Mathematical methods in the analysis of algorithms and data structures

Philippe Flajolet

INRIA
CENTRE DE ROCQUENCOURT

Rapports de Recherche

N° 400

MATHEMATICAL METHODS IN THE ANALYSIS OF ALGORITHMS AND DATA STRUCTURES

Philippe FLAJOLET

Mai 1985

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (3) 954 90 20

MATHEMATICAL METHODS IN THE ANALYSIS OF ALGORITHMS AND DATA STRUCTURES

Philippe Flajolet

Abstract: *This paper is a tutorial review of the main mathematical methods used in the* average case analysis *of algorithms and data structures.*
We survey both generating function and complex analysis techniques, and conclude with several examples related to sorting, searching and tree manipulation algorithms.

Résumé: Cet article constitue une revue des principales méthodes mathématiques utilisées en *analyse des algorithmes*.
On y présente de manière unifiée les techniques de séries génératrices et d'analyse complexe utilisées dans ce domaine. On conclue par plusieurs exemples tirés de l'algorithmique des arbres, du tri et de la recherche rapide d'informations.

MATHEMATICAL METHODS IN THE ANALYSIS OF ALGORITHMS AND DATA STRUCTURES

Philippe Flajolet

INRIA
Rocquencourt
78150 - Le Chesnay (France)

ABSTRACT

This paper is intended both as a tutorial paper and a partial review of advanced mathematical methods in the average case analysis *of algorithms and data structures.*
An analysis usually decomposes into several combinatorial enumeration problems (of words, trees, permutations, distributions ...) whose outcome is then subjected to asymptotic analysis in order to obtain results in a form that is easy to interpret.
The main technique to solve combinatorial enumeration problems is via the use of generating functions. The approach presented here is called the symbolic operator method*: a large set of combinatorial constructions have direct translations as operators on counting generating functions, so that functional equations over generating functions can be obtained rather directly for many combinatorial structures of interest.*
The main technique for asymptotic analysis in this context relies on complex analysis*: analytic function theory and uses of Cauchy's residue theorem. In most cases the asymptotic behaviour of coefficients of a generating function can be recovered directly from the generating function itself with a proper choice of integration contour (singularity analysis, saddle point methods ...).*
These methods are briefly illustrated with several examples relating to: (1) tree manipulation algorithms in compiling and symbolic manipulation systems ; (2) sorting and searching techniques based on comparisons between keys ; (3) digital search algorithms.

Lecture Notes for *A Graduate Course on Computation Theory*, Udine (Italy), September-October 1984. To be published by Computer Science Press (1985)

PART I
AN INTRODUCTION TO THE ANALYSIS OF ALGORITHMS

The task of ***analyzing*** an algorithm consists in predicting the amount of resource that the algorithm will consume when it receives as input, data of some fixed size n. Several ***complexity measures*** corresponding to various notions of resource consumption may be defined:

- time complexity measure (τ): this is the time the algorithm takes to process a particular data on a given machine model ; it may be expressed either in terms of machine cycles or time units (micro-seconds for instance) ; Knuth [Kn 68-73] has defined an abstract machine model MIX, typical of many existing machines in which all the algorithms presented are programmed, time being measured by the number of machine cycles.

- ***storage complexity measure*** (σ): this may be measured by the number of bits, bytes, words or more abstractly records that the algorithm consumes.

Simplified measures may be considered for particular algorithms: for a sorting algorithm, one often restricts attention to the number of comparisons performed or to the number of records moved (these are simplified time complexity measures). For algorithms operating on some external storage device, like a disk, a critical determinant of efficiency is usually the number of disk accesses (again a simplified time complexity measure) or the number of disk pages used (a simplified storage complexity measure).

Let A be an algorithm that operates on a set of inputs $\mathbf{E}$; the size of an element ω of $\mathbf{E}$ is denoted by $|\omega|$ (usually the size of a word is its length, the size of an array its dimension etc ...). Three quantities can be defined to characterise the behaviour of algorithm A over the set $\mathbf{E}_n$ of inputs of size n under a complexity measure μ. With $\mu a[\omega]$ denoting the complexity w.r.t. measure μ of algorithm A on input $\omega \in \mathbf{E}$, we introduce:

- the ***best-case complexity***:

$$\mu a_n^{BEST} = \min\{\mu a[\omega] / \omega \in \mathbf{E}_n\} \tag{1}$$

- the ***worst-case complexity***:

$$\mu a_n^{WORST} = \max\{\mu a[\omega] / \omega \in \mathbf{E}_n\} \tag{2}$$

- the ***average-case complexity***:

$$\mu a_n^{AVERAGE} \equiv \overline{\mu a}_n = \mathbf{E}\{\mu a[\omega] / \omega \in \mathbf{E}_n\}. \tag{3}$$

Quantities (1) and (2) give indications concerning extremal bounds on the complexity of A when applied to data of size n. Their determination usually requires the construction of particular combinatorial configurations that force extremal behaviours of the algorithm. Our main interest here is in the average complexity of some of the classical algorithms and data structure. In (3), we have used the notation $\mathbf{E}\{X\}$ to denote the ***expectation*** of the random variable X ; the determination of the average-case complexity of an algorithm therefore requires introduction the of a ***probabilistic model*** in order for this expectation to be properly defined.

Each class of algorithmic problem usually carries with it one or a few natural probabilistic models. If $\mathbf{E}_n$ is finite, the ***empirical model*** will consist in considering all elements of $\mathbf{E}_n$ to be equally likely; such models are often

considered when analyzing algorithms that operate on words, term trees or expression trees in compiling or symbolic manipulation systems. For comparison based sorting algorithms, a simple model consists in assuming that elements to be sorted are drawn independently from some continuous distribution ; this ***independence model*** is equivalent to assuming that the algorithm is applied to the reduced set $\mathbf{E}_n^*$ of all permutation of $[1..n]$, with each permutation being equally likely (having probability $1/n!$). For hashing algorithms, one will usually assume hashed values to be independent and uniformly distributed over the address space $[1..m]$; there this ***uniform model*** is again equivalent to assuming each of the m^n address sequences to be equally likely.

The preceding discussion indicates that many probabilistic models for analysis are equivalent to a model in which elements of either $\mathbf{E}_n$ or of a finite subset $\mathbf{E}_n^*$ of $\mathbf{E}_n$ are equally likely, having each probability $1/(\mathrm{card}\mathbf{E}_n)$ or $1/(\mathrm{card}\mathbf{E}_n^*)$. In that case, the average case complexity of algorithm A can be reexpressed (identifying here $\mathbf{E}_n$ and $\mathbf{E}_n^*$) as:

$$\overline{\mu a}^n = \frac{1}{\mathrm{card}\mathbf{E}_n} \sum_k k.\sigma_{n,k} \tag{4a}$$

where

$$\sigma_{n,k} = \mathrm{card}\{\omega \in \mathbf{E}_n / \mu a[\omega] = k\} \tag{4b}$$

Formula (4a) is nothing but the standard form of expectations $\mathbf{E}\{X\}=\sum_k k \Pr(X=k)$ since the probability $\Pr(X=k)$ is equal to $\sigma_{n,k}/(\mathrm{card}\mathbf{E}_n)$.

This brief discussion shows that the problem of analyzing algorithms reduces to *counting* various ***classes of combinatorial structures*** (words, trees, permutations, distributions, graphs, ...) according to their sizes and the values of some parameters related to the algorithm under consideration.

1. An example: the max-finding algorithm.

Let $X[1..n]$ be an array of positive real numbers. The following sequence of Pascal instructions returns in max the value of the largest element in $X[1..n]$

```
max:=-1;
for i:=1 to n do
    if max<X[i] then max:=X[i];
```

Apart from its data $X[1..n]$, this simple programme uses two auxiliary variables (max and i) so that its storage complexity is 2 (we do not count the input) or $n+2$ (we count it), the unit being the storage required to keep one integer or real number. A more interesting question is the time complexity of that programme. Knuth analyzes it by translating it into some fixed machine language (*MIX*) which in Pascal notation, is equivalent to using only a very reduced set of Pascal instructions, like:

```
max:=-1;
i:=0;
1 : i:=i+1;
if i>n then goto 2;
if max ≥ X[i] then goto 1;
max:=X[i];
goto 1;
2 : ...
```

This form is also equivalent to a flowchart (graph) like that of Figure 1. Thus on almost any classical (non parallel) computer, a compiled form of the programme will execute:

- a fixed number of assignments to initialise *max* and *i* ;
- $(n+1)$ comparisons of the form $i>n$?
- n increments of index *i*
- n comparisons of the form $max \geq X[i]$?
- a variable number (between 1 and n) of assignments $max:=X[i]$.

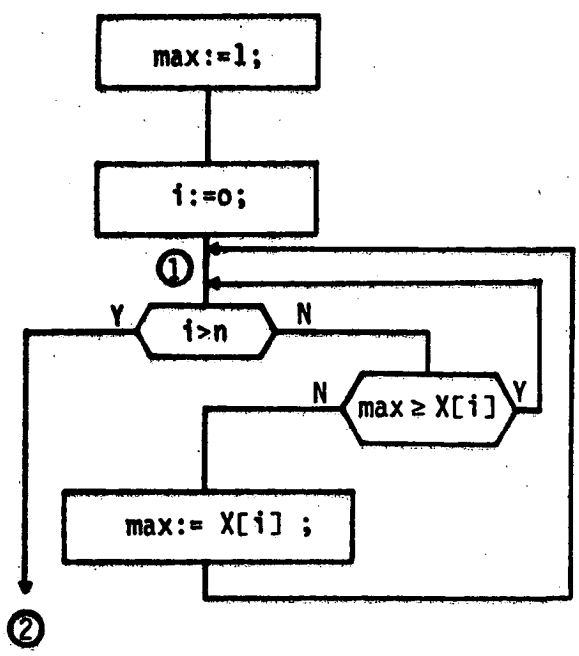


Figure 1: ***The flowchart corresponding to a low-level implementation of the max-finding procedure.***

In summary, the time complexity of max finding (***maxf***) on almost any conceivable machine is going to be of the form:

$$\tau maxf[X] = C_0 + C_1 n + C_2 EXCH[X] \qquad (5)$$

where $EXCH[X]$, for X an array, is the number of times the instruction $max := X[i]$ is executed Quantities C_0, C_1, C_2 are so-called ***implementation constants*** that reflect the execution time of elementary instructions for the machine on which the programme is executed.

The above sketch shows that the analysis of an algorithms starts with a ***flow analysis*** where one determines the number of times each instruction is executed ; taking advantage of the structure of the programme considered reduces the number of independent parameters to a minimum (using the "Kirchhoff's laws", see [Kn 68, pp 95, 167-168]). With some experience, a programme can be analyzed directly at the level of the Pascal source programme, and we shall do so in the rest of this paper. (One could also formally specify costs associated to Pascal constructs for a given machine and a given compiler.).

Formula (5) is our starting point for analysis. We can notice that $EXCH[X]$ is equal to the number of left-to-right maxima of vector X, *i.e.* the number of elements (indices $j \in [1..n]$) such that for all $i<j$: $X[i]<X[j]$. Thus $EXCH[X]$ is equal to 1 iff $X[1]$ is the largest element of the array $X[1..n]$ and $EXCH[X]$ is equal to n iff $X[1..n]$ is already sorted in increasing order: $X[1]<X[2]<X[3]\ldots$. These are obviously the extremal configurations, whence

Proposition 1: *The max finding procedure has extremal complexities described by:*

$$\tau maxf_n^{BEST} = (C_0 + C_2) + C_1 n$$

$$\tau maxf_n^{WORST} = C_0 + (C_1 + C_2) n,$$

where C_0, C_1, C_2 are implementation dependent constants.

To obtain more information on the algorithm when used repeatedly, we proceed to study it under the following probabilistic model:

Model 1: *(Uniform-Independent Model) The n elements of array X are assumed to be independently drawn from a uniform $[0,1]$ distribution.*

Let J denote the unit interval [0,1]. For X a random variable (vector) over J^n, we are interested in the probabilities

$$p_{n,k} = \Pr(EXCH[X] = k).$$

These probabilities can be evaluated by computing multiple integrals. When $n=2$, for instance, one has:

$$p_{2,1} = \Pr(X[1] \geq X[2]) = \int\int_{x_1 \geq x_2} dx_1 dx_2$$

$$p_{2,2} = \Pr(X[1] < X[2] = \int\int_{x_1 < x_2} dx_1 dx_2$$

so that

$$p_{2,1} = \frac{1}{2} \; ; \; p_{2,2} = \frac{1}{2},$$

and the expected value of $EXCH$ is

$$\overline{exch}_2 = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2 = \frac{3}{2}.$$

To avoid computation of multiple integrals, one introduces an alternative model:

Model 2: *(The permutation model). The array X is a permutation of $[1..n]$, each permutation being taken with equal probability $(1/n!)$.*

One has the important:

Lemma 1: *For the analysis of the max-finding procedure, the Uniform-Independent Model and the Permutation Model are equivalent.*

Proof: Associate to each array $X[1..n]$ consisting of (n) distinct elements its *order type* $\tau = \tau_1, \tau_2, \ldots, \tau_n$ defined by

- $\tau_1, \tau_2, \ldots, \tau_n$ is a permutation of $[1..n]$
- for all i,j: $\tau_i < \tau_j$ iff $X[i] < X[j]$.

The order type is a reduced presentation of the order properties of elements of array X. For instance if

$$X = (3.14\ ,\ 2.71\ ,\ 0.55\ ,\ 1.41\ ,\ 1.73)$$

then

$$\tau = (5\ ,\ 4\ ,\ 1\ ,\ 2\ ,\ 3).$$

(Write a 1 under the smallest element of X, a 2 under the second smallest etc...). Obviously, if $\tau(X)$ is the order type of vector X, $EXCH[X]=EXCH[\tau(X)]$.

The first observation is now that under Model 1, the probability that two array elements coincide is 0, so that the order type of a random array (under model 1) is defined with probability 1. The main observation is that each order type is equally likely, by simple symmetry considerations. For instance, if $n=3$.

$$\int_{x_1<x_2>x_3} dx_1 dx_2 dx_3 = \int_{x_1>x_2>x_3} dx_1 dx_2 dx_3.$$

Thus each order type under Model 1 has probability $1/n!$. Since the cost of the algorithm depends only on the underlying order type of the input, the lemma is established. ■

Observations : (1). The equivalence result will hold true for any model where array elements are taken independently from some continuous distribution (*i.e.* no point has a non-zero mass) like Gaussian, exponential *etc* ..., so that the permutation model is really equivalent to a general independence model.

(2). The same equivalence will apply to all algorithms that are only sensitive to the relative order of their input "keys". Thus, comparison based algorithms (bubble sort, heapsort, quicksort...) are always analyzed under the permutation model. ■

The interest of the permutation model is that the analysis reduces to a counting problem. Let $s_{n,k}$ denote the number of permutations of $[1..n]$ such that $EXCH(\sigma)=k$; from Lemma 1, we have:

$$p_{n,k} = \frac{s_{n,k}}{n!}$$

$$\overline{exch}_n = \frac{1}{n!}\sum k s_{n,k}$$

We now proceed to prove:

Theorem 1: *The max-finding procedure has average cost (under the permutation model) given by*

$$\tau max f_n = C_0 + C_1 n + C_2 H_n$$

where H_n denotes the n-th harmonic number:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.$$

Proof: Consider the set of all permutation σ of $[1..n]$ whose parameter *EXCH* has value k (there are $s_{n,k}$ of these). Two cases can occur: (i) the last element σ_n is equal to n so that $\sigma_1\sigma_2\cdots\sigma_{n-1}$ has $(k-1)$ as value of *EXCH* (this can happen in $s_{n-1,k-1}$ ways) ; (ii) the last element has one of the values $1,2,\ldots,n-1$; thus $\sigma_1\sigma_2\cdots\sigma_{n-1}$ already contains value n and has k as value of *EXCH* (this can happen in $(n-1)\times s_{n-1,k-1}$ ways). Whence the *recurrence*:

$$s_{n,k} = s_{n-1,k-1} + (n-1)s_{n-1,k}. \qquad (6)$$

Recurrence (6) is a similar to the recurrence defining elements of Pascal's triangle. It makes it possible to determine all the $s_{n,k}$.

In order to derive information from recurrence (6), we introduce generating functions. We define for each n, the quantity

$$s_n(x) = \sum_{k=1}^{n} s_{n,k} x^k.$$

Multiplying both sides of equality (6) by x^k and summing over k, we get:

$$s_n(x) = x s_{n-1}(x) + (n-1)s_{n-1}(x) = (x+(n-1))s_{n-1}(x).$$

Now from initial values $s_0(x)=1$; $s_1(x)=x$; $s_2(x)=x(x+1)$... we find the explicit form for $s_n(x)$:

$$s_n(x) = \prod_{j=0}^{n-1} (x+j). \qquad (7)$$

From there we can easily conclude since

$$\overline{exch}_n = \frac{1}{n!}\sum k s_{n,k} = \frac{s'_n(1)}{s_n(1)} \qquad (8)$$

where the logarithmic derivative of (7) permits to determine the value of (8) since:

$$\frac{s'_n(x)}{s_n(x)} = \frac{1}{x} + \frac{1}{x+1} + \cdots + \frac{1}{x+n-1}. \qquad \blacksquare$$

2. Generating functions and combinatorial enumerations: A preliminary discussion.

The previous approach is important: to count a class of structures of size n, we decompose it into simpler (smaller) classes. This *decomposition* is

reflected by a main *recurrence* relation (recurrence (6) on the example of max-finding). The recurrence relation is then attacked by the use of *generating functions*, on our example only a technical trick, leading to an explicit form (expression (7) on the example). We shall see later that, for essential reasons, generating functions are a tool of considerable generality.

We first set:

Definition: *Let* $\{a_k\}_{k\geq 0}$ *be a sequence of complex numbers. The* ordinary generating function (o.g.f.) *of sequence* $\{a_k\}$ *is defined as*

$$a(z) = \sum_k a_k z_k. \tag{9}$$

The exponential generating function (e.g.f.) *of sequence* $\{a_k\}$ *is defined as*

$$\hat{a}(z) = \sum_k a_k \frac{z^k}{k!}. \tag{10}$$

Notations: *We let* $[z^n]f(z)$ *denote the coefficient of* z^n *in* $f(z)$ *in the Taylor expansion of* f *around* $z=0$. *Thus*

$$f(z) = \sum_{n\geq 0} f_n z^n \quad => \quad [z^n]f(z) = f_n .$$

We extend that notation by setting:

$$[\frac{z^n}{n!}]f(z) = n![z^n]f(z) .$$

Those notations read as "coefficient of z^n" *(resp. coefficient of* $\frac{z^n}{n!}$*)* in $f(z)$.

Definitions (9), (10) associate power series to sequences of numbers. In the most general case (9) and (10) are to be taken as defining *formal power series* on which the arsenal of classical algebra can be applied. In most cases of interest however, the series defined by (9) and (more often) (10) are convergent, so that methods of classical analysis can further be applied to them.

The advantage of generating functions (series) over sequences satisfying recurrence relations is that they are endowed with a more visible algebraic structure (a field structure essentially). Figure 2 summarises the correspondence between some important operations on sequences and generating functions (we have omitted obvious boundary conditions).

From this table results that a large number of non-linear recurrences (those that obtain by combinations of 1-7 in Figure 2) over number sequences correspond to *functional equations* over generating functions that may often be solved using the classical tools of algebra and analysis.

Examples: (1) If $c_n = \sum_{k=0}^{n} a_k$, then $c_n = \sum_{k=0}^{n} a_k U_{n-k}$ where $U_j \equiv 1$, so that $c(z) = a(z)(1-z)^{-1}$; thus $a(z) = c(z)(1-z)$ whence $a_n = c_n - c_{n-1}$. This is the simplest case of an *inversion* relation. (In this case it could of course have been derived by elementary algebra.)

(2) If $c_n = \sum_{k=0}^{n} \binom{n}{k} a_k$, then $\hat{c}(z) = e^z \hat{a}(z)$; thus $\hat{a}(z) = e^{-z}\hat{c}(z)$ and $a_n = \sum_{k=0}^{n} (-1)^k \binom{n}{k} c_{n-k}$, yet another inversion relation.

	Sequences	o.g.f	e.g.f
1.	$c_n = a_n \pm b_n$	$c(z) = a(z) \pm b(z)$	$\hat{c}(z) = \hat{a}9z) \pm \hat{b}(z)$
2.	$c_n = \sum_{k=0}^{n} a_k b_{n-k}$	$c(z) = a(z) \times b(z)$	———
3.	$c_n = \sum_{k=0}^{n} \binom{n}{k} a_k b_{n-k}$	———	$\hat{c}(z) = \hat{a}(z) \times \hat{b}(z)$
4.	$c_n = a_{n-1}$	$c(z) = z a(z)$	$\hat{c}(z) = \int_0^z \hat{a}(z)\, dz$
5.	$c_n = a_{n+1}$	$c(z) = (a(z) - a(0))/z$	$\hat{c}(z) = \frac{d}{dz}\hat{a}(z)$
6.	$c_n = n\, a_n$	$c(z) = z\frac{d}{dz}a(z)$	$\hat{c}(z) = z\frac{d}{dz}\hat{a}(z)$
7.	$c_n = \frac{a_n}{n}$	$c(z) = \int_0^z [a(t) - a(0)]\, \frac{dt}{t}$	$\hat{c}(z) = \int_0^z [\hat{a}(t) - \hat{a}(0)]\, \frac{dt}{t}$

Figure 2: *The translation of operations on sequences into operators on generating functions: sum (1) ; Cauchy (convolution) product (2) ; binomial Cauchy (convolution) product (3) ; backward and forward shifts (4-5) ; differentiation and integration (6-7).*

(3) The implicit relation ($n \geq 0$)

$$4^n = \sum_{k=0}^{n} a_k a_{n-k}$$

with the initial condition $a_0 = 1$ is equivalent to a non-linear recurrence defining the a_n inductively:

$$a_n = \frac{1}{2}(4^n - \sum_{k=1}^{n-1} a_k a_{n-k}).$$

Introducing generating functions for the original relation, we find:

$$(a(z))^2 = \frac{1}{1-4z}$$

whence

$$a(z) = \frac{1}{\sqrt{1-4z}} \quad ; \quad a_n = \binom{2n}{n}.$$

and the solution to the original recurrence is found, by standard Newton expansion of $(1-4z)^{-1/2}$, to be:

$$a_n = \binom{2n}{n}$$

(4) The relation between generating functions $c(x) = a(x + x^2)$ corresponds (as can be checked by expanding) to the recurrence relation ;

$$c_n = \sum_k a_{n-k} \binom{n-k}{k}. \quad \blacksquare$$

Other operations on sequences have translations into generating functions; sometimes, however, analycity of intervening power series in some domain may be required. A most notable formula is for the Hadamard product: if $c_n = a_n b_n$ then

$$c(z) = \frac{1}{2i\pi} \int a(t) b\left(\frac{z}{t}\right) \frac{dt}{t}$$

for a suitable contour encircling the origin in the t-plane (this therefore assumes that the generating functions have a non zero radius of convergence). Also there is a simple relation between ordinary and exponential generating functions, via the *Laplace-Borel* transform (a mere notational variant of the classical Laplace transform):

$$a(z) = \int_0^z \hat{a}(zt) e^{-t}\, dt.$$

3. Asymptotic methods: A preliminary discussion.

Once an exact expression for the analysis of an algorithm (like that of Theorem 1) has been obtained, it is natural to try and establish approximations that may be of a form simpler to interpret. To that purpose, one determines asymptotic expansions of expressions under consideration w.r.t. to the parameter n, as n gets large. In most cases the expressions so obtained are quite accurate (typically within a few percents of the exact values) as soon as n exceeds 20-50. These asymptotic forms make comparison between algorithms much simpler.

Elementary problems usually require only simple asymptotic methods based on *real approximations*. In the case of the max-finding procedure, we have:

Theorem 2: *The max-finding procedure has average cost given by:*

$$\tau max f_n = C_1 n + C_2 \log n + O(1).$$

Proof: From the standard comparison between a decreasing function and an integral results that†

$$\frac{1}{k+1} < \int_k^{k+1} \frac{dt}{t} < \frac{1}{k}$$

so that, by summation:

$$H_{n+1} - 1 < \log(n+1) < H_n$$

and

$$H_n = \log n + O(1). \qquad \blacksquare$$

† We let log denote the natural logarithm $\log = \log_e$ and occasionally make use of the notation $lg = \log_2$.

Notice that a better approximation for H_n is available, and one has the stronger form

$$H_n = \log n + \gamma + \frac{1}{2n} + O(\frac{1}{n^2}).$$

and the expansion can be pushed to any degree of accuracy.

In the above approximation, we have made use of some of the classical notations of Landau, which we now recall:

Notations: *(1) $f(n)=O(g(n))$ iff for some constant c and for all n larger than some fixed n_0:*

$$|f(n)| < c\, g(n).$$

(2) $f(n)=o(g(n))$ iff for all c there exists a n_0 such that for all n larger than n_0:

$$|f(n)| < c\, g(n).$$

Amongst real analysis methods for obtaining asymptotic expansions, one may mention:

(A) The approximation of finite sums of continuous functions by integrals. For instance, to approximate

$$S_n = \sum_{k=1}^{n-1} \sqrt{k(n-k)}$$

consider

$$\frac{S_n}{n^2} = \sum_{k=1}^{n-1} \sqrt{\frac{k}{n}(1-\frac{k}{n})}.\frac{1}{n}$$

which is a Riemman sum relative to the function $\sqrt{x(1-x)}$. Thus as $n \to \infty$:

$$\frac{S_n}{n^2} \sim \int_0^1 \sqrt{x(1-x)}dx$$

so that

$$S_n = \frac{\pi n^2}{8} + o(n^2)$$

(B) More general expansions are obtained by the use of the *Euler Maclaurin summation formula* (covering for instance the case of H_n above and providing full asymptotic expansions).

Apart from the purpose of simplifying expressions, an equally important reason for performing asymptotic approximations, is that sometimes functional equations over generating functions are available but these only define the functions implicitly and no closed-form expression is available. Nonetheless in many such cases one can still obtain asymptotic expansions for the coefficients using complex analysis.

For instance, Polya in 1937, has obtained the asymptotic expansion of coefficients of a function satisfying a functional equation of the form

$$f(z)=\frac{1}{1-zf(z^2)}$$

for which little more is known beyond the continued fraction expansion ;

$$f(z)=\cfrac{1}{1-\cfrac{z}{1-\cfrac{z^2}{1-\cfrac{z^4}{\cdots}}}}.$$

This function occurs in the enumeration of structurally different isomeres of alcohols of the form $C_nH_{2n+1}OH$. Similarly, the counting of balanced 2-3 trees leads to the functional equation:

$$f(z) = z + f(z^2+z^3)$$

from which Odlyzko [Od 82] has shown that

$$f_n \sim \frac{\varphi^n}{n} w(\log n)$$

for some continuous periodic function w, with φ being the golden ratio $(1+\sqrt{5})/2$.

The major tool in obtaining these asymptotic estimates is the ***Cauchy integral formula*** that relates the values of a generating function in a ***complex domain*** to its coefficients:

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

for a suitable contour of integration Γ.

4. Overview of methods for the analysis of algorithms.

The main paths to be taken when analyzing algorithms and data structures are depicted in Figure 3. Main steps are:

1. Extracting basic combinatorial parameters: the original problem is transformed in this way into a combinatorial enumeration problem of a more or less classical type.
2. Obtaining exact (explicit) expressions for the average cost of the algorithm under consideration when applied to an input of size n, if at all possible.
3. Obtaining asymptotic values of these average costs for large n (this phase may or may not be carried out from the previous one).

The main routes are as follows

1- Flow analysis (*FLOW*): the use of various conservation laws (Kirchhoff's laws) possibly in relation with combinatorial properties of objects (*e.g.* a binary tree with n binary modes has $(n+1)$ external nodes) leads to a minimal set of parameters (random variables) whose expectation/distribution under the probabilistic model of use is sought.

2- Symbolic operator method (*OPER*): this is the method of choice for obtaining generating functions of average values and enumeration quantities. It uses a set of mapping lemmas with which a working kit of combinatorial constructions can be mapped directly into operators over

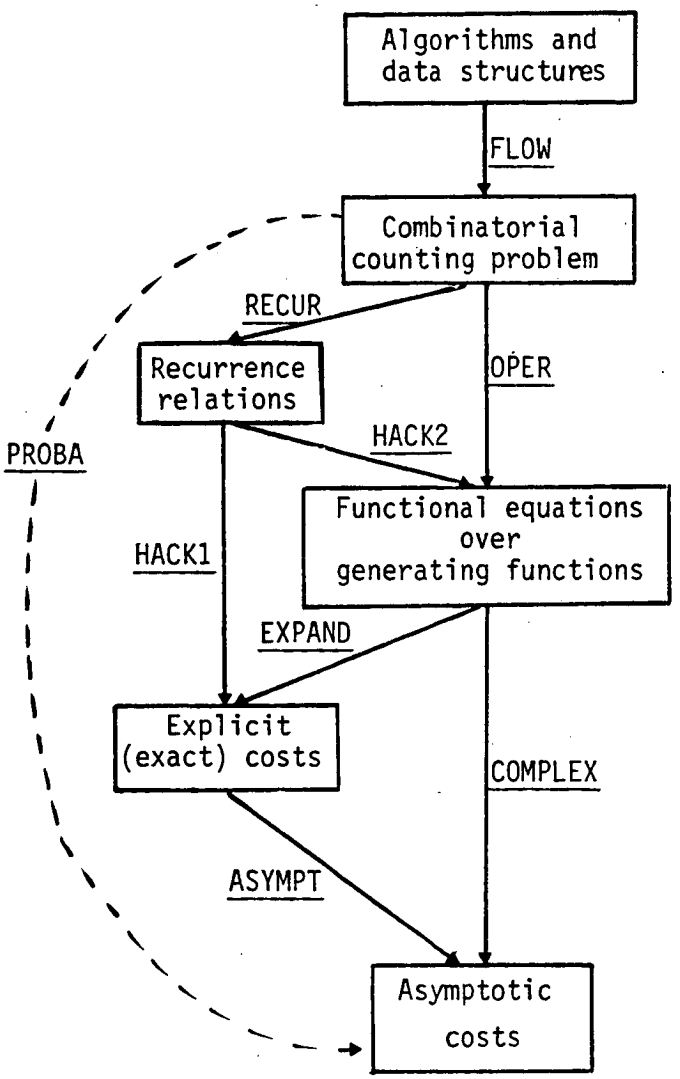


Figure 3: *Main paths for the analysis of algorithms.*

generating functions. In this way rather complicated generating function expressions are obtained often at relatively little cost.

3- Complex analysis methods (*COMPLEX*) for going from functional equations over generating functions to asymptotic of their coefficients. One uses local analysis of generating functions (it is sufficient that these be defined implicitly ; an explicit form is not required) around singularities, saddle points ... . The main tool is Cauchy's integral formula. Another important tool is the Mellin integral transform.

Other classical and important routes are:

4- Recurrences based on decompositions (*RECUR*): by looking at the way structures together with associated parameters decompose into simpler structures, one is often lead to recurrences. In happy cases, the recurrences obtained in this way can be solved explicitly by elementary methods ($HACK_1$). In other cases forming generating functions leads to functional equations after some calculations ($HACK_2$) ; however, in almost all cases where the chain $RECUR+HACK_1$ or $RECUR+HACK_2$ succeeds, it can be bypassed by the simpler symbolic operator method.

5- Taylor expansions (*EXPAND*): this applies essentially to cases where functional equations can be solved explicitly. One then uses the classical tools of algebra and analysis to extract coefficients of generating functions. The asymptotic analysis of these explicit forms (*ASYMPT*) relies then largely on real analysis techniques - like the Euler Maclaurin summation formula - and sometimes on complex analysis methods (Mellin transform techniques, most notably).

6- Direct probabilistic methods (*PROBA*): in many cases - mostly graph algorithms and combinatorial optimisation problems - one can replace the analysis of a complicated parameter by that of a much simpler one which may be asymptotically equal, equal with high probability *etc* ... . This way of approaching problems has been well illustrated by works of Erdos, Renyi and other Hungarian mathematicians whence the name of Hungarian methods sometimes given to them.

PART II
COMBINATORIAL ENUMERATION METHODS
The Symbolic Operator Approach

1. The symbolic operator method

We define here a *class of combinatorial structures* as a pair of a finite or denumerable set $\mathbf{C}$ and a function w: $\mathbf{C} \to N$ called the *size* or weight function, such that for all n, $w^{-1}(n)$ is finite. We let $\mathbf{C}_n$ denote the set of all structures in $\mathbf{C}$ that have size n. The *counting problem* for $\mathbf{C}$ is to determine the integer sequence $\{c_n\}_{n\geq 0}$ defined by

$$c_n = |w^{-1}(n)| = \text{card}\mathbf{C}_n$$

that is to determine for each n how many elements in $\mathbf{C}$ have size n. The size function is often denoted by $|.|$ or $|.|_{\mathbf{C}}$ if the dependence on $\mathbf{C}$ is to be emphasised.

The *ordinary generating function* (*o.g.f.*) of $\mathbf{C}$ (w.r.t. weight w) is the formal power series

$$c(z) = \sum_{n\geq 0} c_n z^n. \tag{1}$$

The *exponential generating function* (*e.g.f.*) of $\mathbf{C}$ (w.r.t. weight w) is the formal power series

$$\hat{c}(z) = \sum_{n\geq 0} c_n \frac{z^n}{n!} \tag{2}$$

It is useful to notice that $c(z)$ and $\hat{c}(z)$ can be expressed alternatively as

$$c(z) = \sum_{\sigma\in\mathbf{C}} z^{w(\sigma)} \quad ; \quad \hat{c}(z) = \sum_{\sigma\in\mathbf{C}} \frac{z^{w(\sigma)}}{w(\sigma)!}. \tag{3}$$

To see it, observe that the term z^n in (3) appears as many times as there are structures of size n in $\mathbf{C}$.

The main approach which we explore here for the counting problem of $\mathbf{C}$ is via the generating functions $c(z)$ or $\hat{c}(z)$.

A *combinatorial construction* Φ of degree k is an operation that associates to k classes of structures $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k$ a class $\mathbf{A} = \Phi(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k)$.

Definition: *A combinatorial construction is* o.g.f. admissible *iff there exists an operator Ψ over formal power series such that*

$$\mathbf{A} = \Phi(\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_k) \implies a(z) = \Psi(c_1(z), c_2(z), \ldots, c_k(z))$$

($a(z), c_1(z) \cdots$ are the o.g.f. of classes $\mathbf{A}, \mathbf{C}_1, \cdots$).

One has the obvious analogous notion of e.g.f. admissible functions.

In other words, a construction is admissible iff the counting sequence $\{a_n\}$ of $\mathbf{A}$ can be determined from the counting sequence $\{c_{1,n}\} \cdots$ of $\mathbf{C}_1 \cdots$

so that no further internal structural information on $\mathbf{C}_1, \mathbf{C}_2 \ldots$ is required to solve the counting problem for **A**.

Notations: In the sequel, we adhere - unless otherwise stated - to the notational convention of representing a class (**C**), the counting sequence ($\{c_n\}$ or $\{C_n\}$) and the corresponding generating functions ($c(z)$ or $C(z)$; $\hat{C}(z)$ or $\hat{c}(z)$) by the same group of letters.

The remainder of this chapter is devoted to the presentation of a certain number of admissible constructions. Admissibility lemmas thus map these combinatorial constructions into operators over generating functions. The counting problem for a class **C** therefore reduces to finding a suitable construction of **C** in terms of simpler structures (and possibly **C** itself if the construction is recursive) by means of admissible constructions : if the construction is non-recursive, then the generating function for **C** will obtain as a functional on simpler functions ; if the definition is recursive, then one obtains a *functional equation* defining $c(z)$ (or $\hat{c}(z)$) implicitly. We call this approach the *symbolic operator method* for counting problems.

Notice that while the classical enumeration approach based on producing recurrences from suitable decompositions is very sensitive to small variations on the formulation of the problem considered, the operator approach is usually far more flexible. Before presenting admissible for ordinary generating functions (Section 2) and for exponential generating functions (Section 3), we illustrate this method informally by an example taken from the counting of permutations.

Let $\mathbf{P}=\cup\mathbf{P}_n$ be the class of all permutation, with $\mathbf{P}_n$ the set of permutations of size n, *i.e.* permutation over $[1..n]$. A direct reasoning (value 1 can appear in any of n positions, value 2 in any of the remaining $(n-1)$ positions ...) shows that

$$p_n = \mathrm{card}\mathbf{P}_n \equiv n!, \tag{4}$$

and the e.g.f of the class of permutations is

$$\hat{p}(z) = \sum_{n\geq 0} n!\frac{z^n}{n!} = \frac{1}{1-z}, \tag{5}$$

equations (4) and (5) being clearly equivalent.

One way to arrive at (5) by the symbolic operator method is to construct permutations at *"sets-of" cyclic permutations* (each permutation has a unique cycle decomposition). If **C** is the class of cyclic permutations, $c_n=(n-1)!$ so that the e.g.f. of **C** is

$$\hat{c}(z) = \sum(n-1)!\frac{z^n}{n!} = \sum\frac{z^n}{n}.$$

Thus one has :

$$\hat{c}(z) = \log(1-z)^{-1} \tag{6}$$

Now the set-of construction in this context is known, in the operator approach to correspond to a left composition with an exponential (see Section 3 for precise statements); here this gives $\hat{p}(z)=\exp(\hat{c}(z))$ or :

$$\hat{p}(z) = \exp\{\log(1-z)^{-1}\}. \qquad (7)$$

Equation (7) is also clearly equivalent to (6) whence to (5).

This seemingly uselessly complicated detour is important. The method behind the derivation of (7) allows for a large number of variations :

A. Restrictions on the number of cycles. Let $\mathbf{P}^{\Gamma}$ be the set of permutations whose number of cycles is in some fixed set $\Gamma \subset \mathbf{N}$; the corresponding exponential generating function is obtained by composing with $\gamma(u) = \sum_{j \in \Gamma} \frac{u^j}{j!}$ (so that $\Gamma = \mathbf{N}$ gives back the exponential) the e.g.f. of cyclic permutations. For instance the e.g.f. of permutations having an even number of cycles is

$$\hat{q}(z) = \cosh(\log(1-z)^{-1}) = \frac{1}{2}\left(\frac{1}{1-z} + 1 - z\right)$$

so that there $q_n = \frac{n!}{2}$ for $n \geq 2$.

B. Restrictions on cycle length : let ${}^{\Lambda}\mathbf{P}$ be the set of permutations whose cycles all have length in a fixed set $\Lambda \subset \mathbf{N}$. The corresponding exponential generating function is obtained by replacing in (7) the function $\log(1-z)^{-1}$ by the function $\alpha(z) = \sum_{j \in \Lambda} \frac{z^j}{j}$. (so that $\Lambda = \mathbf{N}$ gives back equation (7)). For instance to obtain the e.g.f. of permutations without cycles of length 1, replace $\log(1-z)^{-1}$ by $\log(1-z)^{-1} - z$ so that this function is

$$\hat{r}(z) = \frac{e^{-z}}{1-z}.$$

From there follows the (19-th century) result that the number of derangements - permutations without fixed points - is

$$r_n = \sum_{j=0}^{n} \frac{(-1)^j}{j!}.$$

C. Joint restrictions of the two previous types can be combined defining the class ${}^{\Lambda}\mathbf{P}^{\Gamma}$ whose e.g.f. is $\gamma(\lambda(z))$. For instance to obtain the e.g.f. $\hat{s}(z)$ of permutation having an odd number of cycles each of an odd length, take

$$\gamma(u) = \frac{1}{2}(e^{u} - e^{-u})$$

$$\lambda(z) = \frac{1}{2}(\log(1+z) - \log(1-z)) = \log\sqrt{\frac{1+z}{1-z}}$$

so that

$$\hat{s}(z) = \frac{z}{\sqrt{1-z^2}},$$

whence by expanding

$$s_{2n+1} = \frac{(2n)!(2n+1)!}{2^{2n}(n!)^2}.$$

This example illustrates the flexibility of the operator approach, i.e. its *insensitivity* to a large number of changes in definitions of combinatorial structures.

2. Admissible constructions for ordinary generating functions.

A kit of admissible constructions for e.g.f's is displayed on Figure 1, together with the corresponding operators over generating functions. Definitions and mapping lemmas follow.

Construction		Operator
Union	$\mathbf{C = A+B}$	$c(z) = a(z)+b(z)$
Cart. product	$\mathbf{C = A\times B}$	$c(z) = a(z).b(z)$
Diagonal	$\mathbf{C = \Delta(A\times A)}$	$c(z) = a(z^2)$
Sequence-of	$\mathbf{C = A^*}$	$c(z) = (1-a(z))^{-1}$
Marking	$\mathbf{C = \mu A}$	$c(z) = z\frac{d}{dz}a(z)$
Substitution	$\mathbf{C = A[B]}$	$c(z) = a(b(z))$
Set-of	$\mathbf{C = 2^A}$	$c(z) = \exp(a(z)-\frac{1}{2}a(z^2)+\frac{1}{3}a(z^3)\cdots)$
Multiset-of	$\mathbf{C = M\{A\}}$	$c(z) = \exp(a(z)+\frac{1}{2}a(z^2)+\frac{1}{3}a(z^3)\cdots)$

Figure 1: *Admissible constructions for ordinary generating functions.*

Definition 1: *A class* **C** *is the* union (sum) *of two classes* **A** *and* **B** *which we denote by* **C=A+B** *iff:*

- *in the set-theoretic sense* $\mathbf{C=A\cup B}$ *;*
- *sizes* $|.|_{\mathbf{A}}$ *and* $|.|_{\mathbf{B}}$ *are compatible over* $\mathbf{A\cap B}$ *and* $|x|_{\mathbf{C}}$ *= if* $x\in\mathbf{A}$ *then* $|x|_{\mathbf{A}}$ *else* $|x|_{\mathbf{B}}$.

Definition 2: *A class* **C** *is the* cartesian product *of classes* **A** *and* **B**, *denoted by* **C=A×B**, *iff*

- *in the set-theoretic sense* **C=A×B** *;*
- $|(\alpha,\beta)|_{\mathbf{C}}=|\alpha|_{\mathbf{A}}+|\beta|_{\mathbf{B}}$.

Definition 3: *A class* **C** *is the* sequence *class of class* **A** *iff with* ε *a structure of size 0 (called the empty structure):*

$$\mathbf{C} = \{\varepsilon\}+\mathbf{A}+\mathbf{A\times A}+\mathbf{A\times A\times A}+\cdots$$

with size being defined consistently with unions and cartesian products.

Definition 4: *A class* **C** *is the* diagonal *of* **A**×**A** *denoted by* **c**=Δ(**A**×**A**) *iff* **C** *consists of all elements* $(\alpha,\alpha),\alpha\in\mathbf{A}$, *with*

$$|(\alpha,\alpha)|_{\mathbf{C}} = 2|\alpha|_{\mathbf{A}}.$$

Definition 5: *A class* **C** *is the* marking *of class* **A** *denoted by* $\mathbf{C} = \mu\mathbf{A}$ *iff*

$$\mathbf{C} = \sum_{n=0}^{\infty} A_n \times [1..n]$$

with $|(\alpha,\nu)|_{\mathbf{C}} = |\alpha|_{\mathbf{A}}$.

Definition 6: *A class* **C** *is the* composition *of class* **A** *and* **B**, *denoted by* **C**=**A**[**B**], *iff*

$$\mathbf{C} = \sum_{n=0}^{\infty} \mathbf{A}_n \times \mathbf{B} \times \mathbf{B} \times \cdots \times \mathbf{B},$$

the number of factors in the general term being equal to n, *with* $|(\alpha,\beta_1,\beta_2,\ldots,\beta_n)|_{\mathbf{C}} = |\beta_1|_{\mathbf{B}} + |\beta_2|_{\mathbf{B}} + \cdots + |\beta_n|_{\mathbf{B}}$.

Definition 7: *A class* **C** *is the* powerset class *of class* **A** *denoted* $\mathbf{C}=2^{\mathbf{A}}$ *iff, in the set theoretic sense,* **C** *is the class of subsets of* **A** : $\mathbf{C}=2^{\mathbf{A}}$ *and*

$$|\{\alpha_1,\alpha_2,\ldots,\alpha_k\}|_{\mathbf{C}} = |\alpha_1|_{\mathbf{A}} + |\alpha_2|_{\mathbf{A}} + \cdots + |\alpha_k|_{\mathbf{A}}.$$

Definition 8: *A class* **C** *is the* multiset class *of class* **A** *denoted* $\mathbf{C}=M\{\mathbf{A}\}$ *iff* **C** *consists of multisets of elements of* **A** *of the form* $\{\alpha_1^{j_1},\alpha_2^{j_2},\ldots,\alpha_k^{j_k}\}$ (α^j *means* α *repeated* j *times) and*

$$|\{\alpha_1^{j_1},\alpha_2^{j_2},\cdots,\alpha_k^{j_k}\}|_{\mathbf{C}} = j_1|\alpha_1|_{\mathbf{A}} + j_2|\alpha_2|_{\mathbf{A}} + \cdots + j_k|\alpha_k|_{\mathbf{A}}.$$

A few words of explanation are in order. We say that **C** is the *disjoint union* of **A** and **B** if the intersection $\mathbf{A}\cap\mathbf{B}$ is empty. The notion of a cartesian product of classes (and of diagonals) is the standard one with the size of a couple being the sum of the sizes of its components ; the notion extends trivially to the product of any number of factors. The notion of a power class also corresponds to the standard power set construction. The power multiset class of **A**, $M\{\mathbf{A}\}$ is the class of sets of elements of **A** with repetitions allowed ; it thus corresponds to the standard multiset construction.

Composition and marking are useful when dealing with objects like trees, graphs, words consisting of *atomic elements* (nodes, edges, positions *etc* ...) where the size of a structure is the number of elements it comprises. A marked structure from $\mu\mathbf{A}$ in this context is then a structure of **A** augmented by distinguishing one of its elements. Similarly the substitution operation **A**[**B**] is equivalent to constructing all structures obtained from some $\alpha\in\mathbf{A}$ by substituting to all atomic elements of α objects from β still retaining the structural properties of α (this is really a sort of "marked" substitution).

Notice also that, in order for the sequence construction $\mathbf{A}^*$, substitution construction **B**[**A**] and multiset-of construction $M\{\mathbf{A}\}$, to be defined (*i.e.* to

result in sets that are classes of structures satisfying the finiteness condition of $|\ |^{-1}$), one usually has to impose the condition that **A** contains no structure of size 0. We shall also impose a similar restriction on the set-of construction. Such conditions ensure that the operators given in Figure 1 are well defined operators over formal power series.

We have

Theorem 1: *The constructions of disjoint union, cartesian product, diagonal, sequence-of, marking, substitution, set-of (powerset) and multiset-of are admissible.*

The corresponding operators are given in Figure 1. The proof of this theorem proceeds through a chain of easy lemmas.

Lemma 1: *If* $\mathbf{A}\cap\mathbf{B}=\phi$ *and* $\mathbf{C}=\mathbf{A}+\mathbf{B}$, *then* $c(z)=a(z)+b(z)$.

Proof : $c_n=a_n+b_n$. ∎

Lemma 2: *If* $\mathbf{C}=\mathbf{A}\times\mathbf{B}$ *then* $c(z)=a(z).b(z)$.

Proof :

$$c(z)=\sum_{(\alpha,\beta)\in\mathbf{C}} z^{|(\alpha,\beta)|}$$

$$=\sum_{\alpha\in\mathbf{A}}\sum_{\beta\in\mathbf{B}} z^{|\alpha|+|\beta|}=\sum_{\alpha\in\mathbf{A}} z^{|\alpha|}.\sum_{\beta\in\mathbf{B}} z^{|\beta|}.$$ ∎

Lemma 3: *If* $\mathbf{C}=\Delta(\mathbf{A}\times\mathbf{A})$ *then* $C(z)=a(z^2)$.

Proof : $c_{2n}=a_n; c_{2n+1}=0$. ∎

Lemma 4: *If* $\mathbf{C}=\mathbf{A}^*$, *then* $c(z)=(1-a(z))^{-1}$

Proof : $c(z)=\sum_{k\geq 0}(a(z))^k$. ∎

Lemma 5: *If* $\mathbf{C}=\mu\mathbf{A}$, *then* $c(z)=z\frac{d}{dz}a(z)$.

Proof : $c_n=na_n$. ∎

Lemma 6: *If* $\mathbf{C}=\mathbf{A}[\mathbf{B}]$, *then* $c(z)=a(b(z))$.

Proof : From the union and cartesian product mapping lemmas, one has

$$c(z)=\sum_k b_k c(z)^k.$$ ∎

Lemma 7: *If* $\mathbf{C}=2^{\mathbf{A}}$, *then*

$$c(z)=\exp\{\sum_{j=1}^{\infty}\frac{(-1)^{j-1}}{j}a(z^j)\}$$

Proof : Class **C** is isomorphic to the finite size elements of the (infinite) cartesian product

$$\mathbf{C}=\prod_{\alpha\in\mathbf{A}}\{\{\varepsilon\}+\{\alpha\}\}$$

(ε a null structure of size 0) so that translating to generating functions

$$c(z)=\prod_{\alpha\in A}(1+z^{|\alpha|})$$

and grouping terms :

$$=\prod_{n=1}^{\infty}(1+z^n)^{a_n}.$$

Computing $\log c(z)$, we find :

$$\begin{aligned}\log c(z) &= \sum_{n=1}^{\infty}a_n\log(1+z^n)\\ &= \sum_{n=1}^{\infty}a_n\sum_{j=1}^{\infty}\frac{(-1)^{j-1}}{j}z^{nj}\\ &= \sum_{j=1}^{\infty}\frac{(-1)^{j-1}}{j}\sum_{n=1}^{\infty}a_n z^{nj}\\ &= \sum_{j=1}^{\infty}\frac{(-1)^{j-1}}{j}a(z^j).\ \blacksquare\end{aligned}$$

Lemma 8: *If* $\mathbf{C}=M\{\mathbf{A}\}$, *then*

$$c(z)=\exp\{\sum_{j=1}^{\infty}\frac{1}{j}a(z^j)\}.$$

Proof : The class **C** is isomorphic to the finite size elements of the (infinite) cartesian product

$$\mathbf{C}=\prod_{\alpha\in\mathbf{A}}\{\alpha\}^{*}$$

so that by the mapping lemma for the sequence construction

$$\begin{aligned}c(z) &= \prod_{\alpha\in A}(1-z^{|\alpha|})^{-1}\\ &= \prod_{n=1}^{\infty}(1-z^n)^{-a_n}\end{aligned}$$

and the calculation develops as in the previous case. ■

3. Admissible constructions for exponential generating functions.

We consider here particular classes of combinatorial structures consisting of *labelled objects*. Corresponding constructions have natural interpretation in terms of exponential generating functions.

We shall first motivate our constructions over labelled structures by an example, namely determining the number of connected graph over a set of n distinguished vertices. Let $\mathbf{G}=\bigcup_n \mathbf{G}_n$ be the class of *labelled graphs* where $\mathbf{G}_n$ is the set if all undirected graphs over the set of vertices $[1..n]$. Let $\mathbf{K}$ be the subclass of $\mathbf{G}$ consisting of all graphs of $\mathbf{G}$ that are connected. One interesting question is the relation between the quantities $k_n=\mathrm{card}(\mathbf{K}_n)$ and $g_n=\mathrm{card}(\mathbf{G}_n)$. Notice first that one has directly

$$g_n=2^{\frac{n(n-1)}{2}}=2^{\binom{n}{2}} \tag{8}$$

since a graph over n vertices is obtained by selecting a subset of the set of the $\binom{n}{2}$ possible edges. The graphs corresponding to $n=3$ are depicted in Figure 2.

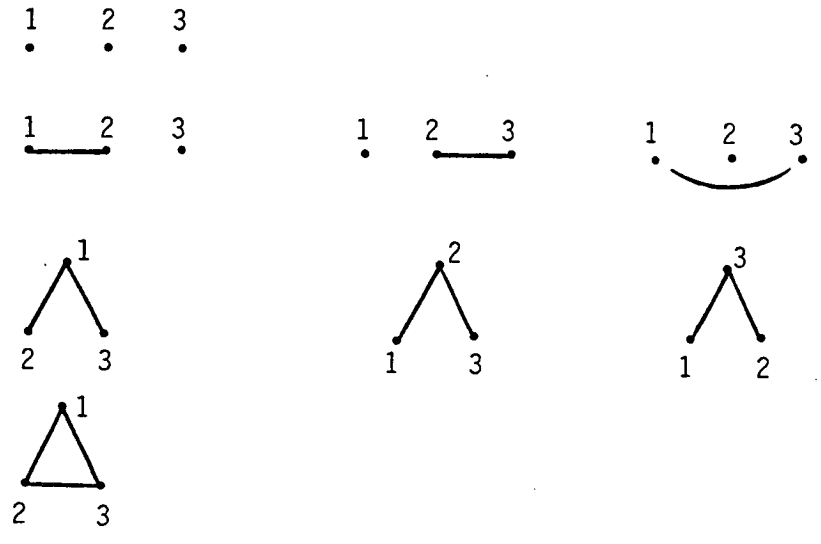


Figure 2 *The labelled graphs over the set of vertices {1,2,3}. From this table results that* $g_3=8$ *and* $k_3=1$

To approach the determination of $\{k_n\}_{n\geq 0}$ we define $\mathbf{G}_n^{(c)}$ as the class of graphs consisting of c connected components so that $\mathbf{G}_n^{(1)}=\mathbf{K}_n$, and we start by relating $g_n^{(c)}=\mathrm{card}\mathbf{G}_n^{(c)}$ to k_n. Let us take first $c=2$.

The cartesian product $\mathbf{K}\times\mathbf{K}$ generates couples of connected graphs. But one has $\mathbf{G}^{(2)}\neq\mathbf{K}\times\mathbf{K}$ for the following two reasons:

(A) Connected components of graphs in $\mathbf{G}^{(2)}$ are not ordered, while components of elements of $\mathbf{K}\times\mathbf{K}$ are, by definition of the cartesian product.

(B) Elements of $\mathbf{K}\times\mathbf{K}$ are *not* well labelled in the sense that structures of size n do not have elements (nodes) labelled with distinct integers from $[1..n]$.

To take care of problem (B), one must ***relabel*** objects from $\mathbf{K}\times\mathbf{K}$ to make them well labelled objects.

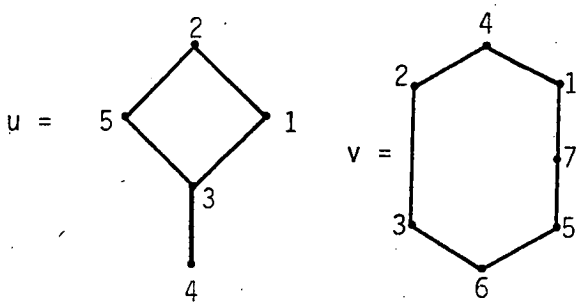
Definition: *A* bipartition Π *of* $[1..n]$ *is a pair* $\Pi=(\alpha,\beta)$ *of subsets of* $[1..n]$ *such that* $\alpha\cup\beta=[1..n]$, $\alpha\cap\beta=\phi$. *The* type *of the bipartition is the integer pair* $(|\alpha|,|\beta|)$.

Let $c=(u,v)$ be a pair of labelled structures, so that u is labelled by elements from $[1..l]$ and v is labelled by elements of $[1..m]$. The *action* of a bipartition $\Pi=(\alpha,\beta)$ of type (l,m) on $c=(u,v)$ is defined as the pair $\bar{c}=(\bar{u},\bar{v})$ where $\bar{u}$ is similarly obtained from u by replacing labels $1,2,\cdots l$ by $\alpha_1,\alpha_2,\cdots\alpha_l$ and $\bar{v}$ is obtained from v by replacing labels $1,2,\cdots,m$ by $\beta_1,\beta_2,\cdots,\beta_m$ where

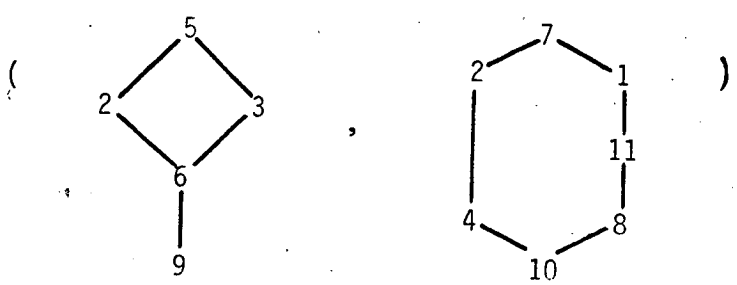
$$\alpha_1<\alpha_2<\cdots<\alpha_m \;;\; \beta_1<\beta_2<\cdots<\beta_m$$

are the elements taken in increasing order of α and β. The action of Π on u and v is denoted by $\Pi<(u,v)>$.

Example : Let $c=(u,v)$ be defined by



Consider the bipartition $\Pi=(\{3,5,6,9,12\},\{1,2,4,7,8,10,11\})$; its type is (5,7) so that its action on (u,v) is defined. The result is the couple :



which is a well labelled structure over $[1..12]$. ■

Definition 9: *The* partitional product *of two labelled structures* $\bar{u}$ *and* $\bar{v}$ *denoted by* $u*v$ *is defined as the set*

$$u*v=\{\ \Pi<(u,v)>\ |\ \Pi \text{ of type } (|u|,|v|)\ \}$$

The partitional product *of two classes of (labelled) structures* **A** *and* **B** *is defined as*

$$\mathbf{A}*\mathbf{B}=\bigcup_{\substack{u\in\mathbf{A}\\ v\in\mathbf{B}}}(u*v).$$

(The size *of any element of* $(u*v)$ *is* $|u|+|v|$*).*

Returning to our original problem concerning the enumeration of labelled graphs, we thus see that the partitional product $\mathbf{K}*\mathbf{K}$ generates all well labelled couples of connected graph. Each graph of $\mathbf{G}^{(2)}$ formed with two connected components K_1,K_2 thus appears in $\mathbf{K}*\mathbf{K}$ twice : once as (K_1,K_2) and once as (K_2,K_1). We can therefore write the symbolic equations :

$$2\mathbf{G}^{(2)}=\mathbf{K}*\mathbf{K}. \tag{9a}$$

$$\mathbf{G}^{(2)}=\frac{1}{2}(\mathbf{K}*\mathbf{K}) \tag{9b}$$

The main interest of the partitional product for enumerations is the following :

Lemma 9: *If classes* **A** *and* **B** *have exponential generating functions* $\hat{a}(z)$ *and* $\hat{b}(z)$*, then the exponential generating function* $\hat{c}(z)$ *of class* $\mathbf{C}=\mathbf{A}*\mathbf{B}$ *satisfies :*

$$\hat{c}(z)=\hat{a}(z).\hat{b}(z).$$

Proof : If u and v are structures of **A** and **B** of respective size l and m, then the number of bipartitions of type (l,m) is the binomial coefficient

$$\binom{l+m}{l}.$$

Thus the cardinality C_n of $\mathbf{C}_n$ satisfies the recurrences :

$$c_n = \sum_{l+m=m}\binom{l+m}{l}a_l b_m,$$

$$c_n = \sum_{l=0}^{n}\binom{n}{l}a_l b_{n-l},$$

$$\frac{c_n}{n!} = \sum_{l=0}^{n}\frac{a_l}{l!}\frac{b_{n-l}}{(n-l)}.$$

From the last equality, the lemma follows. ■

We have thus found for $\hat{g}^{(2)}(z)$ the relation

$$\hat{g}^{(2)}(z) = \frac{1}{2}(\hat{k}(z))^2.$$

A straightforward generalization shows that more generally

$$\hat{g}^{(c)}(z)=\frac{1}{c!}(\hat{k}(z))^c \quad (10)$$

Since $G=\sum_{c=0}^{\infty} G^{(c)}$, we find from (10) the relation

$$\hat{g}(z)=1+\hat{k}(z)+\frac{1}{2!}(\hat{k}(z))^2+\frac{1}{3!}(\hat{k}(z))^3+\cdots$$

($\mathbf{G}^{(0)}$ consists of the empty graph on 0 vertices), or :

$$\hat{g}(z)=\exp(\hat{k}(z)). \quad (11)$$

Using (8) and inverting (11), we have thus found:

Proposition 1: *The exponential generating function of the class of labelled graphs satisfies*

$$\hat{k}(z)=\log\left(1+\sum_{n\geq 1} 2^{n(n-1)/2}\frac{z^n}{n!}\right).$$

Notice that the above series is *divergent* ; however $\hat{k}(z)$ is defined as a formal power series that can be evaluated by $\log(1+u)=u-\frac{u^2}{2}+\frac{u^3}{3}\cdots$. Taking coefficients, we find :

$$k_n=\sum_{j\geq 1}\frac{(-1)^{j-1}}{j}\sum_{n_1+n_2+\ldots+n_j=n\ ,\ n_j\geq 1}\binom{n}{n_1,n_2,\ldots,n_j}2^{\binom{n_1}{2}+\binom{n_2}{2}+\ldots+\binom{n_j}{2}},$$

from which one can conclude for instance:

Proposition 2: *As n tends to infinity the ratio k_n/g_n tends do 1.*

Thus almost all graphs of size n are connected for large n.

Definition 10: *The k-th* partitional power ($k\geq 1$) *of class* **A** *is defined by*

$$\mathbf{A}^{<k>}=\mathbf{A}*\mathbf{A}*\ldots*\mathbf{A}$$

where the number of factors is equal to k. When $k=0$, $a^{<0>}$ is defined as a class consisting of a unique structure of size 0 (called the empty structure or null structure and usually denoted by ε).
The partitional complex $\mathbf{A}^{<\bullet>}$ of **A** *is defined as the sum :*

$$\mathbf{A}^{<\bullet>}=\sum_{k=0}^{\infty}\mathbf{A}^{<k>}.$$

Definition 11: *The k-th Abelian partitional power of class* **A** *is defined as*

$$\mathbf{A}^{[k]}=\{\{w_1,w_2,\ldots,w_k\}\mid(w_1,w_2,\ldots,w_k)\in\mathbf{A}^{<k>}\}$$

The abelian partitional complex $a^{[\bullet]}$ of a is defined as the sum

$$a^{[\bullet]}=\sum_{k=0}^{\infty}a^{[k]}.$$

Notice that the partitional complex construct is for labelled objects the analogue of the sequence construct (order of elements in k-uples count) ; the abelian partitional is a sort of analogue of the power-set construct (orders of elements are *not* taken into account). We then have :

Lemma 10: *Assume that* $\mathbf{C}=\mathbf{A}^{<\bullet>}$, *then*

$$\hat{c}(z)=\frac{1}{1-\hat{a}(z)}.$$

Lemma 11: *Assume that* $\mathbf{C}=\mathbf{A}^{[\bullet]}$, *then*

$$\hat{c}(z)=exp(\hat{a}(z)).$$

Proofs are direct extensions of the previous ones. For the partitional complex, one has :

$$\hat{c}(z) = \sum_{k\geq 0}(\hat{a}(z))^k = \frac{1}{1-\hat{a}(z)}.$$

For the Abelian partitional complex, each element of $\mathbf{A}^{[k]}$ corresponds to $k!$ elements of $\mathbf{A}^{<k>}$. In symbols

$$\mathbf{A}^{<k>}\equiv k!\mathbf{A}^{[k]}.$$

so that, there

$$\hat{c}(z) = \sum_{k\geq 0}\frac{1}{k!}(\hat{a}(z))^k = \exp(\hat{a}(z)).$$

Construction		Operator
Union	$\mathbf{C} = \mathbf{A}\cup\mathbf{B}$	$\hat{c}(z) = \hat{a}(z)+\hat{b}(z)$
Partit. prod.	$\mathbf{C} = \mathbf{A}*\mathbf{B}$	$\hat{c}(z) = \hat{a}(z).\hat{b}(z)$
Partit. complex	$\mathbf{C} = \mathbf{A}^{<\bullet>}$	$\hat{c}(z) = (1-\hat{a}(z))^{-1}$
Abel. part. complex	$\mathbf{C} = \mathbf{A}^{[\bullet]}$	$\hat{c}(z) = \exp(\hat{a}(z))$
Marking	$\mathbf{C} = \mu\mathbf{A}$	$\hat{c}(z) = z\frac{d}{dz}\hat{a}(z)$
Labelled subst.	$\mathbf{C} = \mathbf{A}[\mathbf{B}]$	$\hat{c}(z) = \hat{a}(\hat{b}(z))$
Min-rooting	$\mathbf{C} = \rho\mathbf{A}$	$\hat{c}(z) = \int_0^z \hat{a}(z)\,dz$

Figure 3: *A set of e.g.f. admissible constructs.*

In summary, we get:

Theorem 2: *The constructions: disjoint union, partitional product, partitional complex, abelian partitional complex and marking are e.g.f. admissible.*

The corresponding operators are given in Figure 3. Other operations (min-rooting, labelled substitution) could also be shown to be e.g.f. admissible.

4. Sample applications.

We give here some brief indications on how to derive a collection of combinatorial enumeration results using the symbolic operator method. From our previous discussions, the problem reduces to finding proper constructions (decompositions) for classes of combinatorial structures in terms of admissible set-theoretic constructs.

1. Combinations: Let **C** be the power set of $[1..m]$, where m is a fixed integer; an element of **C** is sometimes called a *combination* of elements of $[1..m]$. Then:

$$\mathbf{C} \approx \prod_{\xi=1}^{m} (\{\varepsilon\}+\{\xi\})$$

with ε the null structure (of size 0). Thus, translating to o.g.f:

$$C(z) = (1+z)^m$$

and the number of n-combinations of a set of m elements is:

$$[z^n](1+z)^m = \binom{m}{n} .$$

2. Combinations with repetitions. Let **M** be the multiset class of $[1..m]$. An element of **M** is sometimes called a *combination with repetitions* of elements of $[1..m]$. Thus:

$$\mathbf{M} \approx \prod_{\xi=1}^{m} \{\xi\}^*$$

$$M(z) = (1-z)^{-m}$$

so that the number of n-combinations with repetitions of a set with m elements is found to be;

$$[z^n](1-z)^{-m} = \binom{n+m-1}{m-1}$$

3. Arrangements. An *arrangement* of n elements of $[1..m]$ is an injective map from $[1..n]$ to $[1..m]$. The set **A** of all arrangements with m fixed has the presentation:

$$\mathbf{A} \approx (\{\varepsilon\}+\{\underline{1}\})^m$$

where $\underline{1}$ represents a labelled structure of size 1. Thus the e.g.f of **A** is:

$$\hat{A}(z) = (1+z)^m$$

and the number of n-arrangements from a set with m elements is:

$$[\frac{z^n}{n!}](1+z)^m = n(n-1)(n-2)\cdots(n-m+1) .$$

4. Set partitions. A *partition* of a set **S** is a family of sets $b = \{\beta_1, \cdots, \beta_k\}$ such that the β_j _called blocks_ are pairwise disjoint and cover **S**. Let **B** be the family of all partitions of an initial segment of $\mathbf{N} = \{1,2,3,\cdots\}$. Then:

$$\mathbf{B} \approx \{\{\underline{1}\} + \{\underline{12}\} + \{\underline{123}\} + \cdots\}^{[\bullet]},$$

thus

$$\hat{B}(z) = \exp(e^z - 1) .$$

From that last equation results that the number of partitions of a set with n elements is the n-th Bell number:

$$B_n = [\frac{z^n}{n!}]\exp(e^z-1)$$
$$= \frac{1}{e}\sum_{k\geq 0}\frac{k^n}{n!} .$$

A similar reasoning shows that the number of partitions of a set of cardinality n comprising k blocks is:

$$S_{n,k} = [\frac{z^n}{n!}]\frac{(e^z-1)^k}{k!}$$
$$= \frac{1}{k!}\sum_{0\leq j\leq k}\binom{k}{j}(-1)^j(k-j)^n ,$$

a Stirling number of the second kind.

5. Permutations. We have already examined the decomposition of *permutations* into cycles. If **C** is the class of all cyclic permutations and **P** the class of all permutations, then $\mathbf{P} \approx \mathbf{C}^{[\bullet]}$. In particular, the class of permutations with k cycles, $\mathbf{C}^{[k]}$, has for e.g.f:

$$\hat{C}^{[k]} = (-\log(1-z))^k .$$

Let $s_{n,k}$ be the number of permutations of $[1..n]$ with k cycles; $s_{n,k}$ is called a Stirling number of the first kind and $s_{n,k} = [z^n/n!]\,\hat{C}^{[k]}(z)$. Using bivariate generating functions, we easily find:

$$\sum_{n,k} s_{n,k}u^k\frac{z^n}{n!} = (1-z)^{-u}$$

whence by expanding the identity:

$$\sum_k s_{n,k}u^k = u(u+1)(u+2)\cdots(u+n-1)$$

Thus these numbers coincide with those appearing in the analysis of the max-finding procedure:

$$s_{n,k} = [u^k]\,u(u+1)(u+2)\cdots(u+n-1) .$$

6. Integer compositions. A *composition* of the integer n is a sequence $(\pi_1,\cdots,\pi_k)$ such that each π_j is an integer larger than 0, and the π_j add up to n. The set **CO** of all compositions satisfies:

$$\mathbf{CO} \approx \mathbf{I}^{\bullet}$$

where **I** is the set of integers ≥ 1, and the weight of integer $k \in \mathbf{I}$ is $w(k)=k$. We

thus have:

$$CO(z) = \frac{1}{1-I(z)} \quad \text{where} \quad I(z) = z+z^2+z^3+\cdots$$

so that the number of compositions of an integer n is:

$$[z^n]\frac{1-z}{1-2z} = 2^{n-1} \quad (n\geq 1) .$$

7. Integer partitions. A ***partition*** of integer n is defined like a composition, except that one imposes the further restriction that the π_j form a non-decreasing sequence. Let $\mathbf{IP}$ be the class of all integer partitions. One can see that:

$$\mathbf{IP} \approx \{1\}^{\bullet}\{2\}^{\bullet}\{3\}^{\bullet}\cdots$$

where again the weight of integer k is equal to k itself. From there one obtains the generating function expression:

$$IP(z) = \prod_{k\geq 1}(1-z^k)^{-1} . \blacksquare$$

As was pointed out in the introduction, a large number of enumeration results follow from these combinatorial constructions. For instance the number of (set) partitions of a set of n elements where each block has size at most h is:

$$[\frac{z^n}{n!}]\exp(e_h(z)-1) ,$$

where e_h is the truncated exponential series:

$$e_h(z) = \sum_{j=0}^{h}\frac{z^j}{j!} .$$

Compositions or (integer) partitions into bounded summands can be dealt with in a similar fashion.

PART III
ASYMPTOTIC METHODS FROM COMPLEX ANALYSIS

The problem we examine here is an ***inversion problem***. Given some information about a generating function $f(z)$ - at best an ***explicit form***, at worst only a ***functional equation*** defining the function ***implicitly*** - how to recover some asymptotic information on the n-th Taylor coefficient f_n of $f(z)$.

One way consists in obtaining explicit forms for the coefficients f_n, if at all possible. For instance, assume we are interested in the probability that a random permutation of $[1..n]$ has no fixed point (*i.e.* no cycle of length 1). From the preceding chapter, this probability is:

$$\pi_n^{<1>} = [z^n] \frac{e^{-z}}{1-z}$$

from which follows the explicit form:

$$\pi_n^{<1>} = \sum_{k=0}^{n} \frac{(-1)^k}{k!} .$$

Observing that $\pi_n^{<1>}$ is a partial sum of the expansion of $\exp(-1)$, we find:

$$\pi_n^{<1>} = e^{-1} + O(\frac{1}{(n+1)!}) .$$

However, if we need the probability that a permutation has no cycles of length 1 or 2, we find

$$\pi_n^{<2>} = [z^n] \frac{e^{-z-z^2/2}}{1-z}$$

and expanding leads to a double sum, whose approximation, though feasible, requires some work. The problem gets worse if cycle lengths of the form $1, 2, \cdots, k$ are prohibited.

In general, the complexity of that method increases drastically as the size of the defining equation grows. With techniques we are going to examine in this chapter, one can reason as follows:

The function $f(z) = \exp(-z-z^2/2)/(1-z)$ has a unique ***singularity*** (a pole) at $z=1$. Around that pole, one has:

$$f(z) \sim \frac{e^{-3/2}}{1-z} ,$$

therefore:

$$[z^n] f(z) \sim [z^n] \frac{e^{-3/2}}{1-z}$$

and the quantity on the r.h.s is equal to $e^{-3/2}$, independently of n.

The basic inversion theorem to be used to justify that reasoning is *Cauchy's* ***residue theorem***, or equivalently *Cauchy's* ***integral formula*** for coefficients of analytic functions, namely:

$$[z^n] f(z) = \frac{1}{2i\pi} \int_{\Gamma} f(z)\, dz \qquad (1)$$

for Γ a simple closed contour around the origin.

The choice of the integration contour Γ in Formula (1) is guided by several principles to be detailed below. In many cases, that formula makes it possible to extract useful asymptotic information about $f_n = [z^n]f(z)$.

a. If $f(z)$ is *meromorphic* in the complex plane **C**, extend Γ to a circle of large radius, taking residues of the integrand of (1) into account.

b. If $f(z)$ has *non polar singularities* on its circle of convergence, take for Γ a contour that comes close to the singularity in order to extract information from the singular behaviour of the function. If the function is small around its singularity, take a contour that extends beyond the circle of convergence; if it is "moderately" large, take a contour that partly coincides with the circle of convergence; if it is "very" large, take a contour properly contained in the disk of convergence and use saddle point methods.

c. If $f(z)$ is *entire*, take Γ to be a circle that crosses the *saddle point(s)* of $f(z)$.

Notice that these methods do not always require $f(z)$ to be explicitly determined: it is often sufficient that some *local* properties of $f(z)$ be obtained from defining equations.

Finally, a number of combinatorial sums can be studied asymptotically by means of the *Mellin (integral) transform*. The method applies well to the asymptotic analysis of *harmonic sums* that are of the form:

$$F(x) = \sum_k \alpha_k g(\beta_k x) .$$

1. The exponential order formula for coefficients of analytic functions.

We start by recalling a few basic definitions:

Definition: *A function $f(z)$ of the complex variable z is said to be* analytic *at $z=a$ if it has a power series expansion, also called* Taylor expansion *convergent in a neighbourhood of a:*

$$f(z) = \sum_{n\geq 0} c_n (z-a)^n . \tag{2}$$

A function $f(z)$ of the complex variable z is said to be meromorphic *at $z=a$ if in a neighbourhood of a it has for $z\neq a$ a convergent expansion (called a* Laurent expansion*) of the form:*

$$f(z) = \sum_{n\geq -M} c_n (z-a)^n . \tag{3}$$

If $c_{-M}\neq 0$, then $f(z)$ has a pole *of order M at $z=a$.*

A function is analytic (meromorphic) in a domain iff it is analytic (meromorphic) at very point of the domain. A point at which an analytic function

ceases to be analytic is called a ***singularity*** of the function. We let Sing(f) denote the set of singularities of function f.

Definition: *If $f(z)$ has a pole of order $M \geq 1$ at $z=a$, the coefficient c_{-1} of its Laurent expansion at $z=a$ is called the* residue *of f at $z=a$ and is denoted by:*

$$\mathrm{Res}(f(z);z=a) .$$

With a slight extension of our previous notations, we could write:

$$\mathrm{Res}(f(z);z=a) = [\frac{1}{z-a}]f(z) .$$

Examples: A. $f(z)=exp(z+\frac{z^2}{2})$ has no singularity in the whole of the complex plane **C**; it is an *entire* function.
B. $f(z) = e^{-z-z^2/2}/(1-z)^2$ has a double pole at $z=1$, where local expansions reveal that

$$f(z) = \frac{e^{-3/2}}{(z-1)^2} - \frac{2e^{-3/2}}{(z-1)} + O(1)$$

so that $\mathrm{Res}(f(z);z=1) = -2e^{-3/2}$.
C. $f(z) = e^{-z/2-z^2/4}/\sqrt{1-z}$ is analytic in $|z| \leq 1$ except for a non polar singularity at $z=1$. ■

We can now state the celebrated Cauchy residue theorem:

Theorem 1: [Cauchy's residue theorem] *Let Γ be a simple closed curve oriented positively, and assume that f is meromorphic in a domain D containing Γ in its interior, and has no poles on Γ. Then:*

$$\frac{1}{2i\pi}\int_{\Gamma} f(z)dz = \sum_{s} \mathrm{Res}(f(z);z=s) \qquad (4)$$

where the sum is over the set of all poles s of $f(z)$ in the interior of Γ.

In particular the integral of an analytic function along a closed contour is equal to 0. An immediate consequence of Theorem 1 is:

Theorem 2: [Cauchy's integral coefficient formula] *Let Γ be a simple closed contour, oriented positively, with the origin in its interior, that is contained inside the domain of analyticity of $f(z)$. Then:*

$$f_n \equiv [z^n]f(z) = \frac{1}{2i\pi}\int_{\Gamma} f(z)\frac{dz}{z^{n+1}} . \qquad (5)$$

Proof: By Cauchy's residue theorem, the integral is equal to $\mathrm{Res}(f(z)/z^{n+1};z=0)$ which is exactly f_n. ■

An important property of analytic functions is that the radius of

convergence of the Taylor expansion of f at a denoted $R(f;a)$ is equal to

$$R(f;a) = \min\{\, |s-a| \mid s \in \mathrm{Sing}(f)\,\}.$$

In other words an analytic function always has a singularity on its circle of convergence.

We can now state a theorem relating the *exponential order* of coefficients of an analytic function to the *location of its singularities*.

Theorem 3: [The exponential coefficient bound] *If* $f(z) = \sum f_n z^n$ *is such that:*

$$R = \min_{s \in \mathrm{Sing}(f)} |s|$$

then for any $\varepsilon > 0$*:*

$$R^{-n}(1-\varepsilon)^n \lesssim_{i.o.} |f_n| <_{a.e.} R^{-n}(1+\varepsilon)^n \,. \tag{6}$$

The notation $a_n <_{i.o.} b_n$ *means that* a_n *is smaller than* b_n infinitely often *(for infinitely many values of* n*) while* $a_n <_{a.e.} b_n$ *means that* $a_n < b_n$ almost everywhere *(except for at most a finite number of values of* n*).*

Proof: If the lower bound was not satisfied, then $f(z)$ would be analytic in a larger domain. The upper bound follows from Cauchy's integral formula taking as integration contour a circle of radius $R(1-\eta)$ where $(1-\eta)^{-1} \equiv (1+\varepsilon)$. ∎

Applications: 1. For the exponential generating function of *surjections* † $f(z) = \dfrac{1}{2-e^z}$, we have:

$$\mathrm{Sing}(f) = \{\log 2 + 2ik\pi \mid k \in \mathbf{Z}\}$$

so that $R = \log 2$ and for any ε:

$$\left(\frac{1}{\log 2}\right)^n (1-\varepsilon)^n <_{i.o.} f_n <_{a.e.} \left(\frac{1}{\log 2}\right)^n (1+\varepsilon)^n \,.$$

2. With $f(z) = e^{-z/2 - z^2/4}/\sqrt{1-z}$, the singularity nearest to the origin is $z = 1$, so that:

$$(1-\varepsilon)^n <_{i.o.} f_n <_{a.e.} (1+\varepsilon)^n$$

3. Consider the functional equation $f(z) = z + f(z^2+z^3)$ defining $f(z)$ implicitly. Iterating the equation, we find:

$$f(z) = \sum_{k \geq 0} \sigma^{(k)}(z) \tag{7}$$

where $\sigma(z) = z^2+z^3$, and $\sigma^{(k)}(z)$ denotes the k-th iterate of $\sigma(z)$. The fixed points of σ are:

$$z_0 = 0\,;\; z_1 = \frac{-1-\sqrt{5}}{2}\,;\; z_2 = \frac{-1+\sqrt{5}}{2}\,.$$

One can observe that z_0 is an attractive fixed point of σ so that the sum (7)

† The coefficient of $z^n/n!$ in $f(z)$ is the number of surjections from $[1..n]$ onto an initial segment of **N**

converges fast in a neighbourhood of the origin. On the other hand the sum (7) becomes infinite when $z=z_2$. (There it becomes a sum of infinitely many identical terms that are non-zero). A slightly more refined analysis reveals that z_2 is the singularity of f nearest to the origin. Since $1/z_2$ is the well known *golden ratio* $\varphi = \frac{1+\sqrt{5}}{2}$, we have the bounds:

$$\varphi^n(1-\varepsilon)^n <_{i.o.} f_n <_{a.e.} (1+\varepsilon)^n \quad \blacksquare$$

As a final conclusion to this section, if R is the distance of the origin to the nearest singularity of f, we have:

$$f_n = \vartheta(n)R^{-n} \tag{8}$$

where ϑ grows *i.o.* faster than any decreasing exponential α^n, $\alpha<1$ and grows *a.e.* faster than any increasing exponential β^n, $\beta>1$. It is the purpose of the next sections to indicate methods by which the growth of the *subexponential factor* $\vartheta(n)$ can be precisely quantified. One has for instance for the above examples:

1. $f_n = O((\frac{1}{\log 2})^n)$;
2. $f_n = O(n^{-1/2})$;
3. $f_n = O(\frac{\varphi^n}{n})$;

and fuller asymptotic expansions can be obtained in all cases.

2. Rational fractions and meromorphic functions.

We have seen in the last section that the *location of singularities* of a function determines the *exponential growth* of its coefficients. In this and the next section, we refine on that observation showing that the *nature of the singularities* is related to the growth of the *subexponential factor* $\vartheta(n)$ that appears in the formula:

$$f_n \sim \vartheta(n)R^{-n} .$$

We start with the simplest class of functions, namely the *rational fractions*. These are of the form

$$f(z) = \frac{N(z)}{D(z)} \tag{9}$$

for two (relatively prime) polynomials $N(z)$ and $D(z)$. If $f(z)$ is to be analytic at 0, one should further assume that $D(0)\neq 0$. Since $N(z)\equiv D(z)f(z)$, the coefficients of a rational fraction (9) satisfy a *linear recurrence relation*

$$\sum_k f_{n-k} D_k = 0$$

with initial conditions determined by $N(z)$. Conversely any linear recurrence relation leads to a generating function that is a a rational function.

The asymptotics of coefficients of rational fractions is easy enough. Let $\alpha_1, \alpha_2, \ldots$ bet the (finite) set of zeros of $D(z)$. Then the *partial fraction decomposition* of $f(z)$ is:

$$f(z) = \sum_{j,k} \frac{c_{j,k}}{(z-\alpha_j)^k} \tag{10}$$

$$= \sum_{j,k} \frac{\gamma_{j,k}}{(1-z/\alpha_j)^k} .$$

The sum in (10) is finite and all coefficients $\gamma_{j,k}$ such that k is larger than the order of the root α_j of $D(z)$ are equal to zero.

From (10), taking coefficients, we get:

$$f_n = \sum_{j,k} \gamma_{j,k} \alpha_j^{-n} \binom{n+k-1}{k-1} . \tag{11}$$

and since the binomial coefficient is a polynomial of degree $(k-1)$ in n:

Theorem 4: *If $f(z)=\frac{N(z)}{D(z)}$ is a rational fraction that is analytic at the origin, then the n-th Taylor coefficient of f has the exact expression:*

$$f_n = \sum_j \alpha_j^{-n} \Pi_j(n) \tag{12}$$

where the α_j are the poles of f and each Π_j is a polynomial whose degree is equal to the multiplicity of the pole of f at α_j minus 1.

Notice that if the α_j are arranged in order of increasing modulus, then (12) has the character of an asymptotic expansion in which each term is *exponentially smaller* than the previous one. Notice also that non-real α's will correspond to *fluctuating terms* since, if $\alpha=\rho e^{i\varphi}$:

$$\alpha^{-n} = \rho^{-n}(\cos(n\varphi)-\sin(n\varphi)) .$$

A result very similar to Theorem 4 holds much more generally for *meromorphic* functions. One has:

Theorem 5: *Let $f(z)$ be meromorphic for $|z| \le R$ and analytic for $|z|=R$. Let $\alpha_1, \alpha_2, \ldots$,bet the (finite) set of poles of $f(z)$ with modulus less than R. Then there exist polynomials $\Pi_1, \Pi_2, \ldots$, such that*

$$f_n = \sum_j \alpha_j^{-n} \Pi_j(n) + O(R^{-n}) . \tag{13}$$

The degree of Π_j is equal to the order of the pole α_j minus 1.

Notice that the remainder term is exponentially smaller than any of the terms in the sum (13). We present two proofs of Theorem 5.

Proof 1: [Method of subtracted singularities]
If $f(z)$ has a pole of order δ_j at α_j, then for some function h_j analytic at α_j:

$$f(z) = \frac{h_j(z)}{(z-\alpha_j)^{\delta_j}}.$$

Expanding h_j around α_j up to terms of order δ_j, we find a polynomial Q_j, namely

$$Q_j = \sum_{r \le \delta_j} \frac{1}{r!} \frac{d}{dz^r} h_j(z) \Big|_{z=\alpha_j} (z-\alpha_j)^r$$

such that:

$$f(z) - \frac{Q_j(z)}{(z-\alpha_j)^{\delta_j}}$$

is analytic at $z=\alpha_j$. Thus the rational fraction obtained by collecting singular contributions from poles:

$$Q(z) = \sum_j \frac{Q_j(z)}{(z-\alpha_j)^{\delta_j}}$$

is such that $f(z)-Q(z)$ is analytic for $|z| \le R$. Writing

$$[z^n] f(z) = [z^n] Q(z) + [z^n](f(z)-Q(z))$$

and applying Theorem 4 to the first term, the Cauchy exponential bound to the second term concludes the proof of the theorem.

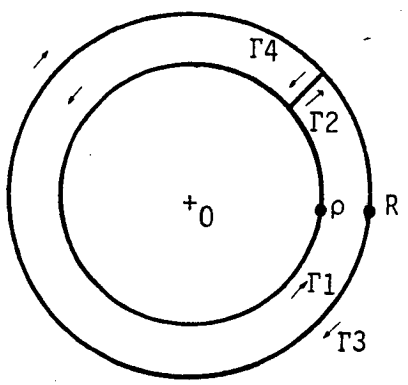


Figure 1: *The integration contour used to extract coefficients of meromorphic functions.*

Proof 2: [Contour integration method]

Let φ be a number $0 \leq \varphi < 2\pi$ be such that the half-ray $\mathrm{Argt}(z)=\varphi$ crosses no pole of $f(z)$ with modulus less than R. Let ρ be smaller than the radius of convergence of f at 0. Consider the contour (see Fig. 1):

$$\Gamma = \Gamma_1+\Gamma_2+\Gamma_3+\Gamma_4$$

where

$$\Gamma_1 = \{ z \mid |z| = \rho \}$$

$$\Gamma_2 = \{ z \mid \rho \leq |z| \leq R \; ; \; \mathrm{Argt}(z) = \varphi \}$$

$$\Gamma_3 = \{ z \mid |z| = R \}$$

$$\Gamma_4 = -\Gamma_2 \, .$$

Γ_1 is oriented positively (anticlockwise), Γ_3 negatively (clockwise); Γ_2 is traversed in the direction from Γ_1 to Γ_3, and Γ_4 is the same as Γ_2 except that the orientation is reversed.

The contour Γ encircles all the poles of $f(z)$ with modulus less than R in a clockwise direction. Therefore, by the residue theorem,

$$\frac{1}{2i\pi} \int_{\Gamma} f(z) \frac{dz}{z^{n+1}} = -\sum_j \mathrm{Res}(f(z) z^{-n-1}; z=\alpha_j) \qquad (14)$$

(Notice the minus sign due to the orientation of Γ). Now the integral decomposes into

$$\frac{1}{2i\pi}\int_{\Gamma} = \frac{1}{2i\pi}\int_{\Gamma_1} + \frac{1}{2i\pi}\int_{\Gamma_2} + \frac{1}{2i\pi}\int_{\Gamma_3} + \frac{1}{2i\pi}\int_{\Gamma_4} .$$

Contributions relative to Γ_2 and Γ_4 cancel each other. The contribution relative to Γ_4 is $O(R^{-n})$ by trivial majorisation. Finally $\frac{1}{2i\pi}\int_{\Gamma_1}$ is equal to f_n. Thus the proof is completed once we check that each of the residues in (14) is of the form $\alpha_j^{-n}\Pi_j(n)$. ∎

Examples: 1. Let **R** be the set of 0−1 strings without 2-*runs* (*i.e.* no 2 consecutive *ones* may appear in these strings). One has the description:

$$\mathbf{R} = (\varepsilon+1)(0(\varepsilon+1))^*$$

so that:

$$r(z) = (1+z)\frac{1}{1-z(1+z)}$$

$$= \frac{1+z}{1-z-z^2} \, .$$

From the partial fraction decomposition of $r(z)$, we get:

$$r_n = \frac{1+\sqrt{5}}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{1-\sqrt{5}}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n .$$

2. Let $f(z)=1/(2-e^z)$ whose set of singularities has been already determined. The residue at $\alpha=\log 2$ of f is $-\frac{1}{2}$. By periodicity of the exponential, this is also equal to the residue of f at any other pole. The sum of residues in

Theorem 5 appears to be convergent, so that one can write the ***exact*** formula:

$$[z^n]\frac{1}{2-e^z} = \frac{1}{2}\left(\frac{1}{\log 2}\right)^{n+1} \sum_{k\in\mathbb{Z}} \chi_k^{-n-1},$$

where:

$$\chi_k = 1+\frac{2ik\pi}{\log 2}.$$

3. The probability that a random permutation of $[1..n]$ has no cycle of length $\leq k$ is:

$$\pi_n^{<k>} = [z^n]\frac{e^{-z-\frac{z^2}{2}-\frac{z^3}{3}\cdots-\frac{z^k}{k}}}{1-z},$$

so that for any positive real R:

$$\pi_n^{<k>} = e^{-H_k} + o(1)$$

with $H_k = 1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{k}$. ∎

Notice finally that these expansions are usually quite good owing to the fast decrease of terms. For instance

$$[z^{10}]\frac{1}{2-e^z} = \frac{34082521}{1209600} = 28.1766873346$$

while the ***first*** term of the asymptotic expansion yields:

$$\frac{1}{2}\left(\frac{1}{\log 2}\right)^{11} = 28.1766873361\ !$$

3. Non polar singularities.

Assume that a function $f(z)$ has a unique singularity of smallest modulus α. Results from the last section entail that, when f is meromorphic, we can ***translate*** an asymptotic relation for the function:

$$f(z) \sim g(z) \quad z\to\alpha, \tag{15}$$

where α is the singularity nearest to the origin, into the corresponding relation for coefficients:

$$[z^n]f(z) \sim [z^n]g(z) \quad n\to\infty. \tag{16}$$

We propose here to describe general conditions under which the transition from (15) to (16) can be effected, relaxing the conditions that functions be meromorphic or singularities be of a polar type.

Developments in this section assume that asymptotic information is available for the function in some area of the complex plane around its singularity. They make it possible to translate $O(.)$ estimates for functions into $O(.)$ estimates for coefficients, whence the name of ***transfer*** or ***translation***

lemmas given to them.

One of the main uses of transfer lemmas is as follows. Assume we have an asymptotic expansion for f around α in the form:

$$f(z) = \sigma_1(z) + \sigma_2(z) + \cdots + \sigma_k(z) + O(g(z)), \tag{17}$$

for some elementary functions σ_1 ... belonging to an asymptotic scale. Then, if proper conditions are satisfied, (17) translates into

$$f_n = \sigma_{1,n} + \sigma_{2,n} + \cdots + \sigma_{k,n} + O(g_n). \tag{18}$$

Application of this method therefore calls for two types of results:

1. Building up a ***catalogue*** of coefficients of standard singular functions appearing in asymptotic expansions, in exact or asymptotic form, using real or complex analysis.
2. Establishing conditions under which transfer lemmas hold true.

Notice also that these methods can be trivially extended when a function has a finite number of singularities on its circle of convergence. Just add up the contributions to the coefficients coming from each singularity.

Function	Coeff.	
$\log(1-z)^{-1}$	$\frac{1}{n}$	
$(1-z)^r$	$\frac{n^{-r-1}}{\Gamma(-r)}$	$r \neq 0, 1, 2 \cdots$
$(1-z)^r \log^s(1-z)^{-1}$	$c_{r,s} n^{-r-1} \log^s$	$r \neq 0, 1, 2, \cdots$
$(1-z)^r \log^s(1-z)^{-1}$	$r! n^{-r-1}$	$r = 0, 1, 2, \cdots$

Table 2: *A simplified catalogue of the asymptotic form of coefficients of some standard singular functions.*

Table 2 provides a simplified catalogue of the asymptotic form of coefficients of some standard functions. Such a catalogue can be built from direct expressions available for coefficients or from contour integration techniques. For instance:

$$[z^n] -\log(1-z) = \frac{1}{n}$$

$$[z^n] (1-z)^{-s} = \frac{s(s+1)(s+2)\cdots(s+n-1)}{n!} = \binom{s+n-1}{s-1}.$$

As to transfer lemmas, they are summarised in Table 3. (Notice, in passing, the analogy between Tables 2 and 3). We shall prove here:

Theorem 6: *(i) Assume that $g(z)$ is analytic in the domain*

$$D = \{ z \mid |z| \leq 1, z \neq 1 \}$$

and that as z tends to 1 inside D, one has:

Function	*Coeff.*	*Cond.*
$O(\log(1-z))$	$O(\frac{\log n}{n})$	
$O((1-z)^{-s})$	$O(n^{s-1})$	$s\geq 0$
$O((1-z)^{r})$	$O(n^{-r-1})$	$r\geq 0$

Table 3: *A simplified set of transfer results (see Theorem 6 for validity conditions).*

$$g(z) = O(|1-z|^{-s})$$

with $s>1$. *Then:*

$$[z^n]g(z) = O(n^{s-1}) .$$

(ii) Assume that $g(z)$ *is analytic in the indented disk:*

$$D = \{ z \mid |z|\leq 1+\delta , \vartheta<|\mathrm{Arg}(z-1)|<2\pi \}$$

where δ,ϑ *are such that* $\delta>0$, $0<\vartheta<\frac{\pi}{2}$. *Assume that, as* z *tends to* 1 *inside* D*:*

$$g(z) = O(|1-z|^{r})$$

with $r>0$. *Then:*

$$[z^n]g(z) = O(n^{-r-1}) .$$

Proof: (Sketch)

(i)_ Use the Cauchy formula with a contour Γ that consists of the circle $|z|=1$ except for a small notch at distance $1/n$ of $z=1$: $\Gamma = \Gamma_0+\Gamma_1$ where

$$\Gamma_0 = \{ z \mid |z|\leq 1 , |z-1|=\frac{1}{n} \}$$

$$\Gamma_2 = \{ z \mid |z|=1 , |z-1|\geq\frac{1}{n} \} .$$

Next evaluate each integral using trivial bounds. One has:

$$\int_{\Gamma_0} f(z)\frac{dz}{z^{n+1}} = O(\frac{1}{n}(\frac{1}{n})^{-s})$$

$$\int_{\Gamma_1} f(z)\frac{dz}{z^{n+1}} = O(\int_{\vartheta_0}^{\pi}|1-e^{i\vartheta}|\,d\vartheta .$$

There ϑ_0 is the argument of the intersection of Γ_0 and Γ_1 in the upper half plane.

(ii)_ Use likewise the Cauchy integral formula with a contour

$$\Gamma = \Gamma_{0,\omega} + \Gamma_{1,\omega} + \Gamma_2$$

where

$$\Gamma_{0,\omega} = \{ z \mid |z-1|=\omega , |\mathrm{Arg}(z-\rho)|>\vartheta_1 \}$$

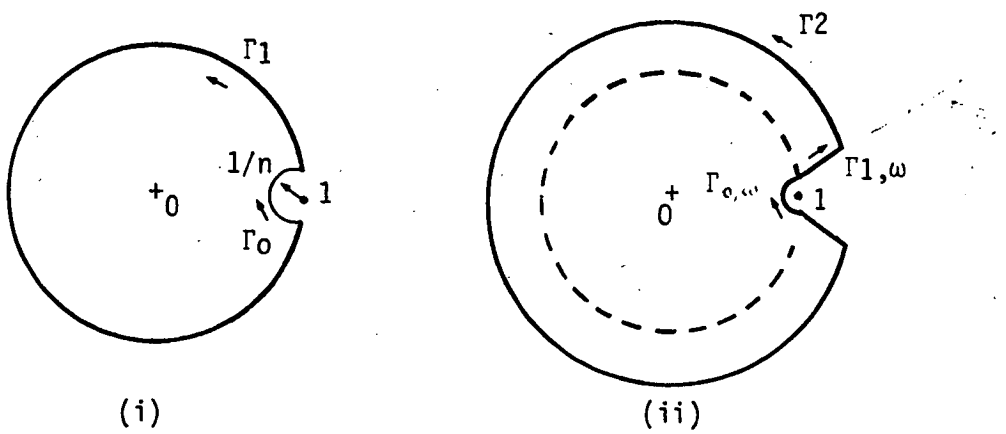


Figure 4: *The contours used in order to establish Theorem 6.*

$$\Gamma_{1,\omega} = \{ z \mid |z-1| \geq \omega \, , \, |z| < r_1 \, , \, \mathrm{Arg}(z-\rho) = \vartheta 1 \}$$
$$\Gamma_2 = \{ z \mid |z| = r_1 \, , \, |\mathrm{Arg}(z-\rho)| \geq \vartheta 1 \}$$

for some r_1: $1 < r_1 < 1+\delta$ ($\delta > 0$) and ϑ_1: $\vartheta < \vartheta_1 < \frac{\pi}{2}$, letting ω shrink to 0. ■

A full discussion of the proof is given in papers by Odlyzko and Flajolet-Odlyzko.

Notes: 1. Many more results are available using these techniques. The underlying idea is to take a contour of integration that comes close to the singularity. It the function is ***small*** (*i.e.* tends to 0) as the argument approaches the singularity then one tries to extend the contour of integration outside the disk of convergence in a manner similar to what was done for polar singularities. If the function is ***large*** then the contour can stay within the disk of convergence of the function.

2. The fact that the singularity of the function was assumed to be at $z=1$ is of course not restrictive (otherwise, normalise the function). Also the case where the function has a ***finite*** number of singularities on its circle of convergence can be dealt with using composite contours. The outcome is that contributions from each of the singularities cumulate.

3. The classical *Darboux-Polya* method is in the same spirit. It however assumes ***smoothness conditions*** while here our conditions concern ***orders of growth***. The present approach lends itself nicely to generalisations; also in some cases, only order-of-magnitude informations are available and it appears to be well suited to combinatorial enumeration problems.

4. *Tauberian theorems* assume much weaker conditions on the function g: basically all that is required is some information on the function as $z \to 1^-$ along the *real axis* However, in the context of combinatorial enumerations, they do not seem to provide information for the same variety of singular behaviours while they require some so-called Tauberian side conditions that may be hard to establish. ■

Examples: 1. When counting certain combinatorial configurations ("clouds"), one encounters the generating function:

$$f(z) = \frac{e^{-z/1-z^2/4}}{\sqrt{1-z}}$$
$$= \frac{e^{-3/4}}{\sqrt{1-z}} + O(|1-z|^{1/2}).$$

Thus, by the transfer lemma (Theorem 6.i):

$$[z^n] f(z) = [z^n] \frac{e^{-3/4}}{\sqrt{1-z}} + O(n^{-3/2})$$
$$= \frac{e^{-3/4}}{\sqrt{\pi n}} + O(n^{-3/2}).$$

2. Let $f(z)$ be the solution analytic at the origin of equation:

$$f = z(1+f+f^2),$$

that is:

$$f(z) = \frac{1-z-\sqrt{1-2z-3z^2}}{2z}.$$

Function f is the o.g.f. of unary binary trees, in which each node has degree 0, 1 or 2. The dominant singularity of $f(z)$ is at $z=1/3$ (the other one is at $z=-1$), where locally f admits an expansion of the form:

$$f(z) = A\sqrt{1-3z} + O(|1-3z|^{3/2})$$

from which one finds:

$$[z^n] f(z) = \frac{3^{n+1}}{2\sqrt{\pi n^3}} + O(3^{n-5/2}). \quad ■$$

4. Saddle point bounds.

We only give here a brief introduction to the subject of *saddle point methods* which allow derivation of asymptotic expansions for integrals of analytic functions depending on a (large) parameter. In the context of extracting coefficients of analytic functions, one way of conceiving these methods is as a refinement of trivial bounds on the Cauchy integral formula.

Assume throughout this section that $f(z)$ is an analytic function that is *entire* and has *positive coefficients*. By Cauchy's integral formula, one has:

$$f_n \equiv [z^n] f(z) = \frac{1}{2i\pi} \int_\Gamma f(z) \frac{dz}{z^{n+1}}. \qquad (19)$$

Take as contour of integration Γ a circle of radius R. Since $f(z)$ has positive coefficients, we have for any z such that $|z|=R$: $|f(z)| \leq f(R)$, and thus, from trivial majorisations of (19):

$$f_n \leq \frac{1}{2\pi} \frac{f(R)}{R^{n+1}} 2\pi R \qquad (20)$$
$$\leq \frac{f(R)}{R^n} .$$

The bound (20) is valid for *any* positive R. Notice that $f(R)/R^n$, which is infinite at $R=0$ and $R=\infty$ is unimodal over $\mathbf{R}^+$. Thus there is a real number ρ: $0<\rho<\infty$ that minimises $f(R)/R^{n+1}$. That number is a root of equation:

$$\frac{d}{dR} \frac{f(R)}{R^n} = 0$$

so that is satisfies:

$$\rho f'(\rho) - n f(\rho) = 0 . \qquad (21)$$

In other words:

Theorem 7: *Let f be entire and have positive coefficients. Define the function $w(u)$ by:*

$$w(u) = \frac{u f'(u)}{f(u)} .$$

Then the n-th Taylor coefficient of $f(z)$ satisfies the bound:

$$f_n \leq \frac{f(w^{<-1>}(n))}{(w^{<-1>}(n))^n} , \qquad (22)$$

where $w^{<-1>}(.)$ denotes the functional inverse of $w(.)$.

Example: Take $f(z) = \exp(z)$. Then $[z^n] f(z) \equiv 1/n!$. We have trivially: $w(u) = u$ so that $w^{<-1>}(n) \equiv n$, whence by Theorem 7 the (expected) bound:

$$\frac{1}{n!} \leq \frac{e^n}{n^n} ,$$

a weak form of Stirling's formula. ■

Note: Since the bound (2) is valid for any R, the function $w(.)$ need not be inversed exactly and may be solved only asymptotically or *approximately*. Of course, the better the approximation, the better the bound. ■

Example: Let f_n be the number of involutions in the set of permutations of $[1..n]$. Involutions are characterised by the fact that they have only cycles of length 1 and 2. Thus their e.g.f is:

$$\hat{f}(z) \equiv \sum_n f_n \frac{z^n}{n!} = \exp(z + \frac{z^2}{2}) .$$

In that case, $w(u) = u + u^2$, so that an approximation to $w^{<-1>}(n)$ is $\sqrt{n}/2$, whence:

$$f_n \le 2^n n! n^{-n-1} e^{-n/8+\sqrt{n}/2} . \quad \blacksquare$$

The estimates we have just seen can be refined. In many cases of interest, only a ***small fraction of the contour*** contributes significantly to the integral. There, local approximations can be performed and an asymptotic estimate of the integral (instead of just an upper bound) can be obtained.

The saddle point method applies to integrals of the form:

$$I = \frac{1}{2i\pi}\int_{\Gamma} e^{h(z)}dz , \tag{23}$$

where $h(z)=h_n(z)$ depends on a large parameter n. The case of Cauchy's formula (19) corresponds to the particular form:

$$h(z) \equiv h_n(z) = \log f(z) - (n+1)\log z .$$

The method proceeds as follows:

1. Determine $R = R_n$ such that:

$$\frac{d}{dz}h(z)\Big|_{z=R} = 0 . \tag{24}$$

Quantity R is called a ***saddle point*** of the integrand due to the local topography of the surface defined by $|h(z)|$ and $|e^{h(z)}|$. Notice that with the notations of Theorem 7, one has $R_n = w^{<-1>}(n+1)$ which is expected to be close to the quantity $w^{<-1>}(n)$ appearing in Theorem 7. The idea is to evaluate integral (23) using as contour Γ a circle of radius R satisfying (24).

2. Select an adequate angle $\vartheta = \vartheta_n$ (usually ϑ will be small), satisfying the two (conflicting) requirements:

$$\int_{\Gamma/\Gamma[\vartheta]} e^{h(z)}dz \ll \int_{\Gamma} e^{h(z)}dz . \tag{C1}$$

$$e^{h(z)} \approx e^{h(R)+\frac{1}{2}h''(R)(z-R)^2} \qquad z\in\Gamma[\vartheta] . \tag{C2}$$

There $\Gamma[\vartheta]$ denotes the part of the circle $|z|=R$ consisting of points z such that $|\mathrm{Arg}(z)|\le\vartheta$. Condition (C1) requires ϑ to be large enough so that the dominant part of the integral comes from $\Gamma[\vartheta]$ while condition (C2) requires ϑ to be small enough that local expansions be valid.

3. If (C1) and (C2) are satisfied, then one has:

$$I \approx \frac{1}{2i\pi}\int_{\Gamma[\vartheta]} e^{h(R)+\frac{1}{2}h''(R)(z-R)^2}dz .$$

The last step is now to ***complete*** the integral; setting $z=R+it$:

$$I \approx \int_{-\infty}^{+\infty} e^{-\frac{t^2}{2}h''(R)}dt , \tag{C3}$$

a Gaussian integral that can be evaluated, leading to:

$$I \approx \frac{e^{h(R)}}{\sqrt{2\pi h''(R)}} .$$

What we have seen above is general enough to apply to a wide class of integrals depending on a (large) parameter. Restricting ourselves to the special form of integral (19), we can state:

Theorem 8: *Assuming approximations (C1), (C2), (C3) to be valid, one has:*

$$[z^n]f(z) \sim \frac{e^{h(R_n)}}{\sqrt{2\pi h''(R_n)}}, \tag{24}$$

where:

$$h(z) = \log f(z) - (n+1)\log z$$

$$R_n = w^{<-1>}(n+1)$$

and $w^{<-1>}(.)$ *is the functional inverse of:*

$$w(u) = \frac{uf'(u)}{f(u)}.$$

Applications: 1. Stirling's formula:

$$[z^n]e^z \sim \frac{e^n n^{-n}}{\sqrt{2\pi n}}.$$

2. The number of involutions:

$$[\frac{z^n}{n!}]e^{z+z^2/2} \sim \frac{1}{\sqrt{2}} n^{n/2} e^{-n/2+\sqrt{n}-1/4}.$$

3. The number of set partitions:

$$[z^n]\exp(e^z-1) \sim e^{-1}(\frac{n}{\log n})^n. \quad \blacksquare$$

5. Mellin transform techniques.

The *Mellin transform* associates to a real function $f(x)$ defined over $[0;+\infty]$ a complex function $f^*(s)$ written $\mathbf{M}[f(x);s]$ or $\mathbf{M}[f(.)]$, and given by:

$$f^*(s) = \int_0^\infty f(x)x^{s-1}dx. \tag{25}$$

If f is continuous and satisfies:

$$f(x) = O(x^\alpha) \quad x \to 0$$

$$f(x) = O(x^\beta) \quad x \to \infty$$

then, it is easy to see that the transform (25) is defined in the strip - called the *fundamental strip* - $-\alpha<\mathrm{Re}(s)<-\beta$. Let $\delta(x)$ be the function whose value is 1 for $0\le x\le 1$ and 0 for $1<x$; it is easy to see that:

$$\mathbf{M}[\delta(x)x^\alpha] = \frac{1}{s+\alpha}$$

and thus the Mellin transform associates to a (particular) function that is $O(x^{\alpha})$ at 0 a transformed function with a pole at $s=-\alpha$. From this observation, proceeding by linearity, it is easy to see that more generally, the transform of a function with an ***asymptotic expansion*** around 0 of the form:

$$f(z) \sim \sum_j c_j x^{\alpha_j}$$

is a function meromorphic in a left half plane that has a pole of residue c_j at $s=-\alpha_j$. Smaller terms in that expansion correspond to poles that are farther to the left. A similar reasoning applies to asymptotic expansions towards ∞ (poles farther to the right correspond to smaller contributions). In other words:

A. *The asymptotic expansions of a function at 0 (or ∞) are reflected by the poles of its Mellin transform in a left half-plane (a right half plane respectively).*

The converse of that property is also (mildly conditionally) true. To prove this, one starts with the ***inversion formula***, that corresponds to the classical Fourier inversion;

$$f(x) = \frac{1}{2i\pi} \int_{c-i\infty}^{c+i\infty} f^*(s) x^{-s}\, ds \tag{26}$$

where c is in the fundamental strip of f. (Notice the analogy with (25)). If $f^*(s)$ is meromorphic, one can evaluate the integral in (26) by residues: take as contour of integration the vertical line $\mathrm{Re}(s)=c$ completed by a large contour in the left half-plane. Under (often satisfied) suitable conditions, one can apply Cauchy's residue theorem to that integral and get:

$$f(x) \sim \sum_{\alpha} \mathrm{Res}[f^*(s) x^{-s}\ ;\ s=\alpha] \tag{27}$$

where the sum is extended to all poles α of $f^*(s)$ to the right of the vertical line $\mathrm{Re}(s)=c$. Notice that if f^* has only simple poles, then (27) can be rewritten as:

$$f(x) \sim \sum_{\alpha} \mathrm{Res}[f^*(s); s=\alpha]\, x^{-\alpha} \tag{28}$$

where the asymptotic nature of expansion (28) is obvious. If $f^*(s)$ has multiple poles, then generalised expansions with powers of $\log x$ appear. In summary:

B. *The poles of a Mellin transform in a left half-plane (right half plane) translate (under certain smallness conditions of $f^*(s)$ towards $i\infty$) into terms of an asymptotic expansion of $f(x)$ at 0 (resp. $+\infty$).*

Thus the correspondence between asymptotic properties of f and singularities of f^* fares both ways. This justifies the importance of that transform for asymptotic analysis.

The usefulness of the Mellin transform is also due to a very elementary functional property, namely:

$$\mathbf{M}[f(ax); s] = a^{-s} f^*(s)\ ,\quad a \geq 0\ ,$$

which using linearity (assuming summations and integrations may be

interchanged) extends to:

$$\sum_k \lambda_k f(a_k x) \quad \rightarrow \quad (\sum_k \lambda_k a_k^{-s}) f^*(s) . \tag{29}$$

Sums on the left hand side of (29) are called *harmonic sums*. Equation (29) shows that:

C. *Harmonic sums are transformed by the Mellin transform into the product of a generalised Dirichlet series and the transform of the basis function.*

We shall only illustrate some of these points by means of a few elementary examples.

Examples. 1. Let $f(x)=e^x$. The transform of f is the classical Gamma function:

$$\Gamma(s) = \int_0^\infty \exp(x) x^{s-1} dx ,$$

with $<0;=\infty>$ as fundamental strip. To the term $(-1)^k \frac{x^k}{k}$ in the expansion of f around 0 there corresponds (Point A above) a simple pole of $\Gamma(s)$ at $s=-k$ with:

$$\Gamma(s) \sim \frac{(-1)^k}{k!} \frac{1}{s+k} \qquad s \rightarrow -k .$$

In other words, the expansion:

$$e^{-x} \sim \sum_{k \geq 0} \frac{(-1)^k}{k!} x^k$$

translates into the *meromorphic* expansion:

$$\Gamma(s) \approx \sum_{k \geq 0} \frac{(-1)^k}{k!} \frac{1}{s+k} ,$$

and in that case both expansions are actually convergent.

2. The following sum appears in relation to the analysis of the expected height of a planar tree with n nodes:

$$S(x) = \sum_{k \geq 1} d(k) e^{-k^2 x^2} \tag{30}$$

where $d(k)$ is the number of *divisors* of k. Sum (30) is typically a harmonic sum whose transform is ($\zeta(s)$ is the Riemann zeta function);

$$S^*(s) = \frac{1}{2} \zeta^2(s) \Gamma(\frac{s}{2}) , \tag{31}$$

where the fundamental strip of (31) is $<1;+\infty>$. Function S^* has a double pole at $s=1$ and a simple pole at $s=0$. Hence the meromorphic expansion:

$$S^*(s) \approx \frac{1}{2} \Gamma(\frac{1}{2}) \frac{1}{(s-1)^2} + \frac{1}{4} \Gamma'(\frac{1}{2}) + \frac{1}{4s} ,$$

whence:

$$S(x) \sim -\sqrt{\pi} \frac{\log x}{x} + \sqrt{\pi} (\frac{3}{4} - \frac{\log 2}{2}) \frac{1}{x} + \frac{1}{4} + O(x^M) \qquad x \rightarrow 0 ,$$

for any positive M. ■

The Mellin transform has a host of applications to: (1) situations where number-theoretic functions appear (like above the divisor function); (2) non-standard asymptotic expansions corresponding to periodicities. Examples are: height of trees, carry propagation, digital trees or tries ... .

PART IV
APPLICATIONS

It is our purpose here to offer a brief guide to some of the existing literature on the subject of analysis of algorithms and data structures, putting enumeration and asymptotic methods in perspective. Since we cannot afford the space necessary to discuss the vast existing literature, we shall restrict ourselves to examining a few ***data structures***, closely related to trees, and corresponding algorithmic processes.

1. Trees and tree manipulation algorithms.

This sections discusses uniform statistics on trees of various compositions. It corresponds to what was called in Part I, the empirical model. The trees we condider are ***term trees*** in some algebraic structure.

Consider first the family **B** of (planar) ***binary trees***; it is a data structure recursively defined by:

$$\mathbf{B} = \blacksquare + \langle o,\mathbf{B},\mathbf{B}\rangle \tag{1}$$

where "■" denotes an empty tree (nullary node), and "o" denotes an internal (binary) node. (We have used an obvious linearised notation for trees). Define the size of a binary tree to be the number of internal nodes it comprises. Equation (1) translates into the fixed point equation for the corresponding generating function $B(z)$:

$$B(z) = 1 + z\,B^2(z)\,, \tag{2}$$

a quadratic equation that has the solution:

$$B(z) = \frac{1-\sqrt{1-4z}}{2z}\,. \tag{3}$$

Whence the explicit result:

$$B_n = \frac{1}{n+1}\binom{2n}{n}\,, \tag{4}$$

and from Stirling's formula:

$$B_n \sim \frac{4^n}{\sqrt{\pi n^3}}\,. \tag{5}$$

The transition from (1) to (2) is general enough. Let Ω be a subset of the non-negative integers. Consider the family $\mathbf{T} \equiv \mathbf{T}[\Omega]$ of trees such that (out)degrees of nodes are restricted to be in the set Ω (binary trees correspond to $\Omega = \{0,2\}$). Such a family is called, after Meir and Moon [MM78], the ***simple family of trees*** associated to degree constraints Ω. Define ω_k to be equal to 1 if $k \in \Omega$ and 0 otherwise. One can write for **T** the symbolic equation:

$$\mathbf{T} = \sum_k \omega_k \langle o,\mathbf{T},\mathbf{T},\cdots,\mathbf{T}\rangle\,. \tag{6}$$

where the number of occurrences of **T** in the general term of the sum is equal to k. With the size of a tree now defined as the total number of nodes that

tree comprises, equation (6) translates into:

$$T(z) = z\,\omega(T(z)) \tag{7}$$

where $\omega(u) = \sum_k \omega_k x^k$.

The Taylor coefficients of the solution $T(z)$ of (17) can be obtained exactly using the *Lagrange inversion Theorem* [He77] that relates the coefficients of the multiplicative powers of a function ($\omega(.)$) to those of its functional inverse (related to T). Hence:

Theorem 1: *The number of trees of size (total number of nodes) n in the family defined by degree constraints Ω is:*

$$T_n \equiv \frac{1}{n}[u^{n-1}]\,\omega(u)^n \,. \tag{8}$$

Notice that in Formula (8), the ω_k need not be 0-1 parameters. Allowing for general integral ω_k will make it possible to count term trees, that is trees whose nodes are labelled with operators; in that case ω_k represents the number of operators of degree (arity) k.

If $\omega(u)$ is simple enough, then Theorem 1 will provide useful counting results. We mention here:

- The number of general trees ($\omega_k \equiv 1$ for all k, *i.e* $\omega(u)=(1-u)^{-1}$) of size n is:

$$\frac{1}{n}\binom{2n-2}{n-1} .$$

- The number of t-ary trees ($t \geq 2$), *i.e.* $\omega(u) = 1+u^t$ with a total of $tn+1$ nodes (and thus with n t-ary internal nodes) is:

$$\frac{1}{tn+1}\binom{tn+1}{n} .$$

In case $\omega(u)$ has a more complex form, one has to resort to asymptotic analysis, and indeed [MM78] have shown that Formula (5) obtained here by elementary methods nicely generalises.

Function $T(z)$ in Equation (7) is the solution (in y) of:

$$F(z,y) = 0 \quad , \quad F(z,y) \equiv y - z\,\omega(y) \,. \tag{9}$$

Thus (9) defines y implicitly as a function of z. From the *implicit function theorem*, we know that a solution y with value y_0 at a point z_0 ($F(z_0,y_0)=0$) is analytically continuable provided:

$$\frac{\partial}{\partial y}F(z,y)\,\Big|_{(z_0,y_0)} \neq 0 \,.$$

From there can be seen that the singularity (-ies) of y closest to the origin is (are) the quantity (-ies) of smallest modulus ρ such that (ρ,τ) are a set of solutions of the system:

$$F(\rho,\tau)=0 \quad ; \quad \frac{\partial}{\partial y}F(z,y)\,\Big|_{(\rho,\tau)} = 0 \,.$$

Hence, here:

$$\rho = \frac{\tau}{\omega(\tau)} \tag{10}$$

where τ is one of the roots of equation:

$$\omega(\tau) - \tau\omega'(\tau) = 0 \qquad (11)$$

Assume for simplicity that there is a unique τ of smallest modulus satisfying (11). Then, around (ρ,τ) the dependency between y and z is locally of the form:

$$(\tau - y)^2 - A\,(z - \rho) = 0\,, \qquad (12)$$

as can be checked using the expansion of F. From (12), one can establish formally that y has the form:

$$y(z) = h_1(z) + h_2(z)\sqrt{1 - \frac{z}{\rho}} \qquad (13)$$

where h_1 and h_2 are analytic at $z=\rho$. That form lends itself nicely to a singularity analysis (of a "square-root" type) and one gets the very general result of [MM78] which we state in the little restrictive case where $T(z)$ has a unique singularity on its circle of convergence (the same assumption is made in the rest of this section):

Theorem 2: [MM78] *If $t(z)$ has a unique singularity on its circle of convergence, the number of trees in $\mathbf{T}[\Omega]$ with size n satisfies asymptotically:*

$$t_n \equiv t_n[\Omega] \sim C\,\rho^{-n} n^{-3/2}$$

where the constants C and ρ are given explicitly by $\rho = \tau/\varphi(\tau)$ and $C = (\varphi(\tau)/(2\pi\varphi''(\tau)))^{1/2}$ with τ the smallest positive root of the equation $\varphi(\tau) - \tau\varphi'(\tau) = 0$.

The main methods for estimating tree parameters are as follows:

1. The symbolic operator approach is a convenient tool for writing symbolic equations in the style of (1), (7). One may have though to extend it to equations over *mulisets* (elements are taken with multiplicities corresponding to values of the parameter to be analysed [Fl81, SF83, FS82]).
2. Most generating functions have expressions in terms of the implicitly defined function $t(z)$. Thus, the Lagrange inversion theorem is an important tool that often leads to exact counting results otherwise difficult to attain.
3. Singularity analysis of intervening generating functions is also of constant use in this context. Since function $t(z)$ has algebraic singularities, the methods of Chapter III often apply here. Other important techniques are saddle point methods and Mellin transform techniques in those cases where, in summations, there appear coefficients of an arithmetical nature.

Some examples follow. We only sketch the main steps of derivations.

The simplest of all tree algorithms is certainly *recursive tree traversal*: to traverse a tree in preorder, visit its root, then recursively traverse all its root subtrees in left-to-right order. The time complexity of that procedure is clearly linear in the size of the tree, while its storage complexity is equal to the maximum size of the recursion stack, a quantity that coincides with the *height* of the tree.

The first result on the expected height of planar trees has been obtained by De Bruijn *et al.*.

Theorem 3: [DBKR71] *The expected height of a general planar tree (all node degrees allowed) with n nodes satisfies:*

$$\bar{H}_n = \sqrt{\pi n} + O(1) .$$

Proof: Let **G** be the family of general trees:

$$\mathbf{G} = o + \langle o,\mathbf{G}\rangle + \langle o,\mathbf{G},\mathbf{G}\rangle + \langle o,\mathbf{G},\mathbf{G},\mathbf{G}\rangle \cdots \tag{14}$$

An equation similar to (14) describes the family $\mathbf{G}^{[h]}$ of trees with height at most h:

$$\mathbf{G}^{[h+1]} = o + \langle o,\mathbf{G}^{[h]}\rangle + \langle o,\mathbf{G}^{[h]},\mathbf{G}^{[h]}\rangle + \langle o,\mathbf{G}^{[h]},\mathbf{G}^{[h]},\mathbf{G}^{[h]}\rangle \cdots \tag{15}$$

whence the equations:

$$g(z) = \frac{z}{1-g(z)} \quad ; \quad g^{[h+1]}(z) = \frac{z}{1-g^{[h]}(z)} \tag{16}$$

from which follows that:

$$g(z) = \frac{1-\sqrt{1-4z}}{2} \quad ; \quad g^{[h]}(z) = z\frac{F_{h+1}(z)}{F_{h+2}(z)} \tag{17}$$

where the F's satisfy the linear recurrence relation:

$$F_{h+2}(z) = F_{h+1}(z) - z\,F_h(z) .$$

The F's can be expressed as functions of $g(z)$ itself, and using Lagrange inversion, one gets:

$$g_{n+1} - g^{[h]}_{n+1} = \sum_j \binom{2n}{n+1-j(h+2)} - 2\binom{2n}{n-j(h+2)} + \binom{2n}{n-1-j(h+2)} \tag{18}$$

and:

$$\bar{H}_{n+1} = \sum_k d(k)\left[\binom{2n}{n+1-k} - 2\binom{2n}{n-k} + \binom{2n}{n-1-k}\right] . \tag{19}$$

The asymptotic evaluation of (19) calls for evaluations of sums of the form:

$$S_n = \sum_k d(k)\frac{\binom{2n}{n-k}}{\binom{2n}{n}} . \tag{20}$$

Using the Gaussian approximation of binomial coefficients, (20) is approximated by $T(1/\sqrt{n})$ where:

$$T(x) = \sum_k d(k)e^{-k^2x^2} . \tag{21}$$

The problem is thus to evaluate asymptotically $T(x)$ given by (21) when x tends to 0. The Mellin transform of $T(x)$ is readily determined to be

$$T^*(s) = \frac{1}{2}\zeta^2(s)\Gamma\left(\frac{s}{2}\right) . \tag{22}$$

It has a double pole at $s=1$, a simple pole at 0 whence the asymptotic expansion

$$T(x) = \frac{1}{2x}(\log x + C_1) + C_2 + O(x^m) , \tag{23}$$

as $x \to 0$, for any $m>0$. A combination of expansions of the form (23) leads to the statement of the theorem. ∎

That result has been generalised by Flajolet and Odlyzko who proved:

Theorem 4: [FO83] *The expected height of a tree of size n in a simple family of trees satisfies:*

$$\overline{H}_n \sim A\sqrt{n}$$

where the explicitly computable constant A is $A=(2\pi/(\varphi(\tau)\varphi''(\tau)))^{1/2}\varphi'(\tau)$.

Returning to the notations of equations (6), (7), we see that the generating function of trees of height at most h, $T^{[h]}$, is defined by the recurrence:

$$T^{[h+1]}(z) = z\,\omega(T^{[h]}(z)) \tag{24}$$

with $T^{[0]}(z)=z$, and the generating function of height of trees is:

$$H(z) = \sum_h [T(z)-T^{[h]}(z)]\,. \tag{25}$$

The scheme (25) is nothing but an iterative approximation scheme to the fixed point equation (7) determining T. A singularity analysis of (24) leads to the result. This necessitates determining the behaviour of the iterative scheme (24) near $z=\rho$, which is a *singular iteration problem*, from which one can prove that:

$$H(z) \sim \frac{K}{1-\frac{z}{\rho}} \log \frac{1}{1-\frac{z}{\rho}}\,,$$

and the result of Theorem 3 follows directly.

Methods similar to those employed in the proof of Theorem 4 had been introduced in an earlier analysis of Odlyzko [Od83], where he counted the number of balanced 2–3 trees of size n.

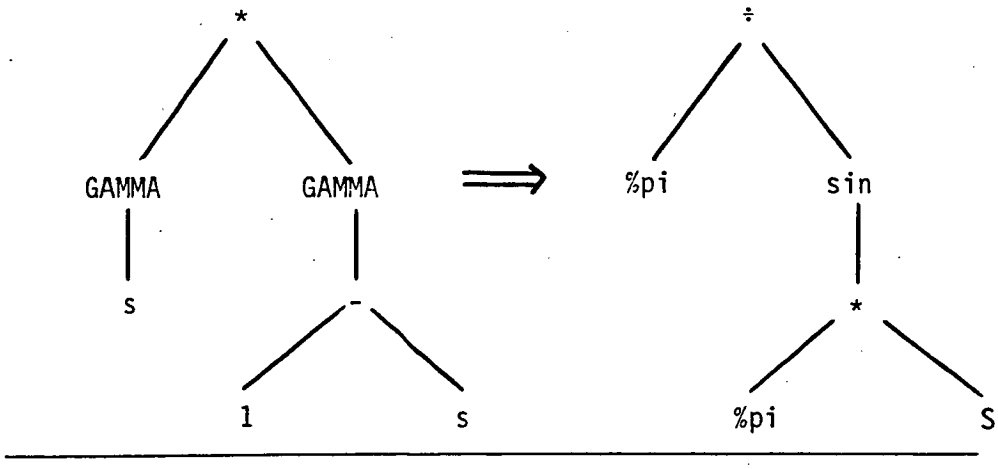
Theorem 5: [Od83] *The number of balanced 2–3 trees with n external nodes satisfies:*

$$E_n \sim \frac{\varphi^n}{n} W(\log n)$$

where φ is the golden ratio $\frac{1+\sqrt{5}}{2}$ and $W(.)$ is a continuous and periodic function.

Odlyzko's result actually includes the counting of a variety of balanced trees. Such trees occur in the management of "dictionaries" and they allow insertions, deletions and queries to be performed in guaranteed $O(\log n)$ time. The occrrence of the golden ratio in Theorem 5 is to be expected after the discussion in Chapter 3 of the equation $f(z) = z + f(z^2+z^3)$ that is satisfied by the o.g.f of the E_n.

The next algorithm to be examined is ***pattern matching*** on trees. The problem is to detect occurrences of a given pattern tree in a larger text tree. For instance, in symbolic manipulation systems, one may look for cases of application of a rewrite rule of the form:



and recognizing cases where the pattern on the l.h.s. appears calls for a pattern-matching algorithm.

Contrary to what happens in the case of strings where efficient worst case linear time algorithms are known, it is conjectured here that no linear time algorithm may exist for tree-matching. The *sequential tree matching algorithm* corresponds to a simple backtracking search. It operates as follows:

1. For each node of the text tree, examine the subtree rooted at that node to see if it matches the pattern, using the comparison procedure (2).
2. To compare a subtree against a pattern, traverse simultaneously the pattern tree and the text's subtree in preorder and abort that traversal as soon as a mismatch is detected.

The sequential matching algorithm has clearly a quadratic worst case complexity of the form $O(n^2)$. However, one can prove in contrast that the expected case is linear, namely:

Theorem 6: [SF83] *The sequential tree matching algorithm, when applied to a fixed pattern P and all trees of size n, has expected cost given by:*

$$\overline{\tau match}_n \sim w(P)\, n$$

where $w(P)$ is a function of the structure of pattern P that is uniformly bounded by an absolute constant: $w(P) \leq W$.

The proof of the theorem depends on the following lemma [SF83]:

Lemma: *For a simple family of tres and a fixed pattern P with i internal nodes and e external nodes, the asymptotic probabilty of occurrence of P at a random node of a large random tree of size n satisfies:*

$$occ_n^{<P>} \sim \tau^{e-1}\rho^i \ .$$

The algebraic part of the proof is a direct application of the symbolic operator method applied to multisets of trees. Generating functions for the number of occurrences of trees have simple expressions in terms of the function $T(z)$

and a singularity analysis yields the statement of the Lemma.

The same type of analysis can be applied to a large variety of of tree algorithms. In [FS82], the authors set up a general framework within which a number of algorithms on trees can be (semi-) automatically analysed. As an illustration, we cite:

Theorem 7: *The symbolic differentiation algorithm has, for any set Ω of operators and any set Δ of differentiation rules with at least one "expanding" rule, the average case complexity;*

$$\tau\overline{diff}_n = C(\Omega,\Delta)n^{3/2} + O(n) .$$

A less standard singular behaviour occurs in the problem known as the *common subexpression problem* or *tree compaction* where a tree is compacted into a *dag* by avoiding duplication of identical substructures. The singularity in that case is of the form [FSS85]

$$\frac{1}{\sqrt{(1-z)\log(1-z)^{-1}}}$$

and one finds:

Theorem 8: [FSS85] *The expected size of the (maximally compacted) dag representation of a random tree of size n in a simple family of trees satisfies:*

$$\overline{K}_n = \gamma\frac{n}{\sqrt{\log n}} + \frac{O(n}{\log n}) .$$

Thus the gain to be expected when compacting trees into dags should be expected to approach 100% as the trees get large, although convergence may be quite slow.

Finally, *register allocation* in compiling is the subject of [FRV79], [Ke79]. The optimal register allocation strategy for expressions involving binary operators has been determined by Ershov as early as 1958. We have:

Theorem 9: [FRV79],[Ke79] *The expected number of registers to evaluate a binary tree of size n using Ershov's algorithm satisfies:*

$$\overline{R}_n = \log_4 n + P(\log_4 n) + o(1),$$

where $P(u)$ is a periodic function of its argument that has period 1 and small amplitude.

In the analysis, there appears the combinatorial sum:

$$V_n = \sum_{k\geq 1} v_2(k)\binom{2n}{n-k}$$

in which $v_2(k)$ is the exponent of 2 in the prime number decomposition of k. Exponential approximations lead to analogous sums with the binomial coefficient replaced by an exponential (*vide* Eqns (20)-(21)). The Mellin transform of the approximation is:

$$\frac{1}{2}\frac{\zeta(s)}{2^s-1}\Gamma(\frac{s}{2})$$

and its line of regularly spaced poles $s=\frac{2ik\pi}{\log 2}$ corresponds to periodic

fluctuations in the form of a Fourier series.

Notice on this example the first occurrence of periodicity phenomena of a non-trivial nature.

2. Digital searching and sorting algorithms.

Let S be a finite set of distinct binary strings (or *keys*), each of some fixed length $l \leq +\infty$. To the set S is canonically associated a special type of tree, called a *trie* and denoted by *trie*(S), that is defined recursively as follows:

- if card(S)=0 then *trie*(S) is the empty tree;
- if card(S)=1 then *trie*(S) consists of a unique node (leaf) labelled with the unique element of S;
- If card$(S) \geq 2$ let S_0 and S_1 be the subsets of S formed by elements beginning with a 0 and a 1 respectively; let S_j^* $(j=0,1)$ denote the set of elements of S_j stripped of their initial bit; then *trie*(S) is defined as:

$$trie(S) = < o, trie(S_0^*), trie(S_1^*) > .$$

If leftmost edges in a tree are labelled with zeros and rightmost edges are labelled with ones, then the set of all labellings from the root of the tree to the leaves is a minimal prefix set of S. For this reason tries are also known in coding theory as *prefix trees*.

Tries as a data structure have been discovered by Fredkin (see [Kn73]) and they support *insertions*, *deletions* and *queries*: to retrieve a key from a trie, for instance, follow a path from the root of the tree that is guided by the successive bits of the key to be found, branching left on 0's and right on 1's. By construction, if l is finite, the worst case cost of these operations is $O(l)$ which represents a logarithmic cost if $l \approx \log_2 n$. If $l=\infty$ (the results will basically apply for finite l as soon as $l \gg \log_2 n$) then, under the assumption that bits of keys are uniform and independent, the expected cost of any of the above operations is $\log_2 n + O(1)$ as we propose to show.

The probability that a trie formed with n random keys has a leftmost trie of size k and a rightmost trie of size $n-k$ is the *Bernoulli probability*:

$$p_{n,k} = \frac{1}{2^n}\binom{n}{k}. \tag{26}$$

Let $v[t], w[t], \cdots$ denote parameters of tries, like path length, number of nodes... . Let $v_n, w_n, \cdots$ be the expectations of $v[t], w[t], \cdots$ when the tries t are built from a set a n random keys, and let finally $v(z), w(z), \cdots$ denote the corresponding exponential generating functions. From the form (26) of splitting probabilities, we find the following relations between structural definitions of parameters and *exponential generating functions* of expected values:

$$v[t] = w[t] + x[t] \quad => \quad v(z) = w(z) + x(z) \tag{27}$$

$$v[t] = w[t_0] \times x[t_1] \quad => \quad v(z) = w(\frac{z}{2}) \times x(\frac{z}{2}) \tag{28}$$

where t_0 and t_1 denote the left and right subtrees of t. There (27) is nothing but the additive property of expectations and generating functions while (28) comes from the equalities

$$v_n = \sum_{k=0}^{n} p_{n,k}\, w_k\, x_{n-k} \tag{29}$$

or equivalently:

$$\frac{v_n}{n!} = \frac{1}{2^n} \sum_{k=0}^{n} \frac{w_k}{k!} \cdot \frac{x_{n-k}}{n-k!} .$$

Let us first analyse the storage occupation of tries. The number of internal nodes of a trie t, denoted by $s[t]$, satisfies the recursive definition:

$$s[t] = s[t_0].U[t_1] + U[t_0]s[t_1] + 1 , \tag{30}$$

where U is the constant *unit valuation* $U[t] \equiv 1$, and (30) holds as soon as the number of keys in t exceeds 1. Thus using the general scheme (27)-(28) in (30), observing that $U(z)=e^z$ and taking care of initial conditions, we find for the corresponding e.g.f $s(z)$ the equation:

$$s(z) = 2e^{z/2}s(\frac{z}{2}) + e^z - 1 - z \tag{31}$$

since the e.g.f. of $U[.]$ is $U(z)=e^z$. Equation (31) can be solved by iteration, and we get the explicit form:

$$s(z) = \sum_{k \geq 0} 2^k [e^z - (1+\frac{z}{2^k})e^{(1-\frac{1}{2^k})z}] \tag{32}$$

so that taking coefficients in (31):

$$s_n = \sum_{k \geq 0} 2^k [1-(1-\frac{1}{2^k})^n - \frac{n}{2^k}(1-\frac{1}{2^k})^{n-1}] . \tag{33}$$

The next step in the derivation is to use Mellin transforms. To that purpose, the simplest way consists in introducing the function:

$$S(x) = \sum_{k \geq 0} 2^k [1-e^{-x/2^k}(1+\frac{x}{2^k})] \tag{34}$$

which derives from (34) when we use the exponential approximation:

$$(1-a)^n \approx e^{-an}$$

substituting x for n. One can justify that approximation here and show that $s_n = S(n)+O(n^{1/2})$ (see [Kn73]).

The interest of the form (34) is that it is a harmonic sum. Its Mellin transform is defined for $-2<\mathrm{Re}(s)<-1$ and from the preceding chapter we find that it is

$$S^*(s) = -\frac{(s+1)\Gamma(s)}{1-2^{s+1}} . \tag{35}$$

Poles to the right of the fundamental strip of S^* determine the asymptotic behaviour of $S(x)$ as x gets large. There is a simple pole $s=0$ that is due to $\Gamma(s)$ and poles at points $\chi_k = -1+\frac{2ik\pi}{\log 2}$ for $k \in \mathbf{Z}$ due to the denominator of (35). Computing residues, we find the following theorem of Knuth (using suggestions by De Bruijn, see [Kn73, pp. 131ff]):

Theorem 10: [Kn73] *The expected storage occupation (measured by the number of internal nodes) of a trie built on n uniform and independent keys is:*

$$s_n = \sum_{k\geq 0} 2^k \left[1-(1-\frac{1}{2^k})^n - \frac{n}{2^k}(1-\frac{1}{2^k})^{n-1}\right],$$

a quantity that is asymptotic to:

$$\frac{n}{\log 2}(1+Q(\log_2 n)) + O(\sqrt{n})$$

where $Q(u)$ is a periodic function with period 1, mean value 0 and Fourier expansion given by:

$$Q(u) = \sum_{k\in\mathbb{Z}/\{0\}} q_k e^{-2ik\pi u} \quad ; \quad q_k = \frac{1}{\log 2}(1+\chi_k)\Gamma(\chi_k)$$

with $\chi_k = -1+\frac{2ik\pi}{\log 2}$.

The expected cost of a positive search in a trie is p_n/n where p_n is the expected path length when n keys are present in the trie. Path length $p[t]$ is defined inductively by:

$$p[t] = p[t_0]U[t_1] + U[t_0]p[t_1] + |t| \tag{36}$$

which, as before leads to

$$p(z) = 2e^{z/2}p(\frac{z}{2}) + z(e^z-1)$$

whence the exact expression:

$$p_n = n\sum_{k\geq 0}\left[1-(1-\frac{1}{2^k})^{n-1}\right].$$

One has $p_n \sim nP(n)$ where:

$$P(x) = \sum_{k\geq 0}(1-e^{-x/2^k})$$

whose Mellin transform is given by:

$$P(x) = -\frac{\Gamma(s)}{1-2^s}$$

Thus a residue calculation shows that:

Theorem 11: [Kn73] *Under the uniform model, the expected cost of a positive search in a trie of size n is:*

$$\sum_{k\geq 0}\left[1-(1-\frac{1}{2^k})^{n-1}\right]$$

a quantity that is asymptotic to:

$$\log_2 n + \frac{\gamma}{\log 2} + \frac{1}{2} + R(\log_2 n) + O(\frac{1}{\sqrt{n}})$$

where $R(u)$ is a periodic function with period 1, mean value 0 and Fourier expansion given by:

$$R(u) = \sum_{k\in\mathbb{Z}/\{0\}} r_k e^{-2ik\pi u} \quad ; \quad r_k = \frac{1}{\log 2}\Gamma(\chi_k)$$

with $\chi_k = \frac{2ik\pi}{\log 2}$.

An important use of tries is as an access method for large files stored on disk. A *b-trie* with leaf capacity equal to b ($b \geq 1$) is obtained by modifying the initial definition of tries in such a way that the recursive splitting is stopped as soon as a subset of size b or less is encountered. Leaves can thus contain up to b elements and can be stored in pages on disk. *Dynamic Hashing* is obtained in that way when the trie is built on hashed values of records instead of records themselves (thus ensuring uniformity of pseudo-keys on which the trie is built). The previous methods easily generalise, and one finds [Kn73],[La78],[FNPS79] for Dynamic Hashing and the closely related *Extendible Hashing* scheme:

Theorem 12: [Kn73],[La78],[FNPS79] *Under the uniform model, the number of pages necessary to store the file using a Dynamic or Extensible Hashing scheme with page capacity b is:*

$$\frac{n}{b \log 2}(1+Q_1(\log_2 n))+O(\sqrt{n}),$$

where Q_1 is a periodic function with mean value 0.

Thus under both schemes pages tend to be about 70% full ($\log 2 = 0.69...$).

Extendible hashing relies on a further paging of the internal nodes of the trie. The corresponding analysis have been given by Flajolet [Fl83] and Regnier (under a Poisson model) [Re83]. The analysis is closely related to the analysis of height in tries. Letting $\pi_{n,h}$ denote the probability that a trie with n keys has height $\leq h$, one finds with $e_b(z)$ denoting the truncated exponential:

$$\pi_{n,h} = [\frac{z^n}{n!}] e_b(\frac{z}{2^h})^{2^h} . \qquad (37)$$

From there, limiting distributions can be determined using saddle point methods. In this way, one obtains:

Theorem 13: [Fl83],[Re83] *Under the uniform model, the expected size of the paged directory in the Extendible Hashing scheme is asymptotic to:*

$$n^{1+1/b} Q_2(\log n)$$

where Q_2 is a periodic function with mean value close to $4/b$.

Many more results follow using these techniques. The underlying splitting process with the Bernoulli splitting probabilities of (26) appears as a model of some *polynomial factorisation* algorithms, of *communication protocols* and more classically of *radix exchange sort*. We can cite here [Kn73, pp. 131ff]:

Theorem 14: [Kn73] *Radix-exchange sort of n keys when applied to infinitely long strings uses an average of $n \log_2 n + O(n)$ comparisons.*

A systematic discussion of algebraic methods involved in all these analyses is given in [FRS84a]. The corresponding asymptotic methods are discussed in [FRS84b]. A detailed analysis of Dynamic and Extendible Hashing is given in Regnier's thesis [Re83].

3. Comparison based searching and sorting

Binary search trees are also amongst the oldest known data structures. Let S be a sequence of distinct real numbers (or of any totally ordered set): $S=(s_1,s_2,\cdots,s_n)$. The ***binary search tree*** built on S is denoted by $bst(S)$ and is defined recursively as follows:

- Make the first element s_1 of S the root of the tree.
- Separate the remaining elements $(s_2,s_3,\cdots,s_n)$ into two subsequences $S_<$ and $S_>$, where $S_<$ ($S_>$) is the subsequence consisting of elements smaller (larger) than s_1. Then:

$$bst(s) = <s_1, bst(S_<), bst(S_>)> . \tag{38}$$

Observe that once a binary search tree has been built, the sequence is almost sorted since a preorder traversal, that takes only linear time, will list the elements in increasing sorted order.

Binary search trees support insertions, deletions and queries [Kn73] as we shall now see in expected $O(\log n)$ time under the uniform-independence model (or equivalently under the permutation model, where S is taken to be a random permutation of $[1..n]$).

The basic principle is that a tree of size n is formed of two similar subtrees of size K and $n-1-K$ where K is a random variable between 0 and $n-1$ with probability distribution:

$$\Pr(K=k) = \frac{1}{n} \tag{39}$$

independently of n. Equation (39) reflects the fact that the first element of a random permutation can take any of the possible values with equal probability $(1/n)$. As in the preceding section it is easy to set up schemes that associate to parameters of trees generating functions of expected values.

Let $v[t],w[t],\cdots$ be functions of trees; let $v_n,w_n,\cdots$ be their corresponding average values and let $v(z),w(z),\cdots$ be the corresponding ***ordinary generating functions***. With t_0 and t_1 denoting the left and right subtrees of tree t, one has (compare with (27), (28)):

$$v[t] = w[t]+x[t] \;\Rightarrow\; v(z) = w(z)+x(z) \tag{40}$$

$$v[t] = w[t_0]\times x[t_1] \;\Rightarrow\; v(z) = \int_0^z w(t)x(t)\,dt . \tag{41}$$

Thus again any additive-multiplicative valuation over binary search trees can be analysed, and in general one will have a set of ***integral equations*** that reduce to a ***differential system*** for associated generating functions.

As a first example, consider the problem of determining the expected path length of binary search trees. Path length is here defined inductively by:

$$p[t] = p[t_0]+p[t_1]+|t|-1 \tag{42}$$

From (40), (41), we find:

$$p(z) = 2\int_0^z p(t)\,\frac{dt}{1-t} + \frac{z}{(1-z)^2} \tag{43}$$

which differentiates into:

$$p'(z) - 2\frac{p(z)}{1-z} - \frac{1+z}{(1-z)^3} = 0 . \tag{44}$$

Equation (44) can be solved by the *variation-of-constant* method, and we find:

$$p(z) = 2\frac{\log(1-z)^{-1}-z}{(1-z)^2}$$
$$= 2H'(z)-\frac{2+z}{(1-z)^2}$$

where $H(z)\equiv\sum_n H_n z^n$ is the generating function of the harmonic numbers. Hence, expanding and performing simple asymptotics:

Theorem 15: *The expected number of comparisons to sort a sequence of n elements building a binary search tree is under the uniform-independent permutation model*

$$p_n = 2(n+1)H_{n+1} - 3n - 2$$

and asymptotically;

$$p_n = 2n\log n + (2\gamma-3)n + O(\log n)\,.$$

As an immediate corollary to Theorem 15, we get that the expected cost of a positive search in a b.s.t of size n is $2\log_2 n+O(1)$.

Height of binary search trees leads to interesting equations over generating functions. Let h_n denote the expected height of a binary search tree with n nodes. Then from (39), one finds:

$$h(z) = \sum_{k\geq 0}[y(z)-y_k(z)] \tag{45}$$

where $y_0(z)=1$ and

$$y_{k+1}(z) = \int_0^z y_k^2(t)\,dt \tag{46}$$

and $y(z)\equiv y_\infty(z)=(1-z)^{-1}$.

Thus the y_k form a sequence of *Picard approximants* to y_∞. Although it is natural to conjecture that:

$$h(z) \sim \frac{c}{1-z}\log\frac{1}{1-z} \tag{47}$$

for some constant c, the singular expansion (47) appears to be amazingly difficult to establish. Devroye [De85], using the theory of certain types of branching processes, has determined directly the asymptotic form of h_n:

Theorem 16: [De85] *The expected height of a binary search tree with n nodes satisfies:*

$$h_n \sim c\log n$$

where $c=4.311070\cdots$ *is the root of* $(2e/c)^c=e$ *that is* >2.

Returning to the scheme (40), (41), we see that it will apply to any additive-multiplicative function of a splitting process whose probabilities satisfy (39). There are at least three instances where the splitting probabilities have this specific form:

A. *Quicksort*: it is a way of sorting that resembles closely the recursive definition (38) of binary search trees. Essentially quicksort is characterised by an *in place* partitioning of S into $S_<$ and $S_>$. (Also these two sets are replaced by their mirror images).

B. *Heap-ordered trees* or non-balanced heaps. they are trees canonically associated to sequences of distincts elements. Let S be such a sequence, then it can be decomposed into:

$$<S_{left}, \min(S), S_{right}> \qquad (48)$$

with S_{left} (S_{right}) being the factor of S formed with elements to the left (right) of $\min(S)$. Using decomposition (48) recursively, a tree is canonically associated to a sequence; it is characterised by the fact that labels increase along any branch starting at the root, and so constitutes a heap-ordered tree. Since in a random permutation, the minimum value occurs at any place with equal probability, (39) is satisfied, so that again the scheme (40)-(41) can be used.

C. *Multidimensional search trees* or $k-d$-trees. They serve to represent sets of multidimensional records consisting of several fields.: a search tree is formed by using successive fields cyclically as discriminators as one proceeds along a branch from the root.

We shall only cite here a few results along those lines:

Theorem 17: *The expected number of comparisons to sort n elements using Quicksort is:*

$$\bar{C}_n = 2(n+1)(H_{n+1}-\frac{4}{3}) \sim 2n\log n + 2(\gamma-\frac{4}{3}) + O(\log n) .$$

The reader is referred to [Kn73] and Sedgewick's papers [Se77a], [Se77b], [Se80] for a complete discussion of the complexity of Quicksort.

Theorem 18: *The expected number of comparisons required to perform extraction of the minimum in a heap-ordered tree of size n is:*

$$\bar{C}_n = O(\log n) .$$

Heap-ordered trees serve to implement mergeable priority queues. An efficient representation is in the form of *pagodas* [FVV78].

Theorem 19: [FP85] *The expected number of elementary field comparisons required to perform a partial match query in a $k-d$-tree of size n when records have dimension k and s fields are specified in the query satisfies asymptotically:*

$$\bar{C}_n^{[s/k]} \sim K n^{1-s/k+\vartheta(s/k)}$$

where $\vartheta(u)$ is the root in $[0;1]$ of equation:

$$(\vartheta+3-u)^u\,(\vartheta+2-u)^{1-u} - 2 = 0 .$$

The proof of Theorem 19 proceeds by first setting a system of integral equations for generating functions of costs using (40)-(41). That system reduces to a differential system of order $2k-s$. It cannot be solved explicitly in terms

of standard transcendental functions. However, using the classical theory of regular singular points of differential systems, a singularity analysis can be performed and Theorem 19 follows.

A result akin to Theorem 19 has recently been established for *quad-trees* [FGPR85]. See also Puech's work [Pu84] for related applications.

We should finally mention that decomposition (48) which corresponds to the symbolic equation:

$$\mathbf{P} \approx \varepsilon + \{\min\} \times (\mathbf{P} \star \mathbf{P})$$

for the set **P** of all permutations is an important starting point for obtaining many statistics over permutations (runs, left-to-right minima ...).

4. Conclusions.

We have tried to demonstrate on a few cases the role of generating functions as a crucial tool in the analysis of algorithms and data structures. The general pattern behind these analyses can be described as follows:

> Each class of simple data structure carries with it a natural class of generating functions with a particular algebraic structure and a set of analytic properties that can be used both for exact and asymptotic analysis.

Table 1 illustrates the algebraic translation mechanisms for multiplicative valuations of trees in each of the three cases considered previously: planar binary trees with the uniform statistics, digital tries and binary search trees.

The set of resolution techniques, as we have seen, are for each case:

1. Lagrange inversion and singularity analysis of functions with algebraic singularities.
2. Difference equations, iteration and Mellin transform techniques.
3. Differential equations: exact solution methods (operators, variation-of-constant) and the theory of regular singular points.

Trees	Splitting of n	Splitting Pb.	$v[t] = w[t_0].x[t_1]$
1. Planar Bin.	$<k, n-1-k>$	$\frac{B_k B_{n-k}}{B_n}$	$v(z) = zw(z)x(z)$
2. Tries	$<k, n-k>$	$\frac{1}{2^n}\binom{n}{k}$	$v(z) = w(z/2)x(z/2)$
3. Bin. Search	$<k, n-1-k>$	$\frac{1}{n}$	$v(z) = \int_0^z w(t)x(t)\,dt$

Table 1: *For each class of trees, description of the splitting sizes and probabilities; translation over generating functions of a multiplicative valuation on subtrees: (1) for o.g.f. of cumulated values; (2) for e.g.f. of expected values; (3) for o.g.f. of expected values.*

Amongst the many areas in the analysis of algorithms that are natural applications of these methods and that we have not had time to discuss, we would like to mention:

- The cycle structure of permutations and the problem of *in situ* permutation [Kn71], [Se83].
- Inversion tables for permutations and sorting algorithms: bubble sort, insertion sort [Kn73] and shellsort [Ya80].
- 2-sorted permutations, lattice path and merging algorithms [Kn73], [Se78].
- Distributions, occupancy statistics and hashing algorithms [Kn73], [GM85], [KW66], [Go81].
- String statistics [G081].
- Random graphs and set-merging ("Union-Find") algorithms [KS78].

Problems in the area of the exact analysis of algorithms may be of several types:

1. ***Finding proper decompositions*** of combinatorial problems in a way that lends itself to treatment by generating functions. If that approach succeeds, it usually has a high yield since, as we have tried to demonstrate, a large number of analyses will be amenable to a uniform treatment.

2. ***Finding approximate models*** that fall into category (1) if the combinatorial structure of the original problem is too intricate to lead to an exact analytic model.

3. ***Finding exact or asymptotic solutions for functional equations*** over generating functions, for models arising from (1) or (2).

To the category of (1) or -most probably- (2) there belongs the analysis of AVL trees, 2-3 trees and other ***balanced structures*** under the permutation model. See [JK77] for an analysis of a data structure that does not have a randomness preservation property and [Se85] for an analysis of heapsort.

A simple example of (3) is provided by the problem of the distribution of the number of comparisons in Quicksort. The bivariate generating function satisfies:

$$\frac{\partial}{\partial q} C(z,q) = qC(qz,q)^2$$

and the problem there is to determine the asymptotic behaviour of the coefficients of polynomials $[z^n]\,C(z,q)$. Related limiting distribution results have been obtained by Louchard [Lo84], Jacquet and Regnier [JR85]. However, despite the practical importance of Quicksort (there are several hundred thousand implementations running since Quicksort is part of the standard sort available on the Unix system) the form of the limiting distribution is yet unknown.

PART V
BIBLIOGRAPHY

Instead of giving here a complete bibliography, we shall restrict ourselves to indicating general references for the subject covered in Parts I-III together with brief historical and bibliographical comments and citing the set of papers whose results are mentioned in Part IV.

1. General References.

The subject of analysing algorithms is as old as algorithms, and thus predates the advent of computers. For instance, in his discussion of the analytical engine, Babbage evaluates the complexity of his (mechanical) integer multiplication method in terms of the number of "turns of the handle" (a measure certainly very relevant to his application). After computers became used for non-numerical data processing, it became obvious that some algorithms performed in a greatly varying manner depending on the specific configuration of the input data, a fact not so frequent with numerical algorithms. Average case analysis naturally emerged as a simple way of obtaining global information on the effectiveness of an algorithm, when it is used repeatedly. It is the merit of Knuth, in volume 1 of *The Art of Computer Programming* (first published in 1968) to have shown that a large number of classical algorithms could be exactly analysed, even at the very detailed level of assembly language programs. Knuth also demonstrated the importance of combinatorial enumeration techniques and asymptotic analysis in that context. For the subjects covered here, the basic references are thus:

[Kn68] D. E. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, Addison Wesley, Reading Mass., 1968.

[Kn73] D. E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*, Addison Wesley, Reading Mass., 1973.

For a presentation of many algorithms of interest in computer science, one may refer to:

[Se83] R. Sedgewick. *Algorithms*, Addison-Wesley (1983).

[Go84] G. Gonnet. *Handbook of Algorithms and Data Structures*, Addison-Wesley (1984).

The goals and methods of average case analysis of algorithms are discussed in Knuth's invited lecture at the 1971 IFIP Congress. An interesting recent survey is given by Sedgewick in:

[Se83] R. Sedgewick. "Mathematical Analysis of Combinatorial Algorithms", in *Probability Theory and Computer Science*, Louchard and Latouche Editors (1983).

The following booklet, corresponding to lecture notes from the course on analysis of algorithms at Stanford University, discusses in greater detail some of the points studied here (most notably saddle point methods):

[GK81] D. Greene and D. E. Knuth. *Mathematics for the Analysis of Algorithms* Birkhaeuser Verlag (1981).

Two other books on that subject are:

R. Kemp. *Fundamentals of the Average-Case Analysis of Particular Algorithms*, Wiley-teubner Series in Computer Science, J. Wiley, New-York (1984)

and for an elementary introduction:

P. Purdom, C. Brown. *The Analysis of Algorithms* in print (1985).

Concerning the combinatorial enumeration problems, the 19th century technique was almost invariably the set-up of recurrences. In a book (*Combinatory Analysis*) published in 1915, Major Percy MacMahon was the first one to systematically depart from the recurrence approach. MacMahon developped a very personal algebraic view of the field of combinatorial analysis. That approach was revived in the sixties through works by Rota, Foata and Schutzenberger. The symbolic operator approach is exposed systematically in the reference book of Jackson and Goulden:

[GJ83] I. Goulden and D. Jackson. *Combinatorial Enumerations* J. Wiley, New-York (1983).

The reading of that book may be complemented by the encyclopedic (and generating function oriented) book of Comtet:

[Co74] L. Comtet. *Advanced Combinatorics* D. Reidel, Dordrecht (1974).

A short survey of the domain of combinatorial enumerations appears in:

[St78] R. Stanley. *Generating Functions*, *M.A.A. Monographs*, (1978).

The field of asymptotic analysis is much closer to classical (pure and applied) mathematics, so that many classical references exist that we do not have space to cite. Two very useful problem solving oriented books are:

[DB60] N. G. De Bruijn. *Asymptotic Methods in Analysis* reprinted by Dover (1984).

[BO78] C. Bender and S. Orszag. *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill (1978).

and the necessary background from complex analysis can be found in:

[He77] P. Henrici. *Applied Computational and Complex Analysis*, J. Wiley, New-York, 2 Vol. (1974,1977).

A concise survey of asymptotic counting techniques is given by:

[Be74] E. Bender "Asymptotic Methods in Enumerations", *SIAM Review* 1974.

and the book of Sachkov provides a complete exposition of probabilistic and asymptotic methods in combinatorial analysis:

[Sa78] X. Sachkov. *Verojatnostnie Metody v Kombinatornom Analize*, Nauka Moscow (1978).

Finally, for further applications of the symbolic operator method to the analysis of algorithms, one may refer to the following works:

[Fl81] P. Flajolet. *Analyse d'algorithmes de manipulation d'arbres et de fichiers*, Cahiers du B.U.R.O **34-35**, Paris (1981), 209p.

[Gr83] D. Greene. "Labelled Formal Languages and Their Uses" (Thesis), Stanford University Rep. STAN-CS-83-982 (1983), 148p.

[St84] J-M. Steyaert. *Complexite' et Structure des Algorithmes*, Thesis, University of Paris VII (1984), 215p.

Additional references are to be found in the list that follows.

2. Specialised References.

[DBKR72] N. G. De Bruijn, D. E. Knuth and S. O. Rice: The Average Height of Planted Plane Trees, In: *Graph Theory and Computing*, R. C. Read Ed. (1972), pp. 15-22.

[De85] L. Devroye: "The Average Height of Binary Search-Trees, *submitted* (1985).

[FGPR85] P. Flajolet, G. Gonnet, C. Puech, M. Robson: Variations on Quad-trees, *in prep.* (1985).

[Fl83] P. Flajolet: On the Performance Evaluation of Extendible Hashing and Trie Searching, *Acta Informatica*, **20** (1983), pp. 345-369.

[FNPS79] R. Fagin, J. Nievergelt, N. Pippenger, R. Strong: Extendible Hashing- A Fast Access Method for Dynamic Files, *ACM Trans. on Database Systems*, **4**, (1979), pp. 315-344.

[FO83] P. Flajolet, A. Odlyzko: The Average Height of Binary Trees and Other Simple Trees, *J. of Comp. and Sytem Sc.* **25** (1982), pp. 171-213.

[FP85] P. Flajolet, C. Puech: Partial Match Retrieval of Multidimensional Data, *J.A.C.M.* (1985), to appear.

[FRS84a] P. Flajolet, M. Regnier, D. Sotteau: Algebraic Methods for Trie Statistics, *Annals of Discr. Math.* (1984), to appear.

[FRS84b] P. Flajolet, M. Regnier, R. Sedgewick: Some Uses of the Mellin Integral Transform in the Analysis of Algorithms, In: *Proc. NATO Adv. Res. Workshop on Combinatorics on Words*, Springer-Verlag (1984), to appear.

[FRV79] P. Flajolet, J-C. Raoult, J. Vuillemin: The Number of Registers Required to Evaluate Arithmetic Expressions, *Theoret. Comp. Sc.* **9**, (1979), pp. 99-125.

[FS82] P. Flajolet, J-M. Steyaert: A Complexity Calculus for Classes of Recursive Search Programs over Tree Structures, In: *Proc. 22nd IEE Symp. on Found. of Comp. Sc. (F.O.C.S)*, Nashville (1982), pp. 386-393.

[FSS85] P. Flajolet, P. Sipala, J-M. Steyaert: Compacted Representations of Trees, *in prep.* (1985).

[FVV78] J. Francon, G. Viennot, J. Vuillemin: Pagodas (1978).

[GM85] G. Gonnet, I. Munro: The Analysis of Linear Probing Sort by the Use of a New Mathematical Transform, *J. of Alg.* (1985).

[Go81] G. Gonnet: Expected Length of the Longest Probe Sequence in Hashing, *j.A.C.M.* **28** (1981), pp. 289-304. [GO81] L. Guibas, A. Odlyzko: Strings Overlaps, Pattern Matching and Non-transitive Games, *J. Comb. Theory (A)* **30** (1981), pp. 183-208.

[JK77] A. Jonassen, D. E. Knuth: "A Trivial Algorithm Whose Analysis Isn't", Stanford Univ. Tech. Rep. STAN-CS-77-498 (1977).

[JR85] P. Jacquet, M. Regnier: Limiting Distributions for Trie Parameters, *in prep.* (1985).

[Ke79] R. Kemp The Average Number of Registers Needed to Evaluate a Binary Tree Optimally, *Acta Informatica*, **11** (1979), pp. 363-372.

[KW66] A. G. Konheim, B. Weiss: An Occupancy Discipline and Applications, *S.I.A.M. J. Applied Math.* **14** (1966), pp. 1266-1274.

[KS78] D. E. Knuth, A. Schonage: The Expected Linearity of a Simple Equivalence Algorithm, *Theoretical Comp. Sc.* **6**, (1978).

[La78] P. A. Larson; Dynamic Hashing, *BIT* **18**, (1978), pp. 184-201.

[MM78] A. Meir, J.W. Moon: On the Altitude of Nodes in Random Trees, *Canad. J. Math* **30** *(1978), pp. 997-1015.*

[Od83] A. Odlyzko: Periodic Oscillations of Coefficients of Power Series that Satisfy Functional Equations, *Adv. in Math.* **44** (1982), pp. 180-205.

[Pu84] C. Puech: *Methodes d'Analyse de Structures de Donnees Dynamiques*, Thesis, Univ. of Orsay (1984).

[Re83] M. Regnier: *Evaluation des Performances du Hachage Dynamique*, Thesis, Univ. of Orsay (1983).

[Se77a] R. Sedgewick: The Analysis of Quicksort Programs, *Acta Informatica* **7** (1977), pp. 327-355.

[Se77b] R. Sedgewick: Quicksort with Equal Keys, *S.I.A.M. J. Computing* **6** (1977).

[Se78] R. Sedgewick: Data Movement in Odd-Even Merging, *S.I.A.M. J. on Computing* **7** (1978).

[Se80] R. Sedgewick: *Quicksort*, Garland Pub. Co., New-York (1980).

[Se85] R. Sedgewick: "The Asymptotic Behaviour of Heapsort", *in prep.*.

[SF83] J-M. Steyaert, P. Flajolet: Patterns and Pattern-Matching in Trees: An Analysis, *Inf. and Control* **58** (1983), pp. 19-58.

[Ya80] A. C. Yao: Analysis of (h,k,l) -Shellsort, *J. of Alg.* **1** (1980).

Acknowledgements: This work has benefited of numerous discussions with participants of the Algorithms Seminar at INRIA especially C. Puech, J-M. Steyaert and M. Regnier, as well as notable influences from A. Odlyzko, R. Sedgewick, H. Prodinger, G. Gonnet and from the lecture notes of the Course of Analysis of Algorithms taught at Stanford University by D. E. Knuth and A. C. Yao.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique