

J.-P. ALLOUCHE

F. DRESS

Tours de Hanoï et automates

Informatique théorique et applications, tome 24, n° 1 (1990), p. 1-15

<http://www.numdam.org/item?id=ITA_1990__24_1_1_0>

© AFCET, 1990, tous droits réservés.

L'accès aux archives de la revue « Informatique théorique et applications » 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/>

TOURS DE HANOÏ ET AUTOMATES (*)

par J.-P. ALLOUCHE ⁽¹⁾ et F. DRESS ⁽¹⁾

Communiqué par J.-E. PIN

Résumé. – *Nous montrons que l'on peut engendrer les mouvements donnés par l'algorithme classique des tours de Hanoï au moyen d'un automate fini, ce qui éclaire les propriétés arithmétiques de cet algorithme et permet en particulier de retrouver tous les résultats de « périodicité ». Le problème des tours de Hanoï « cycliques », intrinsèquement plus complexe, est abordé par des méthodes analogues.*

Abstract. – *We show that the sequence of moves in the classical towers of Hanoï algorithm can be generated by a finite automaton. This throws light on the arithmetical properties of this algorithm and contains in particular all the "periodicity" results. Similar methods are used to study the intrinsically more complex problem of "cyclic" Hanoï towers.*

1. SITUATION DU PROBLÈME ET PANORAMA DE LA LITTÉRATURE

Le problème des tours de Hanoï s'énonce ainsi : on dispose de 3 piquets et de N disques de diamètres tous différents, troués en leur centre; les disques sont empilés, par ordre de diamètres décroissants, sur un piquet, et il faut les déplacer et reconstituer la pile sur un autre piquet; le seul type de mouvement autorisé est le déplacement du disque du dessus d'une des trois piles sur une autre, à condition de respecter la décroissance des diamètres (sur chaque piquet).

Le jeu des tours de Hanoï semble avoir été imaginé par Lucas [11] en 1883. Il existe aussi une légende des tours de Brahma, rapportée en 1884 et dont nous ne connaissons pas l'ancienneté réelle, qui annonce la fin du monde après le déplacement d'une pile de 64 disques. Ces détails sont donnés par

(*) Reçu septembre 1987, révisé juillet 1988.

(¹) C.N.R.S. URA 0226 et U.E.R. de Mathématiques et d'Informatique, Université Bordeaux-I, 351, cours de la Libération, 33405 Talence Cedex.

W. Ball dans l'ouvrage cité ci-dessous (*voir* aussi les livres de A. Sainte-Lague [39], 1942, de H. E. Dudeney [14], 1958, qui appelle ce jeu le « Reeve's puzzle », de M. Gardner [25], 1958, et de F. Schuh [40], 1968 — *voir* également l'article de T. Roth [38], 1974).

On sait, au moins depuis W. Ball ([6], 1892), qu'un algorithme récursif fournit une solution en $2^N - 1$ mouvements. W. Ball semble considérer comme évidente la démonstration de la minimalité de cette valeur, ce qui est effectivement le cas (*voir* aussi J. Frame et B. Stewart, [24], 1941, et D. Wood, [42], 1981).

La succession des mouvements donnée par cet algorithme possède diverses propriétés « cycliques » : la suite des ensembles de 2 piquets concernés par chaque mouvement est périodique de période 3, le plus petit disque est déplacé une fois sur deux, et de façon cyclique, les déplacements des autres disques ont des propriétés analogues; ces résultats sont mentionnés et démontrés par J. Arsac [2], 1977 (*voir* aussi P. J. Hayes, [29], 1977, pour la 3-périodicité). On en trouvera un exposé systématique dans l'article de I. Lavallée ([32], 1985).

Plusieurs algorithmes itératifs donnant la même suite de mouvements ont été proposés, en particulier par J. Arsac ([2], 1977). Certains utilisent explicitement l'une des périodicités notées ci-dessus. Nous pouvons citer par exemple P. J. Hayes ([29], 1977), et P. Buneman et L. Levy ([9], 1980), qui utilisent l'alternance des déplacements (eux-mêmes 3-périodiques) du plus petit disque avec des coups obligés.

Remarquons enfin que si le problème des tours de Hanoï fournit l'exemple peut-être le plus rebattu pédagogiquement d'algorithme récursif, un article récent de J.-C. Fournier ([23], 1986) étudie de façon très fouillée la dérécursivisation dans ce cas particulier (*voir* aussi le livre de A. Aho, J. Hopcroft et J. Ullmann ([1], 1974, p. 72), l'article de D. Barnard ([7], 1980), le livre de J. S. Rohl ([37], 1984) et l'article général de H. Partsch et P. Pepper ([35], 1986).

De nombreuses généralisations ont été étudiées, par exemple augmentation du nombre de piquets (J. Frame et B. Stewart [24], 1941, Br. Brousseau [8], 1976, D. Wood [42], 1981, et J. Arsac [4], 1985), utilisation de disques colorés (D. Wood, M. Er [20], 1985), tours de Hanoï « cycliques » (M. Atkison, [5], 1981, M. Er, [17] à [19], 1982, et [22], 1985), « arbres » de Hanoï (J. Engelfriet [15], 1981). On pourra aussi consulter les travaux de M. Krishnamoorthy et S. Biswas ([31], 1978), et de S. Minker ([34], 1983).

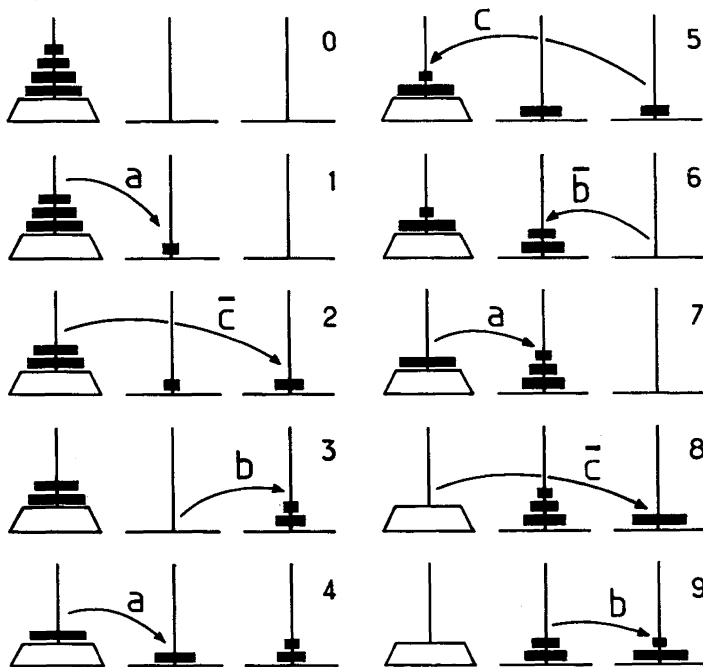


Figure 1

Soit A_N le mot de longueur $2^N - 1$ (sur l'alphabet $\{a, b, c, \bar{a}, \bar{b}, \bar{c}\}$) qui décrit les mouvements effectués pour transférer une pile de N disques du piquet 1 au piquet 2 (si N est impair) ou au piquet 3 (si N est pair). En notant σ la permutation qui transforme $(a, b, c, \bar{a}, \bar{b}, \bar{c})$ en $(c, a, b, \bar{c}, \bar{a}, \bar{b})$, l'algorithme récursif classique se traduit par les relations

$$A_{2m+1} = A_{2m} a \sigma(A_{2m}),$$

$$A_{2m+2} = A_{2m+1} \bar{c} \sigma^2(A_{2m+1}).$$

Remarquons au passage que ces relations démontrent l'existence annoncée plus haut d'une suite infinie $U = U(1)U(2)U(3)\dots$ dont les A_N sont les facteurs gauches de longueur $2^N - 1$. Nous définissons enfin l'application p par $p(a) = p(b) = p(c) = 0$ et $p(\bar{a}) = p(\bar{b}) = p(\bar{c}) = 1$, de sorte que la suite $V = p(U) = V(1)V(2)V(3)\dots$ est la suite « des barres » de U .

THÉORÈME 1 : *La suite U et la suite V sont 2-automatiques. De plus, aucune de ces deux suites n'est ultimement périodique.*

Nous présentons ci-dessous deux automates « à lecture inverse » qui engendrent les suites U et V (on verra ultérieurement qu'ils sont minimaux en nombre d'états).

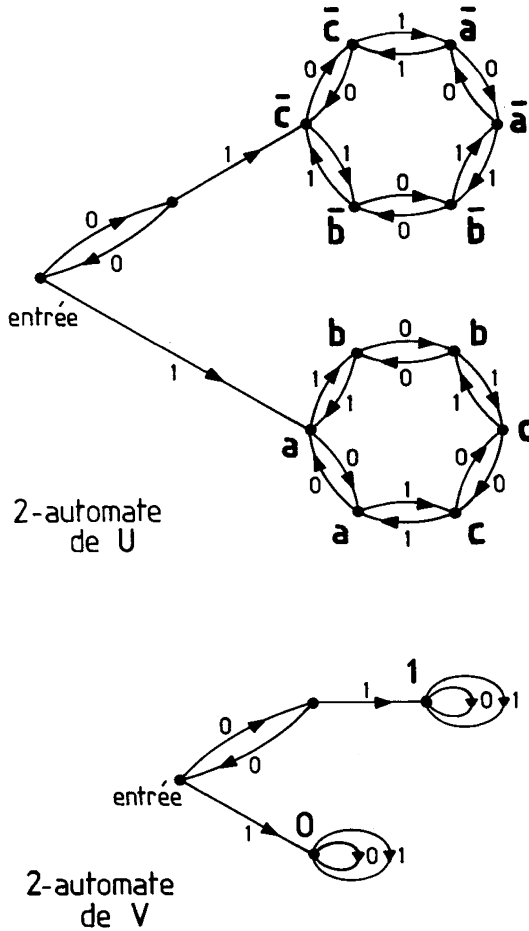


Figure 2

Rappelons (voir [10]) le fonctionnement pratique d'un tel automate : pour calculer $U(n)$, on part de l'état initial et on progresse sur le graphe de l'automate en lisant de droite à gauche les chiffres du développement binaire de n et en suivant chaque fois la flèche 0 ou la flèche 1 selon le chiffre lu; la valeur fournie par l'automate en fin de lecture est la valeur $U(n)$ désirée. Par

exemple, on pourra vérifier que la lecture de $(19)_{10} = (10011)_2$ fournit la valeur a .

3. TRANSFORMATIONS DE TOEPLITZ

Une autre propriété vérifiée par la suite U est d'être de Toeplitz (cf. K. Jacobs et M. Keane [30], 1969), *i.e.* de pouvoir être engendrée en itérant une transformation qui remplace dans une suite la sous-suite des occurrences d'une lettre « surnuméraire » par une suite donnée.

THÉORÈME 2 : *A toute suite B à valeurs dans l'alphabet $\Gamma = \{a, b, c, \bar{a}, \bar{b}, \bar{c}, \omega\}$, on associe une transformation T_B , définie sur les suites C également à valeurs dans Γ , de la façon suivante :*

soit $n_1 < n_2 < \dots < n_k < \dots$ la suite (éventuellement vide) des entiers j tels que $C(j) = \omega$; on pose

$$T_B C(j) = C(j) \quad \text{si } C(j) \neq \omega$$

$$T_B C(j) = B(k) \quad \text{si } j = n_k$$

(autrement dit on remplace dans l'ordre les ω de C par les termes de la suite B).

Alors, si l'on choisit pour B la suite $(a \bar{c} b \omega c \bar{b} a \omega b \bar{a} c \omega)^\omega$, on a

$$U = \lim_{n \rightarrow \infty} (T_B)^n B.$$

4. DÉMONSTRATIONS

Les suites U et V sont automatiques et non ultimement périodiques

D'après A. Cobham [12], la 2-automaticité de la suite U équivaut à la finitude de l'ensemble des sous-suites

$$F = \{ n \mapsto U(2^k n + r) \mid k \geq 0, 0 \leq r \leq 2^k - 1 \}.$$

Remarquons d'abord que le mot $B_k = A_{2^k}$ converge aussi vers la suite infinie U et que l'on a

$$B_{k+1} = B_k a \sigma(B_k) \bar{c} \sigma^2(B_k) b B_k$$

et

$$|B_k| = 2^{2k} - 1.$$

Comme on a

$$B_k = U(1) \dots U(2^{2k} - 1),$$

on obtient immédiatement, quels que soient $k \geq 0$ et $j \in [1, 2^{2k} - 1]$:

$$\begin{aligned} U(2^{2k}) &= a \\ U(2^{2k} + j) &= \sigma(U(j)) \\ U(2^{2k+1}) &= \bar{c} \\ U(2^{2k+1} + j) &= \sigma^2(U(j)) \\ U(3 \cdot 2^{2k}) &= b \\ U(3 \cdot 2^{2k} + j) &= U(j). \end{aligned}$$

Ces relations s'écrivent de façon condensée :

$$\left. \begin{aligned} \forall k \geq 0, \quad \forall x \in [1, 3], \quad U(2^{2k}x) &= U(x) \\ \forall k \geq 0, \quad \forall j \in [1, 2^{2k} - 1], \quad \forall x \in [1, 3], \quad U(2^{2k}x + j) &= \sigma^x(U(j)) \end{aligned} \right\} (\star)$$

Introduisons maintenant l'écriture en base 4 : $n = \sum_{M(n)}^{N(n)} e_k(n) 2^{2k}$, (avec donc $0 \leq e_j(n) \leq 3$ et $e_N(n) e_M(n) \neq 0$); puis posons

$$\begin{aligned} t(n) &= \sum e_k(n) \quad (\text{somme des chiffres}) \\ w(n) &= e_{M(n)}(n) \quad (\text{plus petit chiffre non nul}), \end{aligned}$$

alors les relations $(\star)$ impliquent, par récurrence facile sur $N(n) - M(n)$,

$$U(n) = \sigma^{t(n) - w(n)} U(e_{M(n)}(n)).$$

Si l'on remarque alors que

$$\begin{aligned} t(4n + j) &= t(n) + j && \text{pour } n \geq 0 \text{ et } 0 \leq j \leq 3, \\ w(4n) &= w(n) && \text{pour } n \geq 0, \\ w(4n + j) &= j && \text{pour } n \geq 0 \text{ et } 1 \leq j \leq 3, \end{aligned}$$

on en déduit que, pour $n \geq 0$ et $1 \leq j \leq 3$,

$$U(4n) = U(n) \quad (\text{ici on suppose } n \neq 0)$$

$$U(4n+j) = \sigma^{t(n)} U(j).$$

Il ne reste plus qu'à remarquer que $t(n)$ est congru à n modulo 3, pour en conclure que l'on a :

$$\begin{aligned} \forall n \geq 1, \quad U(4n) &= U(n) \\ \forall n \geq 0, \quad U(4n+1) &= \sigma^n(a) \\ \forall n \geq 0, \quad U(4n+2) &= \sigma^n(\bar{c}) \\ \forall n \geq 0, \quad U(4n+3) &= \sigma^n(b) \end{aligned} \quad (\star\star)$$

puis que l'ensemble F des sous-suites $U(2^k n + r)$ est fini et composé de 14 suites : $(U(n))$, $(U(2n))$, et les suites 3-périodiques dont les valeurs sur une période sont, soit une permutation de abc , soit une permutation de $\bar{a}\bar{b}\bar{c}$.

Ceci prouve la 2-automaticité de la suite U et donne en outre, en suivant la construction indiquée par exemple dans A. Cobham [12], l'automate à lecture inverse et à 14 états (*cf.* ci-dessus la figure 2) qui reconnaît la suite U . Le nombre d'états est minimal car les 14 sous-suites présentées ci-dessus sont distinctes. On obtient de même un automate minimal qui reconnaît la suite V .

Pour montrer que la suite V (et donc la suite U) n'est pas ultimement périodique, il suffit de constater, en examinant l'automate pour V proposé plus haut, que $V(2^k(2n+1))$ est égal à k modulo 2.

Signalons pour terminer que l'on peut également déduire les relations $(\star)$ et $(\star\star)$ de la formule fondamentale (1) de l'article de Fournier [23].

Où l'on retrouve les propriétés de périodicité

L'algorithme standard montre immédiatement que le k -ième disque est déplacé exactement aux mouvements de numéro $2^k n + 2^{k-1}$. Les relations $(\star\star)$ impliquent

$$(U(2n+1)) = (abc)^\infty \quad \text{et} \quad (U(4n+2)) = (\bar{c}\bar{b}\bar{a})^\infty.$$

On remarque alors que le mot 00 envoie l'état initial de l'automate sur lui-même, et l'on obtient pour la suite $(U_k(n)) = (U(2^k n + 2^{k-1}))$ des mouvements du k -ième disque :

$$(U_k(n)) = (U(2n+1)) = (abc)^\infty \quad \text{si } k \text{ est impair}$$

et

$$(U_k(n)) = (U(4n+2)) = (\bar{c}\bar{b}\bar{a})^\infty \quad \text{si } k \text{ est pair.}$$

Quand on « enlève les barres », la suite U devient la suite $(abc)^\infty$, résultat connu que nous pouvons facilement retrouver. A partir du 2-automate à lecture inverse (donné plus haut) qui engendre la suite U , nous construisons « canoniquement » un 4-automate à lecture inverse qui engendre également U , comme suit : on ne garde que les états accessibles à partir de l'état initial par des chemins de longueur paire, et les nouvelles flèches sont les anciens chemins de longueur 2, étiquetés 0 pour 00, 1 pour 01 (flèche 1 suivie de la flèche 0, se souvenir que l'automate est à lecture inverse), 2 pour 10, et 3 pour 11. L'automate résultant est dessiné ci-dessous. Il est clair que le résultat « sans la barre » obtenu en y entrant le développement d'un entier n en base 4 ne dépend que de la valeur modulo 3 de la somme $t(n)$ des chiffres de ce développement (qui est la valeur modulo 3 de n lui-même); plus précisément, si $t(n) \equiv 1$, $U(n) = a$ ou $\bar{a}$, si $t(n) \equiv 2$, $U(n) = c$ ou $\bar{c}$, si $t(n) \equiv 0$, $U(n) = b$ ou $\bar{b}$.

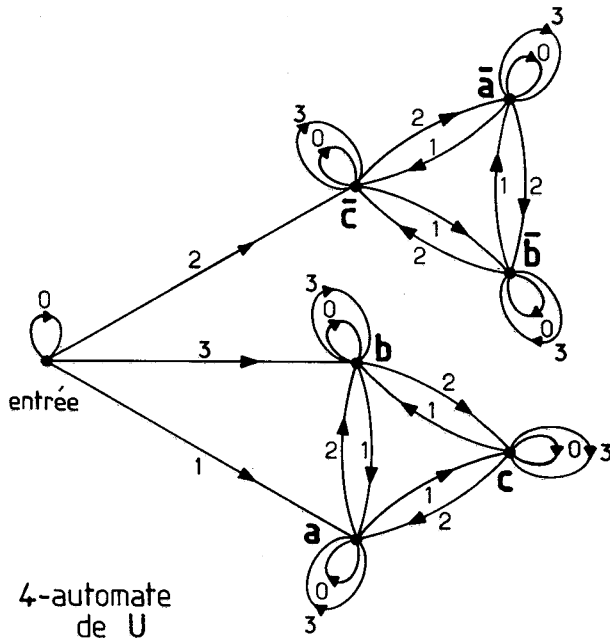


Figure 3

La suite U est de Toeplitz

On remarque que $(T_B)^n B$ converge. Sa limite, que l'on notera U' , est un point fixe de T_B , et vérifie donc les relations

$$\left\{ \begin{array}{l} |\forall n \geq 1, \quad U'(4n) = U'(n) \\ |\forall n \geq 0, \quad U'(6n+1) = a \\ |\forall n \geq 0, \quad U'(6n+3) = b \\ \quad \forall n \geq 0, \quad U'(6n+5) = c \\ |\forall n \geq 0, \quad U'(12n+2) = \bar{c} \\ |\forall n \geq 0, \quad U'(12n+6) = \bar{b} \\ |\forall n \geq 0, \quad U'(12n+10) = \bar{a}. \end{array} \right.$$

Ces relations caractérisent complètement la suite U' ; d'après les égalités (★★), elles sont aussi vérifiées par la suite U , ce qui montre que $U = U'$.

5. REMARQUE SUR LES « TOURS DE HANOÏ CYCLIQUES »

La généralisation des tours de Hanoï « cycliques » (proposée par M. D. Atkinson, [5], 1981) consiste à imaginer les piquets disposés sur un cercle, et à ne s'autoriser que des mouvements dans le sens des aiguilles d'une montre (autrement dit, avec les notations ci-dessus, à se limiter aux mouvements a, b, c). L'algorithme proposé par Atkinson imbrique récursivement les suites C_N et A_N qui transportent N disques respectivement du premier au deuxième et du premier au troisième piquet. Le but cherché dans cette généralisation était de fournir un exemple pédagogique d'algorithme récursif qui ne soit pas facilement transformable en algorithme itératif; néanmoins M. C. Er ([17], 1982) a proposé un tel algorithme itératif. Nous montrons quant à nous que la suite des mouvements est engendrée par un tag system (cf. A. Cobham, [12], 1972), non uniforme comme dans le cas classique il est vrai.

Il résulte de l'algorithme donné par Atkinson que l'on a les relations

$$C_{N+1} = A_N a \sigma(A_N) \quad \text{et} \quad A_{N+1} = A_N a \sigma(C_N) b A_N$$

(en conservant la notation σ pour désigner la restriction à $\{a, b, c\}$ de la permutation introduite au paragraphe 2). Cela suffit à montrer la convergence de A_N et C_N vers une limite commune A .

Les choses s'expriment plus simplement en posant $X_N = A_N a$ et $Y_N = C_N c$; on a alors

$$Y_{N+1} = X_N \sigma(X_N) \quad \text{et} \quad X_{N+1} = X_N \sigma(Y_N) X_N.$$

Si l'on note σ' l'application définie sur Φ par

$$\begin{aligned} \sigma'(f) &= g, & \sigma'(g) &= h, & \sigma'(h) &= f, \\ \sigma'(u) &= v, & \sigma'(v) &= w, & \sigma'(w) &= u, \end{aligned}$$

on remarque que

$$\varphi \circ \sigma' = \sigma \circ \varphi \quad \text{et} \quad s \circ \sigma' = \sigma' \circ s.$$

Soient alors L_N et M_N les mots définis sur l'alphabet $\{a, b, c\}$, pour $N \geq 1$, par :

$$L_N = \varphi(s^N(f)) \quad \text{et} \quad M_N = \varphi(s^{N-1}(f)) \varphi(\sigma'(s^{N-1}(f))).$$

On écrit

$$L_{N+1} = \varphi(s^{N+1}(f)) = \varphi(s^N(f v f)) = \varphi(s^N(f)) \varphi(s^N(v)) \varphi(s^N(f)),$$

puis on calcule

$$\begin{aligned} \varphi(s^N(v)) &= \varphi(s^N(\sigma'(u))) = \varphi(\sigma'(s^N(u))) = \varphi(\sigma'(s^{N-1}(f g))) \\ &= \sigma(\varphi(s^{N-1}(f g))) = \sigma[\varphi(s^{N-1}(f)) \varphi(s^{N-1}(g))] \\ &= \sigma[\varphi(s^{N-1}(f)) \varphi(s^{N-1}(\sigma'(f)))] \\ &= \sigma[\varphi(s^{N-1}(f)) \varphi(\sigma'(s^{N-1}(f)))] = \sigma(M_N), \end{aligned}$$

et on en déduit enfin

$$L_{N+1} = L_N \sigma(M_N) L_N.$$

De même

$$M_{N+1} = \varphi(s^N(f)) \varphi(\sigma'(s^N(f))) = L_N \sigma(\varphi(s^N(f))) = L_N \sigma(L_N).$$

Ainsi L_N et M_N vérifient-ils les mêmes relations de récurrence que X_N et Y_N , comme de plus $L_1 = X_1 = a b a$ et $M_1 = Y_1 = a c$, on a, pour tout $N \geq 1$,

$$X_N = L_N = \varphi(s^N(f))$$

et

$$Y_N = M_N = \varphi(s^{N-1}(f)) \varphi(\sigma'(s^{N-1}(f))).$$

On conclut en remarquant que X_N et Y_N tendent vers la même limite A que A_N et C_N , et donc :

$$A = \lim_{N \rightarrow \infty} X_N = \lim_{N \rightarrow \infty} \varphi(s^N(f)).$$

La « complexité » des tours de Hanoï cycliques semble supérieure à celle des tours de Hanoï ordinaires (voir P. Cull et E. Ecklund, [13], 1985, M. C. Er, [22], 1985, et C. Gerety et P. Cull, [27], 1986). Il serait intéressant de montrer que la suite des mouvements dans le cas cyclique n'est pas automatique. L'idée la plus simple est de calculer les fréquences d'apparition des lettres dans la suite infinie, qui (si elles existent) sont nécessairement rationnelles pour une suite automatique. Mais dans l'algorithme cyclique les trois fréquences sont égales à $1/3$ (résultat qui s'obtient classiquement en calculant le vecteur propre normalisé associé à la valeur propre dominante de la matrice associée à la substitution), et cela ne permet pas de conclure. Il n'est pas inintéressant de signaler au passage que les fréquences des lettres dans le cas classique sont égales à $2/9$ pour a, b, c , et $1/9$ pour $\bar{a}, \bar{b}, \bar{c}$.

BIBLIOGRAPHIE

1. A. AHO, J. HOPCROFT et J. ULLMANN, *The Design and Analysis of Computers Algorithms*, Addison-Wesley, Reading, MA, 1974,
2. J. ARSAC, *Le construction de programmes structurés*, Dunod, Paris, 1977.
3. J. ARSAC, *Les bases de la programmation*, Dunod, Paris, 1983.
4. J. ARSAC, *Jeux et casse-tête à programmer*, Dunod, Paris, 1985.
5. M. D. ATKINSON, *The Cyclic Towers of Hanoï*, Inform. Process. Lett., vol. 13, 1981, p. 118-119.
6. W. R. BALL, *Mathematical Recreations and Essays*, McMillan, London, 1892. Voir aussi W. R. BALL et H. S. M. COXETER, *Mathematical Recreations and Essays*, University of Toronto Press, Toronto, 1974, p. 316-317.
7. D. T. BARNARD, *The Towers of Hanoï: an Exercise in Non Recursive Algorithm Development*, Technical Report 80-103, Dept. of Computing and Information Science, Queen's University, 1980.

8. Br. A. BROUSSEAU, *Towers of Hanoi with More Pegs*, J. Recreat. Math., vol. 8, (3), 1976, p. 165-176.
9. P. BUNEMAN et L. LEVY, *The Towers of Hanoi Problem*, Inform. Process. Lett., vol. 10, 1980, p. 243-244.
10. G. CHRISTOL, T. KAMAE, M. MENDÈS FRANCE et G. RAUZY, *Suites algébriques, automates et substitutions*, Bull. Soc. Math. France, vol. 108, 1980, p. 401-419.
11. N. CLAUS (anagramme de Lucas), *La tour de Hanoi, jeu de calcul*, Revue Science et Nature, vol. 1, n° 8, 1884, p. 127-128.
12. A. COBHAM, *Uniform Tag Sequences*, Math. Syst. Theory, vol. 6, 1972, p. 164-192.
13. P. CULL et E. ECKLUND, *Towers of Hanoi and Analysis of Algorithms*, Amer. Math. Monthly, vol. 92, (6), June/July 1985.
14. H. E. DUDENEY, *The Canterbury Puzzles*, Thos. Nelson & Sons, 1919, réédition Dovers Publications Ltd, New York, 1958.
15. J. ENGELFRIET, *The Trees of Hanoi*, 1981, preprint.
16. M. C. ER, *A Representation Approach to the Towers of Hanoi Problem*, The Comput. J., 1982, p. 442-447.
17. M. C. ER, *An Iterative Solution to the Cyclic Towers of Hanoi Problem*, Technical Report, Dept. of Computing Science, University of Wollongang, 1982.
18. M. C. ER, *The Cyclic Towers of Hanoi: a Generalization*, Technical Report, Dept. of Computing Science, University of Wollongang, 1982.
19. M. C. ER, *A Generalization of the Cyclic Towers of Hanoi*, Technical Report, Dept. of Computing Science, University of Wollongang, 1982.
20. M. C. ER, *Towers of Hanoi with Black and White Discs*, J. Inform. Optim. Sci., vol. 6, (1), 1985, p. 87-93.
21. M. C. ER, *The Towers of Hanoi and Binary Numerals*, J. Inform. Optim. Sci., vol. 6, (2), 1985, p. 147-152.
22. M. C. ER, *The Complexity of the Generalised Cyclic Towers of Hanoi*, J. Algorithms, vol. 6, (3), 1985, p. 351-358.
23. J.-C. FOURNIER, *Pour en finir avec la dérécursivation du problème des tours de Hanoi*, Actes Journée A.F.C.E.T. Combinatoire, Lyon-I, 1985.
24. J. S. FRAME et B. M. STEWART, *Solution of Problem n° 3918*, Amer. Math. Monthly, vol. 48, 1941, p. 216-219.
25. M. GARDNER, *Mathematical Puzzles and Diversions*, Simon & Schuster, New York, 1958, p. 55-62.
26. M. GARDNER, *Mathematical Games: the Curious Properties of the Gray Code and How it Can Be Used to Solve Puzzles*, Sci. Amer., août 1972, p. 106-109.
27. C. GERETY et P. CULL, *Time Complexity of the Towers of Hanoi Problem*, SIGACT News, vol. 18, (1), 1986.
28. J. HARDOUIN-DUPARC, *Génération de mots par des piles d'automates*, 1985, preprint.
29. P. J. HAYES, *A Note on the Towers of Hanoi Problem*, The Comput. J., 1977, p. 282-285.
30. K. JACOBS et M. KEANE, *On 0-1 Sequences of Toeplitz Type*, Z. Warsch. Geb., vol. 13, 1969, p. 123-131.
31. M. S. KRISHNAMOORTHY et S. BISWAS, *The Generalized Towers of Hanoi* (preprint), 1978.
32. I. LAVALLÉE, *Note sur le problème des tours de Hanoi*, Rev. Roumaine Math. pures et appl., vol. 30, 1985, p. 433-438.

33. B. MEYER et C. BAUDOUIN, *Méthodes de programmation*, Eyrolles, Paris, 3^e édition, 1984.
34. S. MINKER, *Three Variations on the Towers of Hanoi Problem*, S. M. Thesis, University of Pennsylvania, 1983.
35. H. PARTSCH et P. PEPPER, *A Family of Rules for Recursion Removal*, 1986, preprint.
36. G. RAUZY, Cours de D.E.A. (communication privée), 1986.
37. J. S. ROHL, *Recursion via Pascal*, Cambridge University Press, 1984.
38. T. ROTH, *The Tower of Brahma Revisited*, J. Recreat. Math., vol. 7, n° 2, 1974, p. 116-119.
39. A. SAINTE-LAGUE, *Avec des nombres et des lignes*, Vuibert, Paris, 1942, p. 71-78.
40. F. SCHUH, *The Master Book of Mathematical Recreations*, Dover Publications, Inc., New York, 1968, p. 119-121.
41. T. R. WALSH, *The Towers of Hanoi Revisited: Moving the Rings by Counting the Moves*, Inform. Process. Lett., vol. 15, 1982, p. 64-67.
42. D. WOOD, *The Towers of Brahma and Hanoi Revisited*, J. Recreat. Math., vol. 14, n° 1, 1981, p. 17-24.