B.) and Knopfmacher (A.). – Compositions with distinct parts. *Aequationes Mathematicae*, vol. 49, 1995, pp. 86–97.

Asymptotics and scalings for large product-form networks via the Central limit theorem

Jean-Marc Lasgouttes
INRIA Rocquencourt

May 6, 1996

[summary by Philippe Robert]

1. Introduction

This talk considers the following closed queueing networks: there are n queues and m_n customers traveling in the network, the service rate at queue k when there are q_k customers is $\mu_{k,n}(q_k)$. A customer finishing his service at queue k goes to queue l with probability $p_{k,l}$ where $P_n = (p_{k,l})$ is an irreducible stochastic matrix with invariant measure $\pi_n = (\pi_{1,n}, \ldots, \pi_{n,n})$, defined by $\pi_n P_n = \pi_n$ and $\pi_{1,n} + \cdots + \pi_{n,n} = 1$. The service discipline can be FIFO, LIFO, or Processor sharing. To this network is associated a Markov process given by the vector of the number of customers in the queues $(Q_{k,n})$. It is well known that this Markov process has a unique equilibrium measure P_n such that, if $q_1, \ldots, q_n \geq 0$ and $q_1 + \cdots + q_n = m_n$,

$$P_n(Q_{1,n} = q_1, \ldots, Q_{n,n} = q_n) = Z_{m_n,n}^{-1} \prod_{k=1}^{n} \frac{\pi_{k,n}^{q_k}}{\mu_{k,n}(1) \cdots \mu_{k,n}(q_k)},$$

with the normalizing condition

$$Z_{m,n} = \sum_{q_1 + \cdots + q_n = m} \prod_{k=1}^{n} \frac{\pi_{k,n}^{q_k}}{\mu_{k,n}(1) \cdots \mu_{k,n}(q_k)}.$$

The explicit expression for the equilibrium measure is not really informative because of the normalizing constant which is not easy to handle. It is difficult to get a qualitative insight on the network (such as the mean queue lengths and their variances). A way to cope with this problem is to consider asymptotics. The paper considers the case where the number of queues and the number of customers tend to infinity with some normalization between them.

2. The equivalent network

The main idea is to introduce the open network defined by n independent parallel queues with service rate $\mu_{k,n}(x)$ and arrival intensity $\lambda_n \pi_{k,n}$ at queue k.

The distribution of the number $X_{k,n}$ of clients in queue k is given by

$$P(X_{k,n} = x) = \frac{1}{f_{k,n}} \frac{(\lambda_n \pi_{k,n})^x}{\mu_{k,n}(1) \cdots \mu_{k,n}(x)}$$

where $f_{k,n}$ is a (simple) normalizing constant.

Theorem 1. *For any choice of m_n, there exists a unique λ_n such that if $S_n = X_{1,n} + \cdots + X_{n,n}$, then $E(S_n) = m_n$. In this case for any $q_1, \ldots, q_n \geq 0$ and $1 \leq \ell \leq n$,*

$$P(Q_{1,n} = q_1, \ldots, Q_{n,n} = q_n) = \frac{1}{P(S_n = m_n)} \prod_{k=1}^{n} P(X_{k,n} = q_k),$$

$$P(Q_{1,n} = q_1, \ldots, Q_{\ell,n} = q_\ell) = \prod_{k=1}^{\ell} P(X_{k,n} = q_k) \frac{P(\sum_{k=\ell+1}^{n} X_{k,n} \sum_{k=\ell+1}^{n} m_{k,n})}{P(S_n = m_n)}$$

$$\times P(X_{1,n} = q_1, \ldots, X_{\ell,n} = q_\ell | S_n = m_n).$$

Starting from this representation, the asymptotic results concerning the network are proved via asymptotic results on S_n. Basically, in the same way as Kolchin [2] in another context, the authors use local limit theorems of the following form.

Theorem 2. *Under "suitable" conditions, there exists a distribution with density h and a sequence a_n such that, for any integer x,* $\lim_{n\to\infty} a_n P(S_n - m_n = x) - h(x/a_n) = 0$.

3. Asymptotic expansions

The queues are partitioned into two sets, F_n and I_n. The set F_n contains those queues k for which $\liminf_{q\to\infty} \sqrt[q]{\mu_{k,n}(1)\cdots\mu_{k,n}(q)} < \infty$; the set I_n contains the other ones.

Definition 1. Let

$$\mu_{k,n} = \begin{cases} \liminf_{q\to\infty} \sqrt[q]{\mu_{k,n}(1)\cdots\mu_{k,n}(q)}, & \text{if } k \in F_n, \\ \mu_{k,n}(1), & \text{if } k \in I_n, \end{cases} \qquad \text{and} \qquad \lambda_n^0 = \min_{k\in F_n} \frac{\mu_{k,n}}{\pi_{k,n}}.$$

A sequence m_n^0 is said to be *weakly critical*, if for any $0 < t < 1$, $g(t) = \limsup_{n\to\infty} m_n(t\lambda_n^0)/m_n^0$ exists and $\lim_{t\to 1^-} g(t)$ be either 1 or ∞.

The *critical sequences* m_n^0 allow to distinguish between saturated and non-saturated regimes of the network, depending on the limit of $g(t)$ at 1. One of the main results on the asymptotic expansion of the equilibrium measure is the following theorem.

Theorem 3. *Assume* $\lim_{n\to\infty} \max_{1\leq k\leq n} (\pi_{k,n}/\mu_{k,n})/[\pi_{1,n}/\mu_{1,n} + \cdots + \pi_{n,n}/\mu_{n,n}] = 0$. *Let m_n^0 be a weakly critical sequence, with the associated function $g(t)$. Assume that* $\lim_{t\to 1^-} g(t) = 1$. *If moreover,* $\limsup_{n\to\infty} m_n/m_n^0 < 1$ *then, for any finite index j,*

$$P(Q_{1,n} = q_1, \ldots, Q_{j,n} = q_j) = \prod_{k=1}^{j} P(X_{k,n} = q_k) \left[1 + O\left(\frac{1}{m_n}\right)\right].$$

In particular, for all $k \in \{1, \ldots, n\}$, $E(Q_{k,n})$ is uniformly bounded in n.

Bibliography

[1] Fayolle (Guy) and Lasgouttes (Jean-Marc). – *Asymptotics and Scalings for Large Closed Product-form Networks via the Central Limit Theorem.* – Technical Report n° 2754, Institut National de Recherche en Informatique et en Automatique, 1996.

[2] Kolchin (V. F.). – *Random Mappings.* – Optimization Software, New York, 1986. Translated from *Slučajnye Otobraženija*, Nauka, Moscow, 1984.

Part 4

Analysis of Algorithms and Data Structures

Analysis of Quickselect

Helmut Prodinger
Technical University of Vienna

October 16, 1995

[summary by Bruno Salvy]

Abstract

Quickselect is an algorithm due to Hoare which uses the same partitioning process as Quicksort. As in Quicksort, there is a median-of-three version which reduces the number of comparisons and passes. This is analyzed as well as a variant called multiple Quickselect. All these analyses result in explicit expressions for the number of passes and comparisons.

Quicksort and Quickselect work as follows. The input is an array of n elements. First, one of these elements—the pivot—is selected at random. Then partitioning takes place: the array is rearranged so that its elements smaller than the pivot end up to the left of it, while the elements larger than the pivot end up to the right (see Fig. 1). It is an important hypothesis for the analysis that this partitioning should be *stable*, i.e. the order of the smaller elements and the order of the larger elements should not have been modified during the partitioning. In the next step, Quicksort and Quickselect differ. In Quicksort, whose aim is to sort the array, the same process is applied recursively to both sides of the array. In Quickselect, whose aim is to find the jth element of the array, the process is applied recursively to the side containing it.

In the case of Quicksort, the number of passes and the number of comparisons satisfy recurrences from which follow explicit formulæ in terms of the harmonic numbers $H_n = \sum_{k=1}^{n} 1/k$ [5].

A classical optimization of Quicksort is obtained by selecting the pivot by a median-of-three process: three elements of the array are selected at random, and the pivot is taken to be the median one. The analysis of this optimization is well-known [3, 2]. In [4], the analysis of Quickselect with this optimization is carried out. The same technique is applied to multiple Quickselect in [7]. We now summarize these works.

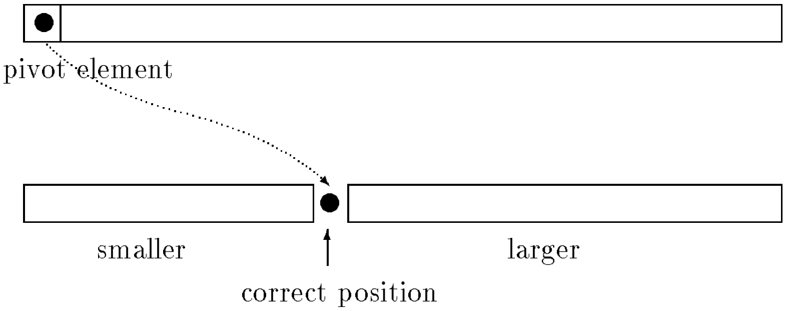


FIGURE 1. The partioning process

1. Number of passes and comparisons

After the pivot has been selected by the median of three process, the probability that the partitioning yields two sub-arrays of sizes $(k-1)$ and $(n-k)$ is

$$\pi_{n,k} = \frac{(k-1)(n-k)}{\binom{n}{3}}.$$

Let $F_{n,j}(z)$ denote the probability generating function of the number of passes necessary to select the jth element out of n under the assumption that all $n!$ permutations of the array are equally likely. Then by a simple generating function argument

$$F_{n,j}(z) = z\left[\sum_{k=1}^{j-1} \pi_{n,k} F_{n-k,j-k}(z) + \pi_{n,j} + \sum_{k=j+1}^{n} \pi_{n,k} F_{k-1,j}(z)\right] \tag{1}$$

for $n \geq 3$ while $F_{1,1}(z) = F_{2,1}(z) = F_{2,2}(z) = z$. The expected number of passes is obtained as $P_{n,j} = F'_{n,j}(1)$ and the generating function $P_j(z) = \sum_{n \geq j} P_{n,j} z^n$ satisfies the following mixed shift-differential equation derived from (1):

$$\frac{1}{6}P_j'''(z) = \frac{1}{(1-z)^4} - \sum_{k=3}^{j-1}\binom{k}{3}z^{k-3} + \sum_{k=2}^{j-1}(k-1)z^{k-2}P'_{j-k}(z) + \frac{P'_j(z)}{(1-z)^2}. \tag{2}$$

Since this is really an equation in P'_j, it is convenient to set $D_j = P'_j$. Then, with the help of Maple, it is possible to find closed-form formulæ for $D_1(z)$, $D_2(z)$, etc. All these functions are linear combinations of $(1-z)^{-2}\log(1-z)$, $\log(1-z)$, $(1-z)^{-2}$ and polynomials in z with simple rational coefficients. It is possible to spot patterns in these coefficients and this suggests studying the bivariate generating function $D(z,u)$ of the $D_j(z)$. From (2), it follows that $D(z,u)$ satisfies a linear differential equation:

$$\frac{1}{6}\frac{\partial^2 D}{\partial z^2} - \left(\frac{1}{(1-z)^2} + \frac{u^2}{(1-uz)^2}\right)D = \frac{u}{1-u}\left(\frac{1}{(1-z)^4} - \frac{u^3}{(1-uz)^4}\right),$$

with initial conditions $D(0,u) = u$, $D'_z(0,u) = 2u(1+u)$. This equation turns out to have a (several pages long) closed-form solution involving the logarithms of $(1-uz)$ and $(1-z)$ and rational functions in u and z. Extracting the coefficients then yields the following theorem.

Theorem 1. *Given a random permutation of n elements and $5 \leq j \leq n-4$, the average* number of passes *needed to select the jth element using Quickselect with a median-of-three partition is*

$$P_{n,j} = \frac{24}{35}H_n + \frac{18}{35}H_j + \frac{18}{35}H_{n+1-j} + \frac{12}{35j} + \frac{12}{35(n+1-j)} - \frac{304}{175} - \frac{6}{7n} + \frac{18j}{35n} - \frac{12(j-1)^2}{35n^{\underline{2}}}$$
$$- \frac{4(2j-3)(j-1)^{\underline{2}}}{35n^{\underline{3}}} - \frac{6(j-2)(j-1)^{\underline{3}}}{35n^{\underline{4}}} + \frac{6(2j-5)(j-1)^{\underline{4}}}{35n^{\underline{5}}} - \frac{4(j-3)(j-1)^{\underline{5}}}{35n^{\underline{6}}},$$

where $n^{\underline{k}} = n(n-1)\cdots(n-k+1)$.

For instance, to compute the median of $2n+1$ elements requires a number of passes $P_{2n+1,n+1} = \frac{24}{35}H_{2n+1} + \frac{36}{35}H_{n+1} + O(1) = \frac{12}{7}\log n + O(1)$ instead of $2\log n$ in the classical case. The savings are thus about 14%.

The number of comparisons is obtained in a similar fashion. In (1), it is sufficient to replace the factor z by z^{n-1} to obtain the generating function of the number of comparisons (at each pass, there are $n-1$ comparisons during the partitioning). Then again, the bivariate generating function of the number of comparisons to select the jth element out of a random permutation of n elements can be found explicitly, and extracting the coefficients yields the following theorem.

THEOREM 2. *Given a random permutation of n elements and $5 \leq j \leq n-4$, the average* number of comparisons *needed to select the jth element using Quickselect with a median-of-three partition is*

$$C_{n,j} = 2n + \frac{72}{35}H_n - \frac{156}{35}H_j - \frac{156}{35}H_{n+1-j} + \frac{36}{35j} + \frac{36}{35(n+1-j)} + \frac{88}{175} + \frac{24}{7n} + 3j - \frac{3(j-1)^{\underline{2}}}{n} - \frac{156j}{35n}$$
$$- \frac{36(j-1)^2}{35n^{\underline{2}}} - \frac{12(2j-3)(j-1)^{\underline{2}}}{35n^{\underline{3}}} - \frac{18(j-2)(j-1)^{\underline{3}}}{35n^{\underline{4}}} + \frac{18(2j-5)(j-1)^{\underline{4}}}{35n^{\underline{5}}} - \frac{12(j-3)(j-1)^{\underline{5}}}{35n^{\underline{6}}},$$

where $n^{\underline{k}} = n(n-1)\cdots(n-k+1)$.

Computation of the median therefore requires $11n/2 + O(\log n)$ comparisons whereas the classical method requires $4(1+\log 2)n + O(\log n)$ comparisons. The savings are thus about 19%.

The same technique also applies to several variants, such as counting only $n-3$ comparisons per partition or selecting the smaller of two random elements as the pivot.

2. Multiple Quickselect

In *multiple Quickselect*, one searches simultaneously for the elements of indices $\{j_1, \ldots, j_p\}$ $(0 < j_1 < \cdots < j_p \leq n)$. The analysis is very similar to the analyses above and results in *explicit* formulæ for the number of passes and the number of comparisons. With obvious notation, one has

$$\begin{aligned} P[n; j_1, \ldots, j_p] &= H_{j_1} + H_{n+1-j_p} + 2\sum_{t=2}^{p} H_{j_t+1-j_{t-1}} - 2p + 1, \\ C[n; j_1, \ldots, j_p] &= 2n + j_p - j_1 + 2(n+1)H_n - 2(j_1+2)H_{j_1} - 2(n+3-j_p)H_{n+1-j_p} \\ &\qquad - 2\sum_{t=2}^{p}(j_t + 4 - j_{t-1})H_{j_t+1-j_{t-1}} + 8p - 2. \end{aligned}$$

Of course, as a special case, we recover the analysis of Quicksort when $p = n$.

A recent work of Lent and Mahmoud [6] gives asymptotic estimates for so-called *grand averages*:

$$\mathcal{P}_{n,p} = \frac{1}{\binom{n}{p}} \sum_{1 \leq j_1 < \cdots < j_p \leq n} P[n; j_1, \ldots, j_p],$$
$$\mathcal{C}_{n,p} = \frac{1}{\binom{n}{p}} \sum_{1 \leq j_1 < \cdots < j_p \leq n} C[n; j_1, \ldots, j_p].$$

Using the formulæ above and summing the harmonic numbers by direct manipulations or standard generating function techniques [1], it is actually possible to derive closed-form formulæ for these averages in terms of harmonic numbers [7].

THEOREM 3.

$$\mathcal{P}_{n,p} = \frac{2p(n+1)^2}{(n+2-p)(n+1-p)}(H_{n+1} - H_p) + 1 - 2p - \frac{2(p-1)^2}{n+2-p},$$
$$\begin{aligned} \mathcal{C}_{n,p} = \frac{1}{(n+2-p)(n+1-p)} &\left[(2H_p+1)n^3 - 8pH_n n^2 + 4((p+2)H_p + p)n^2 \right. \\ &\left. + 2p(p-9)H_n n + (2(4p+5)H_p - 5p^2 + p - 1)n + 2p(p-5)H_n + 4(p+1)H_p - p(p+7)\right]. \end{aligned}$$

Bibliography

[1] Graham (R. L.), Knuth (D. E.), and Patashnik (O.). – *Concrete Mathematics.* – Addison Wesley, 1989.

[2] Greene (D. H.) and Knuth (D. E.). – *Mathematics for the analysis of algorithms.* – Birkhauser, Boston, 1981.

[3] Hennequin (Pascal). – Combinatorial analysis of quicksort algorithm. *RAIRO Theoretical Informatics and Applications*, vol. 23, n° 3, 1989, pp. 317–333.

[4] Kirschenhofer (Peter), Martínez (Conrado), and Prodinger (Helmut). – Analysis of Hoare's find algorithm with median-of-three partition. – 1995. Preprint.

[5] Knuth (Donald E.). – *The Art of Computer Programming.* – Addison-Wesley, 1973, vol. 3: Sorting and Searching.

[6] Lent (J.) and Mahmoud (H. M.). – Average-case analysis of multiple quickselect: An algorithm for finding order statistics. *Statistics and Probability Letters*, vol. 28, n° 4, August 1996, pp. 299–310.

[7] Prodinger (Helmut). – Multiple Quickselect – Hoare's Find algorithm for several elements. *Information Processing Letters*, vol. 56, n° 3, November 1995, pp. 123–129.

Basic hypergeometric series, digital search trees, and approximate counting

Helmut Prodinger
Technische Universitat Wien

October 16, 1995

[summary by Michèle Soria]

The transformation formula of Heine from the theory of basic hypergeometric functions allows very simple and pleasant derivations of explicit forms of the level polynomials of digital search trees [8], as well as of explicit forms of the probabilities in the "approximate counting" problem [7].

1. Basics about hypergeometric functions

This section contains basic notations and results about q-hypergeometric series (see e.g. [1, 3]).

q-Pochhammer symbol. Let us introduce the classical notations:

$$(a)_n = (1-a)(1-aq)\cdots(1-aq^{n-1}), \qquad (a)_0 = 1, \qquad (a)_\infty = \lim_{n\to\infty}(a)_n$$

and observe that

$$(1) \qquad (a)_n = \frac{(a)_\infty}{(aq^n)_\infty}$$

Cauchy's Formula.

$$\sum_{n\geq 0}\frac{(a)_n t^n}{(q)_n} = \frac{(at)_\infty}{(t)_\infty}$$

Euler's identities. The special case $a = 0$ is generally attributed to Euler:

$$\sum_{n\geq 0}\frac{t^n}{(q)_n} = \frac{1}{(t)_\infty}$$

and the so called Euler formula is obtained by first substituting a/b by a and bt by b in Cauchy's formula, then setting $a = -1$ and $b = 0$:

$$(-t)_\infty = \sum_{n\geq 0}\frac{q^{\binom{n}{2}}t^n}{(q)_n}.$$

Heine's transformation. Cauchy's formula and equation (1) lead to Heine's formula:

$$\sum_{n\geq 0} \frac{(a)_n(b)_n}{(q)_n(c)_n} t^n = \frac{(at)_\infty(b)_\infty}{(c)_\infty(t)_\infty} \sum_{n\geq 0} \frac{(c/b)_n(t)_n}{(q)_n(at)_n} b^n$$

Setting $a = q$, $b = y$, $c = 0$ and $t = z$ in Heine's transformation, one gets the simple formula

$$\sum_{n\geq 0} (y)_n z^n = (y)_\infty \sum_{n\geq 0} \frac{y^n}{(q)_n(1 - zq^n)}$$

2. Level polynomials in digital search trees

A digital search tree is constructed like a binary search tree, but the decision to go down to the left or right is done accordingly to the binary representation of the key: if the first bit is 0, the item goes left and otherwise it goes right; then the second bit is used to go down further left or right, etc., until there is an empty node where the item can be stored. In order to study the average search cost, we are interested in $h_{n,k}$, the expected number of nodes on level k (by convention, the root is at level 0), in a tree built from n random data (i.e. in every decision, a bit 0 or 1 is equally likely).

The level polynomial $H_n(u) = \sum_{k\geq 0} h_{n,k} u^k$ satisfies (see e.g. [5]) $H_0(u) = 0$, and for $n \geq 1$

$$H_n(u) = \sum_{k=1}^{n} \binom{n}{k} (-1)^{k-1} (u)_{k-1},$$

By probabilistic arguments, Louchard [6] gave an explicit formula for the coefficients of $H_n(u)$, that we shall derive here by means of hypergeometric functions. We introduce the bivariate generating function $H(u,x) = \sum_n H_n(u)x^n$ and obtain easily:

$$H(u,x) = \frac{x}{(1-x)^2} \sum_{k\geq 0} (u)_k \frac{x^k}{(x-1)^k}.$$

The use of Heine's formula gives

$$H(u,x) = \frac{x}{(1-x)^2} (u)_\infty \sum_{k\geq 0} \frac{u^k}{(q)_k \left(1 - \frac{x}{x-1} q^k\right)}.$$

Then decomposing into partial fractions and applying Euler's formula leads to

$$H(u,x) = \frac{(u)_\infty}{1-x} \frac{1}{(u/q)_\infty} - (u)_\infty \sum_{k\geq 0} \frac{(u/q)^k}{(q)_k} \frac{1}{1 - x(1-q^k)}.$$

From this expression we get

$$H_n(u) \equiv [x^n]H(u,x) = \frac{1}{1 - u/q} - (u)_\infty \sum_{k\geq 0} \frac{(u/q)^k}{(q)_k} (1-q^k)^n,$$

The coefficient of u^l in $H_n(u)$ then transforms by Euler's formula in

$$h_{n,k} \equiv [u^l]H_n(u) = q^{-l} - \sum_{k=0}^{l} \frac{q^{-k}}{(q)_k} (1-q^k)^n (-1)^{l-k} \frac{q^{\binom{l-k}{2}}}{(q)_{l-k}}. \tag{2}$$

Other parameters of interest, such as partial sums $([u^l]H_n(u)/(1-u))$ or leaf levels $([u^l]1 - (1 - u/q)H_n(u))$ can be obtained immediately from (2).

3. Approximate counting via Euler transform

Approximate counting can be described by an automaton with states 1, 2,... Starting in state 1, we proceed step by step. In one step we may either advance from state i to state $i+1$ with probability q^i, or stay in state i with probability $1-q^i$. The interesting parameter is the state reached after n random steps. The original analysis of this problem was done by Flajolet [2] and consists of an enumerative part and an asymptotic part. We will show here how hypergeometric functions allow some shortcuts in the enumerative part. Let $p_{n,l}$ be the probability to be in state l after n random steps, and let $H_l(x) = \sum_{n\geq 0} p_{n,l}x^n$. Using a decomposition path from 1 to l into stages, it is not hard to see that

$$H_l(x) = \frac{x^{l-1}q^{\binom{l}{2}}}{\prod_{i=1}^{l}(1-x(1-q^i))} = \frac{\frac{1}{x}\left(\frac{x}{1-x}\right)^l q^{\binom{l}{2}}}{\left(\frac{xq}{x-1}\right)_l}.$$

We shall go to the expected value after n steps by means of the bivariate generating function $H(x,y) = \sum_{l\geq 0} H_l(x)y^l$. Setting $z = \frac{x}{x-1}$ and applying Heine's formula, we get

$$H(x,y) = \frac{1}{x}\frac{(q)_\infty(yz)_\infty}{(qz)_\infty}\sum_{n\geq 0}\frac{(z)_n q^n}{(q)_n(yz)_n}.$$

One more Heine transform, with $a=0$, $b=z$, $c=yz$ and $t=q$ leads to

$$H(x,y) = \frac{1}{x}\frac{(q)_\infty(yz)_\infty}{(qz)_\infty}\frac{(z)_\infty}{(q)_\infty(yz)_\infty}\sum_{n\geq 0}\frac{(y)_n(q)_n z^n}{(q)_n} = \frac{1}{x}(1-z)\sum_{n\geq 0}(y)_n z^n.$$

The expected value after n steps, $\sum_l lp_{n,l}$, is the coefficient of x^n in the partial derivative $H_y(x,y)$ taken at $y=1$. Since for $n\geq 1$

$$\frac{\partial}{\partial y}(y)_n\big|_{y=1} = -(q)_{n-1},$$

we have

$$\sum_{l\geq 1} lH_l(x) = -\frac{1}{x}(1-z)\sum_{n\geq 1}(q)_{n-1}z^n.$$

And to get the quantity of interest we have to extract the coefficient of z^n in the last expression. This is done by using Euler's transform: if $f(x) = \sum_{n\geq 0} a_n x^n$ then

$$\frac{1}{1-x}f\left(\frac{x}{x-1}\right) = \sum_{n\geq 0}\sum_{k=0}^{n}\binom{n}{k}(-1)^k a_k x^n.$$

Thus

$$\sum_{l\geq 1} lp_{n,l} \equiv [x^n]\sum_{l\geq 1} lH_l(x) = 1 - \sum_{k=1}^{n}\binom{n}{k}(-1)^k q^k (q)_{k-1}.$$

This formula is equivalent to the one given in [4], where its asymptotic value is then obtained by Rice's Method.

Bibliography

[1] Andrews (George E.). – *The Theory of Partitions.* – Addison-Wesley, 1976, *Encyclopedia of Mathematics and its Applications*, vol. 2.

[2] Flajolet (P.). – Approximate counting: A detailed analysis. *BIT*, vol. 25, 1985, pp. 113–134.

[3] Gasper (George) and Rahman (Mizan). – *Basic Hypergeometric Series.* – Cambridge University Press, 1990, *Encyclopedia of Mathematics and its Applications*, vol. 35.

[4] Kirschenhofer (P.) and Prodinger (H.). – Approximate counting: An alternative approach. *RAIRO Informatique Théorique et Applications*, vol. 25, 1991, pp. 43–48.

[5] Knuth (Donald E.). – *The Art of Computer Programming.* – Addison-Wesley, 1973, vol. 3: Sorting and Searching.

[6] Louchard (G.). – Exact and asymptotic distributions in digital and binary search trees. *RAIRO Theoretical Informatics and Applications*, vol. 21, n° 4, 1987, pp. 479–495.

[7] Prodinger (H.). – Approximate counting via Euler transform. *Mathematica Slovaca*, vol. 44, n° 5, 1994, pp. 569–574.

[8] Prodinger (H.). – Digital search trees and basic hypergeometric functions. *EATCS Bulletin*, vol. 56, 1995, pp. 112–115.

Biased Random Walks, Lyapunov Functions, and Stochastic Analysis of Best Fit Bin Packing

Claire Kenyon
CNRS - Villeurbanne

October 23, 1995

[summary by Philippe Robert]

This talk considers the average case behavior of the best fit algorithm for on-line bin packing in the case where the item sizes are uniformly distributed in $\{1/k, \dots, j/k\}$. The best fit algorithm works as follows: the items are packed on-line, each item goes to the bin for which the wasted space is minimized. The bins are of size 1. We focus here on the average wasted space of the algorithm. It is known that this quantity is bounded when j is small compared to k ($j < \sqrt{2k+2.25}-1.5$) or when j is sufficiently close to 1 or k. In the cases where it is known to be unbounded it appears to grow linearly (see [1]). The motivation of this study is to analyze the sensitivity of the performances of best fit algorithm with respect to the probability distribution of the sizes of the items. The main result is the following theorem.

THEOREM 1. *For the uniform distribution on $\{1/k, 2/k, \dots, (k-2)/k\}$, the average wasted space is bounded.*

We sketch the main ideas of the proof. The main variable of interest is the the multi-dimensional Markov chain $S(t) = (s_1(t), \dots, s_{k-1}(t))$, where $s_i(t)$ is the number of bins at time t with a residual space of size i/k. The transitions of this Markov chain are described as follows:
If the $(t+1)$th item is of size x/k,

- If $s_x(t) \neq 0$ then a bin is completely full with this item and so $s_x(t+1) = s_x(t) - 1$;
- if not and $\{s_i(t) \neq 0\}$ is not empty and $\nu = \inf\{i > x / s_i(t) \neq 0\}$ then $s_\nu(t+1) = s_\nu(t) - 1$ and $s_{\nu-x}(t+1) = s_{\nu-x}(t) + 1$;
- Otherwise $s_{k-x}(t+1) = s_{k-x}(t) + 1$.

If $W(t) = \sum_1^{k-1} i s_i(t)$ is the wasted space at time t, the theorem is that $\limsup_{t\to\infty} E(W(t)) < +\infty$. Because of the Markovian context, the first thing to check is whether the Markov chain $(S(t))$ is ergodic or not (i.e. has an equilibrium measure). A classical idea in this domain is to try to construct a Lyapunov function which is decreasing at infinity if the Markov chain is ergodic. The following result (see [2]) gives a useful criterion for our problem.

THEOREM 2. *If $X(t)$ is an irreducible homogeneous Markov chain on a countable state space $\mathcal{S} \subset \mathbb{N}$, and if there exists an integer $b \in \mathbb{N}$ and a function $f : \mathcal{S} \to \mathbb{R}_+$ such that*

(1) *$f(s) > C_1 s^\mu$, for some constants C_1, μ;*
(2) *$P(X(b) = b \mid X(0) = a) = 0$ if $|f(b) - f(a)| > C_2$;*
(3) *there exists a finite subset B of $\mathcal{S}$ such that $E_s(f(X(b)) - f(s)) < -\varepsilon$ if $s \notin B$.*

Then the Markov chain is ergodic with the invariant probability π satisfying

$$\pi(s) \leq C e^{-\delta f(s)},$$

for some constants C and δ.

The function f is usually called a Lyapunov function. The main assumption is condition (3) which expresses that the trajectory of $(f(X(t)))$ goes back ultimately (after b steps) towards the origin in average when it is far away. In our case the Lyapunov function is the wasted space $f(s) = \sum_1^{[j/2]} is_i$. Using stochastic comparisons with simple random walks on the integers, it is proved that the above assumptions are satisfied for this function. The result on the tail of the invariant distribution shows that the wasted space converges in distribution and also in average. Hence the wasted space is bounded.

Bibliography

[1] Coffman (E. G.), Johnson (D. S.), Shor (P. W.), and Weber (R. R.). – Markov chains, computer proofs, and average case analysis of best fit bin packing. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pp. 412–421. – 1993.

[2] Fayolle (G.), Malyshev (V. A.), and Menshikov (M. V.). – *Topics in the constructive theory of countable Markov chains.* – Cambridge University Press, 1995.

[3] Kenyon (C.), Rabani (Y.), and Sinclair (A.). – Biased random walks, Lyapunov functions, and stochastic analysis of best fit bin packing. In *7^th Annual ACM-SIAM Symposium on Discrete Algortihms.* – ACM, 1996.

An urn model from learning theory

Danièle Gardy
Université de Versailles Saint-Quentin

November 13, 1995

[summary by Stéphane Boucheron]

Abstract

The analysis of a learning problem motivates the definition of an urn model. In this model, two kinds of balls representing bad and good data are allocated at random in a collection of urns. This is a variation on the classical occupancy model where one is concerned with allocation of one kind of balls in a family of urns. In this model, the relevant quantities are the number of urns that contain more bad than good balls or as many good as bad balls. We describe the law of those two quantities in the static and dynamic framework. The investigation rely both on complex analysis techniques (generating functions) and probabilistic tools (exchangeability, and finite De Finetti theorems). Using proper normalization, the limiting phenomena are Gaussian random variables. Most interesting is the fact that the moments of the laws are described using modified Bessel functions.

1. The modified urn problem

The modified urn problem was initially motivated by the analysis of the learning curve of symmetric functions under classification noise in the field of computational learning theory [6]. As in the classical random allocation problem, k balls are thrown at random into those n urns. Balls are allocated independently, and the probability to fall into some urn is $1/n$. But here, balls are not only allocated, they are also labelled independently at random as good (with probability $1-\mu > 1/2$) or bad. The balance of one urn is the difference between the number of good balls and the number of bad balls in that urn. All the issues tackled in this investigation have the following flavor: what is the law of linear combinations of the numbers of urns with positive, negative and null balancesΓ This question can be answered in a static context, where k/n remains equal to a positive constant α when n tends to infinity, or in a dynamic context, where urns are allocated one at a time, and where we try to monitor the evolution of the fraction of urns with positive, negative and null balances at different normalized times $\alpha_1, \ldots, \alpha_i, \ldots$ with $\alpha_i = k_i/n$.

The goal of this analysis is to extend results stated in [7] on the empty urn problem. The empty urn problem can be treated by diffusion approximation techniques, or, using implicitly the Markov property, by generating functions. The problem examined here does not share this property. Moreover, the plausible enhancements of the state space that would make the fraction of urns with positive balance a function of a Markov chain, lead to consider processes which take values in infinite-dimensional spaces. The analysis presented in [1] relies both on generating functions and simple principles.

2. Generating functions

The generating functions manipulated here are of exponential type.

2.1. Generating function describing the behavior of one urn. Let y mark the number of balls in that urn. Because balls are indistinguishable, the generating function describing the number of ways of allocating balls in one urn is e^y. To reflect the fact that balls are of two kinds, this is rewritten as $e^{\mu y+(1-\mu)/y}$. Using a second variable z and expanding $e^{y(\mu z+(1-\mu)/z)}$, one notes that the coefficient of $y^k z^p$ is proportional to the probability that the urn has balance p when k balls are thrown into it. We get:

$$e^{y((1-\mu)z+\mu/z)} = \sum_{p\in\mathbb{Z}} a_p(y) z^p.$$

The exponent of z is the balance of the urn. This expression stresses the importance of Bessel functions. Modified Bessel functions of the first kind at order $p \in \mathbb{Z}$ can be defined by:

$$I_p(x) = \sum_{r\geq \max(0,-p)} \frac{(x/2)^{2r+p}}{r!(r+p)!}.$$

Bessel functions obey the following identity: $e^{\frac{y}{2}(u+\frac{1}{u})} = \sum_{p\in\mathbb{Z}} u^p I_p(y)$. Then letting $\sigma = \sqrt{\mu(1-\mu)}$, $e^{y((1-\mu)z+\mu/z)} = e^{\frac{2\sigma y}{2}(\sigma z/\mu+\frac{1}{\sigma z/\mu})} = \sum_{p\in\mathbb{Z}} \left(\frac{\sigma z}{\mu}\right)^p I_p(2\sigma y)$. Marking urns with positive balance by w, null balance by v and negative balance by u , and letting

$$\phi(y) = \sum_{p<0} \left(\frac{\sigma}{\mu}\right)^p I_p(2\sigma y) = \sum_{p>0} \left(\frac{\mu}{\sigma}\right)^p I_p(2\sigma y); \qquad \psi(y) = \sum_{p>0} \left(\frac{\sigma}{\mu}\right)^p I_p(2\sigma y) = e^y - I_0(2\sigma y) - \phi(y),$$

the generating function describing the sign of the balance in one urn is:

$$f(u,v,w,y) = u\phi(y) + vI_0(2\sigma y) + w\psi(y).$$

2.2. Generating function for a sequence of urns. Because urns are exchangeable, the generating function describing the states of a sequence of n urns is:

$$F(u,v,w,y) = f(u,v,w,y)^n = \left(u\phi(y) + vI_0(2\sigma y) + w\psi(y)\right)^n.$$

3. Exchangeability

The balances of different urns follow identical, non-independent but *exchangeable* laws: all permutations of a tuple of balances indexed by different urns have the same probability. Recall that the variation distance between two laws D and D' is defined by:

$$\|D - D'\|_{\text{var}} = \max_{\|\text{Test}\|_\infty \leq 1} |\operatorname{E}_D(\text{Test}) - \operatorname{E}_{D'}(\text{Test})|.$$

The following lemma shows that small sets of urns behave almost independently. Let P_i be the law of a tuple of i independent random variables that are distributed as the difference between two independent Poisson random variables with means $\mu k/n$ and $(1-\mu)k/n$.

Proposition 1. *The vector of balances in urns* 1 *to* i *after throwing* k *balls in* n *urns is distributed according to a law* Q_i *that is within variation distance* $2i/n$ *from* P_i.

The proof relies on the fact that conditionally on the number of balls allocated in urns $1,\ldots,i$, the balances of the i urns are independent and on theorem (5.1) in [2].

4. Static analysis

The cost of an experiment (throwing k balls into n urns) is the sum of the costs of the urns. The cost of an urn with null (resp. negative, positive) balance is C_0 (resp. C_1, C_2). We let $d_0 = C_0 - C_2$ and $d_1 = C_1 - C_0$. For costs relevant to learning theory applications, we have $d_0 = d_1$.

Using either the generating function approach or Proposition 1, one may derive the following equivalents for the expectation and variance of the cost:

$$\begin{aligned}\mathrm{E}(\mathrm{cost}) &\sim n\left[C_2 + d_1 e^{-\alpha}\mathrm{I}_0(2\sigma\alpha) + (d_0 + d_1)e^{-\alpha}\phi(\alpha)\right];\\ \mathrm{Var}(\mathrm{cost}) &\sim n d_1^2 e^{-\alpha}\left[4\phi(\alpha) + I_0(2\sigma\alpha) - e^{-\alpha}\Big(\big(2\phi(\alpha) + I_0(2\sigma\alpha)\big)^2 + \alpha\big((1-2\mu)I_0(2\sigma\alpha)\big)^2\Big)\right].\end{aligned}$$

Using the generating function approach and a theorem in [4], one may also conclude that the normalized and centered variable defined by: $(\mathrm{cost} - \mathrm{E}(\mathrm{cost}))/\sqrt{n}$ is asymptotically Gaussian with variance $\mathrm{Var}(\mathrm{cost})/n$.

5. Dynamic analysis

In the dynamic context, balls are allocated one at a time. If balls are allocated in n urns, the $k = \alpha n$th ball is allocated at time α. The cost is a random function of time. The average function when $n \to \infty$ is given by the above-stated expression for the average cost. The aim of this investigation is to characterize the limiting behavior of the normalized centered processes. To prove (weak) convergence of the processes to a limiting process, one needs to check that the sequence of processes is relatively compact, and that the finite dimensional distributions of the processes converge to the finite dimensional distributions of the limiting process.

Finite dimensional distributions are analyzed using both multivariate generating functions and elementary arguments building on exchangeability of urns.

Balls are assumed to be thrown in two groups. The first group is marked by y_1 and thrown at time α_1; we use variable z_1 to distinguish good balls from bad balls, (a good ball is marked as $(1-\mu)y_1 z_1$ and a bad ball as $\mu y_1/z_1$). Similarly, the second group is thrown at time α_2 and marked by y_2 and z_2. Variables u_i, v_i and w_i indicate the state of the urn after throwing the first group ($i = 1$) and the second group ($i = 2$).

Letting

$$\begin{aligned}\Delta I(y_1, y_2) &= \sum_{n>0} I_n(2\sigma y_1) I_{-n}(2\sigma y_2) = [I_0(2\sigma(y_1 + y_2)) - I_0(2\sigma y_1) I_0(2\sigma y_2)]/2,\\ S(y_1, y_2) &= \sum_{n>0, n+p>0} I_n(2\sigma y_1) I_p(2\sigma y_2)\left(\frac{\sigma}{\mu}\right)^{n+p},\\ T(y_1, y_2) &= \psi(y_1 + y_2) - S(y_1, y_2) - I_0(y_1)\psi(y_2),\end{aligned}$$

the following is derived:

Proposition 2. *The multivariate generating function describing the behavior of a single urn at the times α_1 and α_2 is*

$$\begin{aligned} w_1 w_2 S(y_1, y_2) &+ w_1 v_2 \Delta I(y_1, y_2) + w_1 u_2(\psi(y_1)e^{y_2} - S(y_1, y_2) - \Delta I(y_1, y_2))\\ &+ v_1 w_2 I_0(2\sigma y_1)\psi(y_2) + v_1 v_2 I_0(2\sigma y_1) I_0(2\sigma y_2) + v_1 u_2 I_0(2\sigma y_1)\phi(y_2)\\ &+ u_1 w_2 T(y_1, y_2) + u_1 v_2 \Delta I(y_1, y_2) + u_1 u_2(\phi(y_1)e^{y_2} - T(y_1, y_2) - \Delta I(y_1, y_2)).\end{aligned}$$

The single-urn generating function is used to compute the generating function of a sequence of urns at different instants. Then the limiting value of the characteristic function of the cost at a finite number of instants can be computed using saddle-point approximation methods as in [4, 5].

This allows to conclude that the finite dimensional distributions of the centered normalized processes converge to the finite dimensional distributions of a non-Markov Gaussian process with covariance between times α_1 and α_2:

$$d_0^2 \mathrm{e}^{-\alpha_2} \Bigg(\Bigg(\mathrm{I}_0(2\sigma\alpha_2) + 2\mathrm{I}_0(2\sigma\alpha_1)\phi(\alpha_2 - \alpha_1) + 4 \sum_{i>0,j>0} \left(\frac{\mu}{\sigma}\right)^j \mathrm{I}_i(2\sigma\alpha_1)\mathrm{I}_{j-i}(2\sigma(\alpha_2 - \alpha_1)) \Bigg)$$

$$- \mathrm{e}^{-\alpha_1} \Bigg(\big(\mathrm{I}_0(2\sigma\alpha_1) + 2\phi(\alpha_1)\big)\big(\mathrm{I}_0(2\sigma\alpha_2) + 2\phi(\alpha_2)\big) + \alpha_1(1-2\mu)^2 \mathrm{I}_0(2\sigma\alpha_1)\mathrm{I}_0(2\sigma\alpha_2) \Bigg) \Bigg).$$

Proving the weak convergence of the processes to the above-stated Gaussian process requires the proof of the relative compactness of the sequence of processes. This has not been done although the verification of the Kolmogorov-Centsov criterion raises more cumbersome computations than theoretical difficulties.

6. Questions

A plausible contribution of [1] is the presentation of a new kind of admissible construction: the majority phenomenon that comes from building a combinatorial structure on two types of objects (good and bad in this paper), then deciding on the type of the structure according to the type of the majority of the basic objects. For example, we can have two types of basic objects, build cycles on theses objects and combine these cycles into a set, then ask for the number of cycles of the set that have a majority of elements of one type, or an equal number of elements of each type. It should be possible to extend the distribution results on the number of components presented by Flajolet and Soria [3] to study the number of components of a given type (good, bad or neutral) for various combinatorial constructs.

Bibliography

[1] Boucheron (S.) and Gardy (D.). – An urn model from learning theory. *Random Structures and Algorithms*, 1996. – To appear.

[2] Diaconis (P.) and Freedman (D.). – A dozen de Finetti-style results in search of a theory. *Annales de l'Institut Henri Poincaré*, vol. 23, n° 2, 1987, pp. 397–423.

[3] Flajolet (Philippe) and Soria (Michèle). – Gaussian limiting distributions for the number of components in combinatorial structures. *Journal of Combinatorial Theory, Series A*, vol. 53, 1990, pp. 165–182.

[4] Gardy (D.). – Méthode de col et lois limites en analyse combinatoire. *Theoretical Computer Science, Series A*, vol. 94, n° 2, March 1992, pp. 261–280.

[5] Gardy (D.). – Some results on the asymptotic behaviour of coefficients of large powers of functions. *Discrete Mathematics*, vol. 139, 1995, pp. 189–217.

[6] Kearns (M.) and Vazirani (U.). – *Topics in Learning Theory.* – MIT Press, 1994.

[7] Kolchin (V.), Sevast'yanov (B.), and Chistyakov (V.). – *Random Allocations.* – Wiley & Sons, 1978.

A suboptimal lossy data compression based on approximate pattern matching

Wojciech Szpankowski
Purdue University

December 11, 1995

[summary by Philippe Jacquet]

1. Introduction

A practical algorithm for lossy data compression is presented. It is derived from the lossless Lempel-Ziv data compression. The principle of the scheme consists in considering approximate pattern matching where no more than $D\%$ of mismatches are allowed.

An algorithm is considered to be lossless when $D = 0$. For example Hoffman's algorithm and the Lempel-Ziv algorithm are lossless. Such algorithms are extensively used for text or data transmission or storage every time it is required to have error-free recovery. In this case the compression is limited by information theory. With image or voice/sound compression, there is no need of exact recovery since the noise in the record and/or the limited sensitivity of our eyes or ears will hide the details of the data base. In this case the compression can be limitless, depending only on the degree of *fidelity* one wants to keep in the recovery. Examples of lossy algorithms are JPEG, GIF, and MPEG (for motion pictures), they are based on adaptation of Fourier or wavelet transform, or on self-similarity search as in fractal compression.

The new lossy algorithm can be adapted to numerous applications as image or voice compression. This universality of use simply comes from the fact that the new algorithm proceeds on the digital transcription of the data regardless of their origin. In particular it can be adapted to image compression provided some tuning. An adaptation for voice/sound is under study.

The scheme on image shows performance close to JPEG algorithms and outperforms fractal compression. More importantly, it benefits of a much simpler "on line" decompression algorithm. Another advantage is that the new algorithm is tractable to performance analysis when the database (the text or the image to compress) follows a stochastic model.

2. Measure of fidelity

Before describing the algorithm we will introduce the performance measurement called fidelity. Let x be a text of length n ($|x| = n$). On the transmitter side the compression algorithm encodes x into $c(x)$. The compression rate is the ratio $|c(x)|/n$. With lossless algorithms the average compression rate, $E|c(x)|/n$ cannot be better than the entropy h of the source from which the database is built. In general the lossless algorithms asymptotically attain this theoretical bound when $n \to \infty$. The better the algorithm is, the faster is the convergence:

$$\lim_{n\to\infty} E|c(x)|/n = h.$$

On the receiver side, the code c is decompressed into $\phi(c)$. With lossless compression $\phi(c(x)) = x$. With lossy compression $\phi(c(x)) = \hat{x}$ which is of the same length as x ($|\hat{x}| = |x| = n$) but in general differs from x. In the following, we use the Hamming distance: $d(x, \hat{x})$ is number of mismatches between x and $\hat{x}$, divided by n. We can also accommodate our results to more sophisticated distances where mismatches have different weight per pair of symbols.

3. Lossy Lempel-Ziv compression Algorithm

Let x be a text. We denote x_n the nth suffix of x (starting at position n) and x^n the nth prefix of x (ending at position n). We denote x_i^j the subword starting at position i and ending at position j.

The algorithm is a parsing algorithm. We suppose that at step k the text has been parsed up to position n, i.e. x^n has been compressed into $c(x^n)$. The step $k+1$ will consist in finding the largest prefix x_n^{n+j} of x_n which is a copy within distance D of a substring in x^n. Assume this copy is at position i in x^n. Therefore the new parsed position is $n+j$, and the encoded text is $c(x^n)$ plus the pair (i,j): $c(x^{n+j}) = c(x^n)$."(i,j)". The substring x_n^{n+j} is called the new parsed phrase and j is its length.

4. Results

4.1. Rate-distortion measure. Let A^n be the set of all sequences of length n and let S be a subset of A^n. We call $P(S)$ the probability weight of S in A^n.

The optimal compression ratio depends on the rate-distortion function $R(D)$, which is defined as follows. Let w be a text of length n, we define $B_D(w)$ as the D-ball of center w, i.e. $B_D(w) = \{x : d(x,w) \leq D\}$. We define $N(D,S)$ as the minimum number of D-ball needed to cover S. Then:

$$R_n(D, \varepsilon) = \min_{S \subset A^n, P(S) \geq 1-\varepsilon} \frac{\log N(D,S)}{n},$$

and the rate-distortion is defined as $R(D) = \lim_{\varepsilon \to 0, n \to \infty} R_n(D, \varepsilon)$.

4.2. Generalized entropy. The generalized b-order Rényi entropy $h_b(D)$ is defined as follows:

$$h_b = \lim_{n\to\infty} \frac{-\log E[P^b(B_D(x)) \mid |x| = n]}{bk} = \lim_{n\to\infty} \frac{-\log \sum_{x \in A^n} P^b(B_D(x)) P(\{x\})}{bk}.$$

For $b \to 0$ we understand $h_0(D) = \lim_{n\to\infty} E[-\log P_{(}B_D(x)) \mid |x| = n]/k$, provided the limit exists.

When $D = 0$ (lossless case) we naturally recover the known b-order entropies $h^{(b)}$ defined by $E[-P(\{x\}) \log P(\{x\}) \mid |x| = n]$.

4.3. Asymptotic results on lossy Lempel-Ziv. Under some probabilistic model (Bernoulli, Markov, Mixing conditions), about the already parsed part of the text x^n we can obtain the following result.

THEOREM 1. *The length of the new parsed phrase L_n satisfies:*

$$\lim_{n\to\infty} \frac{L_n}{\log n} = \frac{1}{h_0(D)}$$

The convergence is in probability and/or almost sure convergence.

For the Bernoulli model we prove that $r_0(D)$ is the compression rate of the lossy Lempel-Ziv scheme and that $\lim_{D\to 0} R(D) = \lim_{D \to 0} h_0(D) = h$. In the case of binary uniform database we have $h_0(D) = R(D)$

THEOREM 2. *In the Bernoulli model, the lossy Lempel-Ziv algorithm is asymptotically optimal when $D \to 0$ and is asymptotically optimal for all D in the binary uniform Bernoulli model.*

A Faster Algorithm for Approximate String Matching

Ricardo Baeza-Yates

Department of Computer Science, University of Chile

July 8, 1996

[summary by Mireille Régnier]

Approximate string matching is one of the main problems in classical string algorithms. Given a text of length n, a pattern of length m, and a maximal number of errors allowed, k, we want to find all text positions where the pattern matches the text up to k errors. Errors can be substituting, deleting or inserting a character. The solutions to this problem differ if the algorithm has to be on-line (that is, the text is not known in advance) or off-line (the text can be preprocessed). In this paper the first case is studied, where the classical dynamic programming solution is $O(mn)$.

In the last years several algorithms have been presented that achieve $O(kn)$ comparisons in the worst-case [13, 6, 7] or in the average case [14, 6], by taking advantage of the properties of the dynamic programming matrix. In the same trend is [3], with average complexity $O(kn/\sqrt{c})$ (c is the alphabet size). The algorithms which are $O(kn)$ in the worst case tend to involve too much overhead, and are not competitive in practice. Other approaches attempt to filter the text, reducing the area in which dynamic programming needs to be used [12, 15, 11, 10, 4, 5]. These algorithms achieve sublinear expected time in many cases ($O(kn\log_c m/m)$ is a typical figure) for moderate k/m ratios, but the filtration is not effective for larger ratios. A simple and fast filtering technique is shown in [2], which yields an $O(n)$ algorithm for moderate k/m ratios. Yet other approaches use bit-parallelism [1] in a RAM machine of word length $O(\log n)$ to reduce the number of operations. [9] achieves $O(kmn/\log n)$, which is competitive for patterns of length $O(\log n)$. In [16], the cells are packed differently to achieve $O(mn\log c/\log n)$ complexity.

A new algorithm is presented which combines the ideas of taking advantage of the properties of the matrix, filtering the text and using bit-parallelism, being faster than previous work for moderate size patterns, as we are interested in text searching. One models the search with a non-deterministic finite automaton (NFA) built from the pattern and using the text as input. This automaton is simulated by an algorithm based on bit operations on a RAM machine of word length $O(\log n)$. The algorithm achieves running time $O(n)$, independently of k, for small patterns (i.e. $mk = O(\log n)$). This restricted algorithm is used to design two general algorithms.

The first one partitions the problem into subproblems, and has average time cost $O(mn/\log n)$ for small $\alpha = k/m$ (i.e. $\alpha < 1/\log n$), otherwise it is $O(\sqrt{mk/\log n}n)$ (i.e. $O(\sqrt{k}n)$ for $m = O(\log n)$, else $O(kn)$). It involves also a cost to verify potential matches, which is shown to be not significant for $\alpha < \alpha_1 \approx 1 - m^{1/\sqrt{\log n}}/\sqrt{c}$. This algorithm is a generalization of an earlier heuristic [8, 2], that reduces the problem to exact matching and is shown to be $O(n)$ for $\alpha < \alpha_0 = 1/(3\log_c m)$, and better than problem partitioning for $\alpha < \alpha_0' \approx 1/(2\log_c m)$.

The second one partitions the automaton into subautomata, being $O(k^2n/(\sqrt{c}\log n))$ on average. For $\alpha > 1 - 1/\sqrt{c}$ its worst case, $O((m-k)kn/\log n)$, dominates. This algorithm is shown to be better than dynamic programming for $k > \log(n)/(1-\alpha)$. One studies the optimal way to combine

Condition	Complexity	Method used
$mk = O(\log n)$	$O(n)$	the simple algorithm
$\alpha < \alpha_0$	$O(n)$	reducing to exact match
$\alpha_0 < \alpha < \alpha_1$	$O(\sqrt{mk/\log n}n)$	exact match if $\alpha < \alpha'_0$ else problem partitioning
$\alpha > \alpha_1 \wedge k < \log n/(1-\alpha)$	$O((m-k)kn/\log n)$	automaton partitioning
$\alpha > \alpha_1 \wedge k > \log n/(1-\alpha)$	$O(mn)$	plain dynamic programming

Table 1. Complexity of the hybrid algorithm.

the algorithms. It is shown experimentally that the hybrid algorithm is faster than previous ones, for moderate m. Table 1 shows the complexity.

As a corollary of the analysis, tight bounds are given for the probability of finding an occurrence of a pattern of length m with k errors starting at a fixed position in random text. We also show that the heuristic of [14] works $O(kn)$ on average, with a constant tighter than that of [3].

Bibliography

[1] Baeza-Yates (R.). – Text retrieval: Theory and practice. In *12-th IFIP World Computer Congress*. vol. I:Algorithms, Software, Architecture. – Elsevier Science, 1992.

[2] Baeza-Yates (R.) and Perleberg (C.). – Fast and practical approximate pattern matching. In *CPM'92. Lecture Notes in Computer Science*, vol. 644, pp. 185–192. – Springer-Verlag, 1992.

[3] Chang (W.) and Lampe (J.). – Theoretical and empirical comparisons of approximate pattern matching. In *CPM'92. Lecture Notes in Computer Science*, vol. 644, pp. 172–181. – Springer-Verlag, 1992.

[4] Chang (W.) and Lawler (E.). – Sublinear approximate string matching and biological applications. *Algorithmica*, vol. 12, 1994, pp. 327–344.

[5] Chang (W.) and Marr (T.). – Approximate string matching and local similarities. In *CPM'94. Lecture Notes in Computer Science*, vol. 807, pp. 259–274. – Springer-Verlag, 1994.

[6] Galil (Z.) and Park (K.). – An improved algorithm for approximate string matching. *SIAM Journal on Computing*, vol. 19, n° 6, 1990, pp. 989–999.

[7] Landau (G.) and Vishkin (U.). – Fast string matching with k differences. *Journal of Computer Systems Science*, vol. 37, 1988, pp. 63–78.

[8] Manber (U.) and Wu (S.). – Agrep—a fast approximate pattern matching tool. In *Usenix Technical Conference*, pp. 153–152. – 1992.

[9] Manber (U.) and Wu (S.). – Fast text searching allowing errors. *CACM*, vol. 35, n° 10, 1992, pp. 83–91.

[10] Suntinen (E.) and Tarhio (J.). – On using q-gram locations in approximate string matching. In *ESA'95. Lecture Notes in Computer Science*, vol. 834, pp. 234–242. – Springer Verlag, 1995.

[11] Takaoka (T.). – Approximate pattern matching with samples. In *ISAAC'94. Lecture Notes in Computer Science*, vol. 834, pp. 348–359. – Springer Verlag, 1994.

[12] Tarhio (J.) and Ukkonen (E.). – Boyer-Moore approach to approximate string matching. In *SWAT'90. Lecture Notes in Computer Science*, vol. 447, pp. 348–359. – Springer Verlag, 1990.

[13] Ukkonen (E.). – Algorithms for approximate string matching. *Information and Control*, vol. 64, 1985, pp. 100–118.

[14] Ukkonen (E.). – Finding approximate patterns in strings. *Journal of Algorithms*, vol. 6, 1985, pp. 132–137.

[15] Ukkonen (E.). – Approximate string matching with q-grams and maximal matches. *Theoretical Computer Science*, vol. 1, 1992, pp. 191–211.

[16] Wright (A.). – Approximate string matching using within-words parallelism. *Software-Practice and Experience*, vol. 24, 1994, pp. 337–362.

Rotations of Periodic Strings and Short Superstrings

Dany Breslauer
Max-Planck-Institute für Informatik

June 24, 1996

[summary by Mireille Régnier]

1. State of the Art

Let $S = \{s_1, \ldots, s_m\}$ be a set of strings over some alphabet Σ. A *common superstring*, or simply *superstring*, of S is a string s such that each s_i in S is a substring (*i.e.*, a consecutive block) of s. The shortest superstring problem is to find a superstring of the smallest possible length for any given set of strings S. The problem has applications in a wide range of areas including data compression [6] and DNA sequencing.

Since the problem is *NP-hard* [6] a lot of effort has been taken to find good approximation algorithms with guaranteed performance. Blum *et al.* [4] showed that the problem is *MAX SNP-hard* and thus does not have a polynomial time approximation scheme unless P = NP. Tarhio and Ukkonen [9] and Turner [11] gave several approximation algorithms that achieve $\frac{1}{2}$-approximation with respect to the *compression* measure, or the *total overlap* between adjacent strings in a superstring. This approximation ratio has been improved to $\frac{38}{63}$ by Kosaraju *et al.* [7]. Notice that superstrings have the minimum length if and only if they induce the maximum total overlap. Such relation, however, does not hold for approximations, and a good approximation for the length of the shortest superstring is not necessarily a good approximation for the maximum overlap in the superstring, and *vice versa*.

The first constant-approximation algorithm for the length of the shortest superstring was given by Blum *et al.* [4], who discovered a 3-approximation algorithm and proved that the "Greedy" algorithm by Tarhio and Ukkonen [9] achieves 4-approximation. Their algorithms and analysis rely on the close relation between the shortest superstring problem, that was shown by Turner [11] to be reducible to the *travelling salesman* problem, and the *cycle cover* problem. The same relation was exploited in subsequent papers [10] (≈ 2.89), [5] (≈ 2.83), [7] (≈ 2.79) and [1, 2] (≈ 2.75). Armen and Stein [3] have also recently obtained a $2\frac{2}{3}$-approximation algorithm, independently of our work.

Here we continue this line of work, and further improve the approximation ratio to $2\frac{2}{3} \approx 2.67$ and to $2\frac{25}{42} \approx 2.596$. The improved algorithms are similar to the previous algorithms in the sense that they construct a superstring by computing some optimal cycle covers on the *distance graph* of the given input strings, and then break and merge the cycles to finally obtain a Hamiltonian path representing some superstring. The key to the improvement are new bounds on the overlap between two strings.

2. Preliminaries

Without loss of generality, we assume that the set S is "substring-free" in that no string $s_i \in S$ is a substring of any other $s_j \in S$. For two strings s and t, let y be the longest string such that $s = xy$

and $t = yz$ for some *non-empty* strings x and z. We denote $\mathrm{ov}(s,t) = |y|$ the *overlap* between s and t, $d(s,t) = |x|$ the *distance* from s to t and $\mathrm{pref}(s,t) = x$. Given a list of strings $s_{i_1}, \ldots, s_{i_r}$, we define the superstring $s = \langle s_{i_1}, \ldots, s_{i_r} \rangle$ to be $\mathrm{pref}(s_{i_1}, s_{i_2})\,\mathrm{pref}(s_{i_2}, s_{i_3}) \cdots \mathrm{pref}(s_{i_{r-1}}, s_{i_r}) s_{i_r}$. It is clear that each shortest superstring for S must be $\langle s_{i_1}, \ldots, s_{i_m} \rangle$ for some permutation $i_1, \ldots, i_m$ of $\{1, \ldots, m\}$. Its length, $\mathrm{opt}(S)$, and the total overlap between adjacent strings, $\mathrm{maxov}(S)$, satisfy: $\mathrm{opt}(S) = \sum_{s_i \in S} |s_i| - \mathrm{maxov}(S)$.

2.1. Distance graph and cycle covers. The concept of a *distance graph* is central to all existing approximation algorithms for shortest superstrings. Let $G_S = (V, E, w)$ be a directed graph, where the set of vertices $V = \{s_1, \ldots, s_m\}$, the set of edges $E = \{(s_i, s_j) \mid 1 \leq i \neq j \leq m\}$, and the weight function w is the distance function $d(,)$. G_S is called the distance graph of S. If we denote the cost of a minimum weight Hamiltonian cycle on G_S as $\mathrm{Tsp}(G_S)$, then obviously, for any $s_i \in S$,

$$\mathrm{Tsp}(G_S) \leq \mathrm{opt}(S) \leq \mathrm{Tsp}(G_S) + |s_i|.$$

In other words, a minimum weight Hamiltonian cycle on G_S would be a very good approximation of a shortest superstring of S. Since TSP is NP-hard and has no good approximation algorithms, we try to work with a relaxed version of TSP, the *cycle cover* problem defined below.

Given a directed weighted graph G, a *cycle cover* is a set of (simple) cycles such that each vertex is contained in exactly one cycle. The weight of the cycle cover is the total weight of its cycles. A minimum weight cycle cover can be computed in $O(n^3)$ time using the Hungarian algorithm [8].

Let $\mathrm{Cyc}(G_S)$ be the weight of a minimum weight cycle cover of G_S. Then we have $\mathrm{Cyc}(G_S) \leq \mathrm{Tsp}(G_S) \leq \mathrm{opt}(S)$. To get an upper bound on $\mathrm{opt}(S)$ in terms of $\mathrm{Cyc}(G_S)$ we have to look at the particular structures and properties of strings.

2.2. Periodicity of strings and semi-infinite strings. A string x is *a factor* of a string s if $s = x^i y$ for some positive integer i and prefix y of x (y may be empty). *The factor* of a non-empty string s, denoted $\mathrm{factor}(s)$, is the *shortest* factor of s and the *period* of s is denoted $\mathrm{period}(s) = |\mathrm{factor}(s)|$. A semi-infinite string $s = a_1 a_2 \cdots$ is said to be *periodic* if $s = xs$ for some *non-empty* string x. The shortest such x is called the factor of s. Two (periodic semi-infinite) strings s, t are *equivalent* if their factors are cyclic shifts of each other, *i.e.*, if there are strings x, y such that $\mathrm{factor}(s) = xy$ and $\mathrm{factor}(t) = yx$. Otherwise, they are *inequivalent*. For each string s, let s^∞ denote the semi-infinite string $ss \cdots$, and $s_\infty = \mathrm{factor}(s)^\infty$ denote the periodic semi-infinite string that is equivalent to s and begins with s. Note that in general $s^\infty \neq s_\infty$. For example, $(010)^\infty = 010010 \cdots \neq (010)_\infty = 0101 \cdots$.

Connections between a cycle in G_S and the periodicity of the strings obtained by breaking the cycle are essentially given in [4]. Let $c = s_{i_1}, \ldots, s_{i_r}, s_{i_1}$ be a cycle in G_S, and $w(c)$ be its weight. Without loss of generality, assume that c has the minimum weight among all cycles in G_S containing $s_{i_1}, \ldots, s_{i_r}$. We will use:

LEMMA 1. $w(c) = d(s_{i_1}, s_{i_2}) + \cdots + d(s_{i_{r-1}}, s_{i_r}) + d(s_{i_r}, s_{i_1}) = \mathrm{period}(\langle s_{i_1}, \ldots, s_{i_r} \rangle)$.

2.3. The overlap-rotation lemma. The key to the improved approximation bounds is our overlap-rotation lemma below that follows from the classical *Critical Factorization Theorem.* Given a semi-infinite string $\alpha = a_1 a_2 \cdots$, we denote the rotation $\alpha[k] = a_k a_{k+1} \cdots$.

LEMMA 2. *Let α be a periodic semi-infinite string. There exists an integer k, such that for any (finite) string s that is inequivalent to α,*

$$\mathrm{ov}(s, \alpha[k]) < \mathrm{period}(s) + \frac{1}{2}\mathrm{period}(\alpha).$$

(1) Construct the distance graph G_S for set S.
(2) Find a minimum weight cycle cover C on the graph G_S.
(3) For each cycle $c = s_{i_1}, \ldots, s_{i_r}, s_{i_1} \in C$, choose a string t_c such that for some j, t_c contains $\langle s_{i_{j+1}}, \ldots, s_{i_r}, s_{i_1}, \ldots, s_{i_j} \rangle$, and t_c is contained in $\langle s_{i_j}, \ldots, s_{i_r}, s_{i_1}, \ldots, s_{i_{j-1}}, s_{i_j} \rangle$.
(4) Let T be the set of all strings chosen above and construct the distance graph G_T for T.
(5) Find a minimum weight cycle cover CC on G_T.
(6) Break each cycle of CC arbitrarily to obtain a superstring of the elements in the cycle.
(7) Concatenate the strings found at Step (6) arbitrarily to produce a superstring $\tilde{s}$ of S.

FIGURE 1. The generic shortest superstring approximation algorithm.

In addition, if $\operatorname{period}(s) \leq \operatorname{period}(\alpha)$, *then* $\operatorname{ov}(s, \alpha[k]) < \frac{2}{3}(\operatorname{period}(s) + \operatorname{period}(\alpha))$.

Our proof is *constructive*; it requires two computations of *critical factorizations* done in time that is linear in $\operatorname{period}(\alpha)$. From now on, let $\overrightarrow{\alpha}$ denote a rotation of α satisfying Lemma 2. The bound in the last lemma is roughly tight because for any rotation of the semi-infinite string $(0^n 1 0^{n+1} 1)^\infty$, there exists a string with period at most $n+2$ which overlaps with $(0^n 1 0^{n+1} 1)^\infty$ by at least $2n+2$.

3. Approximation algorithms

Our algorithms are only slightly different from the ones in [1, 2, 3, 4, 5, 7, 10]. The main steps are shown in Figure 1. We show first that this generic algorithm has approximation ratio 3. Noting that

$$\langle s_{i_j}, \ldots, s_{i_r}, s_{i_1}, \ldots, s_{i_{j-1}}, s_{i_j} \rangle = \operatorname{factor}(\langle s_{i_j}, \ldots, s_{i_r}, s_{i_1}, \ldots, s_{i_{j-1}} \rangle) s_{i_j},$$

it is straightforward [4, 10] that: $\operatorname{opt}(T) \leq \operatorname{opt}(S) + \operatorname{Cyc}(G_S) \leq 2\operatorname{opt}(S)$; hence, we have $\operatorname{Cyc}(G_T) \leq \operatorname{opt}(T) \leq 2\operatorname{opt}(S)$. We make use of the following upper bound on the possible overlap between two inequivalent strings s and t: $\operatorname{ov}(s,t) \leq \operatorname{period}(s) + \operatorname{period}(t)$, and show that it applies to the strings in T. Then the total overlap represented by the edges broken in Step 6, OV, is at most the sum of the periods of the strings in T. By Corollary 1, $OV \leq \sum_{c \in C} w(c) = \operatorname{Cyc}(G_S)$.
Putting everything together, we can bound the length of the superstring $\tilde{s}$ as

$$|\tilde{s}| = \operatorname{Cyc}(G_T) + OV \leq \operatorname{Cyc}(G_T) + \operatorname{Cyc}(G_S) \leq 2\operatorname{opt}(S) + \operatorname{opt}(S) \leq 3\operatorname{opt}(S).$$

The $2\frac{2}{3}$-approximation algorithm. Many researchers have tried to improve the performance of the generic algorithm by polishing Steps 5 - 7. Nevertheless, Armen and Stein [1, 2] identified strings that are not much longer than their factors as the bottleneck and tried to avoid them in Step 3. A key difference between our algorithm and all the previous ones actually is Step 3. The previous algorithms all choose one of the strings contained in the cycle c, whereas here we look for a superstring of the strings in c that is not *too long*, to reduce OV. More precisely, we rely on:

LEMMA 3. *For any cycle $c = s_{i_1}, \ldots, s_{i_r}, s_{i_1} \in C$, there exists a string t_c such that for some j,*
(1) *t_c contains the string $\langle s_{i_{j+1}}, \ldots, s_{i_r}, s_{i_1}, \ldots, s_{i_j} \rangle$.*
(2) *t_c is contained in the string $\langle s_{i_j}, \ldots, s_{i_r}, s_{i_1}, \ldots, s_{i_{j-1}}, s_{i_j} \rangle$.*
(3) $(t_c)_\infty = \langle \overrightarrow{s_{i_1}, \ldots, s_{i_r}} \rangle_\infty$.

The string t_c can be found in linear time. We polish the generic algorithm by choosing t_c in Step 3 and changing step 6 into: For each cycle of CC, break the cycle by deleting an edge that goes from a string to a string of *equal or larger period*, to obtain a superstring of the elements in the cycle.

Note that we do not treat the small cycles of CC specially like the other algorithms do. Instead, we cut the cycles with a bit of care. Clearly, in every cycle there must be an edge that goes from a string to a string of equal or larger period. Applying Lemmas 2 and 1, we get

$$OV \leq \frac{2}{3} \sum_{c \in C} \operatorname{period}(t_c) = \frac{2}{3} \sum_{c \in C} w(c) = \frac{2}{3} \operatorname{Cyc}(G_S) \leq \frac{2}{3} \operatorname{opt}(S).$$

Hence, $|\tilde{s}| = \operatorname{Cyc}(G_T) + OV \leq 2\frac{2}{3} \operatorname{opt}(S)$.

The $2\frac{25}{42}$-approximation algorithm. Steps 5, 6 and 7 now become: *Construct a superstring of T using a good overlap approximation algorithm.* It was proven in [4] that the length $\operatorname{apx}(T)$ of the superstring of T produced by a δ overlap approximation algorithm satisfies: $\operatorname{apx}(T) \leq \operatorname{opt}(T) + (1-\delta)\operatorname{maxov}(T)$. Our special choice of the cycle representatives t_c in Step 3 allows to improve on the standard bound used in all previous papers, e.g. $\operatorname{maxov}(T) \leq 2 \operatorname{Cyc}(G_S)$. By Lemma 2, we prove that: $\operatorname{maxov}(T) \leq \frac{3}{2} \operatorname{Cyc}(G_S)$. We use the $\frac{38}{63}$ overlap approximation algorithm in [7], and get: $\operatorname{apx}(T) \leq \operatorname{opt}(T) + (1 - \frac{38}{63})\operatorname{maxov}(T) \leq 2 \operatorname{opt}(S) + \frac{25}{63}\frac{3}{2}\operatorname{Cyc}(G_S) \leq 2\frac{25}{42} \operatorname{opt}(S)$.

Concluding remark. We are still a long way from reaching the conjectured ratio 2 for approximating shortest superstrings.

Bibliography

[1] Armen (Chris) and Stein (Clifford). – Improved length bounds for the shortest superstring problem. In Akl (S. G.), Dehne (F.), Sack (J. R.), and Santoro (N.) (editors), *Algorithms and Data Structures. Proceedings. Lecture Notes in Computer Science*, pp. 494–505. – Berlin, Heidelberg, New York, 1995. 4th International Workshop, WADS '95, Kingston, Canada, 16–18 Aug. 1995.

[2] Armen (Chris) and Stein (Clifford). – Short superstrings and the structure of overlapping strings. *Journal of Computational Biology*, 1995. – To appear.

[3] Armen (Chris) and Stein (Clifford). – A $2\frac{2}{3}$-approximation algorithm for the shortest superstring problem. In *Combinatorial Pattern Matching. Proceedings. Lecture Notes in Computer Science.* – Berlin, Heidelberg, New York, 1996. 7th International Workshop.

[4] Blum (A.), Jiang (T.), Li (M.), Tromp (J.), and Yanakakis (M.). – Linear approximation of shortest superstrings. *Journal of the ACM*, vol. 41, n° 4, 1994, pp. 630–647.

[5] Czumaj (A.), Gąsieniec (L.), Piotrow (M.), and Rytter (W.). – Parallel and sequential approximation of shortest superstrings. In Schmidt (E. M.) and Skyum (S.) (editors), *Algorithm Theory - SWAT '94. Lecture Notes in Computer Science*, pp. 95–106. – Berlin, Heidelberg, New York, 1994. 4th Scandinavian Workshop on Algorithm Theory, Aarhus, Denmark, July 6–8, 1994.

[6] Gallant (J.), Maier (D.), and Storer (J.). – On finding minimal length superstrings. *Journal fo Computer System Sciences*, vol. 20, 1980, pp. 50–58.

[7] Kosaraju (S. R.), Park (J.), and Stein (C.). – Long tours and short superstrings. In *Proceedings 35th IEEE Symposium on Foundations of Computer Science.* – 1994.

[8] Papadimitriou (Christos H.) and Steiglitz (Kenneth). – *Combinatorial optimization : algorithms and complexity.* – Prentice Hall, Englewood Cliffs, N. J., 1982.

[9] Tarhio (J.) and Ukkonen (E.). – A greedy approximation algorithm for constructing shortest common superstrings. *Theoretical Computer Science*, vol. 57, 1988, pp. 131–145.

[10] Teng (S. H.) and Yao (F.). – Approximating shortest superstrings. In *Proceedings 34th IEEE Symposium on Foundations of Computer Science*, pp. 158–165. – 1993.

[11] Turner (J.). – Approximation algorithms for the shortest common superstring problem. *Information and Computation*, vol. 83, 1989, pp. 1–20.

Searching patterns: combinatorics and probability

Mireille Régnier
INRIA-Rocquencourt

July 8, 1996

[summary by Pierre Nicodème]

Abstract

We formally define a class of sequential pattern matching algorithms that includes all variations of the Morris-Pratt algorithm. We prove for the worst case and the average case the existence of a complexity bound which is a linear function of the text string length for the Morris-Pratt algorithm, using the *Subadditive Ergodic Theorem*. We establish some structural property of Morris-Pratt-like algorithms, proving the existence of "unavoidable positions" where the algorithm must stop to compare. We compute also the complexity of the Boyer-Moore algorithm.

1. Sequential pattern matching algorithms

1.1. Basic Definitions. Throughout we write $\mathbf{p}$ and $\mathbf{t}$ for the pattern and the text which are of lengths m and n, respectively. The ith character of the pattern $\mathbf{p}$ (text $\mathbf{t}$) is denoted as $\mathbf{p}[i]$ ($\mathbf{t}[i]$), and by $\mathbf{t}_i^j$ we denote the substring of $\mathbf{t}$ starting at position i and ending at position j, that is $\mathbf{t}_i^j = \mathbf{t}[i]\mathbf{t}[i+1]\cdots\mathbf{t}[j]$. We also assume that the length m of a given pattern $\mathbf{p}$ does not vary with the text length n.

We want to investigate the complexity of string matching algorithms [2]. We define it formally as follows.

Definition 1 (Complexity).

(1) For any string matching algorithm that runs on a given text $\mathbf{t}$ and a given pattern $\mathbf{p}$, let $M(l,k) = 1$ if the lth symbol $\mathbf{t}[l]$ of the text is compared by the algorithm to the kth symbol $\mathbf{p}[k]$ of the pattern. We assume in the following that this comparison is performed at most once.

(2) For a given pattern matching algorithm, a partial complexity function $c_{r,s}$ is defined as

$$c_{r,s}(\mathbf{t},\mathbf{p}) = \sum_{l\in[r,s],k\in[1,m]} M[l,k]$$

where $1 \le r < s \le n$. For $r = 1$ and $s = n$ the function $c_{1,n} := c_n$ is simply called the *complexity* of the algorithm. If either the pattern or the text is a realization of a random sequence, then we denote the complexity by a capital letter, that is, we write C_n instead of c_n.

An Alignment Position AP is a position of the text which is aligned with the first character of the pattern during the processing of the algorithm, and such that, with the corresponding alignment, at least one character of the pattern is compared with the text.

Definition 2. A string searching algorithm is said:

(1) *semi-sequential* if the text is scanned from left to right;

(2) *strongly semi-sequential* if the order of text-pattern comparisons actually performed by the algorithm defines a non-decreasing sequence of text positions (l_i) and if the sequence of alignment positions is non-decreasing.

(3) *sequential* (respectively *strongly sequential*) if they satisfy, additionally for any $k > 1$

$$M[l,k] = 1 \Rightarrow \mathbf{t}_{l-(k-1)}^{l-1} = \mathbf{p}_1^{k-1}.$$

Note that condition (3) forbids unnecessary comparisons.

Example (Naive or brute force algorithm). The simplest string searching algorithm is the naive one. All text positions are alignment positions. For a given one, say AP, the text is scanned until the pattern is found or a mismatch occurs. Then, $AP + 1$ is chosen as the next alignment position and the process is repeated.

This algorithm is sequential but not strongly sequential. Condition (2) is violated after any mismatch on a alignment position l with parameter $k \geq 3$, as comparison $(l+1, 1)$ occurs after $(l+1, 2)$ and $(l+2, 3)$.

Example (Morris-Pratt-like algorithms [3]). Morris-Pratt like algorithms are strongly sequential; when a mismatch is found, they shift the pattern by the largest periodicity of the prefix of the pattern examined at the corresponding alignment position. The Knuth-Morris-Pratt variant remembers the last question concerning the mismatch position of the text and does not ask it again; the Simon variant remembers all the questions at the mismatch position, and does not ask them again. The efficiency of these algorithms is slightly better as the number of remembered questions increases.

It was already noted [3] that after a mismatch occurs when comparing $\mathbf{t}[l]$ with $\mathbf{p}[k]$, some alignment positions in $[l+1, \ldots, l+k-1]$ can be disregarded without further text-pattern comparisons. Namely, the ones that satisfy $\mathbf{t}_{l+i}^{l+k-1} \neq \mathbf{p}_1^{k-i}$, or, equivalently, $\mathbf{p}_{1+i}^{k} \neq \mathbf{p}_1^{k-i}$, and the set of such i can be known by a preprocessing of $\mathbf{p}$. Other i define the "surviving candidates", and choosing the next alignment position among the surviving candidates is enough to *ensure* that condition (2) in Definition 2 holds.

Example (Illustration to Definition 2). Let $\mathbf{p} = abacabacabab$ and $\mathbf{t} = abacabacabaaa$. The first mismatch occurs for $M(12,12)$. The comparisons performed from that point are:

1. *Morris-Pratt variant:* $(12,12); (12,8); (12,4); (12,2); (12,1); (13,2); (13,1)$, where the text character is compared in turn with pattern characters (b, c, c, b, a, b, a) with the alignment positions $(1, 5, 9, 11, 12, 12, 13)$.
2. *Knuth-Morris-Pratt variant:* $(12,12); (12,8); (12,2); (12,1); (13,2); (13,1)$, where the text character is compared in turn with pattern characters (b, c, b, a, b, a) with the alignment positions $(1, 5, 11, 12, 12, 13)$.
3. *Simon variant:* $(12,12); (12,8); (12,1); (13,2); (13,1)$, where the text character is compared in turn with pattern characters (b, c, a, b, a) with the alignment positions $(1, 5, 12, 12, 13)$.

Positions 1, 5 and 12 are unavoidable for all these Morris-Pratt-like algorithms.

Definition 3. For a given a pattern $\mathbf{p}$, a position i in the text $\mathbf{t}$ is an *unavoidable alignment position* for an algorithm if for any r, l such that $r \leq i$ and $l \geq i + m$, the position i is an alignment position when the algorithm is run on $\mathbf{t}_r^l$.

THEOREM 1. [7] *Given a pattern* $\mathbf{p}$ *and a text* $\mathbf{t}$*, all strongly sequential algorithms have the same set of unavoidable alignment positions* $U = \bigcup_{l=1}^{n} \{U_l\}$*, where*

$$U_l = \min\{\min_{1 \le k \le l} \{\mathbf{t}_k^l \preceq \mathbf{p}\}, l+1\}$$

and $\mathbf{t}_k^l \preceq \mathbf{p}$ *means that the substring* $\mathbf{t}_k^l$ *is a prefix of the pattern* $\mathbf{p}$*.*

1.2. Analysis. In the "average case analysis" we indicate that under assumption of *Stationary Model* (both strings $\mathbf{p}$ and $\mathbf{t}$ are random realizations of a *stationary* and *ergodic* sequence), the average complexity C_n may be computed by a direct application of an extension of Kingman's *Subadditive Ergodic Theorem* due to Derriennic [4] . See also [5].

LEMMA 1. [7] *A strongly semi-sequential algorithm satisfies the following basic inequality for all* r *such that* $1 \le r \le n$*:*

$$|c_{1,n} - (c_{1,r} + c_{r,n})| \le 2m^2,$$

provided any comparison is done only once.

We get also:

THEOREM 2. *With* $\mathbf{p}$ *a pattern of size* m*,* $\mathbf{t}$ *a text of size* n*, and a strongly-sequential algorithm, the number of comparisons is given by:*

(a) *worst case:* $\lim_{n \to \infty} \max_t c_n(t,p)/n = \alpha_1(p)$*,*
(b) $\mathbf{p}$ *given,* $\mathbf{t}$ *random:* $C_n(p)/n \overset{p.s.}{\to} \alpha_2(p)$ *(on the average),*
(c) $\mathbf{p}, \mathbf{t}$ *random:* $\lim_{n \to \infty} E_{t,p} C_n/n = \alpha_3 \ge 1$*.*

In the Boyer-Moore algorithm [1], a window of size equal to the size of the pattern is moved from left to right, with shifts depending of the text and pattern contents; inside the window, scanning is performed from right to left; the Boyer-Moore algorithm gives a counterexample to the preceding theorem, inside the class of pattern-matching algorithms: given the text $\mathbf{t} = \{\cdots y^{10} a z^4 (bazbzz)^n \cdots\}$, and a pattern $\mathbf{p} = \{x^4 a x^2 b x^2 a\}$, it is impossible to find a set of unavoidable positions for the Boyer-Moore algorithm.

2. Boyer-Moore algorithm

For the Boyer-Moore algorithm, a *head* is the rightmost position of the text in the window after a shift; let H_n be the number of heads in a text of length n. We show by a Laplace transform method the convergence of H_n to a variable with normal distribution.

Both expectation and variance of H_n are functions of the *shift polynomial*, defined as $f_p(z) = \sum_a q_a z^{d(a)}$, where $d(a)$ is the shift of the first occurrence of letter a from the right extremity of the pattern and q_a is the probability of occurrence of letter a. With this definition, the shift polynomial of the pattern 10001 is $\frac{1}{2}(z + z^4)$, with uniform distribution for letters 0 and 1.

When considering the complexity $C_n^{[P]}$ of the algorithm for a fixed pattern P and a text of length n, we define X_i as the number of comparisons done for an alignment at position i, and $Z_j = 1$ when j is a head, 0 otherwise. We have

$$C_n^{[P]} = \sum_{i=m}^{n} X_i Z_i.$$

After an algebraic manipulation, we take the expectation:

$$E\left[\frac{C_n^{[P]}}{n}\right] = \frac{1}{n}\sum_{i=m}^{n} E[X_i Z_i] - \frac{1}{n}\sum_{i=m}^{n} E[X_j(1 - Z_j)].$$

From this decomposition, we show that $E\left[\frac{1}{n}C_n^{[P]}\right] \to c_P$, and give an expression for c_P. We show also that the fourth moment is bounded.

With these results for moments, we apply a central limit theorem for dependant variables [5], where the strong mixing condition is equivalent to independence of positions sufficiently distant. This proves the convergence of $C_n^{[P]}$ to a variable with normal distribution.

Unavoidable positions. Almost surely, for a random text, there exists one unavoidable position; formally, we say that Z_k is *determined* by $t_{j+1}\cdots t_{k-1}$ if this string is sufficient to tell whether $Z_k = 0$ or 1. We denote the indicator of this event by

$$\xi_k^{(j)} = 1_{\{Z_k \text{ determined by } t_{j+1}\cdots t_{k-1}\}};$$

we have then:

Lemma 2. *$E[1-\xi_k^{(j)}] \leq \rho^{k-j-2}$, where $\rho < 1$, for $k-j \geq 2m$.*

Proof. [Sketch] If $\xi_k^{(j)} = 0$, then p_{m-1} does not occur $m-1$ times consecutively in $t_{j+1}\cdots t_{k-1}$. Given a fixed set of $m-1$ consecutive characters, the probability that not all of them are equal to p_{m-1} is A, with $A < 1$. The probability of no string of $m-1$ consecutive occurrences of p_{m-1} is at most $A^{\lfloor (k-j-2)(m-1)\rfloor}$; take $\rho = A^{1/(2m)}$. □

3. Number of occurrences of a word

We extended the classical result of Guibas and Odlyzko [6] to the Markovian case, giving all moments. This is done by constructing language expressions that characterize both models, and by analysis on the corresponding generating functions.

Bibliography

[1] Boyer (R.) and Moore (J.). – A fast string searching algorithm. *Communications of the ACM*, vol. 20, 1977, pp. 762–772.

[2] Crochemore (M.) and Rytter (W.). – *Text Algorithms.* – Oxford University Press, 1994.

[3] D. E. Knuth (J. Morris) and Pratt (V.). – Fast pattern matching in strings. *SIAM Journal on Computing*, vol. 6, 1977, pp. 189–195.

[4] Derriennic (Y.). – Un théorème ergodique presque sous additif. *The Annals of Probability*, vol. 11, 1983, pp. 669–677.

[5] Durrett (R.). – *Probability: Theory and Examples.* – Wadsworth & Brooks/Cole Books, Pacific Grove, California, 1991.

[6] Guibas (L. J.) and Odlyzko (A. M.). – Strings overlaps, pattern matching and non-transitive games. *Journal of Combinatorial Theory, Series A*, vol. 30, 1981, pp. 183–208.

[7] Regnier (M.) and Szpankowski (W.). – *Complexity of Sequential Pattern Matching Algorithms.* – Research Report n° 2549, Institut National de Recherche en Informatique et en Automatique, 1995.

Part 5

Miscellany

The (max,+) semiring. An introduction

Stéphane Gaubert
INRIA, Rocquencourt

March 11, 1996

[summary by Marianne Akian]

Abstract

Endowing real (or natural) numbers with max and + laws leads to an idempotent semiring which has been reinvented in many domains: graph optimization, language theory, statistical physics, quantum mechanics, discrete event systems, etc. The talk presents applications together with basic results of the so-called (max,+) algebra.

Introduction

We say that $(\mathbb{S}, \oplus, \otimes)$ is an idempotent semiring or dioid [19, 2] if $\oplus$ and $\otimes$ are associative laws on $\mathbb{S}$ with neutral elements $\mathbf{0}$ and $\mathbf{1}$ respectively, $\oplus$ is commutative and idempotent, that is $a \oplus a = a$, $\otimes$ is distributive with respect to the $\oplus$ law and $\mathbf{0}$ is absorbing with respect to the $\otimes$ law. By the idempotency property, $a \oplus b = \mathbf{0}$ implies $a = \mathbf{0}$. Then, the $\oplus$ law is not symmetrizable (and not simplifiable). However, idempotency leads to "simplifications" that partially compensate the non simplifiability. An idempotent semiring is said commutative when $\otimes$ is commutative and it is a semifield if the $\otimes$ law is invertible. Examples of commutative idempotent semifields are $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$ with $\mathbf{0} = -\infty$ and $\mathbf{1} = 0$, $\mathbb{R}_{\min} = (\mathbb{R} \cup \{+\infty\}, \min, +)$, $(\mathbb{R}^+, \max, \times)$ which are isomorphic. They are called respectively $(\max, +)$, $(\min, +)$ and $(\max, \times)$ algebra and are used in operations research [7], graph theory [19], discrete event systems [2, 14, 13], dynamic programming, Hamilton-Jacobi-Bellman equations [28, 1, 8], exponential asymptotics [29, 23, 5, 4]. The subsemiring $\mathbb{N}_{\min} = (\mathbb{N} \cup \{+\infty\}, \min, +)$ of $\mathbb{R}_{\min}$, called tropical semiring, is used in language theory [21, 22, 33, 34, 25, 24]. Concerning theoretical results on $(\max, +)$ algebra, an historical reference is [7]. More recent accounts can be found in [2, 28], collections of survey papers will be presented in [20] and a general and complete bibliography can be found in [26].

1. Some applications

1.1. Shortest path problem. The traditional application of the $(\min, +)$ algebra concerns the shortest path problem in a graph [19]. Let G be a graph with nodes denoted $\{1, \ldots, n\}$ representing towns and arcs representing roads. Let A_{ij} denote the time to go from i to j (or the length of arc (i, j)) with $A_{ij} = +\infty$ when there is no arc. If $A = (A_{ij})$ is considered as a $(\min, +)$ matrix,

$$(A^k)_{ij} = \bigoplus_{i_1, \ldots, i_{k-1}} A_{ii_1} \otimes \cdots \otimes A_{i_{k-1}j} = \min_{i_1, \ldots, i_{k-1}} A_{ii_1} + \cdots + A_{i_{k-1}j}$$

represents the minimal time from i to j (or the minimal distance between i and j) in k steps. If $A^* = \oplus_{k=0}^{\infty} A^k$, then $(A^*)_{ij}$ represents the minimal time from i to j.

A similar problem arises in discrete deterministic optimal control. Let now A_{ij} represent the cost of i to j transition, b_i the final cost in state i at time N and let v_i^n denote the minimal cost of a trajectory starting in i at time $n \leq N$. The value function v^n satisfies the backward dynamic programming (or Hamilton-Jacobi-Bellman) equation

$$v_i^n = \min_j A_{ij} + v_j^{n+1}, \qquad v_i^N = b_i$$

that is $v^n = A \otimes v^{n+1}$ with $v^N = b$, which is the $(\min, +)$ analogue of the Kolmogorov or backward Fokker-Planck equation, (final or transition) costs replacing probabilities [1, 8]. More generally, dynamic programming equations with continuous time and state are solved using $(\min, +)$ algebra in [28, 23].

1.2. Synchronization problems. Let us consider a manufacturing system where 2 types of parts are assembled, taking a fixed duration τ. Let $u_i(t)$ denote the number of parts of type $i = 1, 2$ arrived at time t and $y(t)$ the number of parts assembled. Then

$$y(t) = \min(u_1(t-\tau), u_2(t-\tau)) = u_1(t-\tau) \oplus u_2(t-\tau)$$

in the $(\min, +)$ algebra. If now $u_i(n)$ (resp. $y(n)$) denotes the date of the n-th arriving of part i (resp. of the n-th assemblage of parts), we obtain

$$y(n) = \tau + \max(u_1(n) + u_2(n)) = \tau \otimes (u_1(n) \oplus u_2(n))$$

in $(\max, +)$ algebra. More generally, any problem that can be modelled by a timed event graph (a subclass of timed Petri nets modelled synchronization features) can also be represented by a $(\min, +)$-linear dynamical system (for counter variables)

$$\begin{cases} x(t) = A \otimes x(t-1) \oplus B \otimes u(t), \\ y(t) = C \otimes x(t) \end{cases}$$

or by a $(\max, +)$-linear dynamical system (for dater variables $y(n)$, $x(n)$ and $u(n)$). A linear system theory in $(\min, +)$ and $(\max, +)$ algebras analogous to the classical linear control theory is developed in [2].

1.3. Exponential asymptotics. Let us consider a one-dimensional system of n atoms with energy $H_n(q_1, \ldots, q_n) = V(q_1) + \sum_{k=2}^n K(q_{k-1}, q_k)$, where q_n is the position (state) of the n-th atom with $q_1 < \cdots < q_n$ and $K(q, q') = V(q') + W(q' - q)$ is the sum of the potential V in position q and the potential energy W linking nearest neighbours. The Gibbs distribution of this system has density $\exp(-\beta H_n(q_1, \ldots, q_n))/Z_n$, where β is the inverse of the temperature and $Z_n = \sum_{q_1, \cdots, q_n} \exp(-\beta H_n(q_1, \ldots, q_n))$ is the partition function. Let T be the transfer matrix

$$T_{qq'} = \exp\left(-\beta K(q, q')\right),$$

Q be the row vector with entries $Q_q = \exp(-\beta V(q))$ and e the vector with entries 1. Then $Z_n = QT^{n-1}e$ and the probability for the first atom to be in position q is $P(q) = Q_q(T^{n-1}e)_q/Z_n$. For good matrices T, $P_n(q)$ tends to $P(q) = Q_q R_q$ when n goes to infinity, where R is a right eigenvector of the transfer matrix such that $Q \cdot R = 1$. Similarly, the probability of the n-th atom tends to L_q, where L is a left eigenvector of the transfer matrix such that $L \cdot e = 1$. Moreover, for any transfer matrix, $\log Z_n/n$ tends to $\log \rho$, where ρ is the Perron root of T. The free energy by atom is then $\lambda = \log \rho/\beta$.

If now the temperature is zero ($\beta = +\infty$), either the previous results have to be obtained passing to the limit in β using the property that the $(\min, +)$ algebra is the limit of the $(\mathbb{R}^+, +, \times)$ semifield:

$$\lim_{\beta \to +\infty} \frac{-1}{\beta} \log(e^{-\beta a} + e^{-\beta b}) = \min(a, b), \qquad \frac{-1}{\beta} \log(e^{-\beta a} \cdot e^{-\beta b}) = a + b;$$

or a similar reasoning has to be done directly in the $(\min, +)$ algebra. In this last case, the transfer matrix method is replaced by the effective potential method [5, 4]. Let us consider the $(\min, +)$-matrix K in place of T. The effective potential of the extremal atom of a semi-infinite chain of atoms extending to the right (resp. left) is equal to $F(q) = V(q) + R_q$ (resp. $F(q) = L_q$), where R and L are right and left $(\min, +)$-eigenvectors of K such that $\min_q V(q) + R_q = \min_q L_q = 0$. The energy by atom for a minimum-energy configuration is then the $(\min, +)$-eigenvalue λ of K: $K \otimes R = \lambda \otimes R = \lambda + R$, $L \otimes K = \lambda \otimes L = \lambda + L$. Exponential asymptotics also occur in large deviations and asymptotics of Schrödinger equations (WKB method) [29, 23].

1.4. Language theory. A finite automaton with cost or distance is an automaton with multiplicity over the tropical semiring $\mathbb{N}_{\min}$. For any rational language L over the finite alphabet Σ, a finite automaton with cost A can be constructed, recognizing $L^* = \cup_{n=0}^{\infty} L^n$ (where product means concatenation) and counting for each word $w \in L^*$ the least n such that $w \in L^n$. This has been used by Simon and Hashiguchi [21, 22, 33] to solve positively a long standing problem of J. A. Brzozowski, the decidability for a rational language of the finite power property (FPP) (a language L has the FPP iff there exists N such that $L^* = \cup_{n=0}^{N} L^n$). Indeed, the automaton A has only one initial state and one terminal state and since the language L has the FPP iff A is limited (that is costs of recognized words are bounded), the FPP is equivalent to the finite section property of a finitely generated subsemigroup of matrices of $\mathbb{N}_{\min}^{n \times n}$. Following this first application, other decidability properties for finitely generated subsemigroups of matrices over the tropical semiring and/or automata with cost have been studied [21, 22, 33, 34, 25, 24].

Similarly to cost automata, $(\max, +)$ automata can be also constructed. They allow to represent heaps of pieces and parallel (multitask, multiresource) discrete event systems [17, 16, 27].

2. $(\max, +)$ linear algebra

2.1. Solutions of linear equations and subsemimodules. Since the $\oplus$ law is not symmetrizable in a dioid, general linear equations are of the form $A \otimes x \oplus b = C \otimes x \oplus d$. Important particular cases are $A \otimes x = b$ and $x = A \otimes x \oplus b$. The following result is classical [7] and shows that the first particular equation is not easy to solve.

THEOREM 1. *$A \in \mathbb{R}_{\max}^{n \times n}$ is invertible iff $A = DS$, where D and S are diagonal and permutation matrices.*

THEOREM 2 ([30, 36]). *Any finitely generated subsemimodule of $\mathbb{R}_{\max}^{n}$ has a base (minimal generating family) which is unique up to invertible linear operations.*

THEOREM 3 ([3, 14]). *For any matrices $A, B \in \mathbb{R}_{\max}^{m \times n}$, the set of solutions of $A \otimes x = B \otimes x$ is a finitely generated semimodule.*

Let us solve $x = A \otimes x \oplus b$. To any dioid is associated a partial order: $a \preceq b \Leftrightarrow a \oplus b = b$. In $\mathbb{R}_{\max}$ it is the classical order $\leq$, in $\mathbb{R}_{\min}$ it is the opposite order $\geq$. The dioid $(\mathbb{S}, \oplus, \otimes)$ is complete if any set (even empty) has a least upper bound and if $\otimes$ is distributive with respect to infinite sums. $\mathbb{R}_{\max}$ is not complete but it may be completed in the complete dioid $\overline{\mathbb{R}}_{\max} = (\mathbb{R} \cup \{+\infty, -\infty\}, \max, +)$ with the convention $+\infty + -\infty = -\infty$ ($\mathbf{0}$ is absorbing).

THEOREM 4. *In a complete dioid $\mathbb{S}$, the least solution of $x = a \otimes x \oplus b$ is $a^* \otimes b$, where $a^* = \oplus_{n\in\mathbb{N}} a^n = \sup_{n\in\mathbb{N}} a^n$. Similarly, the least solution of $x = A \otimes x \oplus b$ in $\mathbb{S}^n$ is $x = A^*b$. It can be computed by Gauss algorithm.*

In order to solve the general equation $A \otimes x \oplus b = C \otimes x \oplus d$, a symmetrization of $\mathbb{R}_{\max}$ seems necessary. Although no idempotent field or ring containing $\mathbb{R}_{\max}$ exists, a symmetrized idempotent semiring $\mathbb{S}_{\max}$ has been constructed. It contains positive numbers $x \in \mathbb{R}_{\max}$, negative numbers $\ominus x$, but also doted numbers $\dot{x} = x \ominus x$ which are not invertible. Symmetrizing linear equations in $\mathbb{R}_{\max}$, we obtain balance equations in $\mathbb{S}_{\max}$, where x balances y iff $x \ominus y$ is doted. In $\mathbb{S}_{\max}$, determinants can be calculated and linear balance equations can be solved using Cramer formula or Gauss-Seidel and Jacobi algorithms [2, 14, 31].

2.2. Subsolutions of linear equations: residuation.

DEFINITION 1. Let $f : (E, \leq) \to (F, \leq)$ be a nondecreasing application between lattices. f is residuable iff $\{x \in E, f(x) \leq b\}$ has a maximal element for any $b \in F$.

THEOREM 5. *If $f : \mathbb{S} \to \mathbb{S}'$ is an application between complete dioids such that $f(\mathbf{0}) = \mathbf{0}$ and $f(\sup_{x\in X} x) = \sup_{x\in X} f(x)$ for any subset X of $\mathbb{S}$, then f is residuable.*

As a corollary, any multiplication operation (by a scalar or a matrix) is residuable. Let us denote by $a\backslash b = \max\{x, a \otimes x \preceq b\}$ and $b/a = \max\{x, x \otimes a \preceq b\}$ the residuations of multiplications by the scalar a in any complete dioid. The residuation of the multiplication by a matrix in $\mathbb{R}_{\max}$, $A\backslash b = \max\{x \in \mathbb{R}^n_{\max}, A \otimes x \preceq b\}$ gives the vector with entries $(A\backslash b)_i = \inf_j A_{ji}\backslash b_j = \min_j -A_{ji} + b_j$, that is the $\mathbb{R}_{\min}$ product of the matrix $-A^T$ by b. Applications to system theory can be found in [2]. While linear operators represent the earliest behaviour of a system, the latest behaviour can be represented by a dynamical equation involving residuation.

2.3. Spectral theory. The most useful result of $(\max, +)$ linear algebra is perhaps the following analogue of Perron-Frobenius theorem.

THEOREM 6 ([7, 35, 32, 18, 6, 10]). *Any irreducible matrix $A \in \mathbb{R}^{n\times n}_{\max}$ has a unique eigenvalue $\rho(A)$ and*

$$\rho(A) = \oplus_{k=1}^n \operatorname{tr}(A^k)^{\frac{1}{k}} = \max_{k=1,\dots,n} \max_{i_1,\dots,i_k} \frac{A_{i_1 i_2} + \cdots + A_{i_k i_1}}{k}$$

If A is reducible, the previous formula gives the maximal eigenvalue.

The $(\min, +)$ eigenvalue is then the minimal mean cost (ergodic cost) of a control problem or the asymptotic production rate of a manufacturing system, etc. As in the statistical physics application of section 1.3, it can be obtained as the limit of the Perron root of a matrix.

THEOREM 7 ([12, 11]). *Let A be any $n \times n$ matrix with entries in $\mathbb{R}^+$. If $\rho_{PF}(A)$ is the Perron-Frobenius root of A and $\rho_{(\max,\times)}(A) = \exp(\rho((\log A_{ij})))$ its $(\max, \times)$-eigenvalue, we have*

$$\rho_{(\max,\times)}(A) \leq \rho_{PF}(A) \leq n\rho_{(\max,\times)}(A).$$

COROLLARY 1. *Let $A^{\circ r} = (A^r_{ij})$ and $e^{\circ\beta A} = (\exp(\beta A_{ij})$ denote the r-th power of A and the exponential of βA for the Hadamard product. For any matrix with positive entries*

$$\rho_{(\max,\times)}(A) = \lim_{r\to+\infty} (\rho_{PF}(A^{\circ r}))^{\frac{1}{r}}$$

and for any matrix with entries in $\mathbb{R}_{\max}$

$$\rho(A) = \lim_{\beta\to+\infty} \frac{1}{\beta} \log \rho_{PF}(e^{\circ\beta A}).$$

THEOREM 8 ([6, 9]). *For any irreducible matrix* $A \in \mathbb{R}_{\max}^{n\times n}$*, there exists* c *and* $N \geq 1$ *such that* $A^{n+c} = \rho(A)^c A^n$ *for* $n \geq N$*.*

In the context of timed event graphs, this means that the system reaches after a finite transient behaviour (of length N) a periodic regime of period c in which the production rate is equal to the eigenvalue.

These periodicity results can also be dealt with using rational generating series over the $(\max, +)$ semiring [2, 15].

Bibliography

[1] Akian (M.), Quadrat (J. P.), and Viot (M.). – Duality between probability and optimization. In Gunawardena (J.) (editor), *Idempotency. Publication of the Isaac Newton Institute.* – Cambridge University Press, 1996. To appear.

[2] Baccelli (F.), Cohen (G.), Olsder (G. J.), and Quadrat (J. P.). – *Synchronization and Linearity.* – Wiley, 1992.

[3] Butkovic (P.) and Hegedűs (G.). – The elimination method for finding all solutions of the system of linear equations over an extremal algebra. *Ekonomicko-matematicky Obzor*, vol. 20, 1984.

[4] Chou (W.) and Griffiths (R. B.). – Effective potentials, a new approach and new results for one-dimensional systems with competing lenght scales. *Physical Review Letters*, vol. 56, 1986, pp. 1929–1931.

[5] Chou (W.) and Griffiths (R. B.). – Ground states of one dimensional systems using effective potentials. *Physical Review B*, vol. 34, 1986, pp. 6219–6234.

[6] Cohen (G.), Dubois (D.), Quadrat (J. P.), and Viot (M.). – *Analyse du comportement périodique des systèmes de production par la théorie des dioïdes.* – Rapport de recherche n° 191, Institut National de Recherche en Informatique et en Automatique, Le Chesnay, France, 1983.

[7] Cuninghame-Green (R. A.). – *Minimax Algebra.* – Springer Verlag, 1979, *Lecture notes in Economics and Mathematical Systems.*

[8] Del Moral (P.). – *Résolution particulaire des problèmes d'estimation et d'optimisation non-linéaires.* – Thèse, Université Paul Sabatier, Toulouse, 1994.

[9] Dudnikov (P.) and Samborskiĭ(S.). – Endomorphisms of finitely generated free semimodules. In Maslov (V.) and Samborskiĭ(S.) (editors), *Idempotent analysis.* – Americal Mathematical Society, Rhode Island, 1992.

[10] Dudnikov (P. I.) and Samborskiĭ(S. N.). – Spectra of endomorphisms of semimodules over semirings with an idempotent operation. *Soviet Mathematics Doklady*, vol. 40, n° 2, 1990, pp. 363–366.

[11] Elsner (L.), Johnson (C. R.), and Dias da Silva (J.). – The Perron root of a weighted geometric mean of nonnegative matrices. *Linear Multilinear Algebra*, vol. 24, 1988, pp. 1–13.

[12] Friedland (S.). – Limit eigenvalues of nonnegatives matrices. *Linear Algebra and Applications*, vol. 74, 1986, pp. 173–178.

[13] Gaubert (S.). – *Introduction aux systèmes dynamiques à événements discrets.* – Polycopié de cours donné à l'ENSTA, 1992.

[14] Gaubert (S.). – *Théorie des systèmes linéaires dans les dioïdes.* – Thèse, École des Mines de Paris, July 1992.

[15] Gaubert (S.). – Rational series over dioids and discrete event systems. In *Proceedings of the 11th Conference on Analysis and Optimization of Systems: Discrete Event Systems. Lecture notes in Control and Information Sciences.* – Sophia Antipolis, June 1994.

[16] Gaubert (S.). – Performance evaluation of (max,+) automata. *IEEE Transactions on Automatic Control*, vol. 40, n° 12, December 1995.

[17] Gaubert (S.) and Mairesse (J.). – Task resource systems and (max,+) automata. In Gunawardena (J.) (editor), *Idempotency.* – Cambridge University Press, March 1995. To appear in 1996.

[18] Gondran (M.) and Minoux (M.). – Valeurs propres et vecteurs propres en théorie des graphes. In *Problèmes combinatoires et théorie des graphes, Colloques internationaux CNRS.* – Orsay, 1976.

[19] Gondran (M.) and Minoux (M.). – *Graphes et algorithmes.* – Eyrolles, Paris, 1979. English translation *Graphs and Algorithms*, Wiley, 1984.

[20] Gunawardena (J.) (editor). – *Idempotency.* – Cambridge University Press, 1996, *Publications of the Newton Institute.* To appear.

[21] Hashiguchi (K.). – Limitedness theorem on finite automata with distance functions. *Journal of Computer and System Sciences*, vol. 24, n° 2, 1982, pp. 233–244.

[22] Hashiguchi (K.). – Improved limitedness theorems on finite automata with distance functions. *Theoretical Computer Science*, vol. 72, 1990, pp. 27–38.

[23] Kolokoltsov (V.) and Maslov (V.). – *Idempotent analysis and applications.* – Kluwer Academic Publisher, 1996. To appear.

[24] Krob (D.). – The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, vol. 3, 1993.

[25] Leung (H.). – Limitedness theorem on finite automata with distance function: an algebraic proof. *Theoretical Computer Science*, vol. 81, 1991, pp. 137–145.

[26] Litvinov (G.) and Maslov (V.). – *Correspondence principle for idempotent calculus and some computer applications.* – Technical Report n° IHES/M/95/33, IHES, Bures-sur-Yvette, France, April 1995.

[27] Mairesse (J.). – *Stabilité des systèmes à événements discrets stochastiques. Approche algébrique.* – Thèse, École Polytechnique, June 1995.

[28] Maslov (V.) and Samborskiĭ(S.) (editors). – *Idempotent analysis.* – American Mathematical Society, Rhode Island, 1992, *Advances in Soviet Mathematics*, vol. 13.

[29] Maslov (V. P.). – *Méthodes Opératorielles.* – Mir, Moscou, 1973. French translation 1987.

[30] Moller (P.). – *Théorie algébrique des Systèmes à Événements Discrets.* – Thèse, École des Mines de Paris, 1988.

[31] Plus (M.). – Linear systems in (max, +)-algebra. In *Proceedings of the 29th Conference on Decision and Control.* – Honolulu, December 1990.

[32] Romanovskiĭ(I. V.). – Optimization and stationary control of discrete deterministic process in dynamic programming. *Kibernetika*, vol. 2, 1967, pp. 66–78. – English translation in Cybernetics 3 (1967).

[33] Simon (I.). – Limited subsets of the free monoid. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science.* pp. 143–150. – IEEE, 1978.

[34] Simon (I.). – The nondeterministic complexity of a finite automaton. In Lothaire (M.) (editor), *Mots.* – Hermes, 1990.

[35] Vorobyev (N. N.). – Extremal algebra of positive matrices. *Elektronische Informationsverarbeitung und Kybernetik*, vol. 3, 1967. – In russian.

[36] Wagneur (E.). – Moduloids and pseudomodules. 1. dimension theory. *Discrete Mathematics*, vol. 98, 1991, pp. 57–73.

Computation with DNA

Alain Hénaut et Didier Contamine
Université Versailles-Saint-Quentin

March 25, 1996

[summary by Eithne Murray]

Abstract

In 1994 Leonard Adleman published a paper giving an algorithm to solve the Hamiltonian path problem using DNA manipulations and presented the results of an actual experiment applying this algorithm to a particular graph. The basic operations and the algorithm are described, and the potential of these methods as a means of computation is discussed briefly.

1. Introduction

Using basic techniques of DNA manipulation and standard lab equipment, Adleman finds a Hamiltonian path in a directed graph consisting of 7 nodes and 14 edges (figure 1). Finding such a path, that starts and ends at specified vertices while passing through every other vertex exactly once, is a problem which has no known polynomial time solution. In fact, this problem is NP-complete, and so it is considered unlikely that such a solution will exist. This is the first time biological methods have been used to solve hard computer problems, and it is still unknown to what extent the available DNA operations may be used to solve other problems.

2. Basic Operations

DNA manipulations form the basic operations operations of a DNA computer. It should be emphasized that the biological techniques presented here are routine laboratory procedures, and require no special equipment or expertise. Strands of DNA are made up of sequences of bases represented by the letters $\{A, C, G, T\}$. Each sequence has a (Watson-Crick) complementary sequence, that is, the sequence that binds with the original to form a double strand. In the complementary sequence, each base in the original is replaced by its complement ($A \leftrightarrow T$, $C \leftrightarrow G$).

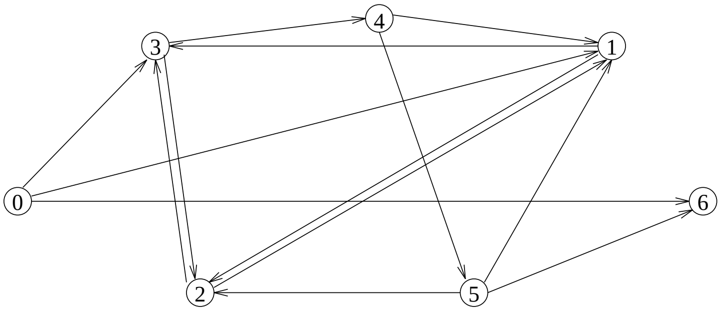


FIGURE 1. The directed graph used in the experiment.

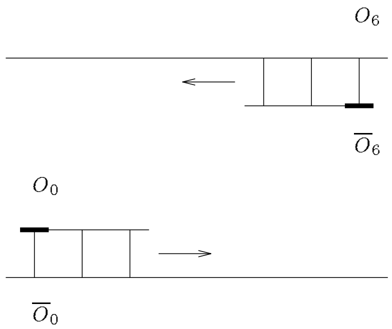


FIGURE 2. PCR - Polymerase Chain Reaction to replicate a specified segment of DNA

The following operations are available. Each one will be discussed briefly, without entering into too many technical details.

creation: A strand of DNA made up of a given sequence of bases can be created. These days, creating a specific short sequence is a matter of filling out the mail-order coupon, writing the check, and sending them off to the laboratory in the catalogue. Here "short" often means of length 20.
joining: Complementary strands will spontaneously join together to form a double strand. Strands can also be concatenated. If two strands are brought into juxtaposition because they have both joined to part of a third, complementary strand, then under the action of a ligase enzyme a bond forms between the first two strands so that they become a single longer strand. An example is found in figure 3. This bond persists even if the strand then separates from its complement.
copying: Many copies of a given strand of DNA can be created by polymerase chain reaction (PCR). The strand to be amplified is defined by two primers, which are segments of DNA. The primers are the start and the complement of the end of the sequence of interest. For example, say O_0 and O_6 are segments of DNA, and the problem is to create copies of every sequence of DNA in the test tube that contains O_0 followed by an unknown sequence of bases followed by O_6. Then the primers for this PCR are O_0 and $\overline{O}_6$, where the bar indicates the Watson-Crick complement. The amplification works roughly in the following way. Many copies of O_0 and $\overline{O}_6$ are added into the test tube. The mixture is heated, which causes the strands of DNA to separate. As it cools, the primers attach themselves where they can, that is, one to the beginning of the edges beginning with 0, the other to the end of the edges ending in 6. The primer then forms the start of a new chain that grows out from it, forwards from O_0 and backwards from $\overline{O}_6$, as shown in figure 2. This process is repeated, and the number of strands consisting of the segments of interest doubles each time. A few hours will suffice to have ample quantities of these strands, though in practise, the duration used is often "one night".
sorting: DNA strands can be sorted by length. This is achieved by gel electrophoresis, a process which involves separating the strands by their electrophoretic mobility, which is a function of the number of base pairs.
extraction: Strands containing a specific segment of DNA can be extracted from the test tube. Extraction is performed by separating the strands, and then using magnetic beads with a complement of the segment to be extracted attached to each bead. Only the DNA containing that segment will attach itself to the bead and be retained.
detection: The existence of DNA in a test tube is determined using PCR.

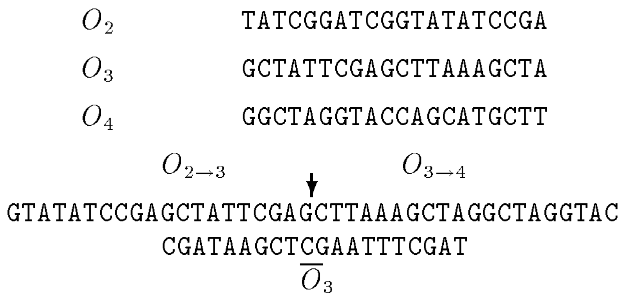

O_2 TATCGGATCGGTATATCCGA
O_3 GCTATTCGAGCTTAAAGCTA
O_4 GGCTAGGTACCAGCATGCTT
$O_{2\to3}$
$O_{3\to4}$
GTATATCCGAGCTATTCGAGCTTAAAGCTAGGCTAGGTAC
CGATAAGCTCGAATTTCGAT
$\overline{O}_3$

FIGURE 3. Encoding a graph in DNA. A path along the edges 2 → 3 and 3 → 4 is formed when each edge becomes attached to half of the complementary vertex $\overline{O}_3$ and a ligation reaction occurs.

3. The Algorithm

Adleman uses a naive brute-force algorithm. Given a directed graph on n vertices, where the path is to start at vertex v_{in} and finish at vertex v_{out}, the following steps will result in a solution if one exists.

(1) Input the graph (creation).
(2) Generate many many random paths through the graph (joining and copying).
(3) Keep only the paths that start at v_{in} and end at v_{out} (copying).
(4) Keep only those paths that enter exactly n vertices (sorting).
(5) Keep only those paths that enter all of the vertices at least once (extraction).
(6) If no paths remain, say “no”, otherwise say “yes”, and the remaining paths are solutions (detection, copying and sorting).

An ordinary computer would not normally attempt such an algorithm, due to the enormous numbers of cases to consider. Using DNA, these cases can be treated in parallel.

The algorithm is performed on the graph in figure 1, and the goal is to construct a path from 0 to 6 while passing through all the vertices exactly once. For convenience, the labels were chosen so that the solution is $0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6$, but of course this does not affect the difficulty of the problem. Obviously, in the case of this graph, the answer can be found by inspection. However, this experiment demonstrates the feasibility of the technique.

Each vertex of the graph is represented by a random 20 base sequence O_i. Using 20 bases means the chances of that sequence appearing elsewhere in the DNA is miniscule. The Watson-Crick complementary sequence is denoted $\overline{O}_i$. Each edge $i \to j$ in the graph is created by creating the 20-letter molecule that starts with the last ten bases of O_i and ends with the first 10 bases of O_j. This sequence is denoted $O_{i\to j}$.

Mixing together all the the edges $O_{i\to j}$ with $\overline{O}_i$ for $i = 1, \ldots, 5$ allows concatenations to occur that forms random paths through the graphs, as required by step 2. For instance, $O_{2\to3}$ and $O_{3\to4}$ are edges in the graph. These edges can be concatenated together by using $\overline{O}_3$ as a splint. This new molecule represents a path from $2 \to 3 \to 4$. See figure 3. Given the number of reactions and the number of molecules formed, it is statistically extremely likely that the Hamiltonian path will be created if it exists.

Step 3 is to keep only those random paths that start at 0 and end at 6. By “keep”, it is meant that these strands are copied so many times that the presence of other strands becomes statistically insignificant in comparison.

Step 4 is achieved by sorting the strands by length, and keeping those that are 140-base pairs long, and thus enter exactly 7 vertices.

In order to keep only the strands that enter each vertex at least once (step 5), first the strands containing O_1 are extracted. Next, those strands containing O_2 are extracted, then O_3 etc.

Then, for step 6, the presence or absence of DNA in the test tube is detected. If absent, there is no Hamiltonian path for this graph. If present, amplification by PCR is performed, first using primers O_0 and O_1 to create copies of the path between 0 and 1, then using O_0 and O_2 to create copies of the path between 0 and 2, etc. Then the lengths are determined. In this case, the length of the molecule starting at O_0 and ending at O_1 is 40, indicating that the vertex 1 comes directly after vertex 0 in the solution. Multiple solutions would show up as multiple lengths for each segment, and by determining the various second vertices from the lengths, these solutions could be separated. A picture in the article [1] shows the result of this step. The solution found is indeed $0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6$.

4. Extensions

Richard Lipton has proposed an algorithm consisting of DNA experiments to solve the satisfaction problem (SAT) [2]. Given a boolean formula involving n variables the problem is to assign values to the variables such that the expression evaluates to true. A graph representation of the problem is used, where each path through the graph gives an assignment to the variables. The paths are generated using the same techniques as before. Briefly, the first step is to extract the DNA that makes the first clause true, then extract the DNA that makes the second clause true, etc. The paths through the graph can also be interpreted as n-bit binary numbers, where x_i is true means the ith bit is a 1, false means 0. Thus any binary number can be stored as a DNA molecule.

5. Advantages of DNA Methods

Both these problems are NP-complete, and so there is no polynomial time algorithm to solve them on traditional computers, and little hope of finding one. The incredible parallelism of the DNA-techniques means that exhaustive searches through all the possibilities can be done relatively rapidly, and may be able to provide a solution to problems that traditional computers cannot solve. For instance, it is estimated that DNA methods may be able to solve the Hamiltonian path problem on graphs of up to 70 edges.

There are also less obvious advantages. DNA techniques are energy efficient. Approximately 2×10^{19} ligation operations per 1 joule of energy can be performed, versus 10^9 operations per joule for existing supercomputers. It is estimated that the energy cost of the other operations is similarly tiny in comparison. Finally, as a storage medium, nothing else comes close. Information can be stored in approximately 1 bit per cubic nanometer. In contrast, videotapes store information at 1 bit per 10^{12} cubic nanometers.

More investigation is needed to determine which kinds of problems can be handled by these methods. The probability and effect of errors during the operations needs to be studied, as well as the possibility of creating new basic operations. It is possible but not yet known if a DNA molecule could encode a Turing machine, where the actions of certain enzymes would perform the operations of the machine.

Bibliography

[1] Adelman (Leonard M.). – Molecular computation of solutions to combinatorial problems. *Science*, vol. 266, 1994, pp. 1021–1024.

[2] Lipton (Richard J.). – DNA solution of hard computational problems. *Science*, vol. 268, 1995, pp. 542–545.

Some applications of the Mellin Transform in Signal processing

Jacques Lévy-Véhel

Projet Fractales, Inria Rocquencourt

April 15, 1996

[summary by Julie Bestel]

Abstract

The Mellin transform has been used in signal processing as a tool to investigate scale invariance. We review some of the recent studies by Wornell [3] and Cohen [2].

1. Introduction and examples

Assume we need to classify ships from radar signals [4]. The echo can be more or less compressed, depending on the angle between the axis of the ship and that of the radar signal. Nevertheless, one would like to be able to compare several echoes with different extension or compression rate, in order to decide whether or not they belong to the same kind of ship. A first approach would be to interpolate the signals, so that they would live on supports of equal size. A second one is to use some kind of transform that would ignore *scale variations*. The Mellin transform fulfils such a requirement; more precisely, the moduli of the Mellin transform of a signal $f(x)$ and of any dilation of $f(x)$ are the same. If time invariance is furthermore required, one may perform the *Fourier-Mellin transform*: Given an original real signal $f(x)$, the analytical signal is defined by $f_a(x) = f(x) + if_h(x)$, where $f_h(x)$ is the Hilbert transform of $f(x)$. Let $\mathcal{F}(f_a)(\omega) = F(\omega)$ be the Fourier transform of f_a. The quantity

$$\left|G_{|F|^2}(ix)\right|^2 = \left|\int_0^{+\infty} \omega^{ix-1}|F(\omega)|^2\,d\omega\right|^2$$

is both shift and scale invariant on the x axis.

Section 2 gives a more detailed description of *scale invariant linear systems*. Section 3 presents a general framework for scale analysis.

2. Linear systems

If $x(t)$ is the input signal, a linear system outputs $y(t)$ as follows:

$$y(t) = S(x(t)) = \int_{-\infty}^{+\infty} x(\tau)K(t,\tau)\,d\tau$$

where $K(t,\tau)$ denotes the kernel of the system.

2.1. Shift invariant systems. As is well known, shift invariant systems are such that:

$$S(x(t-\tau)) = y(t-\tau) \Longleftrightarrow K(t,\tau) = V(t-\tau)$$

where V is the impulse response of the system, i.e., $V(t) = S(\delta(t))$. It follows that y is obtained by convolving x and V:

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)V(t-\tau)\,d\tau = (x \star V)(t).$$

The eigenfunctions of these systems are the exponential functions: $t \mapsto e^{st}$, $s \in \mathbb{C}$. The Laplace transform

$$\mathcal{L}(x)(s) = X(s) = \int_{-\infty}^{+\infty} x(t)e^{-st}\,dt$$

enables to change convolution into multiplication: $\mathcal{L}[(x \star y)](s) = X(s)Y(s)$.

2.2. Scale invariant systems. We are now interested in having $S(x(t/\tau)) = y(t/\tau)$. One can easily check that this is equivalent to $K(t,\tau) = aK(at, a\tau)$. The system S is characterized by two *lagged* impulse responses:

$$\xi_+(t) = S(\delta(t-1)), \qquad \xi_-(t) = S(\delta(t+1))$$

$$y(t) = \int_0^{+\infty} x(\tau)\xi_+(t/\tau)\frac{d\tau}{\tau} - \int_0^{+\infty} x(-\tau)\xi_-(t/\tau)\frac{d\tau}{\tau}.$$

For causal signals and systems with causal response,

$$y(t) = \int_0^{+\infty} x(\tau)\xi_+(t/\tau)\frac{d\tau}{\tau} = (x \diamond \xi)(t) \qquad \text{(scale convolution)}.$$

The kernel K is such that: $K(t,\tau) = \xi(t/\tau)/\tau$. The eigenfunctions of the operator thus defined are the functions $t \mapsto t^s$. The associated eigenvalue is the Mellin transform:

$$\mathcal{M}(x)(s) = M(s) = \int_0^{+\infty} \xi(\tau)\tau^{-s-1}\,d\tau.$$

We can then write: $\mathcal{M}[(x \diamond y)](s) = X(s)Y(s)$. The Mellin transform plays for scale convolution the role that the Laplace transform plays for ordinary convolution.

Application to scale differential equations. One defines the derivative with respect to the scale by:

$$\nabla_s(x)(t) = \lim_{\epsilon \to 1} \frac{x(\epsilon t) - x(t)}{\ln \epsilon}.$$

If x is differentiable with respect to t, $\nabla_s(x)(t) = tx'(t)$. One can check that the derivative with respect to scale corresponds to a multiplication by s in the Mellin domain.

2.3. Generalized scale invariance. More generally, one considers systems such that $S(x(t/\tau)) = \tau^\lambda y(t/\tau)$. This holds if and only if $K(t,\tau) = a^{-(\lambda-1)}K(at, a\tau)$. For causal signals, the *lagged impulse response* ξ_+ is such that:

$$y(t) = \int_0^{+\infty} x(\tau)\xi_+(t/\tau)\frac{d\tau}{\tau^{(1-\lambda)}}.$$

2.4. Jointly time and scale invariant systems. We now wish to have both

$$S(x(t-\tau)) = y(t-\tau) \qquad \text{and} \qquad S(x(t/\tau) = \tau^{\lambda} y(t/\tau).$$

One can show that the kernel should be a generalized homogeneous function of degree $\lambda - 1$:

$$v(t) = a^{-(\lambda-1)} v(at)$$

Hence,

$$v(t) = \begin{cases} C_1 |t|^{\lambda-1} u(t) + C_2 |t|^{\lambda-1} u(-t), & \text{if } -\lambda \notin \mathbb{N}, \\ C_1 |t|^{\lambda-1} u(t) + C_2 |t|^{\lambda-1} u(-t) + C_3 \delta^{(n)}(t), & \text{otherwise,} \end{cases}$$

where the C_i are constants and $u(t)$ is the Heaviside function.

3. The scale representation

The starting point of this approach [2] is the following simple remark:

- The *content* of the signal x at time t is nothing but $x(t)$;
- the *content* of the signal x at frequency f is given by its Fourier transform $X(f)$.

Our purpose is then to define the concept of scale and the *content* of the signal x at scale c. The idea consists in associating a *physical* quantity a with an Hermitian operator $\mathcal{A}$. Let us begin with *common physical* quantities: time and frequency. The operators T and F respectively associated with t and f are:

$$T : x(t) \mapsto tx(t), \qquad F : x(t) \mapsto -i\frac{dx}{dt}.$$

In the frequency domain, we obtain:

$$T : X(f) \mapsto i\frac{dX}{df}, \qquad F : X(f) \mapsto fX(f).$$

It should be noticed that T and F do not commute:

$$[T, F] = TF - FT = i.$$

This is the reason why we get an incertitude principle on t and f. We now define the scale operator as follows:

$$\mathcal{C} = \frac{1}{2}(TF + FT).$$

The following relations justify this definition:

$$e^{i\sigma\mathcal{C}} x(t) = e^{\sigma/2} x(e^{\sigma} t), \qquad e^{i\sigma\mathcal{C}} X(f) = e^{-\sigma/2} x(e^{-\sigma} f).$$

Whereas

$$\begin{aligned} e^{i\tau F} x(t) &= x(t+\tau), & e^{i\theta T} X(f) &= X(f-\theta), \\ [T, \mathcal{C}] &= T\mathcal{C} - \mathcal{C}T = iT, & [T, F] &= F\mathcal{C} - \mathcal{C}F = -iF. \end{aligned}$$

Therefore, there exists an incertitude relation between scale and time, or between scale and frequency:

$$\Delta c \Delta t \geq \frac{1}{2} |\langle t \rangle|$$

where the average time is defined by

$$\langle t \rangle = \int t |x(t)|^2 \, dt.$$

The equality is reached for the signal:

$$x(t) = kt^{\alpha} \exp\left[-\beta t + i \langle c\rangle \ln\left(\frac{t}{\langle t\rangle}\right)\right].$$

Dually, we get $\Delta f \Delta c \geq \frac{1}{2} |\langle f\rangle|$.

Let $\gamma(c,t)$ be the eigenfunctions of $\mathcal{C}$: $\mathcal{C}\gamma(c,t) = c\gamma(c,t)$. We find, for $t \geq 0$:

$$\gamma(c,t) = \frac{1}{\sqrt{2\pi}} t^{ic-\frac{1}{2}}.$$

We can now produce the direct and inverse transforms, for $t \geq 0$:

$$D(c) = \int x(t)\gamma^*(c,t)\,dt = \frac{1}{\sqrt{2\pi}} \int_0^{+\infty} x(t) t^{-ic-\frac{1}{2}}\,dt,$$
$$x(t) = \int D(c)\gamma(c,t)\,dc = \frac{1}{\sqrt{2\pi}} \int_0^{+\infty} D(c) t^{ic-\frac{1}{2}}\,dc.$$

One can notice that we have recovered a Mellin transform, in the special case when $\Re(s) = \frac{1}{2}$. That is why the Mellin transform was commonly renamed *Scale transform* in signal processing.

The average scale of a signal is given by: $\langle c\rangle = \int c|D(c)|^2\,dc$. One obtains

$$\langle c\rangle = \int_0^{+\infty} t\phi'(t)|x(t)|^2\,dt = \int_{-\infty}^{+\infty} f\psi'(f)|X(f)|^2\,df.$$

One can deduce from these relations a notion of *instantaneous scale*, at time t: $c_t = t\phi'(t)$, and at frequency f: $c_f = -f\psi'(f)$.

A more unified presentation can be found in [1, 2].

Bibliography

[1] Baraniuk (Richard G.) and Jones (Douglas L.). – Unitary equivalence: A new twist on signal processing. *IEEE Transactions on Signal processing*, 1996. – To appear.

[2] Cohen (Leon). – *Time-frequency analysis.* – Englewood Cliffs, 1995.

[3] Wornell (Gregory). – *Signal processing with fractals: A wavelet-based approach.* – Prentice Hall, Upper Saddle River, NJ, 1995.

[4] Zwicke (Philip E.) and Kiss (Imre). – A new implementation of the Mellin transform and its application to radar classification of ships. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 5, n° 2, March 1983.

A general upper bound for the satisfiability threshold of random r-SAT formulæ

Olivier Dubois
LAFORIA, CNRS et Université Paris 6, France

October 23, 1995

[summary by Danièle Gardy]

Abstract

It is well known that the general problem of checking the satisfiability of a set of clauses is NP-complete. Experimentations have shown that there is a threshold on the ratio "number of clauses/number of variables" that separates the set of clauses for which a solution can be (easily) found from those for which it is impossible to find a solution. The subject of this talk is the r-SAT problem, in which the clauses have a constant number r of literals. This summary is based on [2].

1. The problem

A literal is either a boolean variable x_i or its negation $\bar{x}_i$. A *clause* is a disjunction of literals over a set of boolean variables; for example $x_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_5$ is a clause on the literals $x_1, \ldots, \bar{x}_5$. A formula is a finite set of clauses, or equivalently a conjunction of clauses. The *satisfiability problem* is to determine whether there exists a truth assignment (each literal is assigned a value *true* or *false*) satisfying a given formula. This famous problem is *NP*-complete as soon as the number n of literals is at least equal to 3; it was the first problem to be proved so [3, 4].

If we cannot find an algorithm that is guaranteed to work in polynomial time (worst-case complexity), what about the average complexityΓ This natural question leads to the notion of random clauses. The first point is to define a model of random clauses, i.e. a probability law on the set of all possible clauses on n literals. Two approaches have been attempted (in both, clauses are chosen independently of each other):

(1) *Constant density:* The literal x_i is present in a clause with probability p_i, its negation $\bar{x}_i$ is present with probability q_i, and the probability that neither x_i nor $\bar{x}_i$ are present is equal to $1 - p_i - q_i$.

(2) *Constant length:* The problem is restricted to all clauses of given length r; there are $C = 2^r\binom{n}{r}$ such clauses, and the probability distribution on this set is uniform: Each clause is chosen with a probability $1/C$.

We choose m clauses amongst C, with replacement. The first model leads to clauses of variable length; an easy analysis shows that, when the number m of clauses and the number n of variables are polynomially related, almost every formula is satisfiable.

The model under active study is the second one, the so-called r-SAT problem. Simulations have shown the importance of the ratio c_r = *Number of clauses/Number of variables:* If c_r is smaller than some threshold value, the probability of finding an assignment of the variables that satisfies the formula is close to 1 for $n, m \to +\infty$; if c_r is larger than this threshold, the probability of finding an assignment that satisfies the set of clauses is close to 0 for $n, m \to +\infty$. This threshold is an increasing function of r; experiments lead to believe that the value for $r = 3$ is $\rho = 4.25...$ Moreover, the backtracking algorithms used to solve r-SAT behave differently according to the ratio c_r. Experimentally, the difficulty of either finding an assignment satisfying a formula or proving that a formula is unsatisfiable is exponentially greater when c_r is close to the threshold than when it is either lower or greater.

The theoretical proof of the existence of a threshold value for the ratio $c_r = n/m$ lags behind. For 3-SAT, the best lower bound presently is 3.003, and the best upper bound is 4.64... (a result established precisely by Dubois and Boufkhad, and presented in this talk). There remains a gap between 3.003 and 4.64..., around the observed threshold 4.25.

2. The result

The main result is as follows:

> A random r-SAT formula ($r \geq 3$) is unsatisfiable with probability asymptotically close to 1, when $n \to +\infty$, as soon as $c_r := m/n$ is at least equal to some specified value $c_{r,min}$.

This lower bound $c_{r,min}$ is defined in terms of the solution x_0 of a transcendental equation, and can be computed numerically with the help of a Computer Algebra System. For $r = 3$, we get $x_0 = 1.924714266...$, which gives the bound $c_r \leq 4.642476157...$

For $r \geq 4$, the bound obtained by Dubois and Boufkhad improves on the general upper bound $c_r \leq -\log 2/\log(1 - 2^{-r})$. For example, with $r = 4$, some minutes of experiment with Maple give $x_0 = 2.69945696....$ and $c_4 \leq 10.2168796...$, which is a slight improvement on the known bound $c_r \leq -\log 2/\log(1 - 2^{-r}) = 10.74005367...$ For $r = 5$, we obtained $x_0 = 3.429641...$ and $c_r \leq 21.32022...$, which is still slightly better than the known bound $c_r \leq -\log 2/\log(1 - 2^{-r}) = 21.83230235...$ For $r = 10$, the known bound gives $c_r \leq 709.436...$, and Dubois's method gives $x_0 = 6.92993239...$ and $c_r \leq 708.935...$ These computations also show that the gain becomes marginal for large r. However, experiments seem to indicate that the difference between the bound $-\log 2/\log(1 - 2^{-r})$ and the threshold is slowly varying, and that the accuracy of the bound of Dubois and Boufkhad actually increases.

3. The proof

The proof relies on the existence of a special type of solutions, called *negatively prime solutions* (NPS), which are defined below, and to which is applied the method of the first moment. The idea behind this method is simple. To show that some problem has no solution, define X as the number of solutions of a random instance and show that the expectation $E[X]$ can be made as close to 0 as desired. This argument, applied to the r-SAT problem, leads to the following reasoning:

- Show that every satisfiable formula has at least one NPS (easy). The average number of NPS of a satisfiable formula is then at least 1.
- Compute the expectation $E[\text{NPS}]$ of the number of NPS on the set of random formulæ with n variables and m clauses.
- If $E[\text{NPS}] = 0$ then a random formula has no negatively prime solution, hence no solution.
- Then we should compute $E[\text{NPS}]$ and study its asymptotic behaviour as $n, m \to +\infty$ with $n/m = c_r$.

3.1. Negatively prime solutions. A solution of a formula F is defined as a set of n literals, each variable appearing either as x_i or as $\bar{x}_i$, such that the assignment of *true* to these literals satisfies F. A *negatively prime solution* is a solution such that, if we substitute x_i for a negative literal $\bar{x}_i$, the resulting set is no longer a solution of F.

It is easy to see that each solution of F either is a NPS, or leads to a NPS (by inverting negative literals as long as possible). Thus the number of solutions of F is greater than or equal to the number of negatively prime solutions; the same holds for expectations, and the method of the first moment, when applied to the number of NPS, will give a better bound than when applied to the number of solutions, as for example in [1, 5].

It is possible to define a *positively prime solution* (PPS) in a similar way (an assignment minimal for the substitution of $\bar{x}_i$ to x_i); as $E[\text{NPS}] = E[\text{PPS}]$, the bound obtained is exactly the same.

3.2. The expectation E[NPS]. Dubois and Boufkhad show that

$$E[\text{NPS}] = \sum_{0 \le i \le j \le n} 2^{i-rm} \binom{n}{i} \binom{m}{j} i!\, S_{j,i} \left(\frac{r}{n}\right)^j (2^r - 1 - r)^{m-j}.$$

In this formula, $S_{j,i}$ is a Stirling number of second kind: $S_{j,i}$ is the number of ways to partition a set of j elements into i nonempty subsets.

In passing, they also remark that *for any set of literals* $\{l_i, i = 1, \ldots, n\}$ *($l_i = x_i$ or $l_i = \bar{x}_i$), there exists at least one formula that has this set as a NPS.*

The next step is to get an upper bound on E[NPS], using a bound on Stirling numbers due to Temme [6]:

$$E[\text{NPS}] \le \left(\frac{2^r - r - 1}{2^r}\right)^{nc_r} + c_r \sqrt{2\pi} n^{5/2} e^{1/12n} A^n (1 + o(1)),$$

with A defined as the maximum of some function. The first term of the r.h.s. is $o(1)$ when $n \to +\infty$; the behaviour of the second term (and of the upper bound) is given by A^n. Then a concavity argument is used to prove that $A < 1$ for c_r greater than a value $c_{r,min}$ that can be precisely defined. This shows that, for $m/n > c_{r,min}$, $E[\text{NPS}] \to 0$, i.e. a random formula cannot be satisfied.

This approach does not give any information for $m/n < c_{r,min}$; however a closer analysis (done by the authors, but not presented in [2]) shows that $E[\text{NPS}] \ge Q(n)A^n$, with a polynomial factor $Q(n)$, and the same exponential basis A; hence E[NPS] is of exponential order A^n.

Bibliography

[1] de la Vega (W. F.) and El Maftouhi (A.). – On random 3-sat. *Combinatorics, Probability and Computing*, 1995, pp. 189–195.

[2] Dubois (O.) and Boufkhad (Y.). – *A general upper bound for the satisfiability threshold of random r-SAT formulæ.* – Technical report, LAFORIA, CNRS and University of Paris 6, 1996.

[3] Garey (M. R.) and Johnson (D. S.). – *Computers and intractability : A guide to the theory of NP-Completeness.* – Freeman, San Francisco, 1979.

[4] Johnson (D. S.). – *A catalog of complexity classes*, Chapter 2, pp. 67–161. – Elsevier, 1990.

[5] Kamath (A.), Motwani (R.), Palem (K.), and Spirakis (P.). – Tail bounds for occupancy and the satisfiability threshold conjecture. In *35th Annual Symposium on Foundations of Computer Science*, pp. 592–603. – 1994.

[6] Temme (N. M.). – Asymptotic estimates of Stirling numbers. *Studies in Applied Mathematics*, vol. LXXXIX, n° 3, 1993, pp. 233–244.

Contents

Part 1
Combinatorics

Part 2
Symbolic Computation

Part 3
Asymptotic Analysis

Part 4
Analysis of Algorithms and Data Structures

Part 5
Miscellany