SIAM J. COMPUT.
Vol. 15, No. 3, August 1986

001

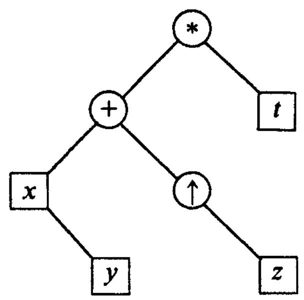
REGISTER ALLOCATION FOR UNARY-BINARY TREES*

P. FLAJOLET† AND H. PRODINGER‡

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 first by 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.

Key words. analysis of algorithms, register allocation, random trees, Mellin transform

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



The problem of *register allocation* consists in finding an *evaluation strategy* for arithmetic expressions using only binary operations applied to elements of an array called *registers*. For the above expression with registers being an array $R[0]$, $R[1]$, $\cdots$ a possible evaluation strategy is

$$\begin{aligned}
&R[0]\leftarrow x\\
&R[1]\leftarrow y\\
&R[2]\leftarrow z\\
&R[1]\leftarrow R[1]\uparrow R[2]\\
&R[0]\leftarrow R[0]+R[1]\\
&R[1]\leftarrow t\\
&R[0]\leftarrow R[0]*R[1]
\end{aligned}$$

There is an optimal strategy with respect to the number of registers used. That strategy has been found by Ershov as early as 1958 [5] and is described by Sethi and Ullman in [23]. 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:

$$\operatorname{Reg}(\square)=0,$$

$$\operatorname{Reg}\left(\begin{smallmatrix} & \bigcirc & \\ \diagup & & \diagdown \\ t_1 & & t_2\end{smallmatrix}\right)=\begin{cases}1+\operatorname{Reg}(t_1) & \text{if } \operatorname{Reg}(t_1)=\operatorname{Reg}(t_2),\\ \max\{\operatorname{Reg}(t_1),\operatorname{Reg}(t_2)\} & \text{otherwise.}\end{cases}$$

The average number D_n of registers needed to evaluate a binary tree of size n (i.e. n internal nodes) assuming that all binary trees of size n are equally likely is a

* Received by the editors January 5, 1984, and in revised form February 19, 1985.
† INRIA, Rocquencourt, 78150 Le Chesnay, France.
‡ Technical University Vienna, Gußhausstraße 27-29, A-1040 Vienna, Austria.

well studied quantity [7], [12], [16]. It satisfies

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

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.

Then we show that this approach extends easily to more general classes of trees: assume that unary operations like $-$, sin, exp, log, etc, $\cdots$, are also permitted. There we have to deal with unary-binary trees, possibly with weights, according to the number of unary and binary operations allowed. (See § 3 for precise definitions.)

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 $\mathrm{Reg}\,(t)$ is defined inductively by:

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

$$\mathrm{Reg}\left(\begin{smallmatrix}\circ\\ |\\ t\end{smallmatrix}\right) = \mathrm{Reg}\,(t),$$

$$\mathrm{Reg}\left(\begin{smallmatrix} & \circ & \\ \diagup & & \diagdown \\ t_1 & & t_2\end{smallmatrix}\right) = \begin{cases} 1+\mathrm{Reg}\,(t_1) & \text{if } \mathrm{Reg}\,(t_1)=\mathrm{Reg}\,(t_2), \\ \max\{\mathrm{Reg}\,(t_1), \mathrm{Reg}\,(t_2)\} & \text{otherwise.}\end{cases}$$

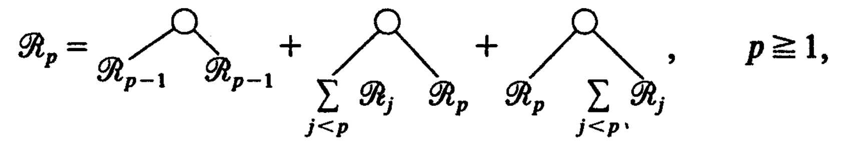
In § 3 we consider the average number of registers needed to evaluate a unary-binary tree. The analysis that we develop for binary trees (§ 2) can be translated to this more general case since the unary-binary trees are obtained from the binary trees by a simple substitution operation. As a consequence, all the generating functions needed for the analysis are obtained from the corresponding ones for binary trees via a simple substitution.

The *singularity analysis* that we are going to use in this paper is based on an extension to complex arguments of the *Mellin transform* inversion theorem. It can be applied to several problems in the analysis of algorithms. We mention height of trees [2], register allocation [7], [12], [16] and odd-even merge [8], [19]. The advantage is that asymptotic expansions to any order can be derived rather simply, as the generating functions are usually much easier to approximate in a neighbourhood of their singularities than their Taylor coefficients; also Mellin transform techniques constitute a rather powerful tool when dealing with number theoretic functions (here the dyadic valuation).

2. The register function of binary trees revisited. In order to rederive the formula for D_n, we need several generating functions, which can be most easily obtained by a simple translation from so-called symbolic equations [6]: If A and B are families of trees, then we write $\begin{smallmatrix} & \circ & \\ \diagup & & \diagdown \\ A & & B\end{smallmatrix}$ for the set of all trees consisting of a root, a left subtree $t_1 \in A$ and a right subtree $t_2 \in B$. The family $\mathcal{B}$ of binary trees is then described by the symbolic equation

$$\mathcal{B} = \square + \begin{smallmatrix} & \circ & \\ \diagup & & \diagdown \\ \mathcal{B} & & \mathcal{B}\end{smallmatrix}.$$

If we define the family $\mathcal{R}_p$ to be the family of all binary trees t with $\operatorname{Reg}(t)=p$, then the definition of the register function easily carries over to:



$\mathcal{R}_0 = \square.$

Let $R_p(z)$ denote the generating function of the family $\mathcal{R}_p$, i.e.

$$R_p(z) = \sum_{t \in \mathcal{R}_p} z^{\operatorname{size}(t)}.$$

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 ith Fibonacci polynomial:

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

The generating function of the cumulated register values is

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

The sought average D_n is then

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

where

$$B(z) = \sum_{t \in \mathcal{B}} z^{\operatorname{size}(t)}$$

is the generating function of all binary trees and $[z^n]f$ denotes the nth Taylor coefficient of the power series f.

From the defining equation for $\mathcal{B}$ one obtains immediately

$$B(z) = 1 + z(B(z))^2, \quad \text{or } B = (1 - r(z))/2z.$$

Using the substitution [2]

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

we easily find

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

where $v_2(k)$ is the dyadic valuation 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.

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

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

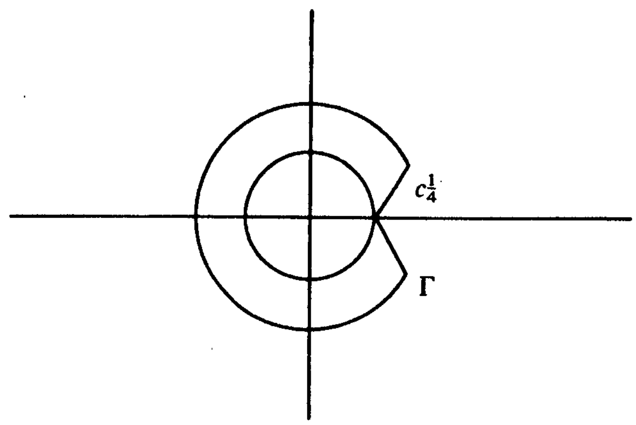


FIG. 1

To be more precise, let $0<\theta<\pi/2$, $\omega>0$ and $\rho>\frac{1}{4}$. Then $\Gamma=\Gamma_0\cup\Gamma_1\cup\Gamma_2$ with

$$\Gamma_0=\{z: |z-\tfrac{1}{4}|=\omega, |\mathrm{Arg}\,(z-\tfrac{1}{4})|>\theta\},$$
$$\Gamma_1=\{z: |z-\tfrac{1}{4}|\geqq\omega, |z|<\rho, |\mathrm{Arg}\,(z-\tfrac{1}{4})|=\theta\},$$
$$\Gamma_2=\{z: |z|=\rho, |\mathrm{Arg}\,(z-\tfrac{1}{4})|\geqq\theta\}.$$

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 developed in [9], we can, provided we have an approximation of $E(z)$ about $\frac{1}{4}$, "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 of obtaining an expansion of the form:

$$E(z)\sim\alpha\cdot r\cdot\log r+\beta\cdot r+\cdots,$$

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

Now, since

$$\sum_{k\geqq1} v_2(k)u^k=\frac{u^2}{1-u^2}+\frac{u^4}{1-u^4}+\frac{u^8}{1-u^8}+\cdots,$$

$z \sim \frac{u}{(1+u)^2}$

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 $\mathrm{Re}\,(z)\geqq\frac{1}{4}$, $\mathrm{Im}\,(z)=0$, 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 origin without crossing the halfray and in particular we can take the contour Γ of Fig. 1.

What remains to do is thus to find a local expansion of $E(z)$ about $z=\frac{1}{4}$. 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

$$V(t)=\sum_{k\geqq1} v_2(k)\,e^{-kt},\qquad V^*(s)=\int_0^\infty x^{s-1}V(x)\,dx.$$

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

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

and so

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

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

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

a reference for this is for example [1, p. 91].

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

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

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

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

By using local expansions of $\Gamma(s)$, $\zeta(s)$, $(2^s - 1)^{-1}$ and t^{-s} we find the residue at $s = 0$, resp. at $s = \chi_k := 2k\pi i/\log 2$:

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

and the residue at $s = \chi_k$.

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

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

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

with

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

Now $e^{-t} = u$ and $u = (1-r)/(1+r)$ with $r = \sqrt{1-4z}$; thus

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

yielding

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

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]:

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

$$[z^n]\log(1-z)\cdot(1-z)^\alpha=\frac{-n^{-\alpha-1}\log n}{\Gamma(-\alpha)}+\frac{n^{-\alpha-1}}{(\Gamma(-\alpha))^2}\Gamma'(-\alpha)+O\left(\frac{\log^* n}{n^{\alpha+2}}\right).$$

Using known values of $\Gamma(-\frac{1}{2})$ and $\Gamma'(-\frac{1}{2})$ this gives us

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

Hence

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

Using the duplication formula for the gamma function [21], we can simplify:

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

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

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

$$D_n=\log_4 n+D(\log_4 n)+O\left(\frac{\log^* n}{n}\right),$$ asympt. exp. ?

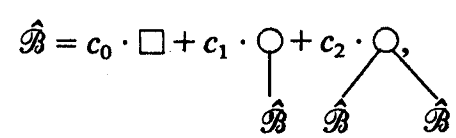
where $D(x)$ is a periodic function with period 1. *This function can be expanded as a convergent Fourier series $D(x)=\sum_{k\in Z} d_k\, e^{2k\pi i x}$, and*

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

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

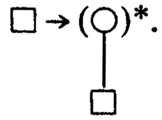
Remark that the constant d_0 was erroneously stated in [7].

3. The register function of unary–binary trees. The symbolic equation:

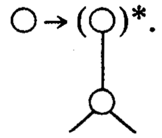


describes a family of unary-binary trees, where the weights fulfill $c_0>0$, $c_1\geqq 0$, $c_2>0$. The interpretation is that we have c_0 different types of nullary nodes, c_1 unary nodes and c_2 binary nodes. For example, if the set of operators and variables is $\{x, y, t, \pi; \sqrt{}, \log, \sin; +, \times\}$, then $c_0=4$, $c_1=3$ and $c_2=2$.

We can obtain $\hat{\mathscr{B}}$ from the family $\mathscr{B}$ 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 $\mathscr{B}$, where y marks a leaf and z marks an internal node. Thus the corresponding series for $\hat{\mathscr{B}}$ is obtained by the substitutions

$$y \to \frac{c_0 y}{1-c_1 z}, \qquad z \to \frac{c_2 z}{1-c_1 z}. \tag{2}$$

Let $\hat{B}$ be the generating function of the trees in $\hat{\mathscr{B}}$ and $\hat{R}_p$ be the generating function of the trees in $\hat{\mathscr{R}}_p$, i.e. the trees in $\hat{\mathscr{B}}$ with register function $=p$. Since the substitutions do not change the register function of the involved trees, we can find $\hat{\mathscr{R}}_p$ from $\mathscr{R}_p$ by the substitutions (2).

We can define the size of a tree in $\hat{\mathscr{B}}$ 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

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

while (2) corresponds to:

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

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

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

where c_0, $c_0' \geqq 0$ 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

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

and in

$$B(z) = 2 - 2r + O(r^2).$$

We are interested in

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

Since the factor $(c_0z+c_0')/(1-c_1z)$ appears both in the numerator and the denominator and is regular at the dominant singularity of $\hat{\mathscr{B}}$, we can write

$$\hat{D}_n=\left[[z^n]E\left(\frac{(c_0z+c_0')c_2z}{(1-c_1z)^2}\right)\right]\Big/\left[[z^n]B\left(\frac{(c_0z+c_0')c_2z}{(1-c_1z)^2}\right)\right]\left(1+O\left(\frac{1}{n}\right)\right).$$

Let $\varphi(z)=(c_0z+c_0')c_2z/(1-c_1z)^2$. 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 $\frac{1}{4}$ plays in the case of binary trees:

$$r(\varphi(z))=\frac{1}{1-c_1z}\sqrt{1-(2c_1+4c_0'c_2)z+(c_1^2-4c_0c_2)z^2};$$

σ is one of the solutions s_1, s_2 of

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

$$s_{1,2}=\frac{c_1+2c_0'c_2\pm2\sqrt{c_2}\cdot\sqrt{c_0'c_1+c_0'^2c_2+c_0}}{c_1^2-4c_0c_2}.$$

(a) We assume first that $c_1^2\neq4c_0c_2$ 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

$$\begin{aligned}
&c_1+2c_0'c_2-2\sqrt{c_2}\sqrt{c_0'c_1+c_0'^2c_2+c_0}>0\\
&\Leftrightarrow c_1^2+4c_0'c_1c_2+4c_0'^2c_2^2>4c_0'c_1c_2+4c_0'^2c_2^2+4c_0c_2\\
&\Leftrightarrow c_1^2>4c_0c_2.
\end{aligned}$$

So we have

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

and thus as $z\to\sigma$

$$\begin{aligned}
(c_1^2-4c_0c_2)z^2-(2c_1+4c_0'c_2)z+1&\sim(c_1^2-4c_0c_2)(z-\sigma)(\sigma-\bar{\sigma})\\
&=4\sqrt{c_2}\sqrt{c_0'c_1+c_0'^2c_2+c_0}\cdot\sigma\cdot\left(1-\frac{z}{\sigma}\right).
\end{aligned}$$

Hence

$$r(\varphi(z))\sim\frac{1}{1-c_1\sigma}\cdot2\sqrt{\sigma}\cdot c_2^{1/4}(c_0'c_1+c_0'^2c_2+c_0)^{1/4}\sqrt{1-\frac{z}{\sigma}}=:A\cdot\sqrt{1-\frac{z}{\sigma}}.$$

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

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

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.

In order to compute $\hat{D}_n$ up to a relative error of $O(1/n)$, we can use

$$\begin{aligned}
&\left[[z^n]\left(2A\sqrt{1-\frac{z}{\sigma}}\log_2\left(A\sqrt{1-\frac{z}{\sigma}}\right)+2(K+1)A\sqrt{1-\frac{z}{\sigma}}\right.\right.\\
&\qquad\qquad\left.\left.+4\sum_{k\neq 0}c_k\left(A\sqrt{1-\frac{z}{\sigma}}\right)^{1-\chi_k}\right)\right]\Big/\\
&\quad\cdot\left[[z^n]-2A\sqrt{1-\frac{z}{\sigma}}\right]\\
&=\left[[z^n]\frac{1}{2\log 2}\sqrt{1-\frac{z}{\sigma}}\log\left(1-\frac{z}{\sigma}\right)+(K+1+\log_2 A)\sqrt{1-\frac{z}{\sigma}}\right.\\
&\qquad\qquad\left.+2\sum_{k\neq 0}c_k\left(\sqrt{1-\frac{z}{\sigma}}\right)^{1-\chi_k}A^{-\chi_k}\right]\Big/\\
&\quad\cdot\left[[z^n]-\sqrt{1-\frac{z}{\sigma}}\right]\\
&=\log_4 n+D(\log_4 n-\log_2 A)-\log_2 A.
\end{aligned}$$

If we consider (according to case (c))

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

this is, up to a relative error of $O(1/n)$, equal to

$$\log_4 n+D(\log_4 n-\tfrac{1}{2}-\log_2 A)-\tfrac{1}{2}-\log_2 A.$$

This leads us to our main theorem.

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

$$\hat{\mathcal{B}}=c_0\cdot\square+c_1\cdot\underset{\hat{\mathcal{B}}}{\overset{\bigcirc}{|}}+c_2\cdot\underset{\hat{\mathcal{B}}\quad\hat{\mathcal{B}}}{\overset{\bigcirc}{\diagup\diagdown}},\qquad c_0>0,\quad c_2>0,\quad c_1\geqq 0,$$

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

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

then

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

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

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

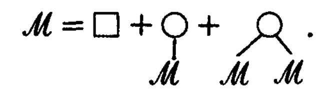
Then

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

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

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

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



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 $\hat{D}_n$ of registers needed to evaluate a Motzkin tree of size n is then

$$\hat{D}_n = \log_4 n + D\left(\log_4 n - \frac{1}{2}\log_2 3\right) - \frac{1}{2}\log_2 3 + O\left(\frac{\log^* n}{n}\right), \qquad (n \to \infty),$$

where $D(x)$ is the periodic function of Theorem 1. We may mention that

$$\log_4 n - \tfrac{1}{2}\log_2 3 = \log_4 \tfrac{4}{3} n.$$

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

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

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

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

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:

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

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

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

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 [8] (problem 3). 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,

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

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].

Acknowledgment. We would like to thank one of the referees for pointing out a reference to a further paper dealing with register problems [22].

REFERENCES

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

[2] 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., Academic Press, New York-London, 1972, pp. 15–22.

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

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

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

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

[7] P. FLAJOLET, J. C. RAOULT AND J. VUILLEMIN, *The number of registers required for evaluating arithmetical expressions*, Theoret. Comput. Science, 9 (1979), pp. 99–125.

[8] P. FLAJOLET AND L. RAMSHAW, *A note on Gray code and odd-even merge*, this Journal, 9 (1980), pp. 142–158.

[9] P. FLAJOLET AND A. ODLYZKO, *The average height of binary trees and other simple trees*, J. Comput. System Sci., 25 (1982), pp. 171–213.

[10] P. FLAJOLET AND C. PUECH, *Partial match retrieval of multidimensional data*, INRIA Research Report 233, Aug. 1983, extended abstract in 24th IEEE Symposium on Foundations of Computer Science, Tucson, AZ, 1983, pp. 282–288.

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

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

[13] P. KIRSCHENHOFER, *On the height of leaves in binary trees*, J. Combin. Inform. Syst. Sci., 8 (1983), pp. 44–60.

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

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

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

[17] A. ODLYZKO, *Periodic oscillations of coefficients of power series that satisfy functional equations*, Adv. Math., 44 (1982), pp. 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*, this Journal, 7 (1978), pp. 239–272.

[20] I. SNEDDON, *The Use of Integral Transforms*, McGraw-Hill, New York, 1952.

[21] E. T. WHITTAKER AND G. N. WATSON, *A Course of Modern Analysis*, Cambridge Univ. Press, Cambridge, 1927.

[22] R. KEMP, *The reduction of binary trees by means of an input-restricted deque*, RAIRO Inform. Theor., 17 (1983), pp. 249–284.

[23] R. SETHI AND J. D. ULLMAN, *The generation of optimal code for arithmetic expressions*, J. Assoc. Comput. Mach., 17 (1970), pp. 715–728.