Register allocation for unary-binary trees

Philippe Flajolet, Helmut Prodinger

INRIA

CENTRE DE ROCQUENCOURT

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

Rapports de Recherche

N° 266

REGISTER ALLOCATION FOR UNARY-BINARY TREES

Philippe FLAJOLET
Helmut PRODINGER

Janvier 1984

REGISTER ALLOCATION FOR UNARY-BINARY TREES

Philippe FLAJOLET
INRIA
Rocquencourt
78150 Le Chesnay
France

Helmut PRODINGER
Technical University Vienna
Gußhausstraße 27-29
A-1040 Vienna
Austria

Abstract:

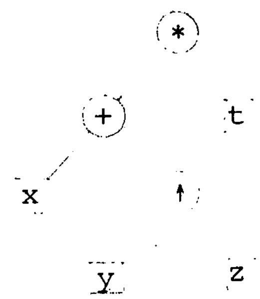
We study the number of registers required for evaluating arithmetic expressions formed with any set of unary and binary operators. Our approach consists in a singularity analysis of intervening generating functions combined with a use of (complex) Mellin inversion. We illustrate it by first rederiving the known results about binary trees and then extend it to the fully general case of unary-binary trees. The method used, as mentioned in the conclusion, is applicable to a wide class of combinatorial sums.

Résumé: *Nous étudions le nombre moyen de registres nécessaire à l'évaluation d'expressions formées d'un ensemble quelconque d'opérateurs unaires ou binaires. L'approche suivie consiste en une analyse de singularité des séries génératrices d'énumération combinée à une utilisation de propriétés de la transformation de Mellin (complexe). La méthode suivie est applicable à l'analyse asymptotique d'une large classe de sommes combinatoires intervenant en analyse d'algorithmes.*

Abstract: *We study the average number of registers required for evaluating arithmetic expressions formed with any set of unary and binary operators. Our approach consists in a singularity analysis of intervening generating functions combined with the use of (complex) Mellin inversion. The method used is also applicable to a wide class of combinatorial sums appearing in the analysis of algorithms.*

1. INTRODUCTION

An arithmetic expression with only binary operations may be described as a binary tree. For instance, (x+y↑z)*t corresponds to



There is an optimal strategy to evaluate binary trees [5]. The minimal number of registers necessary to keep intermediate results is called the register function of the tree t, and is denoted by Reg(t). This function may be defined recursively as follows:

$$\mathrm{Reg}(\square) = 0$$

$$\mathrm{Reg}\Big(\overset{\circ}{t_1\quad t_2}\Big) = \begin{cases} 1 + \mathrm{Reg}(t_1) & \text{if } \mathrm{Reg}(t_1)=\mathrm{Reg}(t_2) \\ \max\{\mathrm{Reg}(t_1),\mathrm{Reg}(t_2)\} & \text{otherwise .} \end{cases}$$

If $\mathcal{B}$ denotes the family of all binary trees and $\mathcal{R}_p$ the family of trees t with Reg(t)=p, one has the following symbolic equation [6]:

$$\mathcal{B} = \square + \overset{\circ}{\mathcal{B}\quad\mathcal{B}}$$

$$\begin{cases} \mathcal{R}_p = \overset{\circ}{\mathcal{R}_{p-1}\quad\mathcal{R}_{p-1}} + \overset{\circ}{\sum_{j<p}\mathcal{R}_j\quad\mathcal{R}_p} + \overset{\circ}{\mathcal{R}_p\quad\sum_{j<p}\mathcal{R}_j} & p\geq 1 \\ \mathcal{R}_0 = \square \end{cases}$$

The average number D_n of registers needed to evaluate a binary tree of size n (i.e. n internal nodes) is a well-studied quantity [7,12,16]: It satisfies

$$D_n = \log_4 n + D(\log_4 n) + \mathcal{O}\left(\frac{\log^* n}{\sqrt{n}}\right) ,$$

where D is a periodic function with period 1 and known Fourier coefficients and $\log^* n$ denotes an unspecified power of $\log n$ (usually different powers in different situations).

The aim of the present paper is twofold: Firstly, we give an alternative proof of this result, which is based on an analytic technique "à la Odlyzko" that has proved to be very helpful in tree enumeration problems (see [9,17]); this alternative proof permits us if needed to derive asymptotic expansions of D_n to any order. Secondly, the approach extends easily to more general classes of trees: assume unary operations like $-$, sin, exp, log, etc. are also permitted. We are then dealing with a family $\hat{\mathcal{B}}$ of trees given by

$$\hat{\mathcal{B}} = c_0 \cdot \square + c_1 \cdot \underset{\hat{\mathcal{B}}}{\overset{\circ}{|}} + c_2 \cdot \underset{\hat{\mathcal{B}}\quad\hat{\mathcal{B}}}{\overset{\circ}{\diagup\;\diagdown}} .$$

Here, we assume $c_0>0$, $c_1\geq 0$, $c_2>0$: there are c_0 nullary node, c_1 unary node and c_2 binary node labels (operators). This is just a special case of a simply generated family of trees [15]. The corresponding enumeration results involve generalized trinomial coefficients and our approach avoids some of the clumsiness involved in approximating these trinomial coefficients.

The register function is also defined on unary-binary trees in an obvious way: it is clear that unary nodes do not affect the register function. More precisely, for a unary-binary tree t, the register function Reg(t) is defined inductively by:

$$\mathrm{Reg}(\square) = 0$$

$$\mathrm{Reg}\Big(\underset{t}{\overset{\circ}{|}}\Big) = \mathrm{Reg}(t)$$

$$\mathrm{Reg}\Big(\underset{t_1\quad t_2}{\overset{\circ}{\diagup\;\diagdown}}\Big) = \begin{cases} 1+\mathrm{Reg}(t_1) & \text{if } \mathrm{Reg}(t_1)=\mathrm{Reg}(t_2) \\ \max\{\mathrm{Reg}(t_1),\mathrm{Reg}(t_2)\} & \text{otherwise} . \end{cases}$$

As we shall see in Section 3, we may find $\hat{R}_p$ ($\hat{}$ refers to the family $\hat{\mathcal{B}}$) from R_p by a suitable substitution. This also means that the method of proof carries over easily to $\hat{\mathcal{B}}$ by means of a change of variable, which is easy to deal with in the context of analytic functions, but uncomfortable (although possible!) if one follows the strategy of extracting coefficients from power series and approximating them afterwards.

2. THE REGISTER FUNCTION OF BINARY TREES REVISITED

Let $R_p(z)$ denote the generating function of the family $\mathcal{R}_p$. It is known [7,12,16] that

$$R_p(z) = \frac{z^{2^p-1}}{F_{2^{p+1}}(z)} ,$$

where $F_i(z)$ is the i-th Fibonacci polynomial:

$$F_i(z) = \frac{y^i-\bar{y}^i}{y-\bar{y}} , \text{ with } y = \frac{1+r}{2} , \ \bar{y} = \frac{1-r}{2} \text{ and } r = \sqrt{1-4z} .$$

The generating function of the cumulated register values is

$$E(z) = \sum_{p\geq 1} p.R_p(z) .$$

The sought average D_n is then

$$D_n = \frac{[z^n]\ E(z)}{[z^n]\ B(z)} ,$$

where $B(z)=(1-r)/2z$ is the generating function of $\mathcal{B}$ and $[z^n]f$ denotes the n-th Taylor coefficient of the power series f.

Using the substitution [2]

$$z = \frac{u}{(1+u)^2} \quad \longleftrightarrow \quad u = \frac{1-r}{1+r}$$

we easily find

$$E(z) = \frac{1-u^2}{u} \sum_{p\geq 1} \frac{u^{2^p}}{1-u^{2^{p+1}}} = \frac{1-u^2}{u} \sum_{k\geq 1} v_2(k)\ u^k ,$$

where $v_2(k)$ is the dyadic evaluation of k, defined as

$$v_2(k) = \max\{i \mid 2^i \text{ divides } k\} .$$

We want to extract $[z^n]E(z)$ by means of Cauchy's formula, viz.

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

where Γ is a path as depicted in Fig. 1:

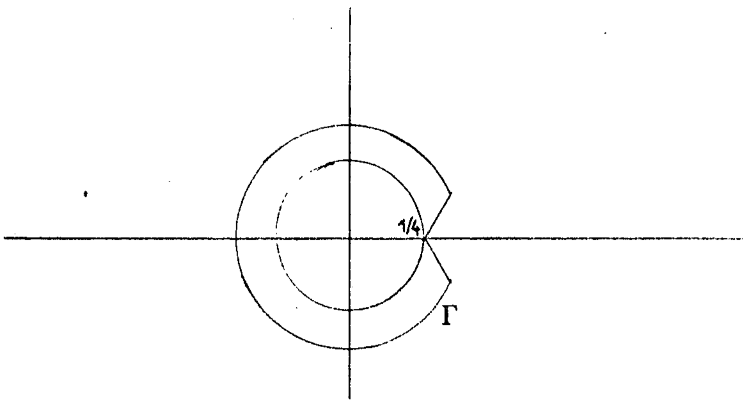


Fig. 1:

For this, we have to show that $E(z)$ has an appropriate analytic continuation in a domain which properly contains Γ.

Following the general strategy developped in [9], we can, provided we have an approximation of $E(z)$ about 1/4, "translate" it to an approximation of the coefficients $[z^n]E(z)$ given by the Cauchy integral (1). This is a fairly straightforward process once the approximation of $E(z)$ is known. So our task is reduced to the problem to obtain an expansion of the form:

$$E(z) \sim \alpha.r.\log r + \beta.r + \ldots ,$$

where $r=\sqrt{1-4z}$, in a sector about 1/4 which contains the line segment of Γ; the contribution of the Cauchy integral (1) of the part of the circle with radius >1/4 is negligible.

Now, since

$$\sum_{k\geq 1} v_2(k)\ u^k = \frac{u^2}{1-u^2} + \frac{u^4}{1-u^4} + \frac{u^8}{1-u^8} + \ldots ,$$

the unit circle $|u|=1$ is a natural boundary of this function. The nature of the mapping $z=z(u)$ is such that the boundary of the unit circle in the u-plane is mapped on the halfray $\mathrm{Re}(z)\geq 1/4$, $\mathrm{Im}(z)=0$, and this halfray thus constitutes a natural boundary for $E(z)$. From the preceding remark we are free to choose any contour that simply encircles the origing without croosing the halfray and in particular we can take the contour Γ of Figure 1. (To be fully precise, we should take a small halfcircle around $z=1/4$ and then let its radius shrink to zero.)

What remains to do is thus to find a local expansion of $E(z)$ about $z=1/4$. This will be done by the use of the Mellin transform. (See [4,20] for more information about the Mellin transform and [6,18] for some applications in Computer Science.)

We set $u=e^{-t}$ and

$$V(t) = \sum_{k\geq 1} v_2(k)e^{-kt} \ , \quad V^*(s) = \int_0^\infty x^{s-1}\, V(x)\, dx \ .$$

Since $v_2(2k)=1+v_2(k)$ and $v_2(2k+1)=0$, we easily find

$$\sum_{k\geq 1} v_2(k)\ k^{-s} = \frac{\zeta(s)}{2^s-1} \ ,$$

and so

$$V^*(s) = \frac{\Gamma(s)\zeta(s)}{2^s-1} \ , \qquad \mathrm{Re}(s)>1 \ .$$

The Mellin inversion formula gives

$$V(t) = \frac{1}{2\pi i} \int_{2-i\infty}^{2+i\infty} V^*(s)\ t^{-s}\ ds \ ,$$

and we can shift the line of integration to the left as far as we please if we only take the residues into account.

The reader might be puzzled that we use the Mellin transform of functions of a complex variable. But we actually do not need more than

$$e^{-t} = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \Gamma(s)\ t^{-s}\ ds \ , \ \mathrm{Re}(t)>0,\ c>0 \ ;$$

a reference for this is for example [1; page 91].

Thus we find an asymptotic series for $V(t)$ via

$$V(t) \sim \sum_{Re(s)\leq 1} \operatorname{Res}(V^*(s)t^{-s}) \ .$$

The main contributions come from $s=1$, $s=0$, $s=2k\pi i/\log 2$, $(k\neq 0)$. The residue at $s=1$ is easily found to be

$$\frac{1}{t} \ .$$

Now $\Gamma(s+1)=s\Gamma(s)$, and so $(s\to 0)$

$$\Gamma(s) = \frac{1}{s}[\Gamma(1)+\Gamma'(1).s+\mathcal{O}(s^2)] = \frac{1}{s} - \gamma + \mathcal{O}(s) \ .$$

Furthermore,

$$\zeta(s) = -\frac{1}{2} - \frac{1}{2}\log 2\pi - s + \mathcal{O}(s^2) \quad , (s\to 0), \quad [21]$$

and

$$\frac{1}{2^s-1} = \frac{1}{\log 2 .s} - \frac{1}{2} + \mathcal{O}(s) \quad , \quad (s\to 0);$$

also

$$t^{-s} = 1 - \log t .s + \mathcal{O}(s^2) \ , \quad (s\to 0).$$

Thus the residue at $s=0$ is

$$\frac{1}{2}\log_2 t - \frac{1}{2}\log_2 2\pi + \frac{1}{4} + \frac{\gamma}{2\log 2} \ .$$

The residue at $s=2k\pi i/\log 2 =: \chi_k$ is simply

$$c_k\, t^{-\chi_k} \quad \text{with} \quad c_k = \frac{1}{\log 2}\,\Gamma(\chi_k)\zeta(\chi_k) \ .$$

Puttings things together we find $(z\to 1/4$, i.e. $t\to 0)$

$$E(z) = (e^t - e^{-t})\ V(t) = (2t + \mathcal{O}(t^2))\ V(t)$$

$$= t.\log_2 t + K.t + \sum_{k\neq 0} 2c_k\, t^{1-\chi_k} + 2 + \mathcal{O}(t^2)$$

with

$$K = -\log_2 2\pi + \frac{1}{2} + \frac{\gamma}{\log 2} \quad .$$

Now $e^{-t} = u$ and $u = \frac{1-r}{1+r}$ with $r = \sqrt{1-4z}$, thus

$$t = -\log\frac{1-r}{1+r} = 2r + O(r^3) \ ,$$

yielding

$$E(z) = 2r\log_2 r + 2(K+1)r + 4\sum_{k\neq 0} r^{1-\chi_k} + 2 + O(r^2) \ .$$

So we find an asymptotic expansion for $[z^n]E(z)$, as announced earlier, by looking at $[z^n]2r\log_2 r$, $[z^n]2(K+1)r$ and so on. For this, we refer to [9,10,11,13]:

$$[z^n]\ (1-z)^{\alpha} = \frac{n^{-\alpha-1}}{\Gamma(-\alpha)}\left(1 + O(\tfrac{1}{n})\right) \ , \quad \alpha\neq 0,1,2,\ldots$$

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

$$+ \frac{n^{-\alpha-1}}{(\Gamma(-\alpha))^2}\,\Gamma'(-\alpha) + O\left(\frac{\log^* n}{n^{\alpha+2}}\right) \ .$$

Remark that [21]

$$\Gamma(-\tfrac{1}{2}) = -2\sqrt{\pi} \quad \text{and} \quad \Gamma'(-\tfrac{1}{2}) = 2\sqrt{\pi}(\gamma+2\log 2 - 2) \ .$$

This gives us

$$[z^n]\log(1-z).(1-z)^{1/2} = \frac{n^{-3/2}\log n}{2\sqrt{\pi}} + \frac{n^{-3/2}}{2\sqrt{\pi}}(\gamma+2\log 2-2) + O\left(\frac{\log^* n}{n^{5/2}}\right) \ .$$

Since $z = \frac{1}{4} + O(r^2)$, and therefore $B(z)=\frac{1-r}{2z} = 2-2r + O(r^2)$, we have

$$D_n = \frac{[z^n]\ E(z)}{[z^n]\ B(z)} = \frac{[z^n]\ \left(2r\log_2 r + 2(K+1)r + 4\sum_{k\neq 0} c_k r^{1-\chi_k} + 2 + O(r^2)\right)}{[z^n]\ (2-2r+O(r^2))}$$

$$= \frac{\frac{1}{\log 2} 4^n \left(\frac{n^{-3/2} \log n}{2\sqrt{\pi}} + \frac{n^{-3/2}}{2\sqrt{\pi}} (\gamma + 2\log 2 - 2) \right) - 2(K+1) \frac{2.4^{n-1} n^{-3/2}}{\sqrt{\pi}} + E}{\frac{4^n}{\sqrt{\pi}} n^{-3/2} (1 + O(\frac{1}{n}))}$$

where

$$E = 4 \sum_{k \neq 0} c_k \, 4^n \, n^{(\chi_k - 3)/2} / \Gamma(\frac{\chi_k - 1}{2}) + O(\frac{\log^* n}{n^{5/2}}) .$$

Hence

$$D_n = \frac{\log n}{2\log 2} + \frac{1}{\log 2}(\frac{\gamma}{2} + \log 2 - 1) - (K+1) + 4\sqrt{\pi} \sum_{k \neq 0} \frac{c_k n^{\chi_k/2}}{\Gamma(\frac{\chi_k - 1}{2})} + O(\frac{\log^* n}{n}) .$$

Now

$$\frac{4\sqrt{\pi} c_k}{\Gamma(\frac{\chi_k - 1}{2})} = \frac{4\sqrt{\pi}\zeta(\chi_k)}{\log 2} \cdot \frac{\Gamma(\chi_k)}{\Gamma(\frac{\chi_k - 1}{2})}$$

$$= \frac{4\sqrt{\pi} \; \zeta(\chi_k)}{\log 2} \cdot \frac{1}{2\sqrt{\pi}} \, 2^{\chi_k - \frac{1}{2}} \, \Gamma(\frac{\chi_k}{2}) . \Gamma(\frac{\chi_k + 1}{2}) \;/\; \Gamma(\frac{\chi_k - 1}{2})$$

$$= \frac{4\sqrt{\pi}\zeta(\chi_k)}{\log 2} \cdot \frac{1}{4\sqrt{\pi}} \Gamma(\frac{\chi_k}{2}) . (\chi_k - 1) = \frac{\zeta(\chi_k)\Gamma(\frac{\chi_k}{2})(\chi_k - 1)}{\log 2}$$

and therefore

$$D_n = \log_4 n - \frac{1}{2} - \frac{\gamma}{2\log 2} - \frac{1}{\log 2} + \log_2 2\pi$$

$$+ \frac{1}{\log 2} \sum_{k \neq 0} \zeta(\chi_k)\Gamma(\frac{\chi_k}{2})(\chi_k - 1) n^{\chi_k/2} + O(\frac{\log^* n}{n}) .$$

Now we notice that $n^{\chi_k/2} = e^{2k\pi i . \log_4 n}$ and state [7,12]:

THEOREM 1. The average number of registers to evaluate a binary tree with n nodes is given by

$$D_n = \log_4 n + D(\log_4 n) + O(\frac{\log^* n}{n}) ,$$

where $D(x)$ is a periodic function with period 1. This function can be expanded as a convergent Fourier series $D(x) = \sum_{k \in \mathbb{Z}} d_k e^{2k\pi i x}$, and

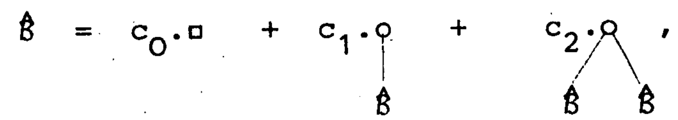
$$d_0 = -\frac{1}{2} - \frac{\gamma}{2\log 2} - \frac{1}{\log 2} + \log_2 2\pi ,$$

$$d_k = \frac{1}{\log 2}\zeta(\chi_k)\Gamma\left(\frac{\chi_k}{2}\right)(\chi_k - 1) \quad , \; k \neq 0, \; \chi_k = \frac{2k\pi i}{\log 2} \quad .$$

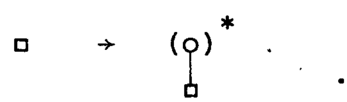
Remark that the constant d_0 was erroneously stated in [7].

3. The Register Function of Unary-Binary Trees

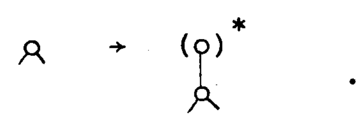
If we have a family



we can obtain it from the family $\mathcal{B}$ of binary trees by means of the following substitution process: Above each leaf insert a sequence of unary nodes, viz.



Above each binary node insert a sequence of unary nodes, viz.



For plain binary trees, $yB(yz)$ is the series enumerating $\mathcal{B}$, where y marks a leaf and z marks an internal node. Thus the corresponding series for $\hat{\mathcal{B}}$ is obtained by the substitutions

$$\begin{aligned} y &\rightarrow \frac{c_0 y}{1-c_1 z} \quad , \\ z &\rightarrow \frac{c_2 z}{1-c_1 z} \quad . \end{aligned} \tag{2}$$

Since the substitutions do not change the register function of the involved trees, we can find $\hat{R}_p$ from R_p etc. by the substitutions (2). ($\hat{R}_p$ is the generating function of the trees in $\mathcal{B}$ with register function p.)

We can define the size of a tree in $\hat{\mathcal{B}}$ in two ways:

(1) we count leaves and internal nodes,

(2) we only count internal nodes.

In terms of generating functions (1) corresponds to the transformation

$$f(z) \quad \to \quad \hat{f}(z) = \frac{c_0 z}{1-c_1 z} f\left(\frac{c_0 c_2 z^2}{(1-c_1 z)^2}\right) ,$$

while (2) corresponds to:

$$f(z) \quad \to \quad \hat{f}(z) = \frac{c_0}{1-c_1 z} f\left(\frac{c_0 c_2 z^2}{(1-c_1 z)^2}\right) .$$

We can treat both cases together by considering the more general transformation:

$$f(z) \quad \to \quad \hat{f}(z) = \frac{c_0 z+c_0'}{1-c_1 z} f\left(\frac{(c_0 z+c_0')c_2 z}{(1-c_1 z)^2}\right) , \qquad (3)$$

where c_0, $c_0' \geq 0$ and $c_0 \neq 0 \Leftrightarrow c_0'=0$. So all we have to do in order to compute the average register function $\hat{D}_n$ of all trees of size n is to perform the transformation in the expansion

$$E(z) = 2r\log_2 r + 2(K+1)r + 4\sum_{k\neq 0} c_k r^{1-\chi_k} + 2 + \mathcal{O}(r^2)$$

and in

$$B(z) = 2 - 2r + \mathcal{O}(r^2).$$

We are interested in

$$\frac{[z^n]\ \hat{E}(z)}{[z^n]\ \hat{B}(z)} .$$

($\hat{E}(z)$ and $\hat{B}(z)$ are obtained from $E(z)$ and $B(z)$ by means of the transformation (3).) The factor $(c_0 z+c_0')/(1-c_1 z)$ appears both in the numerator and the denominator. If we ignore it, we actually multiply both the numerator and the denominator by $1+\mathcal{O}(1-4z)$; this means that we have to multiply the approximation for $\hat{D}_n$ by $1+\mathcal{O}(1/n)$, and this does not affect the accuracy of the approximation for $\hat{D}_n$.

Let $\varphi(z) = (c_0 z+c_0')\, c_2 z/(1-c_1 z)^2$. We have to express $r(\varphi(z))$ in terms of $\hat{r} = (1-z/\sigma)^{1/2}$, where σ is the singularity of $r(\varphi(z))$ nearest to the origin; σ plays the role that 1/4 plays in the case of binary trees:

$$r(\varphi(z)) = \frac{1}{1-c_1 z} \sqrt{1-(2c_1+4c_0'c_2)z+(c_1^2-4c_0c_2)z^2}$$

σ is one of the solutions s_1, s_2 of,

$$(c_1^2-4c_0c_2)z^2-(2c_1+4c_0'c_2)z+1 = 0 \text{ , i.e.}$$

$$s_{1,2} = \frac{c_1+2c_0'c_2 \pm 2\sqrt{c_2}\cdot\sqrt{c_0'c_1+c_0'^2c_2+c_0}}{c_1^2 - 4c_0c_2} .$$

(a) We assume first that $c_1^2 \neq 4c_0c_2$ and $c_1+2c_0'c_2 > 0$. Then $s_1 \neq -s_2$. We set $\sigma = s_2$ and $\bar{\sigma} = s_1$. If $c_1^2 < 4c_0c_2$, then σ is the singularity closest to the origin and $|\bar{\sigma}| > |\sigma|$. If $c_1^2 > 4c_0c_2$, this is also true, because

$$c_1 + 2c_0'c_2 - 2\sqrt{c_2}\sqrt{c_0'c_1 + c_0'^2c_2 + c_0} > 0 \quad \Leftrightarrow$$

$$c_1^2+4c_0'c_1c_2+4c_0'^2c_2^2 > 4c_0'c_1c_2+4c_0'^2c_2^2+4c_0c_2 \quad \Leftrightarrow$$

$$c_1^2 > 4c_0c_2 .$$

So we have

$$(c_1^2-4c_0c_2)z^2-(2c_1+4c_0'c_2)z+1=(4c_1^2-4c_0c_2)(z-\sigma)(z-\bar{\sigma})$$

and thus as $z\to\sigma$

$$(c_1^2-4c_0c_2)z^2-(2c_1+4c_0'c_2)z+1 \sim (4c_1^2-4c_0c_2)(z-\sigma)(\sigma-\bar{\sigma})$$

$$= 4\sqrt{c_2}\sqrt{c_0'c_1+c_0'^2c_2+c_0} \cdot \sigma \cdot (1 - \tfrac{z}{\sigma}) .$$

Hence

$$r(\varphi(z)) \sim \frac{1}{1-c_1\sigma}\cdot 2\sqrt{\sigma}\cdot c_2^{1/4}(c_0'c_1+c_0'^2c_2+c_0)^{1/4}\sqrt{1-\frac{z}{\sigma}} =: A\sqrt{1-\frac{z}{\sigma}} .$$

(b) If $c_1^2 = 4c_0c_2$, then

$$r(\varphi(z)) = \frac{1}{1-c_1\sigma}\cdot \sqrt{1-(2c_1+4c_0'c_2)z} = A\sqrt{1-\frac{z}{\sigma}}$$

with

$$\sigma = \frac{1}{2c_1+4c_0'c_2} \quad \text{and} \quad A = \frac{1}{1-c_1\sigma} .$$

(c) If $c_1+2c_0'c_2=0$, we have $\bar{\sigma} = -\sigma$. This means $c_1=0$ and $c_0'=0$, so that we have to consider

$$\frac{[z^n]c_0zE(c_0c_2z^2)}{[z^n]c_0zB(c_0c_2z^2)} = \frac{[z^{n-1}]\ E(c_0c_2z^2)}{[z^{n-1}]\ B(c_0c_2z^2)} .$$

In this case n has to be odd, $n=2N+1$, and we substitute $z^2=w$ and have to consider

$$\frac{[w^N]\ E(c_0c_2w)}{[w^N]\ B(c_0c_2w)} ,$$

which is as in the other cases. So within the accuracy in which we compute the average register function, we can consider

$$\frac{[z^n]\left(2A\sqrt{1-\frac{z}{\sigma}}\log_2\left(A\sqrt{1-\frac{z}{\sigma}}\right)+2(K+1)A\sqrt{1-\frac{z}{\sigma}}+4A\sum_{k\neq 0}c_k\left(\sqrt{1-\frac{z}{\sigma}}\right)^{1-\chi_k}\right)}{[z^n]\quad -2A\sqrt{1-\frac{z}{\sigma}}} =$$

$$= \frac{[z^n]\frac{1}{2\log 2}\sqrt{1-\frac{z}{\sigma}}\log\left(1-\frac{z}{\sigma}\right)+(K+1+\log_2 A)\sqrt{1-\frac{z}{\sigma}} + 2\sum_{k\neq 0}c_k\left(\sqrt{1-\frac{z}{\sigma}}\right)^{1-\chi_k}}{[z^n]\quad -\sqrt{1-\frac{z}{\sigma}}}$$

$$= \log_4 n \quad + \quad D(\log_4 n) \quad - \quad \log_2 A \ .$$

If we consider

$$\log_4 \frac{n-1}{2} + D\left(\log_4 \frac{n-1}{2}\right) - \log_2 A ,$$

this is within our accuracy equal to

$$\log_4 n \quad + \quad D\left(\log_4 n - \frac{1}{2}\right) \quad - \quad \log_2 A \quad - \quad \frac{1}{2} .$$

This leads us to our main theorem.

THEOREM 2. Given a family $\hat{\mathcal{B}}$ of unary-binary trees:

$$\hat{\mathcal{B}} = c_0 \cdot \square + c_1 \cdot \underset{\hat{\mathcal{B}}}{\overset{\circ}{|}} + c_2 \cdot \underset{\hat{\mathcal{B}}\quad\hat{\mathcal{B}}}{\overset{\circ}{\wedge}} \quad , \quad c_0>0,\ c_2>0,\ c_1\geq 0,$$

the average register function $\hat{D}_n$, where all trees of size n are equally likely (if the size is measured by the number of internal nodes and leaves, we set $c_0'=0$; if the size is just the number of internal nodes, we set $c_0':=c_0$ and $c_0:=0$), is given by:

(a) If $c_1^2 \neq 4c_0c_2$ and $c_1+2c_0'c_2>0$, set:

$$\sigma = \frac{c_1 + 2c_0'c_2 - 2\sqrt{c_2}\sqrt{c_0'c_1+c_0'^2c_2+c_0}}{c_1^2 - 4c_0c_2}$$

and

$$A = \frac{1}{1-c_1\sigma}\, 2\sqrt{\sigma}\,.\,c_2^{1/4}(c_0'c_1+ c_0'^2c_2+c_0)^{1/4} \quad ,$$

then

$$\hat{D}_n = \log_4 n + D(\log_4 n) - \log_2 A + O\left(\frac{\log^* n}{n}\right), \quad (n\to\infty).$$

(b) If $c_1^2 = 4c_0c_2$; set

$$\sigma = \frac{1}{2c_1+4c_0'c_2} \qquad \text{and} \qquad A = \frac{1}{1-c_1\sigma} \quad .$$

Then

$$\hat{D}_n = \log_4 n + D(\log_4 n) - \log_2 A + O\left(\frac{\log^* n}{n}\right), \quad (n\to\infty).$$

(c) If $c_1+2c_0'c_2=0$, then for odd n, we have with σ, A defined as in (a):

$$\hat{D}_n = \log_4 n + D(\log_4 n - \tfrac{1}{2}) - \log_2 A - \tfrac{1}{2} + O\left(\frac{\log^* n}{n}\right), \quad (n\to\infty).$$

EXAMPLE. Let us consider the Motzkin trees, defined by:

$$\mathcal{M} = \square + \underset{\mathcal{M}}{\overset{\circ}{|}} + \underset{\mathcal{M}\ \mathcal{M}}{\overset{\circ}{\wedge}}$$

Let a leaf contribute to the size. The generating function of the numbers of Motzkin trees satisfies $M(z) = z(1+M(z)+M(z)^2)$, whence

$$M(z) = \frac{1 - z - \sqrt{1 - 2z - 3z^2}}{2z},$$

$c_0=1$, $c_0'=0$, $c_1=1$, $c_2=1$, $\sigma=1/3$, $A=\sqrt{3}$. The average number $\mathcal{D}_n$ of registers needed to evaluate a Motzkin tree of size n is then

$$\mathcal{D}_n = \log_4 n + M(\log_4 n) + O\left(\frac{\log^* n}{n}\right), \qquad (n\to\infty),$$

where

$$M(x) = \sum_{k\in\mathbb{Z}} m_k \, e^{2k\pi ix} \text{ with } m_0=-\frac{1}{2}-\frac{\gamma}{2\log 2}-\frac{1}{\log 2}+\log_2 2\pi-\log_2\sqrt{3}$$

and

$$m_k = \frac{1}{\log 2}\, \zeta(\chi_k)\Gamma\left(\frac{\chi_k}{2}\right) (\chi_k-1), \; k\neq 0, \; \chi_k=\frac{2k\pi i}{\log 2}.$$

4. Conclusions

The path we have taken is general enough to enable us to treat the asymptotics of sums of the form

$$S_n = \sum_{k\geq 1} a_k \binom{2n}{n-k} \tag{4}$$

(where instead of binomial coefficients, differences of binomial coefficients may appear), when $\{a_k\}_{k\geq 1}$ is an *arithmetic sequence*, i.e. a sequence such that the Dirichlet generating function

$$\alpha(s) = \sum_{k\geq 1} a_k \, k^{-s}$$

is meromorphic and well enough behaved towards $i\infty$. Such sums appear in the analysis of algorithms in at least the following three cases:

1 - Height of trees [2]

2 - Register allocation [7,12,16]

3 - Odd-even Merge [8,19].

The methods that have been employed to analyse sums of the form (4) are:

A - With the Gaussian approximation of binomial coefficients, replace the study of S_n in (4) by the study of $S^*(1/\sqrt{n})$ where:

$$S^{*}(x) = \sum_{k\geq 1} a_k e^{-k^2x^2} \tag{5}$$

and use Mellin transform techniques to evaluate (5) asymptotically. This is the way taken originally by de Bruijn, Knuth and Rice [2] (problem 1), Kemp [12] (problem 2) and Sedgewick [19] (problem 3).

B - Use real analysis to obtain real expressions for

$$a_k \quad \text{or} \quad A_k^{(1)} = \sum_{j<k} a_j \quad \text{or} \quad A_k^{(2)} = \sum_{j<k} A_j^{(1)} \quad \ldots .$$

Developments based on techniques of Delange constitute the original treatment of register allocation in [7] (problem 2), and have been applied to rederive Sedgewick's solution to problem 3 in [8]. In the context of problem 1, they lead to an elementary derivation of the main terms of the expected height of general trees (this fact has been pointed out to us by L. Guibas).

C - Use singularity analysis of the generating function of the S_n,

$$S(z) = \sum_{n\geq 0} S_n z^n$$

as we have done in this paper. The method has the advantage of allowing rather simply derivation of asymptotic expansions to any order and also generalises easily as we have seen to cases where binomial coefficients are replaced by trinomial coefficients or even more generally to coefficients of powers of some fixed function. It could therefore have been applied to problems 1 and 3 as well; interestingly enough, this is the way Knuth started his partial attack to problem 3 ([14] ex. 5.2.2.16, p.135 and p.607).

REFERENCES

[1] G.ANDREWS, Theory of partitions, Academic Press, New York-London, 1976.

[2] N.G.de BRUIJN, D.E.KNUTH, S.O.RICE, The average height of planted plane trees, in: Graph Theory and Computing (R.C. Read, Ed.), 15-22, Academic Press, New York-London, 1972.

[3] H.DELANGE, Sur la fonction sommatoire de la fonction somme des chiffres, l'Enseignement Mathématique 21 (1975), 31-47.

[4] G.DOETSCH, Handbuch der Laplace Transformation, Birkhäuser,1950.

[5] A.P. ERSHOV , On programming of arithmetic operations, Comm. ACM 1 (1958), 3-6.

[6] P. FLAJOLET, Analyse d'algorithmes de manipulation d'arbres et de fichiers, Cahiers du BURO, 34-35 (1981), 1-209.

[7] P. FLAJOLET, J.C. RAOULT, J. VUILLEMIN The number of registers required for evaluating arithmetical expressions, Theoretical Computer Science 9 (1979), 99-125.

[8] P. FLAJOLET, L. RAMSHAW, A note on Gray code and odd-even merge, SIAM J. on Computing 9 (1980), 142-158.

[9] P. FLAJOLET, A. ODLYZKO, The average height of binary trees and other simple trees, J. Comput. Syst. Sci. 25 (1982), 171-213.

[10] P. FLAJOLET, C. PUECH, Partial retrieval of multidimensional data, INRIA Research Report 233, Aug. 1983. Extended abstract in 24rd IEEE FOCS Symp., Tucson (1983), to appear.

[11] R. JUNGEN, Sur les séries de Taylor n'ayant que des singularités algébrico-logarithmiques sur leur cercle de convergence, Commentarii Math. Helvetici 3 (1931), 266-306.

[12] R. KEMP, The average number of registers to evaluate a binary tree optimally, Acta Inf. 11 (1979), 363-372.

[13] P. KIRSCHENHOFER, On the height of leaves in binary trees, J. Combinatorics, Information and Syst.Sci., to appear.

[14] D.E. KNUTH, The Art of Computer Programming, Vol.3, Addison-Wesley, 1973.

[15] A. MEIR, J.W. MOON, On the altitude of nodes in random trees, Canad. J. Math. 30 (1978) , 997-1015.

[16] A.MEIR, J.W. MOON, J.R. POUNDER, On the order of random channel networks, SIAM J. Alg. Discr. Meth. 1 (1980), 25-33.

[17] A. ODLYZKO, Periodic oscillations of coefficients of power series that satisfy functional equations, Adv. in Math. 44 (1982), 180-205.

[18] M. REGNIER, Evaluation des performances du hachage dynamique, Thèse, Paris-Sud-Orsay, 1983.

[19] R. SEDGEWICK, Data movement in odd-even merging, SIAM J. on Computing 7 (1978), 239-272.

[20] I. SNEDDON, The Use of Integral Transforms, Mc Graw Hill, 1952.

[21] E.T. WHITTAKER, G.N. WATSON, A course of modern analysis, Cambridge University Press, 1927.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique