RAIRO
INFORMATIQUE THÉORIQUE

JOFFROY BEAUQUIER

Un générateur inhéremment ambigu du cône des langages algébriques

RAIRO – Informatique théorique, tome 12, n° 2 (1978), p. 99-108.

<http://www.numdam.org/item?id=ITA_1978__12_2_99_0>

R.A.I.R.O. Informatique théorique/Theoretical Computer Science
(vol. 12, n° 2, 1978, p. 99 à 108)

UN GÉNÉRATEUR INHÉREMMENT AMBIGU DU CÔNE DES LANGAGES ALGÉBRIQUES (*)

par Joffroy BEAUQUIER

Communiqué par J. BERSTEL

Résumé. — *Nous présentons un exemple d'un langage algébrique (context-free), à la fois générateur du cône rationnel des langages algébriques et inhéremment ambigu, qui n'est pas la simple union d'un générateur algébrique et d'un langage inhéremment ambigu, écrits sur des alphabets disjoints.*

Cet exemple n'est connu, à ce jour, ni en tant que générateur algébrique, ni en tant que langage inhéremment ambigu.

1. INTRODUCTION

Le fait, pour un langage algébrique (context-free), considéré comme modèle d'un langage de programmation, d'être inhéremment ambigu, constitue un inconvénient. Cette propriété indique, en effet, que, quel que soit le processus d'analyse (c'est-à-dire la grammaire algébrique) choisi, le langage contient un mot admettant deux analyses distinctes. Aussi, l'étude des langages algébriques inhéremment ambigus est-elle, depuis ses origines, un point fondamental de la Théories des Langages Algébriques.

Plusieurs exemples de langages inhéremment ambigus sont apparus dans différents articles ([6, 5, 8]). C'est un nouvel exemple de langage inhéremment ambigu que nous présentons ici. Cependant, il se différencie des exemples classiques pour des raisons que nous allons maintenant préciser.

Tous les exemples classiques de langages inhéremment ambigus ont la propriété de n'être pas des langages algébriques « compliqués », en ce sens qu'ils peuvent être engendrés par des grammaires linéaires (c'est-à-dire telles que chaque règle admette au plus un non-terminal dans son membre droit). Si l'on utilise la notion de transduction rationnelle (*cf.* [2]) pour comparer les complexités de structure de deux langages algébriques, il existe une autre manière d'exprimer cette constatation. Elle consiste à dire que tous les exemples classiques sont placés bas dans la hiérarchie rationnelle, c'est-à-dire dans la hiérarchie associée à la relation de préordre, entre langages : « dominer par transduction rationnelle ».

(*) Reçu janvier 1978.

(1) Institut de Programmation et Laboratoire associé au C.N.R.S. 248 : Informatique théorique et Programmation.

Néanmoins, il existe un procédé simple pour obtenir un langage inhéremment ambigu rationnellement supérieur à un langage L quelconque : il suffit pour cela de réaliser l'union de L et d'une copie, sur un alphabet disjoint, de l'un des exemples classiques.

Cette méthode est, bien sûr, valable pour obtenir un langage inhéremment ambigu qui est générateur du cône rationnel des langages algébriques, c'est-à-dire qui domine par transduction rationnelle, tout langage algébrique (dans la suite, et par abus de langage, nous parlerons tout simplement de générateur algébrique). Elle est, de plus, la seule actuellement connue pour obtenir un tel résultat. Considérons un exemple. Si l'on réalise l'union du langage de Dyck restreint $D_2'^*$, sur l'alphabet $\{ x, \bar{x}, y, \bar{y} \}$ et du langage inhéremment ambigu de Parikh :

$$L_P = \{ a^n b^r a^n b^s \mid n, r, s \in \mathbf{N} \} \cup \{ a^r b^n a^s b^n \mid n, r, s \in \mathbf{N} \}$$

on obtient un générateur algébrique inhéremment ambigu. Or, la partie du langage obtenu qui fournit le caractère générateur n'interfère pas avec la partie qui fournit le caractère ambigu, puisque ces deux parties sont sur des alphabets disjoints. Ceci montre combien cette méthode est artificielle.

Une réponse, que nous jugerions plus satisfaisante, au problème de l'obtention d'un générateur algébrique inhéremment ambigu, serait donnée par un exemple où ces deux parties seraient plus intimement mêlées. On peut donc se poser le problème de l'existence d'un générateur algébrique inhéremment ambigu, dans lequel la partie génératrice ne puisse pas être séparée par simple projection, de la partie fournissant le caractère ambigu.

Or, un résultat de [1] fixe une limite à ce que l'on peut attendre d'une telle recherche :

THÉORÈME : *Pour tout générateur algébrique G, il existe un langage rationnel K tel que $G \cap K$ soit un générateur algébrique non ambigu.*

Ainsi, même dans des exemples « compliqués » de générateurs algébriques inhéremment ambigus, la partie génératrice et la partie inhéremment ambiguë peuvent être séparées par un langage rationnel.

Le but de cet article est de présenter un exemple de générateur algébrique inhéremment ambigu, pour lequel les parties « génératrice » et « ambiguë » ne peuvent être séparées par simple projection, mais seulement par une intersection rationnelle non triviale. Cet exemple constitue à la fois un nouveau générateur du cône des langages algébriques et un nouveau langage inhéremment ambigu.

2. RÉSULTATS

Soit $L \subseteq \{ a, b \}^*$ le langage engendré par la grammaire algébrique $G = \langle \{ a, b \}, \{ S \}, S, (S \to a\,S\,b\,S\,a,\ S \to 1) \rangle$. Nous allons montrer :

PROPOSITION 1 : *L est un générateur algébrique.*

PROPOSITION 2 : *L est un langage algébrique inhéremment ambigu.*

3. PREUVE DE LA PROPOSITION 1

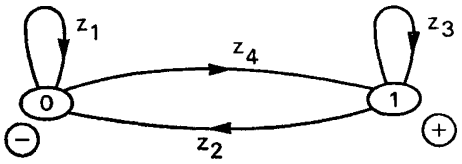
Cette preuve, sans difficultés, consiste à vérifier la propriété suivante. Soit $E \subseteq X^* = \{ z_1, z_2, z_3, z_4 \}^*$ le langage algébrique engendré par la grammaire $G_E = \langle X, \{ S \}, S, P \rangle$ dont les règles sont :

$$S \to z_1\,S\,z_2\,S\,z_3; \qquad S \to z_4.$$

On sait que E est un générateur algébrique [7]. Soit ψ le morphisme de X^* dans $\{ a, b \}^*$ défini par

$$\psi(z_1) = aaab; \qquad \psi(z_2) = aba; \qquad \psi(z_3) = baaa; \qquad \psi(z_4) = a$$

et soit R le langage rationnel reconnu par l'automate fini :



On vérifie alors sans peine que

$$\psi^{-1}\,L \cap R = E,$$

ce qui suffit à assurer que L est lui-même générateur algébrique. ■

4. UN LEMME

Dans la preuve de la proposition 2, nous utilisons le lemme suivant :

LEMME 1 : *Soit $Y = \{ x, y, \#_1, \#_2 \}$ et soit $L' \subset Y^*$ le langage défini par*

$$L' = \{ \#_1\,x^n\,y^{n+1}\,x^{m+1}\,y^m\,\#_2 \mid n \geqq 1,\ m \geqq 0 \}$$
$$\cup \{ \#_1\,x^n\,x^m\,y^m\,y^p\,x^p\,y^n\,\#_2 \mid n,\ m,\ p \geqq 1 \}.$$

L' est un langage algébrique inhéremment ambigu.

Preuve du lemme 1. *Remarque préliminaire :* D'un point de vue strictement technique, la preuve que nous donnons de l'ambiguïté inhérente du langage L', diffère des preuves habituelles. Car si, comme certaines d'entre elles, la preuve fait appel au classique Lemme d'Ogden [5] pour exhiber des arbres de dérivation distincts, elle s'en différencie par le fait que ce Lemme doit, ici, être utilisé de manière répétitive et dans le langage élargi, ce qui introduit un élément de complexité supplémentaire dans la preuve.

Soit G' une grammaire quelconque engendrant L' et soit N l'entier que lui associe le Lemme d'Ogden.

Considérons le mot f de L' : $f = \#_1 x^{3.N!} x^{N.N!} y^{N.N!} y^{N!+1} x^{N!+1} y^{3.N!} \#_2$ et distinguons dans f les N premières occurences du bloc de y le plus à droite. Le lemme d'Ogden nous fournit une paire itérante grammaticale [3] (h_1, u, h_2, v, h_3) du mot f. Faisons trois remarques :

(i) puisque L' est un langage borné, les facteurs u et v ne sont constitués que d'une seule espèce de lettres, x ou y;

(ii) si $u = y^p$ $(p > 0)$ alors, nécessairement, h_2, v et h_3 appartiennent à y^*, ce que contredit le fait que le mot $h_1 u^2 h_2 v^2 h_3$ appartient à L'. C'est donc que $v = y^p$ $(p > 0)$ et $u = x^q$;

(iii) les entiers p et q sont égaux, puisque tout mot w de L' vérifie :

$$|w|_x = |w|_y$$

($|w|_z$ désignant le nombre d'occurrences de la lettre z dans w). Le mot f admet deux blocs constitués de lettres x, que nous appelons blocs 1 et 2 :

$$f = \#_1 \underbrace{x^{3.N!} x^{N.N!}}_{1} y^{N.N!} \underbrace{y^{N!+1}}_{2} x^{N!+1} y^{3.N!} \#_2.$$

Supposons que u soit une occurrence de facteur du bloc 2. Alors :

$$h_1 u^2 h_2 v^2 h_3 = \#_1 x^{3.N!} x^{N.N!} y^{N.N!} y^{N!+1} x^{N!+1+p} y^{3.N!+p} \#_2,$$

n'appartient pas à L'. Cette hypothèse est donc à rejeter. C'est donc que u est une occurrence de facteur du bloc 1 (*fig.* 1).

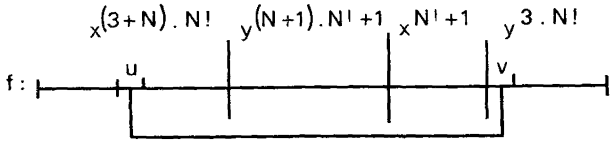


Figure 1.

Puisque le Lemme d'Ogden est valable pour les mots du langage élargi $\hat{L}'$, il peut être appliqué de manière répétitive. Distinguons maintenant les N premières occurrences du premier bloc de y (*fig.* 2).

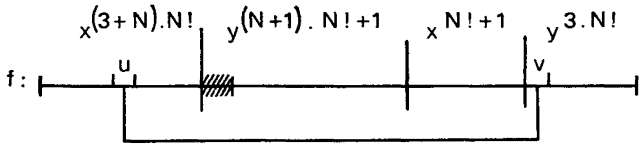


Figure 2.

le Lemme d'Ogden fournit une paire itérante grammaticale $(h'_1, u', h'_2, v'_1 h'_3)$, compatible avec la première que nous avons obtenue (en ce sens que les croisements de paires sont impossibles). Compte tenu des remarques précédentes, deux hypothèses sont envisageables :

(1) $u' = y^q$, $v' = x^q$.

(2) $u' = x^q$, $v' = y^q$.

Nous allons démontrer que l'éventualité (1) ne peut pas se produire. Dans le cas contraire, en effet, v' est nécessairement une occurrence de facteur du bloc 2.

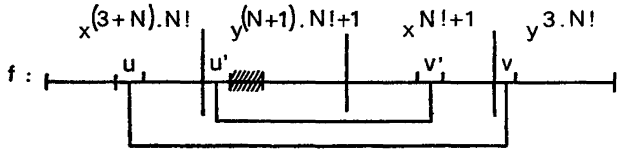


Figure 3.

Distinguons alors les N occurrences de y situées immédiatement à droite de u' dans le premier bloc de y (*fig.* 3). Nous pouvons refaire le raisonnement précédent et nous obtenons une troisième paire dont les facteurs itérants u'' et v'' sont à l'intérieur des facteurs itérants u', v' (*fig.* 4).

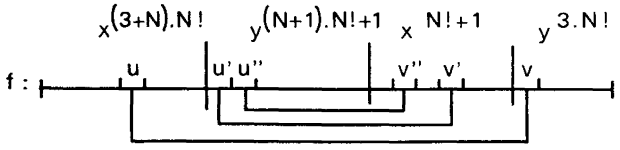


Figure 4.

Ce processus de marquage peut évidemment être itéré. Au bout d'un certain nombre d'itérations, toutes les occurrences du bloc 2 de x ou bien participeront

à une paire itérante, ou bien seront placées entre deux éléments itérants droits de deux paires déjà constituées. A cet égard, le cas le plus défavorable est représenté par le schéma suivant (*fig.* 5) :

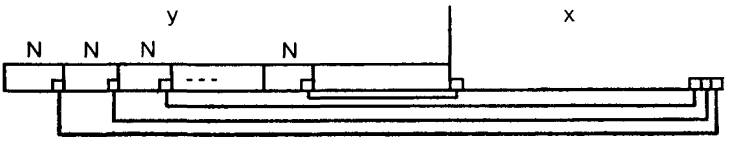


Figure 5.

Une itération du processus de marquage [possible car

$$N.(N!+1) < (N+1).N!+1]$$

fera apparaître une paire itérante dont les deux facteurs itérants figurent dans le bloc de y le plus à gauche, ce qui contredit la remarque (iii).

Donc, l'hypothèse (2) est la seule possible et la configuration est représentée par la figure 6 :

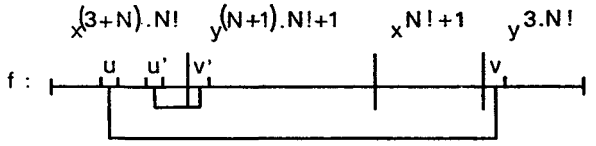


Figure 6.

Distinguons enfin les N dernières occurrences du deuxième bloc de x. Puisque le croisement de paires itérantes grammaticales est impossible, la paire itérante fournie par le lemme d'Ogden, $(h_2'', u'', h_2'', v'', h_3'')$ est telle que : $u'' = y^r$, $v'' = x^r$ et u'' est une occurrence de facteur du bloc de y le plus à gauche (*fig.* 7) :

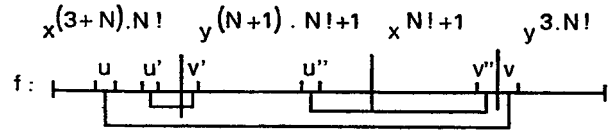


Figure 7.

Par des itérations successives des paires itérantes mises en évidence ci-dessus, nous concluons que le mot

$$f_1 = \#_1 x^{3.N.N!} x^{3.N.N!} y^{3.N.N!} y^{3.N.N!+1} x^{3.N.N!+1} y^{3.N.N!} \#_2$$

appartient à L'.

Considérons maintenant un second mot, f', de L' :

$$f' = \#_1 x^{N!} y^{N!+1} x^{2.N!+1} y^{2.N!} \#_2$$

et distinguons les N dernières occurrences du bloc de y le plus à droite (*fig.* 8) :

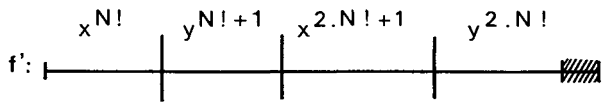


Figure 8.

La paire itérante grammaticale (h_1, u, h_2, v, h_3) fournie par le lemme d'Ogden est, compte tenu des remarques ci-dessus, telle que : $v = y^r$ et $u = x^r$ $(r > 0)$. u ne peut être une occurrence du bloc de x le plus à gauche, car le mot $h_1 u^2 h_2 v^2 h_3$ n'appartient alors pas à L' $(r < N)$. Donc u est une occurrence de facteur du bloc de x le plus à droite.

Distinguons les N premières occurrences du bloc de x le plus à gauche (*fig.* 9) :

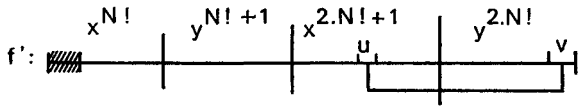


Figure 9.

La paire itérante $(h'_1, u', h'_2, v', h'_3)$ donnée par le lemme d'Ogden vérifie : $u' = x^s$, $v' = y^s$. Pour la même raison que ci-dessus, v' ne peut être une occurrence du second bloc de y :

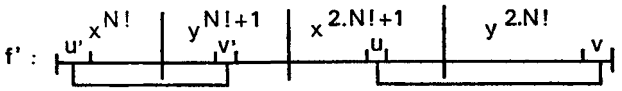


Figure 10.

En itérant les deux paires représentées par la figure 9, on génére le mot

$$f_1 = \#_1 x^{6.N.N!} y^{6.N.N!+1} x^{3.N.N!+1} y^{3.N.N!} \#_2.$$

Examinons maintenant les deux sous-arbres de dérivation du mot f_1, que nous avons mis en évidence (*fig.* 11).

Ces deux sous-arbres ne peuvent être deux sous-arbres d'un même arbre de dérivation du mot f_1, car, dans un même arbre de dérivation, on ne peut avoir deux paires grammaticales croisées, comme il en apparaît dans la figure 11.

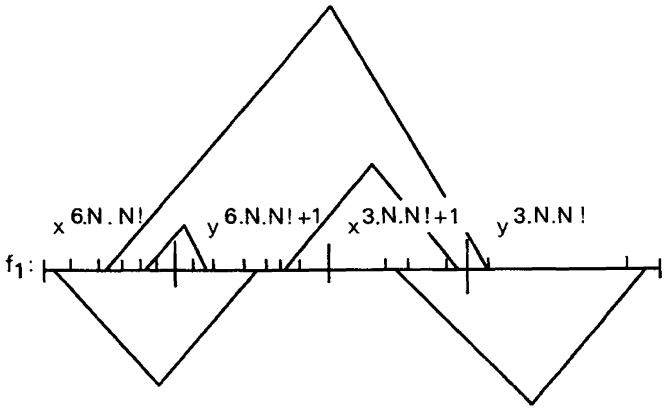
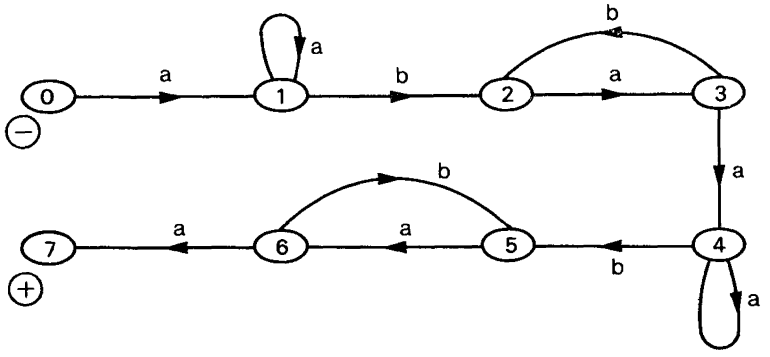


Figure 11.

Par conséquent, le mot f_1 admet dans la grammaire G', (au moins) deux arbres de dérivation distincts. G' est donc une grammaire ambiguë. Nous avons prouvé que toute grammaire engendrant L' est ambiguë : L' est un langage inhéremment ambigu. ■

5. PREUVE DE LA PROPOSITION 2

Soit $K \subseteq \{ a, b \}^*$ le langage rationnel reconnu par l'automate fini suivant :



A partir de la grammaire G engendrant L, nous construisons la grammaire canonique (*cf.* [4]) G'_K engendrant le langage $L \cap K$, puis nous la réduisons, obtenant ainsi une grammaire G_K. Notre convention est la suivante :

$$ {}_iS_j \xrightarrow{\star} f \quad \Leftrightarrow \quad S \xrightarrow{\star} f \qquad \text{et} \quad i \circ f = j. $$

Puisque 0 est état initial de K, 7 état terminal et S axiome de G, l'axiome de G_K est donc ${}_0S_7$. Après calcul, on obtient :

$$G_K = \langle \{a, b\}, \{{}_0S_7, {}_4S_6, {}_1S_4, {}_1S_3, {}_2S_4, {}_2S_6, {}_3S_6, {}_1S_6\}, {}_0S_7, P_K \rangle,$$

où P_K est constitué de l'ensemble des règles suivantes :

$$\begin{cases} {}_0S_7 \to abaa\,{}_4S_6\,babaa, \\ {}_4S_6 \to a\,{}_4S_6\,ba + aba, \end{cases}$$

$$\begin{cases} {}_0S_7 \to a\,{}_1S_4\,babaa, \\ {}_1S_4 \to a\,{}_1S_3\,b\,{}_2S_4\,a + a\,{}_1S_3\,babaa + ab\,{}_2S_4\,a + ababaa, \\ {}_1S_3 \to a\,{}_1S_3\,ba + aba, \\ {}_2S_4 \to ab\,{}_2S_4\,a + ababaa, \end{cases}$$

$$\begin{cases} {}_0S_7 \to a\,{}_1S_3\,b\,{}_2S_6\,a, \\ {}_2S_6 \to a\,{}_3S_6\,ba, \\ {}_3S_6 \to a\,{}_4S_6\,ba + aba, \end{cases}$$

$$\begin{cases} {}_0S_7 \to a\,{}_1S_6\,babaa, \\ {}_1S_6 \to a\,{}_1S_4\,ba + a\,{}_1S_6\,ba. \end{cases}$$

[L'énumération ci-dessus des règles de G_K suggère l'ordre dans lequel se fait la construction. Par exemple, pour aller des états 0 à 7 en lisant successivement a, S, b, S et a on peut faire :

$${}_0a_1\,{}_1S_{**}b_{+\ +}S_6\,{}_6a_7,$$

d'où quatre valeurs possibles pour le couple $(*, +)$: (1, 2), (3, 2), (4, 5) et (6, 5), correspondant aux quatre sous-ensembles de règles représentés].
Au vu de la grammaire G_K, $L \cap K$ se présente comme une quadruple union :

$$\begin{aligned} L \cap K = &\{abaaa^n(ba)^n babaa \mid n \geqq 0\} \\ &\cup \{a^2 a^n (ba)^n b (ab)^m a^m ababaa \mid n \geqq 0,\ m \geqq 1\} \\ &\cup \{aa^n (ba)^n ba^2 a^m (ba)^m (ba)^2 a \mid n \geqq 1,\ m \geqq 0\} \\ &\cup \{aa^n a^2 a^m (ba)^m b (ab)^p a^p aba (ba)^n (ba)^2 a \mid n,\ m,\ p \geqq 1\}. \end{aligned}$$

Posons

$$K' = aa^+ (ba)^+ a^+ (ba)^2 (ba)^+ a \qquad (\text{où } w^+ = w^* \setminus 1).$$

Il vient

$$\begin{aligned} L \cap K' = &\{a^2 a^n (ba)^n ba^2 a^m (ba)^m (ba)^2 a \mid n \geqq 1,\ m \geqq 0\} \\ &\cup \{a^2 a^n a^m (ba)^m b (ab)^p a^p a (ba)^n (ba)^2 a \mid n,\ m,\ p \geqq 1\} \end{aligned}$$

Soit φ le morphisme de $\{ x, y, \#_1, \#_2 \}^*$ dans $\{ a, b \}^*$ défini par

$$\varphi \#_1 = a^2; \qquad \varphi \#_L = (ba)^2 a; \qquad \varphi x = a; \qquad \varphi y = ba.$$

Il vient

$$\varphi^{-1}(L \cap K') = \{ \#_1 x^n y^{n+1} x^{m+1} y^m \#_2 \mid n \geqq 1, m \geqq 0 \}$$
$$\cup \{ \#_1 x^n x^m y^m y^p x^p y^n \#_2 \mid n, m, p \geqq 1 \} = L'.$$

D'après le lemme 1, L' est un langage inhéremment ambigu. Puisque la famille des langages non ambigus est fermée par homomorphisme inverse et intersection avec un langage rationnel [4], nous en déduisons que L est lui-même un langage inhéremment ambigu. ■

BIBLIOGRAPHIE

1. J. BEAUQUIER, *Générateurs algébriques non ambigus* in *Automata, Languages and Programming*, S. MICHAELSON et R. MILNER (éd.) Edinburgh University Press, 1976. p. 66-73.
2. J. BERSTEL, *Transductions and Context-Free Languages*, Teubner Verlag, 1978.
3. L. BOASSON, *Langages Algébriques, Paires Itérantes et Transductions Rationnelles*, Theoretical Computer Science ,Vol. 2, 1976, p. 209-223.
4. S. GINSBURG, *The Mathematical Theory of Context-free Languages*, McGraw Hill, 1966.
5. W. OGDEN, *A Helpful Result for Proving Inherent Ambiguity*, Math. Syst. Theory, Vol. 2, 1967, p. 191-194.
6. R. J. PARIKH, *On Context-Free Languages*, J. Assoc. Comput. Math., Vol. 13, 1966, p. 570-581.
7. M. P. SCHUTZENBERGER, *Sur un langage équivalent au langage de Dyck*, in *Logic, Methodology and Philosophy of Sciences*, Vol. IV, 1973, p. 197-203. North-Holland.
8. E. SHAMIR, *Some Inherently Ambiguous Context-Free Languages*, Information and Control, Vol. 18, 1971, p. 355-363.