PHILIPPE FLAJOLET'S RESEARCH IN ANALYSIS OF ALGORITHMS AND COMBINATORICS

HELMUT PRODINGER AND WOJCIECH SZPANKOWSKI

ABSTRACT. Philippe Flajolet's research in theoretical computer science spans over more than twenty years. He made lasting contributions to analysis of algorithms and analytic combinatorics. Among many of his results we mention here some in such diversified topics as enumeration, number theory, formal languages, continued fractions, automatic analysis of algorithms, Mellin transform, digital sums, recurrences, trees, random generation of combinatorial objects, random graphs and mappings, polynomial factorization, communications, codes, graphics, etc. This gives only a small snapshot of his work, and we encourage the reader to visit Flajolet's homepage `http://www-rocq.inria.fr/algo/flajolet/index.html` for a fuller account. The bibliography was taken from his homepage and may not be totally complete, although we added a few items. Our paper was written without giving any prior notice to Philippe Flajolet. It reflects the view of the authors and any misunderstandings and shortcomings should be put on their account.

1. Register function and Gray code

We start with a section about the register function and Gray codes since Flajolet's research in the analysis of algorithms started with the papers [13, 20] about the register function of binary trees (see also [58] for an extension to more general trees). The register function $\mathrm{reg}(t)$ of a binary tree is the minimal number of auxiliary registers to evaluate the tree t. If t has subtrees t_1 and t_2, then

$$\mathrm{reg}(t) = \begin{cases} \max\{\mathrm{reg}(t_1), \mathrm{reg}(t_2)\} & \text{if } \mathrm{reg}(t_1) \neq \mathrm{reg}(t_2) \\ 1 + \mathrm{reg}(t_1) & \text{if } \mathrm{reg}(t_1) = \mathrm{reg}(t_2), \end{cases}$$

with the initial value $\mathrm{reg}(\Box) = 0$.

Assume now that all binary trees with n internal nodes (=operators) are equally likely (there are $\frac{1}{n+1}\binom{2n}{n}$ of them), then the problem is to compute the average register function R_n. It turns out to be

$$R_n = (n+1)\sum_{k\geq 1} v_2(k)\frac{\binom{2n}{n+1-k} - 2\binom{2n}{n-k} + \binom{2n}{n-1-k}}{\binom{2n}{n}},$$

with the interesting function $v_2(k)$, being defined to be the largest j such that 2^j divides k. By partial summation, $\sum_{k\leq n} v_2(k) = n - \nu(n)$, where $\nu(n)$ is the binary sum of digits function. Delange proved that

$$\frac{1}{n}\sum_{k=0}^{n-1}\nu(k) = \frac{1}{2}n\log_2 n + F(\log_2 n) + o(1),$$

Date: May 5, 1998.

with an explicit periodic function $F(x)$. Using this and an approximation of the fourth differences of binomial coefficients via Hermite polynomials, the final summations are carried out along the lines of the Euler–MacLaurin summation formula. The final result is

$$R_n = \log_4 n + \frac{1}{2} - \frac{1}{\log 2} - \frac{\gamma}{2\log 2} + \delta(\log_4 n) + o(1);$$

again with an explicit periodic function $\delta(x)$.

The same methodology was used in the analysis of odd–even merge [27], where essentially the sum of digits function $\nu(n)$ is to be replaced by the number of ones in the so called Gray code representation.

2. Singularity analysis of generating functions

The papers [26, 33, 106] deal with the average height of binary trees (the maximal length of a path from the root to a leaf). The problem reduces to a study of the following quadratic recurrence

$$e_{h+1}(z) = \bigl(1 - \sqrt{1-4z}\,\bigr)e_h(z)\bigl(1 - e_h(z)\bigr).$$

A very careful analysis of $e_h(z)$ in the neighbourhood of the dominant singularity $z = \frac{1}{4}$ was performed using ideas from de Bruijn. Consequently, it is was shown that a certain generating function of interest has a logarithmic behaviour close to $z = \frac{1}{4}$, from which it follows that the average height is asymptotically $2\sqrt{\pi n}$.

The basic idea of the transfer from the behaviour of generating functions in the vicinity of their dominant singularities to the behaviour of its (Taylor) coefficients was systematically developed in the paper *Singularity Analysis of Generating Functions* [88] which is a major accomplishment and found a multitude of applications in the analysis of algorithms:

Let $A(z)$ be the ordinary generating function of a sequence $\{a_n\}_{n=0}^{\infty}$. We start with the observation that $[z^n]A(z) = \rho^n[z^n]A(z/\rho)$, that is, we need only to study singularities at, say, $z = 1$. The next observation deals with asymptotics of $(1-z)^{-\alpha}$, for α not being a negative integer. When α is a natural number, the function $(1-z)^{-\alpha}$ has a pole of order α at $z = 1$. However, when $\alpha \neq 1, 2, \ldots$, then the function has an *algebraic singularity* (in fact, it is then a multi-valued function). Luckily enough, we can proceed formally as follows:

$$\begin{aligned}[z^n](1-z)^{-\alpha} &= \binom{n+\alpha-1}{n} = \frac{\Gamma(n+\alpha)}{\Gamma(\alpha)\Gamma(n+1)} \\ &= \frac{n^{\alpha-1}}{\Gamma(\alpha)}\left(1 + \frac{\alpha(\alpha-1)}{2n} + O\Bigl(\frac{1}{n^2}\Bigr)\right) .\end{aligned}$$

In the above, $\Gamma(x) = \int_0^\infty e^{-t}x^{t-1}dx$ is Euler's Gamma function, and the latter asymptotic expansion follows from Stirling's formula. Even more generally, let

$$A(z) = (1-z)^{-\alpha}\left(\frac{1}{z}\log\frac{1}{1-z}\right)^{\beta} .$$

Then, as shown by Flajolet and Odlyzko [88]

$$a_n = [z^n]A(z) = \frac{n^{\alpha-1}}{\Gamma(\alpha)}\left(1 + C_1\frac{\beta}{\log n} + C_2\frac{\beta(\beta-1)}{2\log^2 n} + O\Bigl(\frac{1}{\log^3 n}\Bigr)\right)$$

provided $\alpha \notin \{0, -1, -2, \ldots\}$, and C_1 and C_2 are constants that can be calculated explicitly.

The most important aspect of singularity analysis comes next: In many instances we do *not* have an explicit expression for the generating function $A(z)$ but only an expansion of $A(z)$ around a singularity. For example: let $A(z) = (1-z)^{-\alpha} + O(B(z))$. In order to get information about the coefficients a_n of $A(z)$ we need a "transfer theorem" that will allow us to pass to the coefficients of $B(z)$ under the "Big Oh" notation.

To be more specific we need a definition of Δ-analyticity around the singularity $z = 1$:

$$\Delta = \{z : \ |z| < R, \ z \neq 1, \ |\arg(z-1)| > \phi\}$$

for some $R > 1$ and $0 < \phi < \pi/2$ (i.e., the domain Δ is an extended disk around $z = 1$ with a circular part rooted at $z = 1$ deleted). A typical result is then the following theorem.

Theorem 2.1 (Flajolet and Odlyzko 1990). *Let $A(z)$ be Δ-analytical that satisfies in a neighbourhood of $z = 1$*

$$A(z) = O\left((1-z)^{-\alpha}\log^{\beta}(1-z)^{-1}\right).$$

Then

$$[z^n]A(z) = O\left(n^{\alpha-1}\log^{\beta} n\right).$$

There are several theorems of this flavor in [88], for instance replacing the "$O(\ldots)$" by "$o(\ldots)$" in the above formulation.

We will see several other applications in the analysis of algorithms and analytic combinatorics in the following sections.

3. Applications of the Mellin Transform

The first important application of the Mellin transform in Computer Science (Discrete Mathematics) is — to the best of our knowledge — an outcome of the cooperation of de Bruijn and Knuth who in the mid 60's analyzed the height of planar trees and the path length of digital trees. Philippe Flajolet together with his numerous collaborators use the Mellin transform systematically within a wide range of problems in the analysis of algorithms and analytic combinatorics. Excellent surveys arose out of this collaborations, serving also as handbooks for potential analysts; we mention here [53, 118, 119, 123, 127].

We recall that for a real-valued function $f(x)$ defined on $(0, \infty)$ the Mellin transform is defined as

$$f^*(s) = \int_0^\infty f(x)x^{s-1}dx$$

provided the above integral exists. The following subsections reflect the various survey papers on the subject. However, Mellin transforms will also appear in some other sections of the present paper.

Harmonic sums. Sums that occur frequently in the analysis of algorithms like

$$\sum_{k\geq 1} d(k)e^{-kx}, \quad \sum_{k\geq 1} e^{-k^2x}, \ldots$$

are typical examples of a *harmonic sum*

$$\sum_k a_k f(b_k x)$$

whose Mellin transform becomes

$$\sum_k a_k b_k^{-s} f^*(s).$$

From the inversion formula of the Mellin transform one obtains

$$\sum_k a_k f(b_k x) = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} \sum_k a_k b_k^{-s} f^*(s) x^{-s} ds \quad \text{for some real } c \ .$$

Shifting the line of integration and collecting appropriate residues leads to the desired asymptotics of the harmonic sum.

Digital sums. Sums as in the result of Delange can be treated conveniently by the Mellin–Perron summation formula [119]; a particular case that is very useful in applications is

$$\sum_{1\le k<n} \lambda_k (n-k) = \frac{n}{2\pi i}\int_{c-i\infty}^{c+i\infty} \left(\sum_{k\ge 1}\frac{\lambda_k}{k^s}\right) n^s \frac{ds}{s(s+1)}$$

where λ_k are some coefficients. Then, the sum–of–digits function (cf. Section 1)

$$S(n) = \sum_{k<n} \nu(k)$$

has the following representation

$$S(n) = \frac{n(n-1)}{2} - \frac{n}{2\pi i}\int_{2-i\infty}^{2+i\infty} \frac{\zeta(s)}{2^s-1} n^s \frac{ds}{s(s+1)}.$$

Observing that

$$0 = \int_{-\frac14-i\infty}^{-\frac14+i\infty} \zeta(s) n^s \frac{ds}{s(s+1)}$$

Delange's explicit formula for $S(n)$ is most easily rederived by shifting the line of integration to the left and collecting some residues.

But this technique gives much more: Counting of subblocks in the q–ary representation, summatory functions in residue classes (à la Cocquet), etc. are possible. An open problem about odd numbers in Pascal's triangle, viz.

$$\Phi(n) = \sum_{0\le k<n} 2^{\nu(k)}$$

can be solved, since it turns out that

$$\Phi(n) = n^{\log_2 3} F(\log_2 n),$$

with a periodic function $F(x)$ whose mean is ≈ 0.86360.

Divide and Conquer Recursions. A typical recursion

$$U(n) = U\left(\left\lfloor \tfrac{n}{2} \right\rfloor\right) + U\left(\left\lceil \tfrac{n}{2} \right\rceil\right) + n - \gamma_n$$

with

$$\gamma_n = \frac{\left\lfloor \frac{n}{2} \right\rfloor}{\left\lfloor \frac{n}{2} \right\rfloor + 1} + \frac{\left\lceil \frac{n}{2} \right\rceil}{\left\lceil \frac{n}{2} \right\rceil + 1}$$

occurs in the analysis of the average number of comparisons of the (top–down) Mergesort algorithm. More generally, consider the recursion $f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + e_n$ and define $\Delta f_m = f_{m+1} - f_m$, $\nabla f_m = f_m - f_{m-1}$,

$$W(s) = \sum_{n \ge 1} \frac{\Delta \nabla f_n}{n^s},$$

then Mellin's inversion formula leads to

$$f_n = n f_1 + \frac{n}{2\pi i} \int_{3-i\infty}^{3+i\infty} \frac{\Xi(s) n^s}{1 - 2^{-s}} \frac{ds}{s(s+1)},$$

where

$$\Xi(s) = \Delta \nabla f_1 + \sum_{n \ge 2} \frac{\Delta \nabla e_n}{n^s},$$

provided that $e_n = O(n)$. Shifting the line of integration to the left and collecting the residues gives then the desired asymptotics. In particular,

$$U(n) = n \log_2 n + n B(\log_2 n) + O(n^{\varepsilon}),$$

where $B(x)$ is again a periodic function with mean

$$\frac{1}{2} - \frac{1}{\log 2} - \frac{1}{\log 2} \sum_{m \ge 1} \frac{2}{(m+1)(m+2)} \log\left(\frac{2m+1}{2m}\right) = -1.24815.$$

This and more can be found in [107, 118].

4. Alternating sums and Rice's method

In [62] an open problem from Knuth's *The Art of Computer Programming: Sorting and Searching* was solved concerning the number of endnodes in a digital search tree. Digital search trees are built over a sequence of n data, to each of which an infinite bit string is associated. The first data is stored in the root of a binary tree, and consecutive nodes go down to the left/right according to whether to actual bit is 0/1, until they find an available space. The question was to compute the average number E_n of endnodes (having no descendants), assuming that n random data are given. In [62] the following representation is first derived, using e. g. formulæ traditionally belonging to the theory of partitions;

$$E_n = \sum_{k=2}^{n} \binom{n}{k} (-1)^k R_k,$$

with a complicated but explicit quantity R_k involving products like $Q(x) = \prod_{k\ge1}(1 - x2^{-k})$. The second part then deals with asymptotics; it was observed that the method of Rice is very well suited to this kind of sums. It expresses the sum as a contour integral, viz.

$$\sum_{k=2}^{n} \binom{n}{k} (-1)^k R_k = \frac{1}{2\pi i} \int_{\mathcal{C}} \frac{\Gamma(n+1)\Gamma(-z)}{\Gamma(n+1-z)} R(z) dz, \tag{1}$$

where $R(z)$ is a meromorphic continuation of the sequence R_k. In this instance it is not quite trivial to find it. Next, one extends the range of integration, which originally enclosed the points $2, \ldots, n$, to a larger domain, and collects the extra residues. They in turn are the desired terms in the asymptotic expansion for E_n. In this way it was shown that

$$E_n = n\Big(\beta + \delta(\log_2 n)\Big) + o(1)$$

with a constant $\beta = 0.37204$ and a small oscillating term $\delta(x)$.

Over the years, Rice's method found many applications. (In general, the summation as in (1) goes from $k = m, \ldots, n$, and a sequence f_k is extrapolated to a function $f(z)$.)

In many applications the function $f(s)$ has only poles which are easily to handle by the residue theorem. The survey [127] shows that in this instance we can even get explicit formulæ and explains how to deal with e. g. algebraic and logarithmic functions.

5. Some other open problems from Knuth's books

The open problems proposed by D. E. Knuth in his *The Art of Computer Programming* were always a challenge for Philippe Flajolet. We have already discussed some examples and sketch a few others in the sequel.

Storage Allocation Problem. The paper [57] considers what is perhaps the simplest *storage allocation algorithm.* Assume that two stacks are to be maintained inside a shared (contiguous) memory area of fixed size m. A trivial algorithm (dealing with insertions and deletions) will let them grow from both ends until their cumulated sizes fill the initially allocated storage (m cells), and the algorithm stops having exhausted its available memory. Knuth proposed to analyze this problem in Volume 1 of his book.

Probability distributions and expectations of characteristic parameters are obtained in [57]. The problem can also be formulated in terms of a random walk in a triangle with the axes as reflecting barriers (to touch an axis means to have an empty stack) and an absorbing barrier (a diagonal, describing the exhaustion of memory space). One–dimensional analogs can be described by continued fractions and Fibonacci (Chebyshev) polynomials. The two–dimensional case is obtained by merging two one–dimensional instances, whence we have to deal with the Laplace–Borel transform (switching between ordinary and exponential generating functions).

Ramanujan's Q-function. A classical open problem of Ramanujan deals with the function ('Ramanujan's Q–function')

$$Q(n) = 1 + \frac{n-1}{n} + \frac{(n-1)n}{n^2} + \frac{(n-2)(n-1)n}{n^3} + \cdots .$$

It may be rephrased as follows: Let

$$R(n) = 1 + \frac{n}{n+1} + \frac{n^2}{(n+1)(n+2)} + \frac{n^3}{(n+1)(n+2)(n+3)} + \cdots .$$

Then

$$D(n) := R(n) - Q(n) = \frac{2}{3} + \frac{8}{135(n+k)} \quad \text{with} \quad \frac{2}{21} \le k \le \frac{8}{45}.$$

In [124] this conjecture was proved. The key step is the contour integral

$$D(n) = \frac{n!}{n^{n-1}} \frac{1}{2\pi i} \oint_{\mathcal{C}} \log \frac{(1-T(z))^2}{2(1-ez)} \frac{dz}{z^{n+1}}$$

with the classical *tree function* $T(z)$ defined as

$$T(z) = ze^{T(z)} = \sum_{n \ge 1} n^{n-1} \frac{z^n}{n!}.$$

The technique is then to make $y = T(z)$ the new independent variable and pull out a sufficient number of terms. The remainder can then be carefully bounded, using e. g. the saddle point method.

Very recently, in the analysis of hashing with linear probing [140] (this issue!), Ramanujan's Q–function plays a vital rôle again. We will not describe this here, since the present issue is somehow built around this fascinating research (see also Knuth's paper in this issue).

6. Generalized Digital Trees

Tries and digital search trees were covered in several papers, [36, 37, 41, 42, 54, 59, 62]. We have already discussed digital search trees in Section 4. However, the paper [103] is perhaps the most influential and original contribution in this area, and therefore we will briefly sketch it here.

The recursion

$$f_{n+b} = 1 + \sum_{k=0}^{n} 2^{-n} \binom{n}{k} (f_k + f_{n-k}),$$

where $b = 0, 1, 2, \ldots$ is a design parameter, was traditionally attacked by exponential generating functions, leading to a functional differential equation

$$\frac{d^b}{dz^b} f(z) = e^z + 2e^{z/2} f(\tfrac{z}{2}).$$

Consequently, only the cases $b = 0, 1$ were amenable to a satisfying mathematical analysis. The Poisson transformed function $g(z) = e^{-z} f(z)$ plays a vital rôle here. Now, the stupendous idea is to introduce the *ordinary* generating function $F(z)$. Let also $G(z)$ be the ordinary generating function corresponding to $g(z)$. Then by the Euler transform

$$f(z) = \sum_{n \ge 0} a_n z^n \longleftrightarrow \frac{1}{1-z} f\left(\frac{z}{z-1}\right) = \sum_{n \ge 0} \left(\sum_{k=0}^{n} \binom{n}{k} (-1)^k a_k \right) z^n, \tag{2}$$

the functions $F(z)$ and $G(z)$ are related via

$$F(z) = \frac{1}{1-z} G\Big(\frac{z}{1-z}\Big).$$

But the function $G(z)$, which is the easiest of the four functions, satisfies a simpler recurrence, namely

$$G(z)(1+z)^b = 2z^b G(\tfrac{z}{2}) + P(z),$$

where $P(z)$ is a certain polynomial. Consequently, the equation for $G(z)$ can be solved by iteration: Let

$$Q(u) = \prod_{j\ge 0} \Big(1 + \frac{u}{2^j}\Big),$$

then

$$\frac{G(t^{-1})}{Q^b(t/2)} = \sum_{k\ge 0} \frac{2^k (2^k t)^b P(\frac{1}{2^k t})}{Q^b(2^k t)}.$$

But this is a *harmonic sum*, and the Mellin transform strikes again, yielding the behavior of $G(t^{-1})$ for $t \to 0$, which means $z \to \infty$. This in turns gives the behaviour of $F(z)$ for $z \to 1$, from which the coefficients of interest come out by means of transfer theorems (singularity analysis, as discussed above).

7. Quadtrees

The papers [95, 105, 108, 120, 125] deal with quadtrees; a quadtree is a tree to store a geometric object of dimension 2 (there are also generalizations to higher dimensions). Unlike a binary search tree, where an element goes either left or right depending on whether it is smaller or larger than the element in the root, there are now four subtrees, conveniently denoted by north-west, north-east, south-west, south-east.

Paper [105] studies the idea of *paging*; with a design parameter b, a node can hold up to b data (they are stored sequentially, as they come in). Then the expected number of pages is asymptotic to $\gamma_b \cdot n$, with

$$\gamma_b = 18b^2 + 27b + 3 - 18b(b+1)^2\Big[\frac{\pi^2}{6} - \sum_{j=1}^{b} \frac{1}{j^2}\Big].$$

Like in the instance of binary search trees, these problems are in the domain of differential equations, in particular the hypergeometric differential equation.

Papers [95, 108] deal with the average search costs for the general d–dimensional quadtrees. A laborious analysis, based again on the careful study of the underlying hypergeometric differential equations, reveals that the cost is asymptotically equal to $\frac{2}{d}\log n$ for fixed d; the paper [120] treats the (limiting) distribution, which can be described explicitly in the instance $d = 2$ and asymptotically in the general case.

The major breakthrough, however, is paper [125]. The new feature is simple yet extremely powerful. It applies the Euler transformation (2). With this, amazingly, all kind of parameters can now be dealt with relatively easily, since the new variable $t = z/(z-1)$ makes everything much nicer. The differential equations become much simpler in the new variable. The resulting explicit formulæ come out usually as alternating sums

$$\sum_k \binom{n}{k} (-1)^k a_k$$

with some basic sequences a_k. From these explicit forms, asymptotics can be derived either by Rice's method (cf. Section 4) or, as in the paper, a combination of Mellin transforms and singularity analysis. A major ingredient is the following formula which is being related to the names of Mellin, Lindelöf, Ramanujan:

$$\sum_{n=n_0}^{\infty} \varphi(n)(-t)^n = -\frac{1}{2\pi i}\int_{n_0-1/2-i\infty}^{n_0-1/2+i\infty} \varphi(s)t^s \frac{\pi}{\sin \pi s} ds.$$

8. Partial Match Retrieval of Multidimensional Data

Binary search trees can still be used, even when one faces *multidimensional data.* If a key has d components, the components are used (in cyclic order) to make decisions. So the first component might say 'go left,' the second component 'go right,' and the third component 'go right,' etc., until an empty node has been found.

Instead of plain searching one can now deal with partial queries, introducing the `don't care` component. We might search for all keys with first component $= 3$, second component irrelevant, third component $= 8$, etc.

The generalization of the usual *depth* is thus the size of the subtree containing all the nodes visited during such a search. The papers [41, 59] analyze this parameter. Naturally, one faces *systems of differential equations*, of dimension d. For example, one must solve asymptotically the following system

$$\frac{d}{dz}\mathbf{y}(z) = \mathbf{A}(z)\mathbf{y}(z) + \mathbf{b}(z)$$

without being able to solve it explicitly. In the above, $\mathbf{A}(z)$ is a matrix that vanishes at $z = 1$. For instance, for $d = 2$ when one attribute is specified and the other one is not, the average value of the cost is proportional to $n^{(\sqrt{17}-3)/2}$, disproving earlier conjectures.

The paper [76] revisits this issue.

9. Protocols for Multi–Access Broadcast Channels

The paper [66] is a survey that summarizes the earlier works [56, 50, 55] on the analysis of multi-access protocols. We describe below a generic multi-access protocol and show how it can be analyzed.

If people (users) communicate via a (discrete) channel, they face the problem of conflicts, if more than two want to send a packet in the same time slot. In the absence of a supervisor, the conflict parties must resolve their conflict. There are basically two possible schemes: *blocked access* (i.e., no new parties can enter the system until the current conflict has been resolved), and *free access* (i.e., the opposite). Tree algorithms associated with the name Capetanakis, Tsybakov and Mikhailov use consecutive rounds of coin flippings to decide who is allowed to send. Mathematically, this model is close to the digital trees discussed above. The analysis thus involves Mellin transforms, Rice's method, and singularity analysis of generating functions.

The free access protocol is harder to analyze. Typically, the functional equation is here of the following form

$$\Lambda(z) = \Lambda(\lambda + pz) + \Lambda(\lambda + (1-q)z) + \alpha(z)$$

where $\alpha(z)$ is a known function. Solutions of such an equation involve the *iteration semigroup* with two generators. This is analytically quite complicated, but one can then answer question like "what is the maximal arrival rate such that the system is stable?". Further questions involve the delay, steady state probabilities, queue length, etc.

The paper [69] also belongs to this area, as it estimates the *multiplicity of a conflict*. With this knowledge one can get better algorithms. This leads to a typical Mellin transform type analysis.

10. Continued fractions

The papers [22, 23, 32] deal with the interplay of continued fractions and combinatorics. Let us consider lattice paths, consisting of steps NORTHEAST, EAST, SOUTHEAST, starting at the origin, returning to the x–axis after n steps, and never being negative. The possible steps are denoted by the letters $\{a, b, c\}$, and an index i is additionally used when a step starts at altitude i. Thus, such a lattice path is a *word* in the variables $\{a_0, a_1, \dots, b_0, b_1, \dots, c_1, \dots\}$.

The set of all paths (a *formal language*) is given by the infinite continued fraction

$$\cfrac{1}{1 - c_0 - \cfrac{a_0|b_1}{1 - c_1 - \cfrac{a_1|b_2}{1 - c_2 - \cfrac{a_2|b_3}{\cdots}}}},$$

where $(u|v)/w$ denotes $uw^{-1}v$, and w^{-1} is the "quasi–inverse" of languages (or formal power series).

There are many consequences of this *continued fraction theorem*, e. g. finite versions describe lattice paths of bounded height. *Counting* leads to a replacement of noncommuting variables by commuting variables. For instance, replacing all the variables by z gives the continued fraction for $\sum M_n z^n$, M_n being a Motzkin number. Many combinatorial objects can be described by such lattice paths, with suitable specializations. Some examples: Set partitions (also with several restrictions), permutations (via tournament trees), involutions, etc. Some later developments can be found in [78, 89].

There are also applications to Computer Science, since several dynamic data structures can be described in this way, the simplest being a stack, but also Dictionaries, Priority queues, Linear lists, Symbol tables, and subspecies of these. Operations like Insertion, Query, Deletion have then an obvious interpretation in the path diagram. Several notions of costs can be discussed conveniently in terms of continued fractions. These concepts were worked out in collaboration with Chéno, Françon, Puech, and Vuillemin [18, 19, 21, 25, 30, 60].

11. Number theoretic aspects of continued fractions

Gauss studied expansions of complex numbers into continued fractions. Here is an example.

$$\frac{35470}{99661} + \frac{315}{9961} i = \cfrac{1}{3 - \cfrac{1}{5 + \cfrac{1}{\frac{1}{3} + \frac{7}{5} i}}}.$$

Consecutive digits are obtained by the recursive rule

$$\psi(z) = \lceil \Re(z) \rfloor + \frac{\epsilon(z)}{\psi\left(\dfrac{\epsilon(z)}{z - \lceil \Re(z) \rfloor}\right)},$$

where $\lceil x \rfloor = \lceil x - \frac{1}{2} \rceil$ and $\epsilon(z) = \mathrm{sign}(\Re(z) - \lceil \Re(z) \rfloor)$. The algorithm terminates if the resulting number falls into the domain $\{z \in \mathbb{C} \mid 0 \leq \Re(z) \leq \frac{1}{2} \text{ and } |z| \geq 1\}$.

The average number of steps of this algorithm (in various continuous and discrete models) turns out to be linear, and the constant(s) involve the interesting quantity

$$\sum_{d \geq 1} \frac{(-1)^d}{d^2} \sum_{c=1}^{d} \frac{1}{c^2},$$

which is expressible in terms of the remarkable constants $\zeta(3)$ and $\mathrm{Li}_4(\frac{1}{2})$ (a *tetralogarithm*).

This and much more can be found in the papers [92, 116, 132, 135]. The work [135] is a survey paper and covers much more general reduction schemes (transformation), e. g. the binary representation of integers. The average–case analysis of these transformations usually involves interesting numerical constants, like Wirsing's, Lévy's, Hensley's, and Vallée's constant. This is a quite challenging domain, with relations to Functional Analysis.

12. Approximate counting and probabilistic counting

Approximate counting is a method of keeping an approximate count of n using only approximately $\log \log n$ bits. The counter is advanced by 1 according to a random generator with probability $\frac{1}{2^n}$ provided the counter is n. The approximate counter value C_n (on average) after n (random) increments can then be used as an estimator of $\log_2 n$, since it was shown in [38, 49] that

$$C_n = \log_2 n + \frac{\gamma}{\log 2} - \alpha + \frac{1}{2} + \delta(\log_2 n) + o(1),$$

with $\alpha = \sum_{j \geq 1} 1/(2^j - 1) = 1.606695$ and an explicit periodic function $\delta(x)$.

This delicate result was achieved by first computing the exact probabilities $p_{n,l}$ that a counter value l is achieved. Then the sum $\sum_l l\, p_{n,l}$ was approximated by a simpler one using real analysis. Then the asymptotic behavior was grinded out by the Mellin transform technique which we have already discussed above.

Probabilistic counting is a method to count a large population of size n. The special feature is that multiple elements will be counted only once, and one does not have to take extra care of that. Each possible element is mapped via a hash function to a bit string, and the position of the least important bit '1' is reported. Then the OR composition of all these gives one final bit string as the result. The number of trailing ones in this string can be used as an estimator for $\log_2 n$. To justify this, first the probability that the value of the estimator is k was computed via the inclusion–exclusion principle. After some more approximations, the very interesting Dirichlet series

$$N(s) = \sum_{k \geq 1} \frac{(-1)^{\nu(k)}}{k^s}$$

arises with the familiar sum of digits function $\nu(k)$. One of the major challenges of the papers [40, 51] is to find an analytic continuation of this series to the complex plane. This is somehow related to the infinite product

$$\prod_{k \geq 0} \left(1 - x^{2^k}\right) = \sum_{n \geq 0} (-1)^{\nu(n)} x^n.$$

A paper on adaptive sampling [85] might also be seen in this context.

13. Automatic analysis of algorithms

The paper [96] contains the description of this very ambitious project. The system $\Lambda\Upsilon\Omega$ consists of two parts: the Algebraic Analyzer (Alas) and the Analytic Analyzer (Ananas). The implementation of Alas was done by Paul Zimmermann, whereas Bruno Salvy was responsible for Ananas.

Historically, first steps towards this monster project can be found in the papers [31, 68]; in them, a complexity calculus for a class of simple recursive programs over tree structures was proposed (e. g. differentiation). The algebraic part consists in specifications, combinatorial structures, translation rules into generating functions, etc.

The analytic part deals with the (automatic) asymptotic analysis of a class of elementary functions. Well, they are not so elementary after all, containing for instance

$$f(z) = \exp\left(z \log \frac{1}{1-4z} e^{z/(1-z)^3}\right) + e^z \log \frac{1}{1-2z^2} + 1 + z.$$

A major part is the algorithm Equivalent. For a function f from a certain elementary class it computes an asymptotic form of the coefficient $[z^n]f(z)$. It uses modules Radius, Directions, Expansions, Transfer. The analyzer can also deal with entire functions, namely the class of the so called Hayman–admissible functions (via saddle point analysis).

The text [96] describes the theory as well as some parts of the implementation. For the purpose of illustration of the power of the system, it contains 20 *automatic theorems*, from classical combinatorics and the analysis of algorithms.

The system $\Lambda\Upsilon\Omega$ still exists, but it needs probably some adjustments due to changes in the Maple system. However, several modules are of independent interest, and they can be used (after downloading from the Inria server).

14. Coupon Collector and Birthday Paradox

Assume that a set $\mathcal{A} = \{a_1, \dots, a_m\}$ of different items ('coupons') are given. The coupon collector problem asks for the (average) number of drawings until one has a full collection of all coupons.

The birthday paradox asks for the (average) number of drawings until one sees an item for the second time.

These seemingly different and also classical problems are considered in a unified setting in [74, 102]. The items are coming with probabilities $\{p_1, \dots, p_m\}$ and one asks for the time to see k items j times. So $k = m$ and $j = 1$ means coupon collecting and $k = 1$ and $j = 2$ means the birthday paradox.

The following master formula contains it all:

$$\mathbf{E} = \sum_{q=0}^{m-1} \int_0^\infty [u^q] \Bigg(\prod_{i=1}^m \Big(e_{k-1}(p_i t) + u\big(e^{p_i t} - e_{k-1}(p_i t)\big)\Big)\Bigg) e^{-t} dt$$

with the truncated exponential series

$$e_m(t) = \sum_{i=0}^m \frac{t^i}{i!}.$$

It is obtained by Schützenberger's methodology to code events by certain words over a suitable alphabet (here $\mathcal{A} = \{a_1, \dots, a_m\}$). It turns out the shuffle product plays an important rôle. In terms of enumeration, this fits to the realm of exponential generating functions. However, the overall problem must be seen in the context of ordinary generating functions. Henceforth, the Laplace–Borel transform is used to switch between the two different kinds of generating functions.

We also cite a recent paper [129] about coupon collection with group drawings.

15. Random generation of combinatorial objects

The papers [115, 122] contain a general calculus for the random generation of labelled combinatorial structures. The objects must have a representation in terms of a grammar. For instance, Non plane trees are given by $A = Z \cdot \mathrm{SET}(A)$, Plane general trees by $C = Z \cdot \mathrm{SEQUENCE}(C)$, Permutations are $D = \mathrm{SET}(\mathrm{CYCLE})(Z)$, etc. Also, the objects have a 'size,' usually the number of nodes in trees, the numbers of elements in a permutation, etc. Then, a general machinery is developed to create a random generator solely based on the specification. The automatic analysis of the complexity of the generator comes with it for free! This important technique is not only theoretically interesting but was later implemented in MapleV4 under the name CombStruct.

16. Gaussian limiting distributions

The collaboration with Michèle Soria concentrates on limiting distributions [90, 97, 114]. The flavour of the general results is perhaps best described by giving a few examples. The bivariate generating function for the number of cycles in permutations with u marking the number of cycles is

$$\sum_{n,k} \begin{bmatrix} n \\ k \end{bmatrix} u^k \frac{z^n}{n!} = \exp\Big(u \log \frac{1}{1-z}\Big).$$

From that form alone one can conclude that the number of cycles in a random permutation of size n converges to a Gaussian limiting distribution! This was an example of a labelled structure; an example for an unlabelled structure comes from the number of irreducible factors of random polynomials of degree n over a finite field, where, with $P(z) = 1/(1-qz)$, the generating function of interest $I(z)$ is implicitly given by

$$P(z) = \exp\Big(\frac{I(z)}{1} + \frac{I(z^2)}{2} + \frac{I(z^3)}{3} + \dots\Big).$$

Since $I(z)$ has a logarithmic singularity, the Gaussian limiting distribution follows automatically. Large deviations results are also discussed in the above papers.

17. Random graphs and random mappings

The paper [87] concerns random mappings (there are n^n of them, on n points); consequently, the tree function is of major importance here. A thorough analytic treatment is provided, analyzing (cumulative) parameters such as Cycle length, Tree size, Component size, Predecessor size, and also extremal statistics such as Longest Paths, Maximum Tail length, etc.

The paper [79] is the first one that uses analytic methods for the study of random graphs, and may be seen as a step towards the monumental paper by Janson, Knuth, Pittel and Luczak on the *Birth of the Giant Component*.

It answers an old question about the average length of the first cycle in an evolving graph; it turns out that this quantity is $Kn^{1/6}+O(n^{1/22})$, with $K=2.0337$. An analytic expression for K is also provided (and was later simplified by D. E. Knuth).

18. Applications to formal languages

The papers [47, 63, 65, 84] deal with applications of analytic methods to questions from formal language theory. The basic principles are these:

An unambiguous context–free language has an algebraic generating function.

If the generating function of a context–free language considered as an analytic function is a transcendental function, then the language is inherently ambiguous.

That means that context–free languages have generating functions from a rather restricted class. Consequently, if for a certain language the corresponding generating function fails to be in this class, a negative result is thus automatically implied. In a number of cases, it is a good strategy to show that the generating function has a non-algebraic singularity (e. g. logarithmic), since an algebraic function does not have such singularities.

19. Various Papers

Consider the harmonic numbers $H_n = 1+\frac{1}{2}+\cdots+\frac{1}{n}$. There exist striking identities like

$$\sum_{n\geq 1}\frac{H_n}{n^2}=2\zeta(3), \qquad \sum_{n\geq 1}\frac{(H_n)^3}{n^4}=\frac{231}{16}\zeta(7)-\frac{51}{4}\zeta(3)\zeta(4)+2\zeta(2)\zeta(5).$$

The paper [136] gives a complete answer to such identities. The method is the residue calculus. There is independent interest in this line of research, namely by J. Borwein and his team in Vancouver, and by people in Lille.

The paper [138] discusses the classical subject of non–crossing configurations. Assume that the points of a regular n–gon are consecutively labelled $1,\dots,n$ and the edges are drawn as lines in the plane. The notion of 'non–crossing' should then be (intuitively) clear. Euler studied already triangulations, which led to Catalan numbers. The present paper studies in a systematic way various other statistics such as Trees, Forests, Graphs (general and connected), Dissections and Partitions. The paper [137] also belongs to this area.

The paper [98] takes up the subject of increasing trees. They are labelled rooted trees, and labels along any branch from the root to the leaves are increasing. A reasonable level of generality is obtained by introducing a *degree function* $\phi(w)$, which must be seen as an ordinary/exponential generating function if the *variety* of trees is planar/non–planar.

The enumeration of such a variety goes like this. Set

$$Y(z)=\sum_{n\geq 0}Y_n\frac{z^n}{n!},$$

with the Y_n's being the numbers of interest. Then $Y(z)$ is implicitly given by

$$\int_0^{Y(z)}\frac{dw}{\phi(w)}=z.$$

In some special cases the inversion problem is solvable explicitly in terms of special functions. However, this is not necessary, and all kind of information, including asymptotics, can be extracted from this form. More advanced parameters like path length etc. can also be handled in this way. This work resorts to a combination of algebraic and analytic generating function methods which is versatile and widely applicable.

The recent paper [131] deals with patterns in binary search trees. There are some earlier papers dealing with the problem of patterns in several families of trees, [15, 28, 31, 44, 52, 68, 73, 91].

Let $\lambda(u)$ be the probability for a given unlabelled tree u to have the shape of a random binary search tree of size $|u|$, and $\omega[t]$ be the number of occurrences of the (fixed) pattern u in the binary search tree t. As expected, the solution involves bivariate generating function like this one

$$F(z,y) = \sum_t \lambda(t) y^{\omega[t]} z^{|t|}.$$

It satisfies the Riccati equation

$$\frac{\partial}{\partial z} F(z,y) = F^2(z,y) + (y-1)\lambda(u) z^{|u|-1}, \qquad F(0,y) = 1.$$

Analyzing this equation, one gets all the information (moments, limiting distribution) of interest.

20. Survey papers and books

When reading Philippe Flajolet's papers, one gets the impression that he is driven by the desire for a systematic account and a thorough and deep understanding. Even more so, there is a series of survey papers, partially based on courses and invited talks, and they all show a very systematic approach to combinatorics, analysis of algorithms and related areas [39, 64, 70, 93, 99].

Eventually, these surveys were culminating in two books: *An Introduction to the Analysis of Algorithms* [128], and the continuation *Analytic Combinatorics* that is still in preparation. Several chapters can already be found and downloaded from Flajolet's homepage: `http://pauillac.inria.fr/algo/flajolet/index.html`.

References

[1] P. Flajolet and J-M. Steyaert. Complexité des problèmes de decision relatifs aux algorithmes de tri. In M. Nivat, editor, *Automata, Languages and Programming*, pages 537–548. North Holland, 1972. Proceedings of 1st ICALP Colloquium, 1972.

[2] P. Flajolet and J-M. Steyaert. Decision problems for multihead finite automata. In J. Gruska, B. Rovan, and J. Wiedermann, editors, *Mathematical Foundations of Computer Science*, pages 225–230, 1973. Proceedings of the 1st MFCS Symposium 1973.

[3] P. Flajolet and J-M. Steyaert. *Une formalisation de la notion d'algorithme de tri non–récurrent.* Thèse de 3e cycle, Université de Paris VII, 1973. 289 pages. Presented jointly with J–M. Steyaert.

[4] P. Flajolet and J-M. Steyaert. A class of non recursive sorting algorithms. In E. Bianco, editor, *Journées Mathématiques de la Compilation*, pages 42–49. Université de Marseille, 1973.

[5] P. Flajolet and J-M. Steyaert. Generalized immune sets. Technical report, IRIA, 1973. 22 pages.

[6] P. Flajolet and J-M. Steyaert. On sets having only hard subsets. In *Automata, Languages and Programming*, volume 14 of *Lecture Notes in Computer Science*, pages 446–456, 1974.

[7] P. Flajolet and J-M. Steyaert. Une généralisation de la notion d'ensemble immune. *RAIRO Theoretical Informatics and Applications*, 1:37–48, 1974.

[8] P. Flajolet and J-M. Steyaert. Complexity classes of languages and operators. Technical report, IRIA, 1974. 13 pages.

[9] P. Flajolet, editor. *Informatique et Philologie*, 1975. 233 pages. Proceedings of a colloquium organized at IRIA, 1974.

[10] P. Flajolet and J.-P. Kherlakian. Linguistique formelle et linguistique historique. In *Informatique et Philologie*, pages 195–207, 1975.

[11] P. Flajolet and J-M. Steyaert. Classes de complexité et réduction entre problèmes. In *Codici, Complessità di Calcolo e Linguaggi Formali.* Liguori Pub., 1975.

[12] P. Flajolet and J-M. Steyaert. Hierarchies de complexité et réduction entre problèmes. *Asterisque*, 38–39:53–72, 1976.

[13] P. Flajolet, J-C Raoult, and J. Vuillemin. On the average number of registers required to evaluate arithmetic expressions. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 196–205, 1977.

[14] P. Flajolet. Analyse d'algorithmes de manipulation de fichiers. Technical report, IRIA, 1978.

[15] P. Flajolet. On the analysis of tree-matching. In *Proc. CAAP Colloquium*, Lille, 1978.

[16] P. Flajolet. *Analyse d'algorithmes de manipulation d'arbres et de fichiers.* Doctorat ès sciences, Université de Paris XI, Orsay, 1979.

[17] P. Flajolet. Deux problèmes d'analyse d'algorithmes. In *Séminaire de Théorie des Nombres Delange–Pisot–Poitou*, volume 20, pages 1401–1410, 1979.

[18] P. Flajolet, J. Françon, and J. Vuillemin. Towards analysing sequences of operations for dynamic data structures. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, pages 183–195, 1979.

[19] P. Flajolet, J. Françon, and J. Vuillemin. Computing integrated costs of data structures with applications to dictionaries. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pages 49–61, 1979.

[20] P. Flajolet, J-C Raoult, and J. Vuillemin. The number of registers required to evaluate arithmetic expressions. *Theoretical Computer Science*, 9:99–125, 1979.

[21] L. Cheno, P. Flajolet, J. Françon, C. Puech, and J. Vuillemin. Dynamic data structures: Finite files, limiting profiles and variance analysis. In *Eighteenth Annual Conference on Communication, Control, and Computing*, pages 223–232, 1980.

[22] P. Flajolet. Combinatorial aspects of continued fractions. *Annals of Discrete Mathematics*, 8:217–222, 1980. Proceedings of "Colloque Franco–Canadien de Combinatoire", Montreal, 1979.

[23] P. Flajolet. Combinatorial aspects of continued fractions. *Discrete Mathematics*, 32:125–161, 1980.

[24] P. Flajolet and J. Françon. Structures de données dynamiques en reservoir borné. In J. Morgenstern, editor, *III Journées Algorithmiques*, 1980.

[25] P. Flajolet, J. Françon, and J. Vuillemin. Sequence of operations analysis for dynamic data structures. *Journal of Algorithms*, 1:111–141, 1980.

[26] P. Flajolet and A. Odlyzko. Exploring binary trees and other simple trees. In *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, pages 207–216, 1980.

[27] P. Flajolet and L. Ramshaw. A note on Gray code and odd–even merge. *SIAM Journal on Computing*, 9:142–158, 1980.

[28] P. Flajolet and J-M. Steyaert. On the analysis of tree matching algorithms. volume 85 of *Lecture Notes in Computer Science*, pages 208–219, 1980.

[29] P. Flajolet. *Analyse d'algorithmes de manipulation d'arbres et de fichiers*, volume 34–35 of *Cahiers du Bureau Universitaire de Recherche Opérationnelle.* 1981. 209 pages.

[30] P. Flajolet and C. Puech. Analyse de structures de données dynamiques et histoires de fichiers. *Questiió*, 5:31–48, 1981.

[31] P. Flajolet and J-M. Steyaert. A complexity calculus for classes of recursive search programs over tree structures. In *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, pages 386–393, 1981.

[32] P. Flajolet. On congruences and continued fractions for some classical combinatorial quantities. *Discrete Mathematics*, 41:145–153, 1982.

[33] P. Flajolet and A. Odlyzko. The average height of binary trees and other simple trees. *Journal of Computer and System Sciences*, 25:171–213, 1982.

[34] P. Flajolet and D. Sotteau. A recursive partitioning process of computer science. In *Second World Conference on Mathematics at the Service of Man*, pages 25–30. Universidad Politecnica de Las Palmas, June 1982.

[35] P. Flajolet and J-M. Steyaert. Elements d'un calcul de complexité de programmes récursifs d'arbres. In *Les Mathématiques de l'Informatique—Mathematics for Computer Science*, pages 81–92, 1982.

[36] P. Flajolet and J-M. Steyaert. A branching process arising in dynamic hashing, trie searching and polynomial factorization. In *Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 239–251, 1982.

[37] P. Flajolet. On the performance evaluation of extendible hashing and trie searching. *Acta Informatica*, 20:345–369, 1983.

[38] P. Flajolet. On approximate counting. In *International Seminar on Modelling and Performance Evaluation Methodology*, pages 205–236. INRIA, 1983.

[39] P. Flajolet. Methods in the analysis of algorithms: evaluations of a recursive partitioning process. In *Fundations of Computation Theory*, volume 158 of *Lecture Notes in Computer Science*, pages 141–158, 1983.

[40] P. Flajolet and P. N. Martin. Probabilistic counting. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, pages 76–82, 1983.

[41] P. Flajolet and C. Puech. Tree structures for partial match retrieval. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, pages 282–288, 1983.

[42] P. Flajolet and N. Saheb. Digital search trees and the complexity of generating an exponentially distributed variate. In *CAAP'83*, volume 159 of *Lecture Notes in Computer Science*, pages 220–235, 1983.

[43] P. Flajolet and R. Sedgewick. The asymptotic evaluation of some alternating sums involving binomial coefficients. Unpublished memoir, 1983. 11 pages.

[44] J-M. Steyaert and P. Flajolet. Patterns and pattern-matching in trees: an analysis. *Information and Control*, 58:19–58, 1983.

[45] P. Flajolet. Algorithmique. In *Encyclopedia Universalis*, volume 1, pages 758–763. 1984.

[46] P. Flajolet and A. Odlyzko. Limit distributions for coefficients of iterates of polynomials with applications to combinatorial enumerations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 96:237–253, 1984.

[47] P. Flajolet. Ambiguity and transcendence. In *Automata, Languages and Programming*, volume 194 of *Lecture Notes in Computer Science*, pages 179–188, 1985.

[48] P. Flajolet. Elements of a general theory of combinatorial structures. In *Fundamentals of Computation Theory*, volume 199 of *Lecture Notes in Computer Science*, pages 112–127, 1985.

[49] P. Flajolet. Approximate counting: A detailed analysis. *BIT*, 25:113–134, 1985.

[50] G. Fayolle, P. Flajolet, M. Hofri, and P. Jacquet. Analysis of a stack algorithm for random access communication. *IEEE Transactions on Information Theory*, IT-31:244–254, 1985.

[51] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31:182–209, 1985.

[52] P. Flajolet, T. Ottmann, and D. Wood. Search trees and bubble memories. *RAIRO Theoretical Informatics and Applications*, 19:137–164, 1985.

[53] P. Flajolet, M. Régnier, and R. Sedgewick. Some uses of the Mellin integral transform in the analysis of algorithms. In *Combinatorial Algorithms on Words*, volume 12, pages 241–254. Springer Verlag, 1985.

[54] P. Flajolet, M. Régnier, and D. Sotteau. Algebraic methods for trie statistics. *Annals of Discrete Mathematics*, 25:145–188, 1985.

[55] P. Mathys and P. Flajolet. Q–ary collision resolution algorithms in random access systems with free or blocked channel access. *IEEE Transactions on Information Theory*, IT-31:217–243, 1985.

[56] G. Fayolle, P. Flajolet, and M. Hofri. On a functional equation arising in the analysis of a protocol for a multiaccess broadcast channel. *Advances in Applied Probability*, 18:441–472, 1986.

[57] P. Flajolet. The evolution of two stacks in bounded space and random walks in a triangle. In *Mathematical Foundations of Computer Science*, volume 233 of *Lecture Notes in Computer Science*, pages 325–340, 1986.

[58] P. Flajolet and H. Prodinger. Register allocation for unary–binary trees. *SIAM Journal on Computing*, 15:629–640, 1986.

[59] P. Flajolet and C. Puech. Partial match retrieval of multidimensional data. *Journal of the ACM*, 33:371–407, 1986.

[60] P. Flajolet, C. Puech, and J. Vuillemin. The analysis of simple list structures. *Information Sciences*, 38:121–146, 1986.

[61] P. Flajolet and N. Saheb. The complexity of generating an exponentially distributed variate. *Journal of Algorithms*, 7:463–488, 1986.

[62] P. Flajolet and R. Sedgewick. Digital search trees revisited. *SIAM Journal on Computing*, 15:748–767, 1986.

[63] J-M. Autebert, P. Flajolet, and J. Gabarro. Prefixes of infinite words and ambiguous context–free languages. *Information Processing Letters*, 25:211–216, 1987.

[64] P. Flajolet. Mathematical tools for automatic program analysis. Research Report 603, INRIA, 1987.

[65] P. Flajolet. Analytic models and ambiguity of context–free languages. *Theoretical Computer Science*, 49:283–309, 1987.

[66] P. Flajolet and P. Jacquet. Analytic models for tree communication protocols. In *Flow Control of Congested Networks*, volume 38, pages 223–234. Springer Verlag, 1987.

[67] P. Flajolet and H. Prodinger. Level number sequences for trees. *Discrete Mathematics*, 65:149–156, 1987.

[68] P. Flajolet and J-M. Steyaert. A complexity calculus for recursive tree algorithms. *Mathematical Systems Theory*, 19:301–331, 1987.

[69] A. G. Greenberg, P. Flajolet, and R. E. Ladner. Estimating the multiplicities of conflicts to speed their resolution in multiple access channels. *Journal of the ACM*, 34:289–325, 1987.

[70] P. Flajolet. Mathematical methods in the analysis of algorithms and data structures. In *Trends in Theoretical Computer Science*, chapter 6, pages 225–304. Computer Science Press, 1988.

[71] P. Flajolet. L'analyse d'algorithmes ou le risque calculé. In *Journées Scientifiques et Prix U.A.P. 1985,1986,1987*, pages 17–34. Conseil Scientifique de l'UAP, 1988.

[72] P. Flajolet. Evaluation de protocoles de communication: aspects mathématiques. Technical Report 797, INRIA, 1988.

[73] P. Flajolet. Random tree models in the analysis of algorithms. In *PERFORMANCE'87*, pages 171–187. Elsevier, 1988.

[74] P. Flajolet, D. Gardy, and L. Thimonier. Probabilistic languages and random allocations. In *Automata, Languages and Programming*, volume 317 of *Lecture Notes in Computer Science*, pages 239–253, 1988.

[75] P. Flajolet, P. Kirschenhofer, and R. F. Tichy. Deviations from uniformity in random strings. *Probability Theory and Related Fields*, 80:139–150, 1988.

[76] W. Cunto, G. Lau, and P. Flajolet. Analysis of *kdt*–trees: *kd*–trees improved by local reorganisations. In *Algorithms and Data Structures*, volume 382 of *Lecture Notes in Computer Science*, pages 24–38, 1989.

[77] P. Flajolet, P. Kirschenhofer, and R. F. Tichy. Discrepancy of sequences in discrete spaces. In *Irregularities of Partitions*, volume 8 of *Algorithms and Combinatorics*, pages 61–70. Springer Verlag, 1989.

[78] P. Flajolet and J. Françon. Elliptic functions, continued fractions and doubled permutations. *European Journal of Combinatorics*, 10:235–241, 1989.

[79] P. Flajolet, D. E. Knuth, and B. Pittel. The first cycles in an evolving graph. *Discrete Mathematics*, 75:167–215, 1989.

[80] P. Flajolet, B. Salvy, and P. Zimmermann. Lambda–Upsilon–Omega: An assistant algorithms analyzer. In *Applied Algebra, Algebraic Algorithms and Error–Correcting Codes*, volume 357 of *Lecture Notes in Computer Science*, pages 201–212, 1989.

[81] P. Flajolet, B. Salvy, and P. Zimmermann. Lambda–Upsilon–Omega: The 1989 Cookbook. Research Report 1073, INRIA, 1989. 116 pages.

[82] D. Gardy, P. Flajolet, and C. Puech. On the performance of orthogonal range queries in multiattribute and doubly chained trees. In *Algorithms and Data Structures*, volume 382 of *Lecture Notes in Computer Science*, pages 218–229, 1989.

[83] D. Gardy, P. Flajolet, and C. Puech. Average cost of orthogonal range queries in multiattribute trees. *Information Systems*, 14:341–350, 1989.

[84] J-P. Allouche, P. Flajolet, and M. Mendès France. Algebraically independent formal power series: a language theory interpretation. In *Analytic Number Theory*, number 1434 in Lecture Notes in Mathematics, pages 11–18, 1990.

[85] P. Flajolet. On adaptive sampling. *Computing*, 34:391–400, 1990.

[86] P. Flajolet, M. Goldwurm, and J-M. Steyaert. Random generation and context–free languages. Manuscript, 1990.

[87] P. Flajolet and A. Odlyzko. Random mapping statistics. In *Advances in Cryptology*, volume 434 of *Lecture Notes in Computer Science*, pages 329–354, 1990.

[88] P. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics*, 3:216–240, 1990.

[89] P. Flajolet and R. Schott. Non–overlapping partitions, continued fractions, Bessel functions and a divergent series. *European Journal of Combinatorics*, 11:421–432, 1990.

[90] P. Flajolet and M. Soria. Gaussian limiting distributions for the number of components in combinatorial structures. *Journal of Combinatorial Theory*, Series A, 53:165–182, 1990.

[91] P. Flajolet, P. Sipala, and J-M. Steyaert. Analytic variations on the common subexpression problem. In *Automata, Languages, and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 220–234, 1990.

[92] B. Vallée and P. Flajolet. Gauss' reduction algorithm: An average case analysis. In *Proceedings of the 31st Symposium on Foundations of Computer Science*, pages 830–839. IEEE Computer Society Press, 1990.

[93] J. Vitter and P. Flajolet. Analysis of algorithms and data structures. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, chapter 9, pages 431–524. North Holland, 1990.

[94] P. Flajolet. Pólya festoons. Research report, INRIA, July 1991. 7 pages.

[95] P. Flajolet, G. Gonnet, C. Puech, and J. M. Robson. The analysis of multidimensional searching in quad–trees. In *Proceedings of the Second Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 100–109. SIAM Press, 1991.

[96] P. Flajolet, B. Salvy, and P. Zimmermann. Automatic average–case analysis of algorithms. *Theoretical Computer Science*, 79:37–109, 1991.

[97] P. Flajolet and M. Soria. The cycle construction. *SIAM Journal on Discrete Mathematics*, 4:58–60, 1991.

[98] F. Bergeron, P. Flajolet, and Bruno Salvy. Varieties of increasing trees. In *CAAP'92*, volume 581 of *Lecture Notes in Computer Science*, pages 24–48, 1992.

[99] P. Flajolet. Analytic analysis of algorithms. In *Automata, Languages and Programming*, number 623 in Lecture Notes in Computer Science, pages 186–210, 1992.

[100] P. Flajolet. La calculabilité et ses limites. In *La Science au Présent*, pages 216–218. Les Éditions de l'Encyclopedia Universalis, 1992.

[101] P. Flajolet. Introduction à l'analyse d'algorithmes. *Singularité*, 3:5–16, 1992.

[102] P. Flajolet, D. Gardy, and L. Thimonier. Birthday paradox, coupon collectors, caching algorithms, and self–organizing search. *Discrete Applied Mathematics*, 39:207–229, 1992.

[103] P. Flajolet and B. Richmond. Generalized digital trees and their difference–differential equations. *Random Structures and Algorithms*, 3:305–320, 1992.

[104] P. Flajolet and P. Zimmermann. Algorithms seminar, 1991–1992. Research Report 1779, INRIA, 1992. 192 pages.

[105] M. Hoshi and P. Flajolet. Page usage in a quadtree index. *BIT*, 32:384–402, 1992.

[106] P. Flajolet, Z. Gao, A. Odlyzko, and B. Richmond. The distribution of heights of binary trees and other simple trees. *Combinatorics, Probability and Computing*, 2:145–156, 1993.

[107] P. Flajolet and M. Golin. Exact asymptotics of divide–and–conquer recurrences. In *Automata, Languages, and Programming*, number 700 in Lecture Notes in Computer Science, pages 137–149, 1993.

[108] P. Flajolet, G. Gonnet, C. Puech, and J. M. Robson. Analytic variations on quadtrees. *Algorithmica*, 10:473–500, 1993.

[109] P. Flajolet, X. Gourdon, and B. Salvy. Sur une famille de polynômes issus de l'analyse numérique. *Gazette des Mathématiciens*, 55:67–78, 1993.

[110] P. Flajolet, R. Kemp, and H. Prodinger, editors. *Average Case Analysis of Algorithms*, number 68 in Dagstuhl Seminar Reports. IBFI GmbH Schloß Dagstuhl, 1993.

[111] P. Flajolet and B. Salvy. A finite sum of products of binomial coefficients. *SIAM Review*, 35(4):645–646, 1993. Solution to Problem 92–18 by C. C. Grosjean.

[112] P. Flajolet and R. Sedgewick. The average case analysis of algorithms: Counting and generating functions. Research Report 1888, INRIA, 1993. 116 pages.

[113] P. Flajolet and R. Sedgewick. The average case analysis of algorithms: Complex asymptotics and generating functions. Research Report 2026, INRIA, 1993. 100 pages.

[114] P. Flajolet and M. Soria. General combinatorial schemas: Gaussian limit distributions and exponential tails. *Discrete Mathematics*, 114:159–180, 1993.

[115] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus of random generation. In *Algorithms—ESA'93*, number 726 in Lecture Notes in Computer Science, pages 169–180, 1993.

[116] H. Daudé, P. Flajolet, and B. Vallée. An analysis of the Gaussian algorithm for lattice reduction. In *Algorithmic Number Theory Symposium*, number 877 in Lecture Notes in Computer Science, pages 144–158, 1994.

[117] P. Dumas and P. Flajolet. Asymptotique des récurrences mahleriennes: le cas cyclotomique. *Journal de théorie des nombres de Bordeaux*, 8:1–30, 1996.

[118] P. Flajolet and M. Golin. Mellin transforms and asymptotics: The mergesort recurrence. *Acta Informatica*, 31:673–696, 1994.

[119] P. Flajolet, P. Grabner, P. Kirschenhofer, H. Prodinger, and R. Tichy. Mellin transforms and asymptotics: Digital sums. *Theoretical Computer Science*, 123:291–314, 1994.

[120] P. Flajolet and T. Lafforgue. Search costs in quadtrees and singularity perturbation asymptotics. *Discrete and Computational Geometry*, 12:151–175, 1994.

[121] P. Flajolet and R. Sedgewick. The average case analysis of algorithms: Saddle point asymptotics. Research Report 2376, INRIA, 1994. 55 pages.

[122] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132:1–35, 1994.

[123] P. Flajolet, X. Gourdon, and P. Dumas. Mellin transforms and asymptotics : Harmonic sums. *Theoretical Computer Science*, 144:3–58, 1995.

[124] P. Flajolet, P. Grabner, P. Kirschenhofer, and H. Prodinger. On Ramanujan's Q–function. *Journal of Computational and Applied Mathematics*, 58:103–116, 1995.

[125] P. Flajolet, G. Labelle, L. Laforest, and B. Salvy. Hypergeometrics and the cost structure of quadtrees. *Random Structures and Algorithms*, 7:117–144, 1995.

[126] P. Flajolet and B. Salvy. Computer algebra libraries for combinatorial structures. *Journal of Symbolic Computation*, 20:653–671, 1995.

[127] P. Flajolet and R. Sedgewick. Mellin transforms and asymptotics: finite differences and Rice's integrals. *Theoretical Computer Science*, 144:101–124, 1995.

[128] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms.* Addison-Wesley, 1996.

[129] V. Dumas, P. Flajolet, and P. Robert. Coupon collection with group drawings. 1996.

[130] P. Flajolet, X. Gourdon, and D. Panario. Random polynomials and polynomial factorization. In *Automata, Languages and Programming*, number 1099 in Lecture Notes in Computer Science, pages 232–243, 1997.

[131] P. Flajolet, X. Gourdon, and C. Martinez. Patterns in random binary search trees. *Random Structures and Algorithms*, 11:223 – 244, 1997.

[132] H. Daudé, P. Flajolet, and B. Vallée. An average–case analysis of the Gaussian algorithm for lattice reduction. *Combinatorics, Probability, and Computing*, 6:397 –433, 1997.

[133] P. Flajolet and B. Salvy. The sigsam challenges: Symbolic asymptotics in practice. Research report, INRIA, 1997.

[134] E. G. Coffman, P. Flajolet, L. Flatto, and M. Hofri. The maximum of a random walk and its application to rectangle packing. Research report, INRIA, 1997.

[135] P. Flajolet and B. Vallée. Continued fraction algorithms, functional operators, and structure constants. *Theoretical Computer Science*, 194:1–34, 1998.

[136] P. Flajolet and B. Salvy. Euler sums and contour integral representations. *Experimental Mathematics*, 1998.

[137] L. Devroye, P. Flajolet, F. Hurtado, M. Noy, and W. Steiger. Random triangulations. *Discrete and Computational Geometry*, 1998.

[138] P. Flajolet and M. Noy. Analytic combinatorics of non–crossing configurations. *Discrete Mathematics*, 1998.

[139] J. Clément, P. Flajolet, and B. Vallée. The analysis of hybrid trie structures. In *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, 1998.

[140] P. Flajolet, P. Poblete, and A. Viola. On the analysis of linear probing hashing. *Algorithmica*, 1998.

[141] P. Flajolet and H. Prodinger. Stirling numbers for complex arguments and Hankel contours. Research report, INRIA, 1998.

[142] P. Flajolet, X. Gourdon, and D. Panario. The complete analysis of a polynomial factorization algorithm over finite fields. Research Report 3370, INRIA, 1998.

[143] H. Mahmoud, P. Flajolet, P. Jacquet, and M. Régnier. Analytic variations on bucket selection and sorting. Research report, INRIA, 1998.

[144] P. Flajolet. Singularity analysis and asymptotics of Bernoulli sums. Research report, INRIA, 1998.

[145] P. Flajolet and W. Szpankowski. Analytic variations on redundancy rates of renewal processes. Research report, in preparation, 1998.

Institut für Algebra und Diskrete Mathematik, Technical University of Vienna, Wiedner Hauptstrasse 8–10, A-1040 Vienna, Austria.

E-mail address: `Helmut.Prodinger@tuwien.ac.at`

Department of Computer Science, Purdue University, W. Lafayette, IN 47902, U.S.A.

E-mail address: `spa@cs.purdue.edu`