

LAMINE MELKEMI
MAURICE TCHUENTE

**Un réseau linéaire pour la reconnaissance
des mots sans carré**

RAIRO. Informatique théorique et applications, tome 22, n° 2 (1988),
p. 147-161

http://www.numdam.org/item?id=ITA_1988__22_2_147_0

© AFCET, 1988, tous droits réservés.

L'accès aux archives de la revue « RAIRO. 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/>

UN RÉSEAU LINÉAIRE POUR LA RECONNAISSANCE DES MOTS SANS CARRÉ (*)

par Lamine MELKEMI ⁽¹⁾ et Maurice TCHUENTE ⁽²⁾

Communiqué par J. E. PIN

Résumé. – Dans cet article, nous proposons une architecture parallèle spécialisée pour la recherche en temps linéaire d'un carré dans un mot. Le réseau proposé est linéaire et une seule de ses cellules communique avec l'extérieur. Le traitement d'un mot de longueur $n=2p$ s'effectue en temps $T=3n$ sur un réseau de taille p , où le temps de cycle de la cellule de base est pris comme unité de temps.

Abstract. – In this paper, we introduce a parallel special-purpose architecture for the linear time recognition of square free words. The array presented is linear and only one of the basic cells can communicate with the external world. The processing of a word of length $n=2p$ is performed in time $T=3n$ on an array of size p , where the cycle-time of the basic cell is taken as time unit.

1. INTRODUCTION

Dans cet article, nous nous intéressons à la recherche d'un carré dans un mot x de longueur n . De nombreux algorithmes ont déjà été proposés pour résoudre ce problème.

D'abord, dans le cas séquentiel, on peut montrer que, si l'alphabet n'est pas connu à l'avance, alors la complexité du problème est $O(n \log_2 n)$; en outre, des algorithmes optimaux, c'est-à-dire qui atteignent cette borne, ont été proposés dans [2], [3], [6]. Lorsque l'alphabet est connu à l'avance, Crochemore [4] a exhibé une solution séquentielle qui s'exécute en temps linéaire $O(n)$.

Récemment, Apostolico et Negro [1] ont proposé un réseau pipeline qui détecte en temps linéaire tous les carrés dans un mot. Dans cette solution, le

(*) Reçu en juillet 1986, révisé en 1987.

⁽¹⁾ Laboratoire TIM3/INPG, Équipe Algorithmique Mathématique, B.P. n° 68, 38402 Saint-Martin-d'Hères Cédex, France.

⁽²⁾ Faculté des Sciences, Laboratoire d'Informatique, B.P. n° 812, Yaoundé, Cameroun.

traitement d'un mot de longueur $n=2p$ nécessite un réseau de taille p dans lequel toute cellule en position i , $1 \leq i \leq p$, dispose d'un compteur pouvant prendre au moins i valeurs distinctes. En conséquence, si l'on réalise un circuit spécialisé correspondant à cette solution, la taille des mots pouvant être traités sera bornée par la taille des compteurs implantés dans les cellules de base.

Nous proposons ici un réseau uni-dimensionnel permettant lui aussi de résoudre en temps linéaire, le problème de la recherche d'un carré dans un mot. Contrairement à la solution proposée dans [1], la structure de la cellule de base est indépendante de la taille du mot traité. Il convient toutefois de noter que cette solution requiert des cellules de base programmables; en effet, le comportement d'une cellule de base au cours du déroulement de l'algorithme, dépend de sa position dans le réseau. L'intérêt de cette solution par rapport à celle proposée par Crochemore [4], est sa plus grande vitesse d'exécution si elle est réalisée sous forme de circuit VLSI.

Dans toute la suite, nous adoptons les notations suivantes: A^+ désigne l'ensemble des mots de longueur supérieure ou égale à un, sur l'alphabet non vide A .

Un mot x de A^+ de longueur n est noté $x = x_1 x_2 \dots x_n$ où $x_i \in A$ pour $i = 1, \dots, n$.

On appelle carré, un mot x de la forme ww avec $w \in A^+$; ceci revient à dire que $x = x_1 \dots x_p x_{p+1} \dots x_n$, $n=2p$, avec $x_i = x_{i+p}$, $1 \leq i \leq p$. Un mot y est appelé facteur d'un mot $x = x_1 \dots x_n$, s'il est de la forme $y = x_i x_{i+1} \dots x_j$ avec $1 \leq i \leq j \leq n$.

Un mot x est dit sans carré s'il ne contient pas de facteur qui est un carré.

Par exemple, *acabca* est un mot sans carré, alors que *bacaacab* contient les carrés *acaaca* et *aa*.

2. FLUX DES VARIABLES D'ENTRÉE

Commençons par donner la description détaillée du flux des variables d'entrée, que le réseau doit lire pour effectuer les traitements.

Dans toute la suite nous supposons que $n=2p$.

Considérons le schéma de la figure 1 ci-dessous :

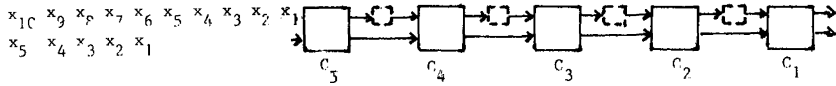


Figure 1

Les cellules sont numérotées $1, 2, \dots, p$ de droite à gauche. Entre deux cellules voisines, on a deux liaisons :

- La liaison inférieure permet la circulation des variables de cellule en cellule. Une variable $x_i, 1 \leq i \leq p$, qui se trouve dans la cellule C_r à l’instant t , est transmise à la cellule C_{r-1} à l’instant $t + 1$.

- La liaison supérieure comporte une cellule de retard. Cela signifie qu’une variable $x_j, 1 \leq j \leq n$, qui se trouve dans la cellule C_r à l’instant t , est transmise à la cellule de retard à l’étape $t + 1$, et ne se retrouve dans C_{r-1} qu’à l’instant $t + 2$.

Ce type de réseau a été introduit pour la première fois par Kung [5] pour le problème de convolution.

Dans le flux supérieur, toute variable $x_j, 1 \leq j \leq n$ est lue par le réseau à l’instant j . Dans le flux inférieur, toute variable $x_i, 1 \leq i \leq p$, est lue par le réseau à l’étape $i + p$. L’absence de cellule de retard sur les liaisons du flux inférieur, permet aux variables de ce flux, de rattraper progressivement les variables du flux supérieur au cours de la traversée du réseau (cf. fig. 2).

Ainsi, lors de la traversée du réseau, $x_i, 1 \leq i \leq p$ rencontre successivement $x_{i+p}, x_{i+p-1}, \dots, x_{i+1}$ dans les cellules C_p, C_{p-1}, \dots, C_1 .

Par exemple dans le cas $p = 5$, le tableau 1 donne les rencontres qui se produisent au cours du déroulement de l’algorithme.

On voit ainsi que la cellule C_r qui est en position r à partir de la droite du réseau, est le lieu de rencontre de tous les couples $(x_i, x_{i+r}), 1 \leq i \leq p$. C’est donc la cellule C_r qui peut détecter les carrés de longueur $2r$. Plus précisément, r égalités consécutives détectées par C_r correspondent à la présence d’un carré de longueur $2r$ dans le mot traité.

Par exemple dans le tableau 1, correspondant à $p = 5$,

- $(x_1 = x_4)$ et $(x_2 = x_5)$ et $(x_3 = x_6)$ correspond au carré $x_1x_2x_3x_4x_5x_6$
- $(x_2 = x_5)$ et $(x_4 = x_6)$ et $(x_4 = x_7)$ correspond au carré $x_2x_3x_4x_5x_6x_7$
- $(x_3 = x_6)$ et $(x_4 = x_7)$ et $(x_5 = x_8)$ correspond au carré $x_3x_4x_5x_6x_7x_8$.

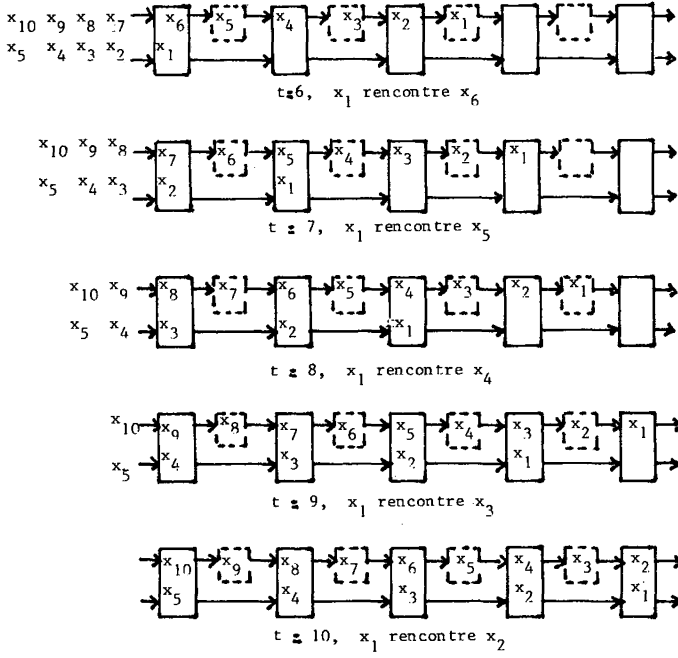


Figure 2

étape	cellule	C_5	C_4	C_3	C_2	C_1
6.....		x_1, x_6				
7.....		x_2, x_7	x_1, x_5			
8.....		x_3, x_8	x_2, x_6	x_1, x_4		
9.....		x_4, x_9	x_3, x_7	x_2, x_5	x_1, x_3	
10.....		x_5, x_{10}	x_4, x_8	x_3, x_6	x_2, x_4	x_1, x_2
11.....			x_5, x_9	x_4, x_7	x_3, x_5	x_2, x_3
12.....				x_5, x_8	x_4, x_6	x_3, x_4
13.....					x_5, x_7	x_4, x_5
14.....						x_5, x_6

TABLEAU 1

3. UNE SOLUTION BI-DIMENSIONNELLE

Dans ce paragraphe, nous présentons un réseau bi-dimensionnel capable de détecter en temps linéaire, tous les carrés dans un mot. Comme nous l'avons remarqué au paragraphe précédent, la détection de tous les carrés peut s'obtenir à partir du réseau de la figure 1, en adjoignant à toute cellule C_r un dispositif permettant de signaler la détection par C_r , de r égalités consécutives.

Pour ce faire, nous adjoignons à toute cellule C_r un sous-réseau linéaire composé de $r-1$ cellules de propagation. Comme leur nom l'indique, les cellules de propagation (représentées en pointillé sur la figure 3), ont pour seul rôle de propager les signaux d'égalité générés par C_r , de manière à les transmettre en sortie à la frontière droite du réseau.

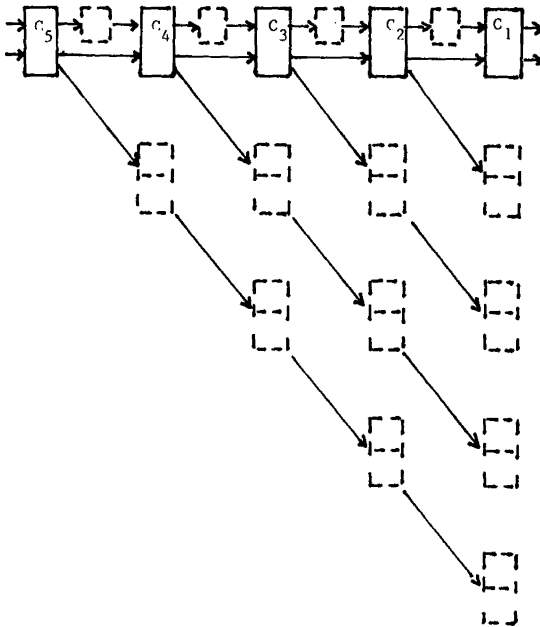


Figure 3

Chaque cellule de propagation est composée de deux registres; nous disons qu'un registre est occupé ou vide selon qu'il contient ou pas un signal d'égalité.

- Initialement, tous les registres des cellules de propagation sont vides.

– Lorsque C_r génère un signal d'égalité il l'envoie à la cellule de propagation qui lui est voisine à droite.

– Lorsqu'une cellule de propagation reçoit un signal d'égalité, il y a deux situations possibles :

. Si le registre inférieur est vide, alors ce signal d'égalité y est stocké;

. Si le registre inférieur est occupé, alors ce signal d'égalité est stocké dans le registre supérieur.

– Tout signal d'égalité stocké dans le registre inférieur d'une cellule de propagation, y reste jusqu'à ce qu'il soit détruit par un signal de destruction; un signal d'égalité stocké dans le registre supérieur d'une cellule de propagation, est transmis l'instant d'après à la voisine de droite.

– Lorsqu'une cellule C_r traite deux symboles x_i, x_{i+r} qui sont différents, elle envoie l'instant d'après, un signal de destruction à la cellule de propagation qui lui est voisine à droite.

– Un signal de destruction se déplace de cellule en cellule à chaque étape, et comme son nom l'indique, il détruit sur son passage tous les signaux d'égalité rencontrés.

Montrons maintenant que le réseau bi-dimensionnel ainsi construit est bien une solution du problème.

LEMME 1 : Si une cellule C_r détecte r égalités consécutives, alors un signal d'égalité est transmis en sortie, via son couloir de propagation.

Démonstration: Le cas $r=1$ est évident, en effet, comme C_1 n'a pas de couloir de propagation, la détection par C_1 de toute égalité $x_i = x_{i+1}$, entraîne la transmission l'instant d'après d'un signal d'égalité en sortie du réseau.

Supposons maintenant que $r > 1$, et que C_r génère r signaux d'égalité à des instants consécutifs. On peut évidemment supposer que ces signaux sont précédés par un signal de destruction.

– Le premier signal d'égalité s'installe dans le registre inférieur de la première cellule de propagation.

– Le second signal d'égalité traverse la première cellule de propagation par le registre supérieur et s'installe dans le registre inférieur de la seconde cellule de propagation.

– Plus généralement, le j -ième signal d'égalité, $j < r$, traverse les $j-1$ premières cellules de propagation par le registre supérieur, et s'installe dans le registre inférieur de la j -ième cellule de propagation.

On voit donc que, à l'arrivée du r -ième signal d'égalité, tous les registres inférieurs des $r-1$ cellules du couloir de propagation seront occupés. Ce

signal traversera donc toutes les cellules du couloir de propagation par les registres supérieurs et se retrouvera en sortie du réseau.

LEMME 2 : Si C_r , $r > 1$, génère successivement un signal de destruction, i signaux d'égalité, $i < r$, et un signal de destruction, alors il n'y a pas de signal d'égalité transmis en sortie du réseau.

Démonstration : Le premier signal de destruction a pour effet de vider tous les registres des cellules du couloir de propagation avant l'arrivée des signaux d'égalité.

Les i signaux d'égalité s'installent dans les registres inférieurs des i premières cellules du couloir de propagation, $i < r$, et sont ensuite détruits par le second signal de destruction.

Comme $i < r$, on voit que, au cours de cette séquence, la $(r - 1)$ -ième cellule de propagation ne contiendra à aucun moment deux signaux d'égalité. Cette séquence ne permet donc pas de transmettre en sortie du réseau, un signal d'égalité.

Les deux lemmes que nous venons de démontrer, prouvent la validité de l'algorithme que nous avons défini sur le réseau bi-dimensionnel.

4. UNE SOLUTION UNI-DIMENSIONNELLE

Le réseau bi-dimensionnel que nous avons présenté au paragraphe précédent, permet de détecter en temps linéaire, tous les carrés d'un mot.

Dans ce paragraphe, nous allons le transformer en un réseau uni-dimensionnel permettant de savoir en temps linéaire, si un mot contient ou pas un carré.

Considérons le réseau uni-dimensionnel de la figure 4, obtenu à partir de la figure 3 en remplaçant par une cellule unique, toutes les cellules de propagation appartenant à une même colonne.

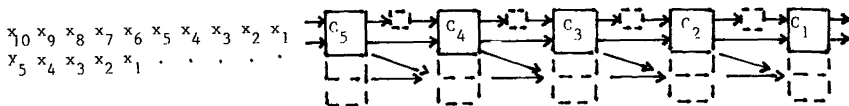


Figure 4

Ce schéma pose un problème de chevauchement que nous allons illustrer maintenant sur deux exemples.

Exemple 1: Supposons que C_k , $k > 2$, génère deux signaux d'égalité aux instants t , $t+1$, et que C_{k-1} génère un signal d'égalité à l'instant $t+2$. On voit que

- les deux signaux d'égalité engendrés par C_k entraînent l'envoi de C_{k-1} vers C_{k-2} , d'un signal d'égalité à l'instant $t+3$;
- l'égalité détectée par C_{k-1} à l'instant $t+2$, entraîne l'envoi de C_{k-1} vers C_{k-2} , d'un signal d'égalité à l'instant $t+3$.

Ainsi, on a chevauchement à l'instant $t+3$, entre deux signaux d'égalité d'origines différentes arrivant dans C_{k-2} .

Exemple 2: Supposons qu'une cellule C_k ait généré k signaux d'égalité correspondant à la présence d'un carré de longueur k ; supposons que le dernier de ces signaux se trouve dans C_{k-1} à l'instant t . Si à l'instant t , C_{k-1} génère un signal de destruction, alors ce signal ira détruire les signaux d'égalité générés par C_k , et le carré de longueur k ne sera pas reconnu par le réseau; ceci peut être catastrophique, si ce carré est le seul contenu dans le mot. Cette situation est illustrée dans la figure 5, où

- $1 = e_{i, i+k}$ désigne le signal d'égalité généré par C_k après la détection de l'égalité $x_i = x_{i+k}$.
- $0 = d_{i, i+k}$ désigne le signal de destruction généré par C_k après la détection de l'inégalité $x_i \neq x_{i+k}$.

Pour éviter ce problème de chevauchement, nous allons inhiber toute cellule dont la partie de propagation contient un signal d'égalité. Autrement dit, une cellule C_k génère un signal d'égalité ou de destruction à l'instant t , uniquement si ses deux registres de propagation ne contiennent pas de signal d'égalité.

Le problème qui se pose cette fois est le suivant: des signaux d'égalité générés par une cellule C_k , et correspondant à un faux début de carré (par exemple $x_i = x_{i+3}$, $x_{i+1} = x_{i+4}$, $x_{i+2} \neq x_{i+4}$, cf. fig. 6) peuvent aller inhiber une cellule C_h telle que $h < k$, et l'empêcher de reconnaître un carré de longueur h .

Dans la suite nous allons restreindre les inhibitions. Plus précisément, nous considérons le réseau de la figure 11 a, dans lequel une cellule C_k peut inhiber une cellule C_h , $h < k$, si et seulement si $\lceil \log_2 h \rceil = \lceil \log_2 k \rceil$ (où $\lceil u \rceil$ désigne le plus petit entier supérieur ou égal à u).

La validité de l'algorithme est basée sur le fait que, si $h < k$, $\lceil \log_2 h \rceil = \lceil \log_2 k \rceil$, C_k inhibe C_h et l'empêche de reconnaître un carré de longueur $2h$, alors le mot traité contient un carré de longueur inférieure à $2h$. La conséquence de ce résultat est que le réseau reconnaît les carrés de longueur minimum.

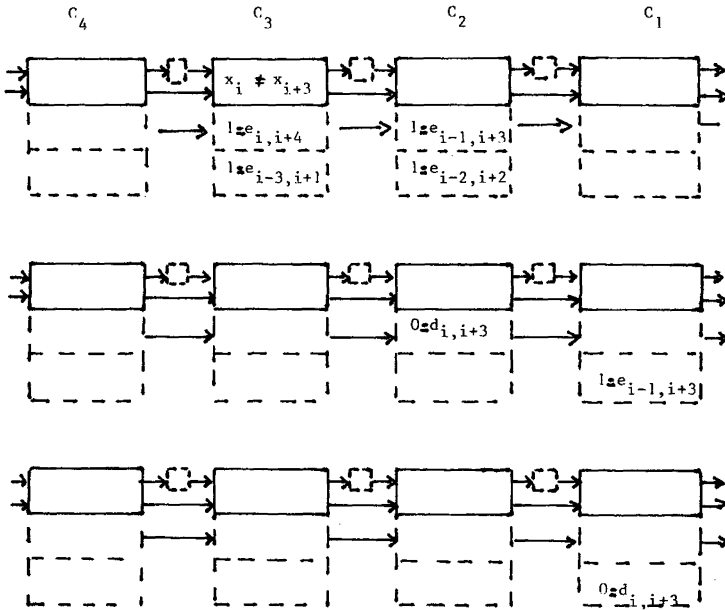


Figure 5

Nous allons maintenant démontrer cette propriété de manière rigoureuse.

LEMME 3 : Pour que C_k inhibe C_{k-r} , $r > 1$, à l'instant t , elle doit générer r signaux d'égalité aux instants $t-r, t-r-1, \dots, t-2r+1$.

Démonstration : Raisonnons par récurrence sur r .

Le cas $r=1$ est évident. Supposons maintenant le résultat vrai pour $r-1$ et montrons-le pour r . Supposons que C_k inhibe C_{k-r} à l'instant t . D'après la définition de l'algorithme, C_{k-r+1} doit contenir à l'instant $t-1$, deux signaux d'égalité générés par C_k . C_{k-r+1} est donc inhibée par C_k aux instants $t-1$ et $t-2$. D'après l'hypothèse de récurrence, C_k doit générer des signaux d'égalité aux instants

$$(t-1)-(r-1), (t-1)-(r-1)-1, \dots, (t-1)-2(r-1)+1$$

et

$$(t-2)-(r-1), (t-2)-(r-1)-1, \dots, (t-2)-2(r-1)+1$$

ce qui montre bien que C_k génère des signaux d'égalité aux instants

$$t-r, t-r-1, \dots, t-2r+1.$$

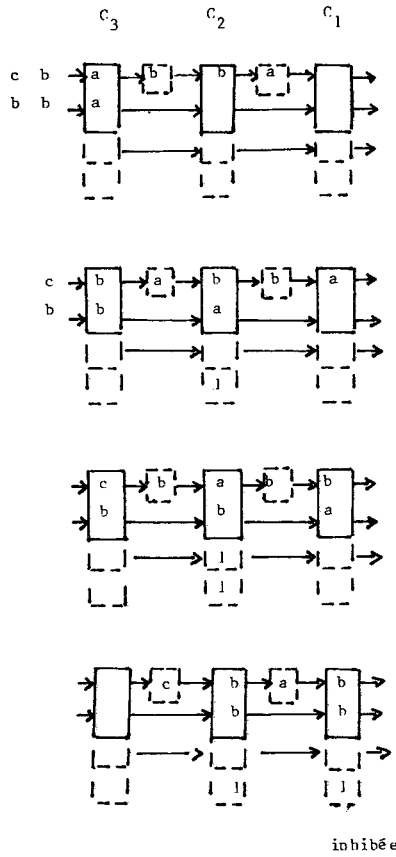


Figure 6

Remarque. — Si les r signaux d'égalité générés par C_k dans le lemme 3, aux instants $t - 2r + 1, \dots, t - r - 1, t - r$, correspondent à

$$x_i = x_{i+k}, x_{i+1} = x_{i+k+1}, \dots, x_{i+r-1} = x_{i+k+r-1}$$

alors on s'aperçoit qu'à l'instant t , la cellule C_{k-r} contient x_{i+r-1} et x_{i+k-1} ; en conséquence, le test inhibé dans C_{k-r} à l'instant t est $(x_{i+r-1} = x_{i+k-1})$.

Nous sommes maintenant en mesure d'énoncer le résultat principal.

PROPOSITION 1 : *Soit h, k deux entiers tels que $1 \leq h < k < 2h$. Si C_k inhibe C_h et l'empêche de reconnaître un carré de longueur $2h$, alors le mot traité, contient un carré de longueur inférieure à $2h$.*

Démonstration : Notons $k = h + r$, $1 \leq r < h$. D'après la remarque précédente, on a la situation suivante :

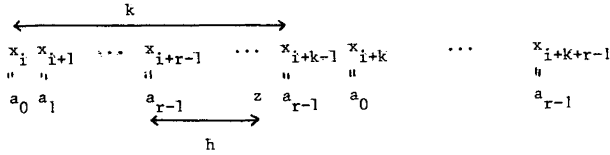


Figure 7

Notons y le carré de longueur $2h$ inhibé dans C_h .

Cas 1 : y ne contient pas de symbole x_j tel que $j > i + k + r - 1$ ou $j < i$. On a la situation illustrée dans la figure 8.

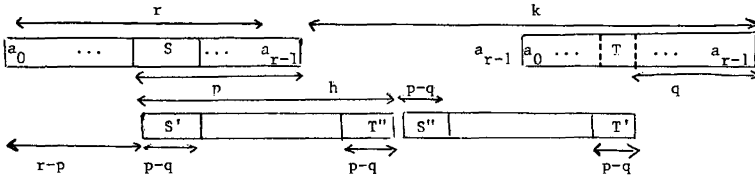


Figure 8. — Les rectangles du haut correspondent aux égalités détectées par C_k .
Les rectangles du bas correspondent au carré inhibé dans C_h .

On a

$$q + 2h + r - p = k + r$$

donc

$$p - q = 2h - k = h - r$$

ce qui montre que

$$q < p \quad \text{et} \quad p - q < h.$$

Dans la figure 8 on voit que :

- S et S' représentent le même facteur.
- S et T sont égaux puisque les rectangles du haut correspondent aux égalités détectées par C_k .
- T et T' représentent le même facteur.
- $T' = T''$ et $S' = S''$ puisque les rectangles du bas correspondent au carré inhibé dans C_h .

Tout ceci montre que $T' = S''$. En conséquence, $T' S''$ est un carré de longueur $2(p - q) < 2h$.

Cas 2: y contient des x_j tels que $j < i$.

Le carré inhibé dans C_h est de la forme $S a_0 \dots a_{r-1} T S a_0 \dots a_{r-1} T$, où $x_{i+r-1} = x_{i+k-1} = a_{r-1}$, et le facteur T de droite est un préfixe de $x_{i+k} \dots x_{i+k+r-1} = a_0 \dots a_{r-1}$. On voit alors que x contient le carré $a_0 a_1 \dots a_{r-1} a_0 a_1 \dots a_{r-1}$, de longueur $2r < 2h$, représenté dans la figure 9 par les deux rectangles en pointillé.

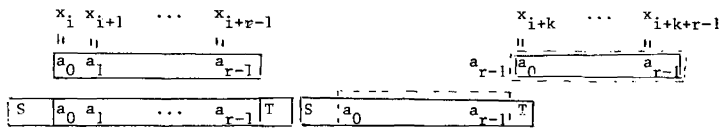


Figure 9. — Les rectangles du haut correspondent aux égalités détectées par C_k .
Les rectangles du bas correspondent au carré inhibé dans C_h .

Cas 3: y contient un symbole x_j tel que $j > i + k + r - 1$.

Ceci est analogue au cas 2. En examinant les deux rectangles de gauche dans la figure 10, on voit que la chaîne traitée contient le carré $a_0 a_1 \dots a_{r-1} a_0 a_1 \dots a_{r-1}$ de longueur $2r < 2h$ représenté sur la figure 10 par les deux rectangles en pointillé.

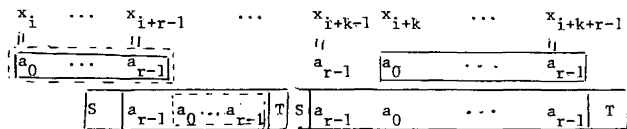


Figure 10. — Les rectangles du haut correspondent aux égalités détectées par C_k .
Les rectangles du bas correspondent au carré inhibé dans C_h .

Les résultats que nous avons établis peuvent se résumer dans le théorème suivant :

THÉORÈME : *La reconnaissance des mots sans carré de longueur au plus $n = 2p$, peut se faire en temps $T = 3n$ sur un réseau uni-dimensionnel de taille p , dans lequel deux cellules C_k, C_h partagent le même couloir de propagation si et seulement si $\lceil \log_2 h \rceil = \lceil \log_2 k \rceil$.*

Démonstration : Supposons que $2^{k-1} < n \leq 2^k$. Nous construisons un réseau dans lequel deux cellules $C_i, C_j, i \neq j$, partagent le même couloir de propagation si et seulement si $\lceil \log_2 i \rceil = \lceil \log_2 j \rceil$ (cf. fig. 11 a). Ce réseau comporte k couloirs de propagation distincts, que l'on peut replier de manière à obtenir

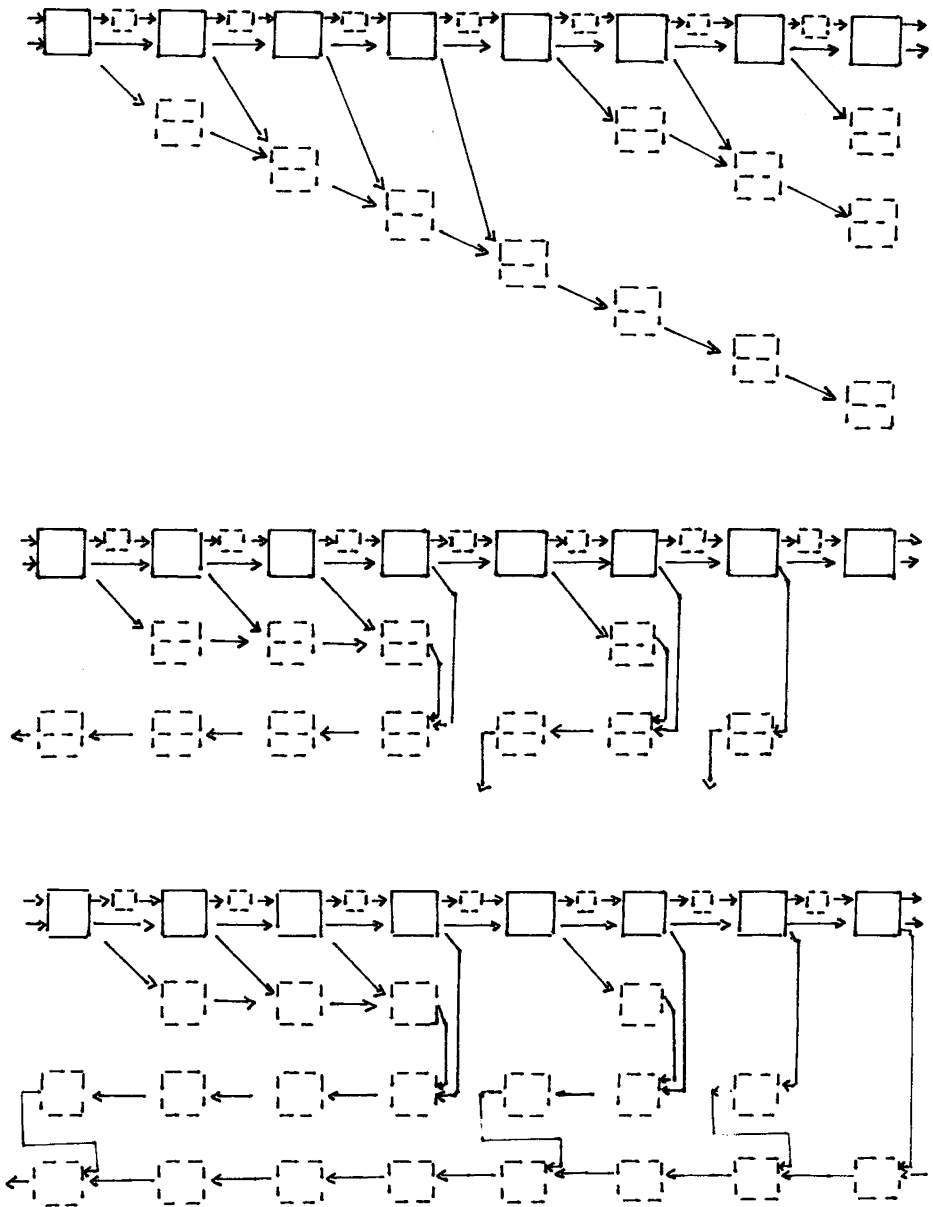


Figure 11

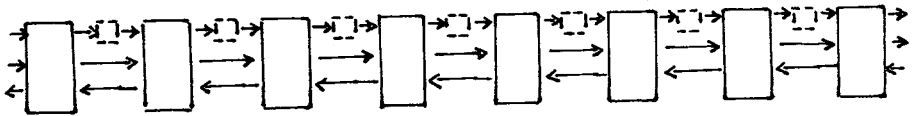
un réseau linéaire bidirectionnel (cf. *fig. 11 b*). D'après les résultats précédents, il nous suffit de montrer que ce réseau détecte les carrés de longueur minimum, lorsqu'il en existe.

Soit donc m la longueur minimum d'un carré de x ; si dans le réseau, la cellule C_m est inhibée, et que cela l'empêche de reconnaître un carré de longueur m , alors, d'après le lemme précédent, x contient un carré de longueur inférieure à m , ce qui est impossible. Ceci montre que le réseau de la figure 11 *b* permet de reconnaître les carrés de longueur minimum.

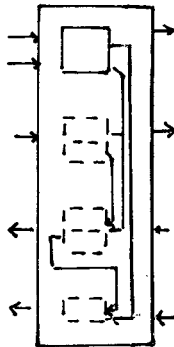
Dans le réseau de la figure 11 *b*, les signaux de sortie sont délivrés par les cellules $C_i, i = 1, 2, 4, \dots, n = 2^k$. Pour obtenir une solution dans laquelle seule la cellule C_p communique avec l'extérieur, il suffit d'ajouter un couloir de sortie, ce qui conduit au réseau de la figure 11 *c*.

En conclusion, la reconnaissance des mots sans carré peut être résolue en temps linéaire, sur un réseau linéaire à connexions bidirectionnelles, dont chaque processeur est programmable et a la structure suivante (cf. *fig. 12*) :

1. un module de traitement des tests ($x_i = x_j$);
2. un registre de mémorisation temporaire;
3. deux cellules de propagation contenant chacune deux registres;



Structure générale du réseau



Structure de la cellule de base

Figure 12

4. une cellule de propagation comportant un seul registre.

Dans ce réseau, le comportement d'une cellule dépend de sa position, et les signaux véhiculés sont les suivants :

- les symboles de l'alphabet $x_i, i = 1, \dots, n;$
- les signaux d'égalité et de destruction, $e = 1,$ et $d = 0.$

Seule la cellule C_p communique avec l'extérieur, et le temps d'exécution est $T = 3n;$ ce temps se décompose en

- $T_1 = 2n$ pour les calculs
- $T_2 = n$ pour la transmission des résultats en sortie.

REMERCIEMENTS

Nous remercions vivement le rapporteur dont les nombreuses remarques ont permis d'améliorer la présentation de cet article.

BIBLIOGRAPHIE

1. A. APOSTOLICO et A. NEGRO, *Systolic Algorithms for String Manipulations*, I.E.E.E. TC, C 33, 4, 1984, p. 361-364.
2. A. APOSTOLICO et F. P. PREPARATA, *Optimal Off-Line Detection of Repetitions in a String*, Theor. Comp. Sci., vol. 22, 1983, p. 297-315.
3. M. CROCHEMORE, *An Optimal Algorithm for Computing the Repetitions in a String*, Information processing letters, vol. 12, 1981, p. 244-250.
4. M. CROCHEMORE, *Recherche linéaire d'un carré dans un mot*, C.R. Acad. Sci. Paris, t. 296, série I, 1983, p. 781-784.
5. H. T. KUNG, *Why Systolic Architectures*, Computer Magazine, vol. 15, n° 1, janvier 1982, p. 37-46.
6. M. MAIN et R. LORENTZ, *Linear Time Recognition of Square-Free Strings*, dans Proceedings of the Nato Advanced Research Workshop on Combinatorial Algorithms on Words, Maratea, Italy, 1984, p. 271-278.