

JEAN FRANÇON

Sur le nombre de registres nécessaires à l'évaluation d'une expression arithmétique

RAIRO. Informatique théorique, tome 18, n° 4 (1984), p. 355-364

http://www.numdam.org/item?id=ITA_1984__18_4_355_0

© AFCET, 1984, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Informatique théorique » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

SUR LE NOMBRE DE REGISTRES NÉCESSAIRES A L'ÉVALUATION D'UNE EXPRESSION ARITHMÉTIQUE (*)

par Jean FRANÇON ⁽¹⁾

Communiqué par R. CORI

Résumé. — *La distribution de la variable « nombre de registres nécessaires à l'évaluation d'une expression arithmétique » sur les arbres binaires est étudiée de façon purement combinatoire. Une nouvelle relation de récurrence est démontrée. Cette relation permet une preuve nouvelle et sans calcul du théorème de Flajolet, Raoult et Vuillemin liant la distribution en question à celle de la « hauteur à gauche » d'un arbre binaire.*

Abstract. — *The distribution of the variable "number of registers required for evaluating an arithmetic expression" on binary trees is studied in a purely combinatorial way. A new recurrence relation is proved. This relation gets a new proof, without calculations, of the theorem of Flajolet, Raoult and Vuillemin linking this distribution to the distribution of the "left height" of a binary tree.*

1. INTRODUCTION

On sait qu'une expression arithmétique dont tous les opérateurs sont binaires est représentable par un arbre binaire (dans la terminologie de [10] section 2.3, p. 305, que nous utilisons dans cette note). On sait aussi, depuis les travaux d'Ershov [1] et de Sethi et Ullman [12], que le nombre minimum de registres de mémoire nécessaire à l'évaluation d'une expression arithmétique, dans les modèles courants de calcul, est une variable d'arbre binaire à valeurs entières, notée ici Reg , qu'on peut définir ainsi :

si b est l'arbre binaire vide, alors $\text{Reg}(b)=0$, sinon, en notant b_1 (resp. b_2) le sous-arbre à gauche (resp. droite) de b ,

Si $\text{Reg}(b_1)=\text{Reg}(b_2)$, alors $\text{Reg}(b)=1+\text{Reg}(b_1)$, sinon

$$\text{Reg}(b)=\max(\text{Reg}(b_1), \text{Reg}(b_2)).$$

Nous renvoyons le lecteur à [5] ou [3] pour plus de détails concernant l'intérêt informatique de cette variable.

(*) Reçu en février 1983, révisé en octobre 1983.

(1) I.S.E.A., Université de Haute Alsace, 4, rue des Frères Lumière, 68093 Mulhouse.

Exemple: La figure 1.1 représente un arbre binaire et à côté de chaque sommet est inscrite la valeur de Reg du sous-arbre enraciné en ce sommet.

On pourrait baptiser cette variable Reg « nombre de Strahler » d'un arbre binaire, du nom du géomorphologiste qui l'inventa [14], modifiant une première idée de [8]. On trouvera dans [3, 5 et 13] (pages 113 et 225 de la traduction française), des références à des travaux montrant l'intérêt de cette variable non seulement en géomorphologie mais aussi en botanique et en anatomie.

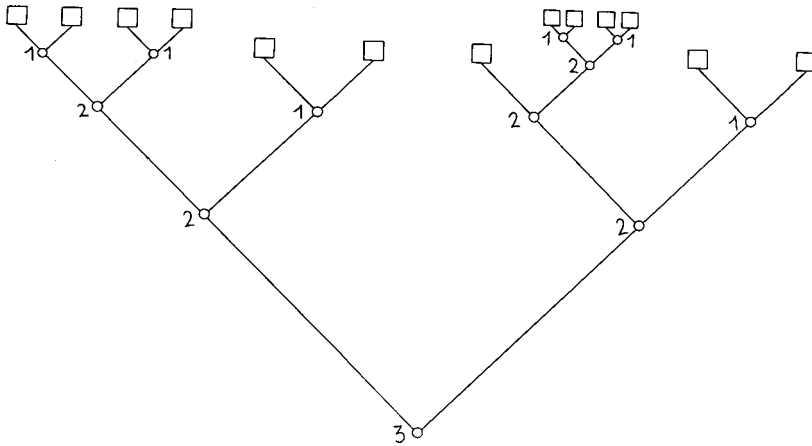


Figure 1.1.

Une autre variable d'arbre binaire à valeurs entières est classique en combinatoire et apparaît souvent en informatique, par exemple lors d'un parcours d'arbre; c'est celle de « hauteur à gauche », notée Hag et ainsi définie :

si b est l'arbre binaire vide, alors $\text{Hag}(b)=0$, sinon, soient b_1 (resp. b_2) le sous-arbre à gauche (resp. droite) de b , alors

$$\text{Hag}(b) = \max(\text{Hag}(b_2), 1 + \text{Hag}(b_1)).$$

Par exemple, la hauteur à gauche de l'arbre de la figure 1.1 vaut 4, celle de son sous-arbre droit vaut 3 ainsi que celle de son sous-arbre gauche.

Un résultat essentiel de [5], démontré indépendamment dans [9], peut s'énoncer ainsi :

THÉORÈME 1.1: *Les variables Reg et $\lfloor \log_2(1 + \text{Hag}) \rfloor$ (*) ont même distribution sur l'ensemble des arbres binaires de taille donnée.*

(*) On note $\lfloor x \rfloor$, pour un réel x , le plus grand entier inférieur ou égal à x .

En d'autres termes : notons B_n l'ensemble des arbres binaires de taille (nombre de sommets internes) n , pour tout entier $n \geq 0$; notons $R_{p,n}$ le nombre d'arbres $b \in B_n$ tels que $\text{Reg}(b) = p$, pour tout entier $p \geq 0$; notons $H_{k,n}$ le nombre d'arbres $b \in B_n$ tels que $\text{Hag}(b) = k$, k entier ≥ 0 ; alors on a

$$R_{p,n} = \sum H_{k,n}$$

où la somme s'étend aux entiers k tels que $p = \lceil \log_2(1+k) \rceil$. Nous appelons les entiers $R_{p,n}$ *nombre de Strahler*.

On peut constater ce fait sur les tables des premières valeurs des nombres $R_{p,n}$ et $H_{k,n}$ ci-dessous : la première colonne des $R_{p,n}$ est égale à la somme des deux premières colonnes des $H_{k,n}$ et la colonne $R_{2,n}$ est la somme des colonnes restantes, soit $H_{3,n}$, $H_{4,n}$ et $H_{5,n}$, puisque les tables sont limitées à $n = 5$.

TABLE 1.1.
 $R_{p,n}$ (tiré de [5])

$n \backslash p$	1	2
1	1	
2	2	
3	4	1
4	8	6
5	16	26

TABLE 2.1.
 $H_{k,n}$ (par dénombrement direct)

$n \backslash k$	1	2	3	4	5
1	1				
2	1	1			
3	1	3	1		
4	1	7	5	1	
5	1	15	18	7	1

Ce théorème est surprenant. En effet, la variable Reg est évidemment invariante par échange des deux sous-arbres d'un arbre binaire, alors que la variable Hag ne l'est en général pas. Cette remarque nous a conduit à examiner de plus près la combinatoire des arbres binaires. L'objet de cette note est de montrer d'abord qu'une construction combinatoire tout à fait élémentaire permet de trouver une nouvelle relation de récurrence pour les nombres de Strahler (section 2), puis (section 3) de montrer que des constructions soit très classiques, soit élémentaires permettent d'obtenir la même relation de récurrence pour la distribution de la variable $\lceil \log_2(1 + \text{Hag}) \rceil$, fournissant ainsi une nouvelle preuve du théorème 1.1.

Construire une bijection (suffisamment simple) de l'ensemble des arbres binaires de taille donnée sur lui-même transformant la variable Reg en la variable $\lceil \log_2(1 + \text{Hag}) \rceil$ reste toutefois un problème ouvert.

2. UNE RELATION DE RÉCURRENCE POUR LES NOMBRES DE STRAHLER

Appelons *point simple* d'un arbre binaire tout sommet dont un seul fils est vide (c'est-à-dire sommet externe) et *feuille* tout sommet dont les deux fils sont vides.

Nous définissons maintenant une opération appelée *suppression des points simples*. Soit α un point simple d'un arbre binaire b , β son père (on admet un père fictif si α est racine de b) et γ son unique fils non vide ; supprimer α dans b consiste à opérer comme le ferait un programmeur sur la représentation classique en informatique, d'un arbre binaire avec deux pointeurs par sommet interne (cf. [10] section 2.3.1), c'est-à-dire remplacer dans b le sommet α , en tant que fils de β , par γ . Ainsi l'arbre de la figure 2.1 (a) n'a qu'un seul point simple ; supprimer ce point simple produit l'arbre de la figure 2.1 (b).

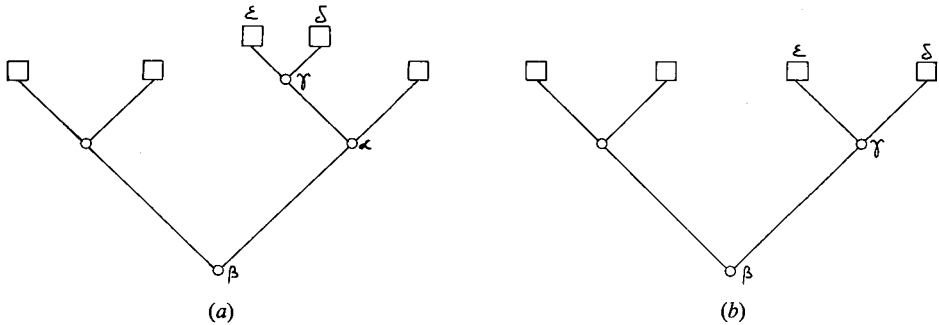


Fig. 2.1. — Suppression d'un point simple.

Remarquons de plus qu'on peut procéder à la suppression de *tous* les points simples d'un arbre binaire en les supprimant un à un, le résultat final étant insensible à l'ordre. Remarquons enfin que le nombre $\text{Reg}(b)$ est inchangé après suppression d'un point simple de b . Énonçons :

LEMME 2.1 : *La suppression des points simples laisse invariante la variable Reg.*

Définissons maintenant la *suppression des feuilles* d'un arbre binaire. La suppression d'une feuille consiste à remplacer cette feuille par un sommet externe ; par exemple, la suppression de la feuille γ dans l'arbre de la figure 2.1 (a) fournit l'arbre de la figure 2.1 (b). La suppression de *toutes* les feuilles d'un arbre binaire peut se faire dans n'importe quel ordre sans affecter le résultat final. Quand on observe l'effet sur la variable Reg de la suppression des feuilles, on obtient aisément le

LEMME 2.2: Soit b un arbre binaire de taille non nulle ; soit b' l'arbre binaire obtenu à partir de b en procédant à une suppression des points simples et b'' l'arbre binaire obtenu en procédant à une suppression des feuilles de b' ; alors on a

$$\text{Reg}(b'') = \text{Reg}(b) - 1.$$

On peut donc caractériser le nombre $\text{Reg}(b)$ comme le nombre d'itérations de l'opération suppression des points simples suivie de l'opération suppression des feuilles pour réduire b à l'arbre vide.

Notons que si l'arbre b du lemme 2.2 est de taille $n > 0$ et a $s \geq 0$ points simples, alors l'arbre b' est de taille $n - s$; b' a donc $\frac{n-s-1}{2} + 1$ feuilles ; b'' est donc de taille $\frac{n-s-1}{2}$

Considérons maintenant un arbre $a \in B_m$, avec $m \geq 0$, tel que $\text{Reg}(a) = p - 1$, avec $p \geq 1$. Le nombre d'arbres $b \in B_n$, avec $n > m$, tels que la suppression des points simples de b suivie de la suppression des feuilles de l'arbre résultant fournit a , est égal à

$$\binom{n-1}{2m} 2^{n-2m-1}$$

le coefficient binomial étant le nombre de façons de rajouter $n - 1 - 2m$ points simples à a , la puissance de 2 étant le nombre de façons de choisir la direction (à droite ou à gauche) du fils vide d'un point simple.

On obtient donc la récurrence :

THÉORÈME 2.3: Les nombres de Strahler vérifient pour $p > 0$ et $n > l'$ égalité

$$R_{p,n} = \sum_{0 \leq n \leq \lfloor \frac{n-1}{2} \rfloor} \binom{n-1}{2m} 2^{n-m-1} R_{p-1,m}$$

avec $R_{0,n} = R_{p,0} = 0$ sauf $R_{0,0} = 1$.

Remarquons encore que cette récurrence est équivalente, pour la série génératrice,

$$R(u, x) = \sum_{n \geq 0} u^n \sum_{p \geq 0} R_{p,n} x^p$$

à l'équation fonctionnelle

$$R(u, x) = 1 + x \frac{u}{1-2u} R\left(\left(\frac{u}{1-2u}\right)^2, x\right).$$

3. PREUVE DU THÉORÈME 1 EN PASSANT PAR LES CHEMINS

3.0. Rappels

Les informaticiens sont familiers des parcours d'arbres binaires à l'aide d'une pile (voir [10] section 2.3.1). On sait que, pour plusieurs algorithmes de parcours, la suite des hauteurs de pile, appelée ci-après un « chemin », est caractéristique de l'arbre parcouru et que certaines propriétés d'arbre se transposent bien sur les chemins. Nous allons considérer deux codages d'arbres binaires par des chemins, résultant de deux façons différentes d'empiler les sommets d'un arbre binaire quand on le parcourt en ordre préfixé. Des constructions élémentaires permettent alors de montrer que la distribution de la variable $[\text{Log}_2(1 + \text{Hag})]$ vérifie la même relation de récurrence que celle des nombres de Strahler (Théorème 3.2).

Nous appelons *chemin de longueur n* toute suite $(c(1), c(2), \dots, (c(n)))$ d'entiers tels que $c(1) = c(n) = 1$ et $|c(i) - c(i+1)| \leq 1$ pour $i = 1, 2, \dots, n-1$; nous appelons *chemin strict* un chemin pour lequel l'inégalité précédente est une égalité pour tout i , *chemin large* sinon. Pour tout chemin c de longueur n nous posons

$$\text{Haut}(c) = \max_{1 \leq i \leq n} c(i)$$

que nous appelons la *hauteur* de c .

3.1. Chemin de Dyck d'un arbre binaire

Numérotons en ordre préfixe les sommets internes et externes d'un arbre binaire b de taille n par les entiers $1, 2, \dots, 2n+1$; associons à l'arbre b le chemin $c = D(b)$, que nous appelons *chemin de Dyck* de b , de la façon suivante : on pose pour $c(1) = 1$ et si le sommet de b de numéro i est interne on pose $c(i+1) = c(i) + 1$, sinon on pose $c(i+1) = c(i) - 1$ pour $i = 1, 2, \dots, 2n$. L'énoncé suivant est à considérer comme classique, bien qu'il ne soit écrit nulle part explicitement (une bibliographie apparaît dans [6]).

THÉORÈME 3.1: *Pour tout entier n positif, l'application qui à tout arbre binaire $b \in B_n$ associe le chemin de Dyck $c = D(b)$ est une bijection de B_n sur l'ensemble des chemins stricts de longueur $2n+1$; de plus on a*

$$\text{Haut}(c) = 1 + \text{Hag}(b).$$

3.2. Chemin coloré codant un arbre binaire

Appelons *palier* d'un chemin c tout couple $(c(i), c(i+1))$, $1 \leq i < n$, tel que $c(i) = c(i+1)$. Nous appelons *chemin coloré* un chemin large dont tous les

paliers sont « colorés » en une « couleur » prise dans l'alphabet $\{ +, - \}$. La figure 3.1 représente le chemin coloré de longueur 9 ayant 3 paliers colorés par + et 1 palier coloré par - :

$$c(1) = 1, c(2) = c(3) = 2, c(4) = c(5) = c(6) = 3, c(7) = 2, c(8) = c(9) = 1.$$

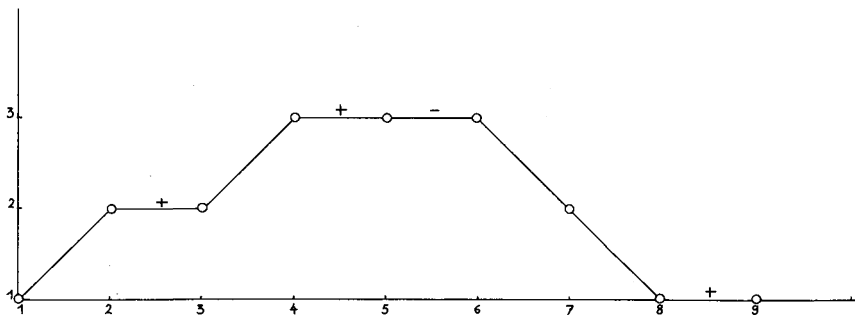


Figure 3.1.

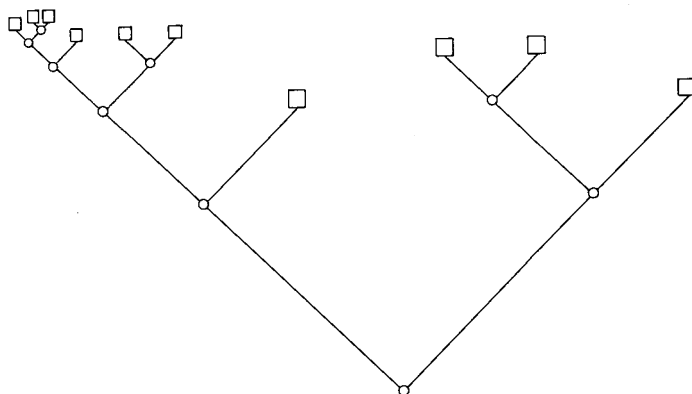


Figure 3.2.

Le codage suivant est une reformulation du théorème A p. 326 de [10]. Numérotons les sommets internes d'un arbre binaire b de taille n par les entiers $1, 2, \dots, n$ en ordre préfixe (d'autres ordres peuvent convenir aussi, voir [6] et [7]). Associons à b le chemin $c = \pi(b)$ de la façon suivante : pour $i = 1, 2, \dots, n-1$ posons

- $c(i+1) = c(i) + 1$ si les deux fils du sommet i sont internes
- $c(i+1) = c(i) - 1$ si les deux fils du sommet i sont externes (i est feuille)
- $c(i+1) = c(i)$ sinon et ce palier est coloré par + (resp. -)
si le fils à droite (resp. gauche) du sommet i est vide.

Par exemple, l'arbre b dont le chemin de la figure 3.1 est le codage $\pi(b)$ est celui de la figure 3.2.

On se convainc facilement que l'application π est bien un codage, c'est-à-dire

THÉORÈME 3.2: *Pour tout entier $n > 0$, l'application π qui à tout arbre binaire $b \in B_n$ associe le chemin coloré $c = \pi(b)$ est une bijection de B_n sur l'ensemble des chemins colorés de longueur n .*

Considérons maintenant l'opération de *suppression des paliers* d'un chemin coloré. Supprimer les paliers d'un chemin large $c(1), c(2), \dots, c(n)$ consiste à prendre la plus grande sous-suite $c(i_1), c(i_2), \dots, c(i_n)$ qui ne contient aucun palier. Ainsi, supprimer les paliers du chemin de la figure 3.1 fournit le chemin strict $c'(1)=1, c'(2)=2, c'(3)=3, c'(4)=2, c'(5)=1$.

PROPRIÉTÉ 3.3: *Étant donné un chemin strict $d \in C_{2m+1}$, $m > 0$, le nombre de chemins colorés de longueur n , n entier donné $\geq 2m+1$, tels que d est obtenu en supprimant les paliers de c est égal à*

$$\binom{n-1}{2m} 2^{n-1-2m}$$

Preuve: Le binomial dénombre les façons de rajouter à d $n-1-2m$ paliers pour obtenir un chemin large de longueur n et la puissance de 2 est le nombre de façons de colorer ces paliers. Q.E.D.

Observons aussi trivialement :

PROPRIÉTÉ 3.4: *La suppression des paliers conserve la hauteur de tout chemin.*

3.3. La contraction d'un chemin strict en un chemin coloré

On appelle contraction l'application qui à tout chemin c strict de longueur $2n+1$ fait correspondre le chemin coloré c' de longueur n ainsi construit : pour tout $i=1, 2, \dots, n$ posons

$$c'(i) = \frac{1}{2} c(2i)$$

et si $c'(i) = c'(i+1)$ ce palier de c' est coloré par $+$ quand $c(2i+1) = c(2i) + 1$ et par $-$ quand $c(2i+1) = c(2i) - 1$.

On vérifie aisément le

THÉORÈME 3.5: *La contraction est une bijection de l'ensemble des chemins stricts de longueur $2n+1$ sur l'ensemble des chemins colorés de longueur n ; de plus on a*

$$\text{Haut}(c') = \left\lceil \frac{\text{Haut}(c)}{2} \right\rceil.$$

Par exemple, le chemin coloré de la figure 3.1, de hauteur 3, est obtenu par contraction du chemin strict de la figure 3.3, de hauteur 7.

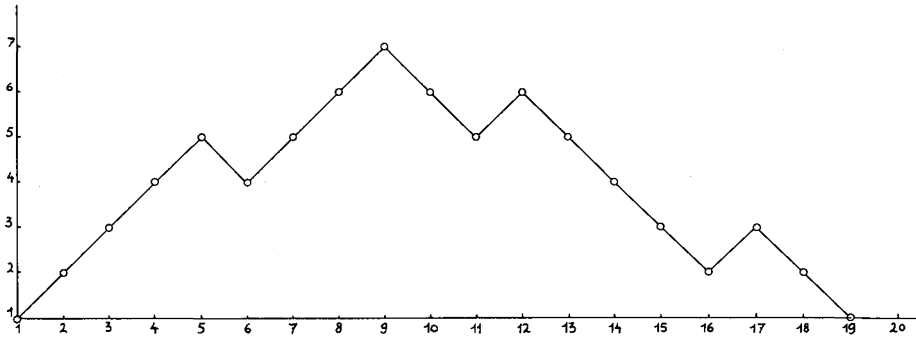


Figure 3.3.

On peut montrer ([2, 3]) que la série génératrice des chemins stricts comptés par longueur, ainsi que celle des chemins colorés, admettent un développement en fraction continue, et que ([4]) le théorème 3.5 se traduit par l'égalité de deux fractions continues, la seconde étant obtenue à partir de la première par une opération que Perron [11] p. 200 appelle « contraction ».

Remarquons enfin qu'en composant une contraction et une suppression de paliers, tout chemin c strict est transformé en un nouveau chemin strict, et qu'en itérant cette transformation k fois, avec $k = \lfloor \text{Log}_2(\text{Haut}(c)) \rfloor$, on obtient l'unique chemin de longueur 1.

3.4. Preuve du théorème 1.1

Il nous suffit de montrer que la distribution de la variable $\lfloor \text{Log}_2(1 + \text{Hag}) \rfloor$ sur B_n , c'est-à-dire les arbres binaires de taille n , vérifie la récurrence du théorème 2.3. Par le théorème 3.1, il suffit de montrer que la distribution de la variable $\lfloor \text{Log}_2 \text{Haut} \rfloor$ sur les chemins stricts de taille $2n+1$ vérifie ladite récurrence. Le théorème 3.5 indique que la variable $\lfloor \text{Log}_2 \text{Haut} \rfloor$ diminue de 1 par contraction et la propriété 3.4 que cette variable ne change pas par suppression des paliers. En d'autres termes et en notant $C_{n,k}$ le nombre de chemins stricts c de longueur $2n+1$ tels que $\lfloor \text{Log}_2 \text{Haut}(c) \rfloor = k$, on a

$$C_{n,k} = \sum_{m < n} \alpha_{m,n}^k C_{m,k-1}$$

où $\alpha_{m,n}^k$ est le nombre de chemins stricts de longueur $2n+1$ et de hauteur k fournissant un chemin strict de longueur $2m+1$ donné par contraction suivie

de suppression des paliers. La contraction étant une bijection, le nombre $\alpha_{m,n}^k$ n'est autre que celui de la proposition 3.3, à savoir $\binom{n-1}{2m} 2^{n-1-2m}$ et est ainsi indépendant de k .

Nous obtenons donc

$$C_{n,k} = \sum_{m < n} \binom{n-1}{2m} 2^{n-1-2m}$$

qui est la relation cherchée. Il est facile de vérifier que les conditions initiales des entiers $C_{n,k}$ sont celles des nombres de Strahler.

REMERCIEMENTS

L'auteur remercie Jean Vuillemin et Philippe Flajolet pour l'avoir intéressé à cette question, les mêmes et Gérard Viennot pour d'intéressantes discussions.

BIBLIOGRAPHIE

1. A. P. ERSHOV, *On programming of arithmetic operations*, CACM, vol. 1, n° 8, 1958, p. 3-6.
2. P. FLAJOLET, *Combinatorial aspects of continued fractions*, Discrete Math., vol. 32, 1980, p. 125-161.
3. P. FLAJOLET, *Analyses d'algorithmes de manipulation d'arbres et de fichiers*, Cahiers du B.U.R.O., p. 34-35, 1981.
4. P. FLAJOLET et J. FRANÇON, Notes non publiées.
5. P. FLAJOLET, J. C. RAOULT et J. VUILLEMIN, *The number of registers required for evaluating arithmetic expressions*, Theor. Comp. Sc., vol. 9, 1979, p. 99-125.
6. J. FRANÇON, *Des codes pour arbres binaires*, Actes du 2^e Colloque de Lille, Les arbres en algèbre et en programmation, 17-19 février 1977.
7. J. FRANÇON et G. VIENNOT, *Permutations selon leurs pics, creux, doubles montées et doubles descentes, nombres d'Euler et nombres de Genocchi*, Discrete Math., vol. 28, 1979, p. 21-35.
8. R. E. HORTON, *Erosional development of streams and their drainage basins: hydro-physical approach to quantitative morphology*, Bull. of the Geological Soc. of America, vol. 56, 1945, p. 275-370.
9. R. KEMP, *The average number of registers needed to evaluate a binary tree optimally*, Acta Informatica, vol. 11, 1979, p. 363-372.
10. D. E. KNUTH, *The Art of Computer Programming*, vol. 1, Addison-Wesley, 1968.
11. O. PERRON, *Die Lehre von den Kettenbrüchen*, Teubner, Leipzig und Berlin, 1929.
12. R. SETHI et J. D. ULLMAN, *The generation of optimal code for arithmetic expressions*, JACM, vol. 17, 1970, p. 715-728.
13. P. S. STEVENS, *Patterns in Nature*, Little, Brown and Co., 1974. Traduction française : *Les formes dans la nature*, Seuil, Paris, 1978.
14. A. N. STRAHLER, *Hypsometric (area-altitude) analysis of erosional topology*, Bull. of the Geological Soc. of America, vol. 63, 1952, p. 1117-1142.