second world conference on mathematics at the service of man

PROCEEDINGS

UNIVERSIDAD POLITECNICA DE LAS PALMAS

Las Palmas (Canary Islands), Spain

June 28 to July 3, 1982

A RECURSIVE PARTIONNING PROCESS

OF COMPUTER SCIENCE

Philippe FLAJOLET
INRIA - Rocquencourt
78150 Le Chesnay (France)

Dominique SOTTEAU
LRI - Université Paris-Sud
91405 Orsay (France)

ABSTRACT

We informally review some of the algebraic and analytic techniques involved in investigating the properties of a combinatorial process that appears in very diverse contexts in computer science including digital sorting and searching, dynamic hashing methods, communication protocols in local networks and some polynomial factorization algorithms.

1 - INTRODUCTION

The basic combinatorial process which is studied here is the following : one starts with a finite set of individuals ; in the first stage each individual tosses a coin ; individuals are then split into two groups : the "heads" group and the "tails" group. Each subgroup then recursively repeats the process until some termination condition is met. Various control policies are conceivable. The simplest ones are :

- halting the process for groups that are of size 1.
- halting the process for subgroups that reach a size less than or equal to a fixed integer b.

One may also consider situations where biased or unbiased coins are used, cases where dice of various configurations are used, or situations where the set of individuals varies dynamically.

The succession of splittings can easily be described in the form of a tree. A group X created at some stage of the partitionning process is represented by a node in the tree ; if X is split into X_H (the heads group) and X_T (the tails group), then X_H is represented as the left son-node of X_H and X_T is represented as the right-son node of X. Figure 1 represents a possible partitionning tree when the process is stopped on subgroups of size 1.

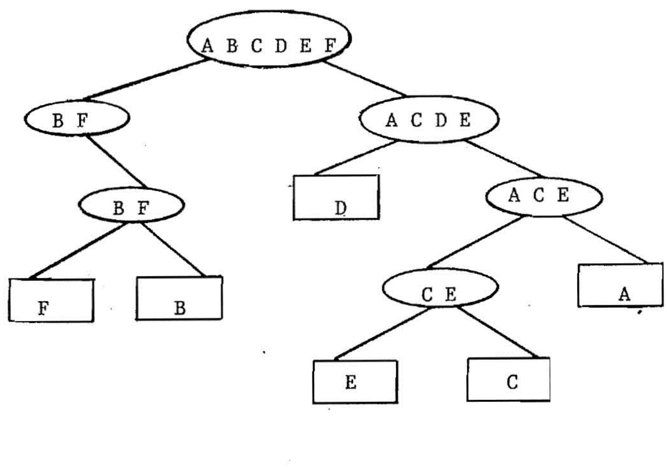

Figure 1 : A recursive partitionning tree on the set {A,B,C,D,E,F}. Terminating subgroups are represented by squares.

This process appears to underlie a large number of computer algorithms which we now mention :

(i) Collision resolution in networks : In a decentralized network, several users have access to a common channel which they use to broadcast information and on which collisions that occur have to be resolved. Basically, when a collision occurs, colliding senders separate themselves into two groups (using a stochastic decision procedure). Members of the first group first recursively resolve their conflicts and broadcast, and members of the second group wait until the channel is free to transmit. These protocols are considered for instance in [Ca 79], [TV 80], [FH 81].

(ii) Digital sorting and searching : The prototype is digital sorting (see [Kn 73] for several applications) : to sort a set of binary sequences, one first separates them into two groups depending on the values of their leading digit. One then recursively proceeds to recursively sort each of the two groups on their next leading digits. The algorithm is isomorphic to the construction of a digital search tree -or "trie"- on a set of binary sequences.

(iii) Dynamic Hashing : Such algorithms are used to manage large files kept on secondary storage ([Li 78], [La 78], [FNPS 79]). Larson's dynamic hashing algorithms starts like classical hashing with a fixed page capacity (or bucket size) b. When a bucket overflows, the hashing function is refined locally on this bucket, and two new buckets are allocated. A tree -called the index- retraces the history of successive splittings and is used to guarantee direct access to records on the secondary storage device.

(iv) Polynomial Factorization : Some recent developments of Berlekamp's Factorization algorithm for polynomials over a finite field due to Cantor, Zassenhauss [CZ 81] and Lazard [La 81] are based on an iterative construction of primitive idempotents. The construction is a refinement process that can be shown equivalent to the generation of a partition tree with biased probabilities on splittings.

Most of the characteristic parameters of these algorithms are expressible in terms of classical parameters of the corresponding partition tree, like : path length, number of nodes, height, number of unary nodes, left path length... We thus examine in the next sections general methods for the analysis of these parameters.

2 - ALGEBRAIC METHODS

Developments in this section permit, in a large number of cases, to determine in a simple, and quasi-automatic, way generating functions associated to parameters of partition trees. We use the classical representation of finite sets of (finite or infinite) binary sequences by trees. With 0's corresponding to a left branching edges and a 1's to right branching edges, the tree of Figure 1 would be associated to any set of sequences of the form :

A = 111... B = 011... C = 1101...
D = 10... E = 1100... F = 010...

We start with trees associated to binary sequences of a fixed length s. For each $s \geq 0$, we let

$$B^{(s)} = \{0;1\}^s \quad ; \quad P^{(s)} = \mathcal{P}(B^{(s)}).$$

($P^{(s)}$ is the power set of $B^{(s)}$).

We also introduce the sets of variables

$$X^{(s)} = \{x_u / u \in B^{(s)}\} \; ; \; X = \bigsqcup_s X^{(s)}$$

which are used to encode corresponding sequences. A set of sequences $\omega \in P^{(s)}$ will be represented by the <u>product</u>

$$\rho(\omega) = \prod_{u\in\omega} x_u, \tag{1}$$

and a family Ω of elements of $P^{(s)}$ is represented by the generating polynomial

$$\sum_{\omega\in\Omega} \rho(\omega). \tag{2}$$

For instance, when $s = 2$, $\Omega = P^{(2)}$ is represented by the polynomial

$$1+x_{00}+x_{01}+x_{10}+x_{11}+x_{00}x_{01}+x_{00}x_{10}+x_{00}x_{11}+x_{01}x_{10}$$
$$+x_{01}x_{11}+x_{10}x_{11}+x_{00}x_{01}x_{10}+x_{00}x_{01}x_{11}$$
$$+x_{00}x_{10}x_{11}+x_{01}x_{10}x_{11}+x_{00}x_{01}x_{10}x_{11}$$

which appears to be equal to :

$$(1+x_{00})(1+x_{01})(1+x_{10})(1+x_{11}).$$

Similarly, a multiset on $P^{(s)}$:

$$M = \sum_{\omega\in P^{(s)}} \mu(\omega) \cdot \dot{\omega} \tag{3}$$

is represented by the polynomial

$$M^{(s)}(X) = \sum_{\omega\in P^{(s)}} \mu(\omega)\ \rho(\omega). \tag{4}$$

In order to express recurrences based on the size s of binary sequences we introduce the functions

$$\sigma_0 : B^{(s)} \to B^{(s+1)} \quad ; \quad \sigma_1 : B^{(s)} \to B^{(s+1)}$$

defined by

$$\sigma_0(u) = \underline{0}u \quad ; \quad \sigma_1(u) = \underline{1}u \;.$$

with normal extension to sets and multisets.

For β in $B^{(s)}$ with $s > 0$, we introduce the dual operations $\beta/0$, $\beta/1$ by :

$$\beta/0 = \{u\} \text{ if } \omega = 0u, \quad \beta/0 = \emptyset \text{ if } \omega = 1u$$
$$\beta/1 = \{u\} \text{ if } \omega = 1u, \quad \omega/1 = \emptyset \text{ if } \omega = 0u,$$

with extensions to sets ; for $\omega \in P^{(s)}$, we have

$$\omega/0 = \sum_{\beta\in\omega} \beta/0 \qquad \omega/1 = \sum_{\beta\in\omega} \beta/1 \;.$$

so that

$$\omega = \sigma_0(\omega/0) + \sigma_1(\omega/1) \;.$$

The σ operations are applied to generating polynomials interpreting these polynomials as multisets. For instance

$$\sigma_1(1+2x_{00}+4x_{01}x_{10}) = 1+2x_{100}+4x_{101}x_{110} \;.$$

In the sequel, we adhere to the notational conventions of (3), (4), representing valuations on the $P^{(s)}$ by lower case letter, corresponding multisets by script letters and associated generating polynomials by capital letters. Whenever convenient, we also identify multisets and generating polynomials that encode them.

<u>Lemma 1</u> : Let $v^{(s)}$, $w^{(s)}$ be two valuations on $P^{(s)}$. The valuations on $P^{(s)}$, $P^{(s+1)}$ defined by

$$a^{(s)}(\omega) = v^{(s)}(\omega) + w^{(s)}(\omega)$$
$$b^{(s+1)}(\omega) = v^{(s)}(\omega/0) \,.\, w^{(s)}(\omega/0)$$

have corresponding generating polynomials :

$$A^{(s)}(X) = V^{(s)}(X) + W^{(s)}(X)$$
$$B^{(s+1)}(X) = \sigma_0(V^{(s)}(X)) \,.\, \sigma_1(W^{(s)}(X)) \;.$$

In particular since $P^{(s)}$ is the multiset that corresponds to the constant valuation $p(\omega) = 1$, we have

$$p(\omega) = p(\omega/0) \,.\, p(\omega/1)$$

whence the recurrence

$$P^{(s+1)}(X) = \sigma_0(P^{(s)}(X)) \,.\, \sigma_1(P^{(s)}(X))$$

with $P^{(0)}(X) = 1 + x_\varepsilon$

(ε denoting the empty sequence). Thus by recurrence the expected result :

$$P^{(s)}(X) = \prod_{u\in B^{(s)}} (1+x_u) \;. \tag{5}$$

As a consequence of Lemma 1, if

$$a^{(s+1)}(\omega) = v^{(s)}(\omega/0) + w^{(s)}(\omega/1) \tag{6}$$

we have

$$a^{(s+1)}(\omega) = v^{(s)}(\omega/0) \,.\, p(\omega/1) + p^{(s)}(\omega/0)\ w^{(s)}(\omega/1),$$

so that

$$A^{(s+1)}(X) = \sigma_0(V^{(s)}(X))\sigma_1(P^{(s)}(X)) + \sigma_0(P^{(s)}(X)) \,.\, \sigma_1(W^{(s)}(X)) \;. \tag{7}$$

The power of Lemma 1 and of (6), (7) comes from the fact that most parameters of interest on trees are definable as additive-multiplicative combinations of similar or simpler parameters on subtrees, so that a large number of equations can be written systematically. Also these equations on multivariate generating functions yield more classical generating functions by various sorts of morphisms. Starting again from (3), (4), and denoting by M(x) the polynomial obtained from M(X) by replacing all subscripted variables by the variable x, we see that

$$[x^n]\ M^{(s)}(x) = \sum_{\substack{\omega\in P^{(s)} \\ |\omega|=n}} \mu(\omega)$$

where $|\omega|$ is the number of sequences in ω and $[x^n]\ f(x)$ is the coefficient of x^n in $f(x)$.

Equations obtained in this way are much simpler and can usually be solved by iterating the recurrence on s. We give below a few examples :

1. The univariate generating polynomial of $P^{(s)}$ is from (5)

$$P^{(s)}(x) = (1+x)^{2^s} \tag{8}$$

and accordingly the number of n-subsets of $B^{(s)}$ is

$$[x^n]\ (1+x)^{2^s} = \binom{2^s}{n}$$

2. Let $P_k^{(s)}$ be the family of sets whose associated tree has height $\leq k$. Clearly

$$P_0^{(s)} = 1 + \sum_{u\in B^{(s)}} x_u \quad ;$$

$$P_{k+1}^{(s+1)}(X) = \sigma_0(P_k^{(s)}(X))\ \sigma_1(P_k^{(s)}(X))$$

so that

$$P_k^{(s)} = (1+2^{s-k}\ x)^{2^k} \tag{9}$$

and the probability that a set of n sequences of length s gives rise to a tree of height $\leq k$ is :

$$2^{n(s-k)}\binom{2^k}{n}\binom{2^s}{n}^{-1} . \tag{10}$$

3. Similarly if the tree growth is stopped when groups have size $\leq b$, the probability that the height is $\leq k$ for a tree formed with n sequences is

$$\frac{1}{\binom{2^s}{n}}\ [x^n]\ \left(B_b^{(2^{s-k})}(x)\right)^{2^k}$$

where

$$B_b^{(m)}(x) = 1 + \binom{m}{1}x + \binom{m}{2}x + \binom{m}{2}x^2 + \dots + \binom{m}{b}x^b . \tag{11}$$

4. The previous examples showed the use of the product rule of Lemma 1. A large number of parameters of interest in applications are defined as additive and multiplicative combinations. The simplest example is the statistics of the total number of nodes in the tree constructed from binary sequences of fixed length. This parameter $no(\omega)$ is defined by the recurrences

$$no^{(0)}(\omega) = |\omega|$$

$$no^{(s+1)}(\omega) = no^{(s)}(\omega\backslash 0) + no^{(s)}(\omega\backslash 1) + 1 - \delta_{|\omega|,0} - \delta_{|\omega|,1} ,$$

whence the equations

$$NO^{(s+1)}(x) = 2P^{(s)}(x)\ NO^{(s)}(x) + P^{(s+1)}(x) - 1 - 2^{s+1}x . \tag{12}$$

The equation can be solved by iteration, and we find

$$NO_n^{(s)} = \binom{2^s}{n}(n+2^s-1) - \sum_{j=1}^{s-1}\left[2^j\binom{2^s-2^{s-j}}{n} + 2^s\binom{2^s-2^{s-j}}{n-1}\right] \tag{13}$$

5. Path length is another parameter of interest since it is related to the time necessary for sorting a set of sequences by constructing the associated tree. From the classical inductive definition of path length, we find equations similar to those of part (4) above, and solving one has for the cumulated path length of trees constructed on n binary sequences :

$$LCE_n^{(s)} = \binom{2^s-1}{n-1}2^s(s+1) - 2^s\sum_{j=0}^{s-1}\binom{2^s-2^{s-j}}{n-1} . \tag{14}$$

For a fixed cardinality n, letting s tend to infinity, one notices that average values of most practical parameters on trees tend to well defined limits. These limiting values coincide with the average values of the corresponding parameters on trees constructed from infinite sequences, under the usual statistic on $\{0;1\}^\infty$ (The trees are finite with probability 1). They are also useful as they constitute good approximations to the finite case when $n \ll 2^s$ and are themselves easier to estimate by the methods of section 3.

For a parameter $\pi^{(s)}$ on $P^{(s)}$, we define the cumulative values

$$\Pi_n^{(s)} = \sum_{\substack{|\omega|=n \\ \omega\in P^{(s)}}} \pi^{(s)}(\omega) \tag{15}$$

and we are interested in computing the quantities (of which we assume the existence) :

$$\Pi_n^{(\infty)} = \lim_{s\to\infty} \frac{\Pi^{(s)}}{\binom{2^s}{n}} . \tag{16}$$

<u>Lemma 2</u> : The average values of parameters corresponding to infinite sequences have exponentiel generating function given by :

$$\Pi^{(\infty)}(z) = \sum_{n\geq 0} \Pi_n^{(\infty)}\frac{z^n}{n!} = \lim_{s\to\infty} \Pi^{(s)}\left(\frac{x}{2^s}\right) .$$

In particular the exponential generating function of the parameter $p(\omega) \equiv 1$ is e^x and one has

$$e^x = \lim_{s\to\infty}\left(1+\frac{x}{2^s}\right)^{2^s}$$

in accordance with Lemma 2.

Lemma 2 used in conjunction with Lemma 1 permits a direct calculation of generating functions associated to parameters defined on $\mathcal{P}_f(B^{(\infty)})$ (finite sets of infinite sequences). So that, if :

$$a(\omega) = v(\omega/0) + w(\omega/1) \ ; \quad b(\omega) = v(\omega/0)\ .\ w(\omega/0), \tag{17}$$

one has for the associated exponential generating functions :

$$A^{(\infty)}(x) = e^{x/2}\left(V^{(\infty)}\left(\frac{x}{2}\right) + W^{(\infty)}\left(\frac{x}{2}\right)\right) ;$$

$$B^{(\infty)}(x) = V^{(\infty)}\left(\frac{x}{2}\right) . W^{(\infty)}\left(\frac{x}{2}\right) . \tag{18}$$

For recursively defined parameters, the translation schemes (17), (18) lead to functional difference equations of the form

$$\phi(x) = a(x)\ \phi\left(\frac{x}{2}\right) + b(x) . \tag{19}$$

These can normally be solved by iteration, so that

$$\phi(x) = \sum_{k\geq 0} b(x2^{-k}) \prod_{j=0}^{k-1} a(x2^{-j}) .$$

In the frequent case $a(x) = Ae^{cx}$, (20) further simplifies to

$$\phi(x) = \sum_{k\geq 0} A^k\, b(x2^{-k})\, \exp(c(1-2^{-k})x) \qquad (20)$$

whose Taylor coefficient have an explicit form. Some examples follow :

1. The cumulative distribution of height in trees corresponding to Example 3 of the finite case, becomes in the infinite case :

$$[\tfrac{x^n}{n!}] \quad e_b\left(\frac{x}{2^k}\right)^{2^k}$$

where

$$e_b(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^b}{b!} \qquad (21)$$

which agrees with the classical result on occupancy problems in probability theory.

2. External path length leads to the equation

$$\phi(x) = 2e^{x/2}\phi\,\frac{x}{2} + x(e^x-1),$$

which when solved by (20) (21) leads to the two equivalent forms :

$$LCE_n^{(\infty)} = n\sum_{k\geq 0}\left(1-(1-2^{-k})^{n-1}\right) \qquad (22)$$

$$= \sum_{p=2}^{n}\binom{n}{p}\frac{(-1)^p p}{2^{p-1}-1}$$

Example 1 above appears in the analysis of extendible hashing [FS 82] ; Example 2 is a classical result [Kn 73].

Amongst the several possible extensions of these algebraic methods, we mention :

(i) Translation schemes corresponding to a biased distribution on bits of sequences. For instance, in the infinite case, if p and q are the probabilities of zeros and 1 respectively, then (17) translates into :

$$A^{(\infty)}(x) = e^{qx}\,V^{(\infty)}(px) + e^{px}\,W^{(\infty)}(qx) \quad ;$$

$$B^{(\infty)}(x) = V^{(\infty)}(px)\,W^{(\infty)}(x) \quad . \qquad (23)$$

For instance if $h_m(\omega)$ is 1 if the height of the tree associated to ω is $\leq$ m and 0 otherwise, we have

$$H_m^{(\infty)}(x) = H_{m-1}^{(\infty)}(px)\,H_{m-1}^{(\infty)}(qx)$$

whence

$$H_m^{(\infty)}(x) = \prod_{k=0}^{m}(1+p^k q^{m-k} x)^{\binom{m}{k}} \qquad (24)$$

This result is used in [FS 82] to analyze a polynomial factorization algorithm. The methods extend to alphabets of cardinality larger than 2.

(ii) Extension to functions of several sets of binary sequences n applications, these occur for instance in algorithms for performing set-theoretic operations (see [TP] for several examples). Thus using trees to compute the intersection of two sets ω, ω' by the relation :

$$\omega \cap \omega' = \sigma_0(\omega\backslash 0 \cap \omega'\backslash 0) + \sigma_1(\omega\backslash 1 \cap \omega'\backslash 1) \qquad (25)$$

with the corresponding computation time :

$$I(\omega,\omega') = I(\omega\backslash 0,\, \omega'\backslash 0) + I(\omega\backslash 1,\, \omega'\backslash 1)$$
$$+ 1 - \delta_{|\omega|,0} - \delta_{|\omega'|,0} + \delta_{|\omega|,0}\,\delta_{|\omega'|,0}\,,$$

we find for the bivariate exponential generating fonction :

$$I^{(\infty)}(x,y) = 2e^{x/2}\,e^{y/2}\,I\left(\frac{x}{2},\frac{y}{2}\right) + (e^x-1)\,(e^y-1) \qquad (26)$$

This equation can be solved by techniques described above.

(iii) Variances and higher moments can also be derived. With q a formal variable, we have for external path length :

$$q^{e(\omega)} = q^{e(\omega/0)}\, q^{e(\omega/1)}\, q^{|\omega|}\, q^{-\delta_{|\omega|,1}}$$

whence for the bivariate generating function of probabilities the recurrence

$$L^{(s)}(z;q) = (L^{s-1)}(qz,q))^2 + 2^s\, q(1-q)z\ldots \quad .$$

3 - ANALYTIC METHODS

3.1 - Multiplicative valuations

Purely multiplicative valuations on trees lead to generating functions that have product forms. A typical example is tree height with equations (9), (11), (21), (24). The saddle point method is well suited to the derivation of limit distributions (whence averages and variances). The starting point is Cauchy's integral form of coefficients of analytic functions :

$$[z^n]\; f(z) = \frac{1}{2i\pi}\int_\Gamma f(z)\,\frac{dz}{z^{n+1}} \qquad (27)$$

which can be put under the form :

$$\frac{1}{2i\pi}\int_\Gamma e^{h(z)}\,dz \quad .$$

The saddle point heuristic (see [He 78]) consists in selecting for Γ a contour that crosses some saddle point of h(z), i-e a point s such that $h'(s) = 0$

In the case of integrals of the form (27) with f an entire function depending on a parameter, a circle centered around the origin and crossing the saddle point of smallest modulus leads to good localization properties of the integral (27) : the main contribution is shown to come from a small fraction of the contour around the saddle-point ; there local expansions are used to approximate the integral.

One proves in this way [FS 82] that the probability π_n^m of having a tree of height $\leq$ m with n keys when subtrees of size $\leq$ b are grouped in a single leaf (page) satisfies :

$$\pi_n^m = e^{-\beta(n)2^{-b\delta}}$$

with

$$\delta = m - \lfloor (1+\tfrac{1}{b}) \log_2 n \rfloor \qquad (28)$$

and $\beta(n)$ a bounded fluctuating function of n. This formula shows in particular that $\pi_n^m - \pi_n^{m-1}$ has a strong peak around $(1+\frac{1}{b}) \log_2 n$, with periodicities (in n) appearing in the probability distribution.

3.2 - Additive valuations

Examples of additive valuations have been given when discussing statistics on the number of nodes and external path length. The algebraic paradigm is summarized by equations (19), (20) ; results appear as sums of which (22) is typical. Using thus path length as an illustration, the problem is to approximate sums like

$$S_n = \sum_{k\geq 0} [(1-2^{-k})^n - 1] \quad . \qquad (29)$$

The classical way of dealing with such sums is first to introduce an exponential approximation, using

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

valid for small a. Thus one first approximates S_n by :

$$T_n = \sum_{k\geq 0} (e^{-n2^{-k}} - 1) \quad . \qquad (30)$$

An expression like (30)belongs to the category of harmonic sums of the general form

$$F(x) = \sum \alpha_k \, f(\beta_k X) \quad . \qquad (31)$$

To determine the asymptotic behaviour of (31) for large values of x, following [Kn 73], one computes the Mellin transform of F :

$$F^*(s) = \int_0^\infty F(x) \, x^{s-1} \, dx \qquad (32)$$

which in this case factorizes as

$$F^*(s) = f^*(s) \sum_{k\geq 0} \alpha_k \, \beta_k^{-s} \; , \qquad (33)$$

i-e as the product of the Mellin transform of the fundamental function and of a Dirichlet series related to the amplitudes and phases of the harmonics. As is known the singularities of $F^*(s)$ -which are easy to determine using the factor form (33)- are related to the terms in the asymptotic behaviour of F(x) at 0 and at ∞ : a combination of the Mellin inversion theorem with Cauchy's residue theorem shows that

$$F(x) = \sum \mathrm{Res}(F^*(s) \, x^{-s})$$

$$+ \text{ remainder of smaller order} \qquad (34)$$

where the summation is extended to poles of F (s) in a stripe. Equation (34) is based on the inversion theorem for the Mellin transform :

$$F(x) = \frac{1}{2i\pi} \int_{c-i\infty}^{c+i\infty} F^*(s) \, x^{-s} \, ds \quad . \qquad (35)$$

In the case of

$$T(x) = \sum_{k\geq 0} e^{-x2^{-k}} - 1 ,$$

one finds :

$$T^*(s) = \frac{\Gamma(s)}{1-2^s} \qquad -1 < \mathrm{Re}(s) < 0 \quad .$$

This function has a double pole at s = 0 and simple poles at

$$s = \frac{2ik\pi}{\log 2} \; , \quad k \in Z/\{0\} \; .$$

Hence,

$$T(x) = -\log_2 x + \frac{1}{2} - \gamma$$

$$+ \frac{1}{\log 2} \sum_{k\neq 0} \Gamma\left(\frac{2ik\pi}{\log 2}\right) e^{-2ik\pi(\log_2 n)} + O(x^{-M})$$

the sum being a Fourier series in $\log_2 n$.

Such periodicities are of frequent occurrence in the analysis of algorithms and they have here a clear origin in regularly spaced singularities of functions of the type $(1-2^s)^{-1}$. An alternative derivation, which avoids the exponential approximation is based on the observation that S_n is itself a harmonic sum :

$$S(x) = \sum (e^{-x\log(1-2^{-k})^{-1}} - 1)$$

which can be dealt with by the preceding techniques, leading sometimes to simpler derivations.

Let us last mention that the results of the exponential approximations (30) coincides in a large number of cases with expressions derived by assuming the sequences to be generated by a Poisson process.

4 - APPLICATIONS

We have made an attempt at summarizing the main techniques that can be used to analyze a general partitionning process of computer science. We now conclude by informally mentionning some typical applications.

The first application concerns the stack protocol for resolving collisions in networks sharing a single communication channel :

Theorem 1 [FFH 82] : *The time necessary to resolve n collisions in an open stack protocol network satisfies*

$$\alpha_n = An + n\phi(n) + O\left(\frac{n}{\log n}\right) \quad .$$

where ϕ is a fluctuating function of small amplitude and mean value 0.

The exponential generating function of the α_n satisfies a functional equation of the form

$$\alpha(z) = c^{-qz}\alpha(\lambda+pz) + e^{-pz}\alpha(\lambda+qz) + \beta(z) \qquad (36)$$

where λ is the Poisson anival rate on the channel. The interest of this non-local difference equation is the non-commutative character of its iteration group.

Theorem 2 [Kn 73] *: The total number of bit inspections in radix-exchange sort applied to n uniformly distributed binary keys satisfies :*

$$\overline{E}_n = n\log_2 n + \left(\frac{\gamma}{\log 2} - \frac{1}{2}\right) n - \frac{n}{\log 2} \sum_{k\neq 0} \Gamma\left(\frac{2ik\pi}{\log 2}\right) e^{-2ik\pi \log_2 n} + o(n)$$

Theorem 2 and companion results show that radix exchange sort is in the class of quasi-optimal sorting algorithms.

Theorem 3 [FS 82] *: The expected size of the directory of an extendible hashing file of n elements satisfies :*

$$\overline{S}_n = Q((1+\tfrac{1}{b})\log_2 n)\ n^{1+1/b} + o(n^{1+1/b})$$

where Q *is a periodic function with Fourier coefficients*

$$q_0 = -\frac{1}{b\log 2}\ \beta^{1/b}\,\Gamma\left(-\frac{1}{b}\right)\ ;$$

$$q_k = \frac{-1}{b\log 2}\ \beta^{-\chi_k+1/b}\ (\chi_k - \tfrac{1}{b})$$

with

$$\chi_k = \frac{2ik\pi}{b\log 2} \quad \textit{and} \quad \beta = (b+1)!^{-1}\ .$$

This result shows a non-linearity of the growth of the index which is very perceptible for small b, in which case compromise chaining solutions should be used.

The last result is relative to the "idempotent algorithm" [CZ 81], [La 81] for factoring a polynomial over a finite field.

Theorem 4 [FS 82] *: The idempotent algorithm factorizes a polynomial with n irreducible factors in*

$$\overline{H}_n = \frac{2\log_2 n}{\log_2(\alpha^2+\beta^2)^{-1}} + O(1) \qquad \alpha = \frac{1}{2} + \frac{1}{2p} \qquad \beta = \frac{1}{2} + \frac{1}{2p}$$

"main" steps.

This result is a refinement of bounds given in the original papers. The interest of the algorithm when used on polynomials over GF_p is to avoid an exhaustive scan over the field elements which leads to considerable improvements over Berlekamp's algorithm for large q.

References

[Ca 79] Capetanakis J.I. "Tree Algorithms for Packet Broadcast Channels", IEEE Trans. Inf. Th. IT-25 pp. 505-515 (1979).

[CZ 81] Cantor D., Zassenhauss H. "A New Algorithm for Factorizing Polynomials over Large Finite Fields", Math. of Comp. 36 (1981), pp. 581-592.

[FFH 82] Fayolle G., Flajolet P., Hofri M. "An Evaluation of a Stack Based CSMA protocol", in preparation (1982).

[FH 81] Fayolle G., Hofri M. "On the Capacity of a CSMA Channel under Stack-Based Collision Resolution Algorithms" Manuscript (1981).

[FNSP 79] Fagin R., Nievergelt J., Pippenger N., Strong H.R. "Extendible Hashing - A Fast Access Method for Dynamic Files", ACM Trans. on Database System, 4 (1979), pp. 289-344.

[FS 82] Flajolet P., Steyaert J.M. "A Branching Process Arising in Dynamic Hashing, Trie Searching and Polynomial Factorization", in 9th ICALP, Aarhus (1982).

[He 78] Henrici P. "Applied and Computational Complex Analysis", Vol. 2, J. Wiley, New York (1978).

[La 78] Larson P.A. "Dynamic Hashing", BIT, 18 (1978), pp. 184-201.

[La 81] Lazard D. "Factorisation des Polynômes", in 4ème Journées Algorithmiques, Poitiers (1981), also submitted for publication.

[Li 78] Litwin W. "Virtual Hashing : A Dynamically Changing Hashing", in Proc. Very Large Data Bases conf., Berlin (1978), pp. 517-523.

[TV 80] Tsybakov B.S., Vvedenskaya N.D. "Network Theory and Large Systems", in Problemy Peredachi Informatsii, 16 (1980), pp. 80-94.