

RAIRO

INFORMATIQUE THÉORIQUE

JEAN-MICHEL AUTEBERT

JOFFROY BEAUQUIER

LUC BOASSON

MICHEL LATTEUX

Indécidabilité de la condition IRS

RAIRO – Informatique théorique, tome 16, n° 2 (1982), p. 129-138.

http://www.numdam.org/item?id=ITA_1982__16_2_129_0

© AFCET, 1982, 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/legal.php>). 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/>

INDÉCIDABILITÉ DE LA CONDITION IRS (*)

par Jean-Michel AUTEBERT ⁽¹⁾, Joffroy BEAUQUIER ⁽²⁾,
Luc BOASSON ⁽³⁾ et Michel LATTEUX ⁽⁴⁾

Communiqué par la Rédaction

Résumé. — *Nous démontrons :*

- 1) *il est indécidable de déterminer si un langage linéaire est un langage IRS;*
- 2) *il est décidable de déterminer si un langage déterministe est un langage IRS;*
- 3) *il existe un langage linéaire IRS qui contient le produit de deux langages algébriques infinis.*

Abstract. — *We prove:*

- 1) *for any linear language L , it is recursively unsolvable to determine whether L is an IRS language;*
- 2) *for any deterministic context-free language L , it is recursively solvable to determine whether L is an IRS language;*
- 3) *there exists an IRS linear language containing the concatenation product of two infinite context-free languages.*

INTRODUCTION

La plupart des problèmes importants de la théorie des langages algébriques sont indécidables. Citons, par exemple, les problèmes de la rationalité, de l'égalité ou de la disjonction de langages algébriques (*cf.* [1, 6]). Si l'on se restreint à la sous-famille des langages déterministes, certains de ces problèmes deviennent décidables, comme celui de la rationalité [12], mais d'autres restent indécidables, comme ceux de l'inclusion ou de la disjonction [14, 4]. C'est pourquoi différents auteurs ont défini des sous-classes encore plus restreintes, de manière à obtenir des résultats de décidabilité positifs. Citons les langages super-déterministes de Friedmann et Greibach [10], les langages reconnus par les automates non singuliers de Valiant [13] ou les langages reconnus par les automates à pile à un nombre fini de tours de Ginsburg et Spanier [7].

(*) Reçu février 1980 et dans sa forme révisée novembre 1980.

⁽¹⁾ L.I.T.P.

⁽²⁾ L.I.T.P. et Université de Picardie.

⁽³⁾ L.I.T.P. et Université de Paris-VII.

⁽⁴⁾ Université de Lille-I.

Pour toute correspondance : Laboratoire Informatique théorique et Programmation, Université Paris-VII, U.E.R. de Mathématiques, 2, place Jussieu, 75221 Paris Cedex 05.

Ce sont pour des raisons toute différentes, propres à la théorie des langages elle-même, qu'ont été étudiées plusieurs familles « naturelles » de langages : langages linéaires, à un compteur, quasi rationnels. Dans ces familles, il apparaît qu'une condition particulière, baptisée condition IRS dans [9], joue un rôle très important. Un langage L vérifie la condition IRS s'il ne contient pas de langage rationnel infini. Greibach a, par exemple, montré dans [9] que le fait d'imposer à un langage à un compteur de vérifier la condition IRS entraînait la quasi rationalité de ce langage. La condition IRS est étudiée dans plusieurs articles ([2, 3, 5]).

Ainsi, puisque la condition IRS joue un rôle important dans la famille des langages algébriques, il est naturel de se poser à propos de la famille des langages algébriques qui vérifient cette condition, les mêmes questions que pour les familles classiques, et, en particulier, les questions de décidabilité. Rappelons que la question de savoir si un langage algébrique satisfait la condition IRS est indécidable [6]. Dans cet article, nous précisons ce résultat en établissant qu'il vaut pour les langages linéaires.

Notons qu'ici, un parallèle peut être fait avec le problème de la vacuité de l'intersection de deux langages algébriques, dont l'indécidabilité est prouvée sur les langages linéaires. Comme bien souvent, si l'on se restreint au cas des langages déterministes, alors le problème devient décidable. La proposition 2 énonce que l'on peut décider si un langage déterministe satisfait la condition IRS.

Notons, enfin, qu'à l'occasion des traitements concernant les langages linéaires, nous sommes amenés à résoudre certains problèmes techniques et à faire apparaître un objet un peu surprenant : il s'agit d'un langage satisfaisant la condition IRS qui, bien que linéaire, contient le produit de deux langages algébriques infinis.

1. DÉFINITIONS ET RAPPELS

Une grammaire est un triple $\langle X, V, P \rangle$ où X (alphabet terminal) et V (alphabet non terminal) sont deux alphabets finis disjoints et P est un ensemble fini de règles de réécriture. Une grammaire est dite algébrique (ou context-free) si chaque membre gauche de règle est réduit à un non-terminal, c'est-à-dire si P est un sous-ensemble de $V \times (X \cup V)^*$. Une grammaire algébrique est dite linéaire si, de plus, chaque membre droit de règle contient au plus un non-terminal, c'est-à-dire si P est un sous-ensemble de $V \times (X^* V X^* \cup X^*)$.

Nous utilisons les définitions classiques (données dans [6]) de dérivation directe (notée \rightarrow), de dérivation (notée $\xrightarrow{*}$) et de langage engendré par la grammaire G à partir d'un axiome v appartenant à V [noté $L_G(v)$].

Un langage est dit algébrique (resp. linéaire) s'il existe une grammaire algébrique (resp. algébrique linéaire) qui l'engendre.

Une grammaire algébrique $G = \langle X, V, P \rangle$ est dite réduite pour l'axiome v_0 , si et seulement si les deux conditions suivantes sont satisfaites :

- (i) $\forall v \in V, L_G(v) \neq \emptyset$;
- (ii) $\forall v \in V, \exists f, g \in X^*, v_0 \xrightarrow{*} fvg$.

On sait que tout langage algébrique peut être engendré par une grammaire algébrique réduite.

Une grammaire algébrique G , d'axiome v_0 est ambiguë s'il existe un mot f de $L_G(v_0)$ admettant dans G deux dérivations gauches distinctes. Un langage algébrique L est ambigu si toute grammaire algébrique qui l'engendre est ambiguë.

Nous noterons Alg la famille des langages algébriques et Lin la famille des langages linéaires. On sait que : $\text{Lin} \not\subseteq \text{Alg}$.

Outre ces définitions classiques, posons :

DÉFINITION 1 [9] : *Un langage est dit IRS s'il ne contient aucun langage rationnel infini.*

2. RÉSULTATS

Nous commençons par démontrer un lemme qui caractérise les grammaires algébriques engendrant un langage qui n'est pas IRS, c'est-à-dire qui contient un langage rationnel infini.

LEMME 1 : *Une grammaire algébrique $G = \langle X, V, P \rangle$ réduite, engendre un langage qui n'est pas IRS si et seulement si on peut trouver :*

– soit (1) une dérivation :

$$v \xrightarrow{*} fv \quad \text{où } v \in V \text{ et } f \in X^+;$$

– soit (2) une dérivation :

$$v \xrightarrow{*} vf \quad \text{où } v \in V \text{ et } f \in X^+;$$

– soit (3) deux dérivations :

$$v \xrightarrow{*} (f_1 f_2)^p f_1 \quad \text{où } v \in V, f_1 \in X^*,$$

$$f_2 \in X^+, p \geq 0 \quad \text{et} \quad v \xrightarrow{*} (f_1 f_2)^m v (f_2 f_1)^n \quad \text{où } m, n > 0.$$

Preuve : La condition suffisante est triviale. Puisque la grammaire G (que nous supposons d'axiome v_0) est réduite, il existe certainement une dérivation $v_0 \xrightarrow{*} gvh$ où $g, h \in X^*$ et une dérivation $v \xrightarrow{*} k$ où $k \in X^*$.

Dans le cas (1) gf^*kh , dans le cas (2) gkf^*h et dans le cas (3) $g((f_1f_2)^{m+n})^*(f_1f_2)^pf_1h$ sont des langages rationnels infinis contenus dans $L_G(v_0)$.

Démontrons maintenant que la condition est nécessaire. D'après le lemme de l'étoile pour les langages rationnels, si $L = L_G(v_0)$ contient un langage rationnel infini, il contient un langage rationnel infini de la forme $K = g_1f^*g_2$ où $f \in X^+$, $g_1, g_2 \in X^*$. Une construction classique (cf. [6]) permet d'obtenir, à partir de la grammaire G , une grammaire $G_{\cap K} = \langle X, V_{\cap K}, P_{\cap K} \rangle$ engendrant $L \cap K$. Rappelons que $V_{\cap K}$ est l'ensemble des triplets (p, v, q) où p et q sont deux états de l'automate minimal reconnaissant K et v un non-terminal de G . Le classique lemme d'itération d'Ogden, permet d'associer à $G_{\cap K}$ un entier, soit N_0 .

Considérons le mot $g_1f^{N_0}g_2$ de $L \cap K$, distinguons dans ce mot les N_0 premières occurrences des facteurs f mis en évidence ci-dessus et utilisons le lemme d'Ogden. Ce lemme fournit deux dérivations $w \xrightarrow{*} h_1wh_2, w \xrightarrow{*} k$ telles que $w \in V_{\cap K}, h_1, h_2, k \in X^*, h_1h_2 \neq \varepsilon$ et :

$$(1) \quad g_1f^{N_0}g_2 = g_1h'_1h_1kh_2h'_2g_2,$$

$$(2) \quad \forall n \in \mathbb{N}, \quad g_1h'_1h_1^nkh_2^n h'_2g_2 \in L \cap K.$$

Si $h_2 = \varepsilon$ (resp. $h_1 = \varepsilon$) on obtient donc une dérivation $w \xrightarrow{*} h_1w$ (resp. $w \xrightarrow{*} wh_2$) dans $G_{\cap K}$. D'après la construction de la grammaire $G_{\cap K}$ cela entraîne qu'il existe une dérivation $v \xrightarrow{*} h_1v$ (resp. $v \xrightarrow{*} vh_2$) dans la grammaire G .

Si $h_1 \neq \varepsilon$ et $h_2 \neq \varepsilon$, on prouve aisément que la formule (2) entraîne que les mots h_1 et h_2 sont puissances conjuguées, c'est-à-dire que $h_1 = (f_1f_2)^m, h_2 = (f_2f_1)^n$ et donc que $k = (f_1f_2)^pf_1$.

Il existe donc dans $G_{\cap K}$ des dérivations de la forme :

$$w \xrightarrow{*} (f_1f_2)^pf_1; \quad w \xrightarrow{*} (f_1f_2)^mw(f_2f_1)^n.$$

De la construction de $G_{\cap K}$, il ressort qu'il existe dans G des dérivations :

$$v \xrightarrow{*} (f_1f_2)^pf_1; \quad v \xrightarrow{*} (f_1f_2)^mv(f_2f_1)^n. \quad \square$$

Dans [6], il est démontré la proposition suivante :

PROPOSITION 1 : *Il est récursivement indécidable de savoir si un langage algébrique, donné par une grammaire algébrique qui l'engendre, est un langage IRS.*

Dans [6], la preuve de cette proposition consiste à coder le problème de correspondance de Post dans un langage défini comme le produit de deux langages linéaires mais n'étant pas lui-même linéaire. Nous allons étendre le résultat implicitement contenu dans cette preuve, en démontrant :

PROPOSITION 1' : *Il est récursivement indécidable de savoir si un langage linéaire donné par une grammaire linéaire qui l'engendre est un langage IRS.*

Preuve : Nous allons démontrer que décider si un langage linéaire est IRS équivaut à décider si le problème de correspondance de Post n'a pas de solution. Prenons un alphabet X ayant au moins deux lettres et deux suites de p mots de X^+ , f_1, f_2, \dots, f_p et g_1, g_2, \dots, g_p .

Considérons le langage linéaire $L = L_G(t)$ engendré par la grammaire linéaire $G = \langle Z, \{t, u, v, w\}, P \rangle$, où :

$$Z = X \cup A \quad \text{avec} \quad A = \{a_1, \dots, a_p\} \quad \text{et} \quad X \cap A = \emptyset$$

et où P est l'ensemble de règles :

$$\begin{aligned} t &\rightarrow f_i t a_i + f_i a_j u g_j a_i + 1, & i, j \in \{1, \dots, p\}, \\ u &\rightarrow a_i u g_i + a_i f_j v a_j g_i, & i, j \in \{1, \dots, p\}, \\ v &\rightarrow f_i v a_i + f_i a_j w g_j a_i, & i, j \in \{1, \dots, p\}, \\ w &\rightarrow a_i w g_i + a_i f_j t a_j g_i, & i, j \in \{1, \dots, p\}. \end{aligned}$$

Le langage L dépend de l'entier p et des mots $f_1, \dots, f_p, g_1, \dots, g_p$. Lorsque nous faisons varier ces paramètres, L décrit une famille \mathcal{F} et, bien sûr, $\mathcal{F} \subset \text{Lin}$. Clairement G ne peut vérifier l'une des conditions (1) ou (2) du lemme 1. Par contre, G peut vérifier la condition (3) du lemme 1, de quatre manières différentes, selon que le non-terminal impliqué est t, u, v ou w . Nous allons traiter l'un des quatre cas, les autres pouvant se traiter de la même façon.

Supposons, donc, qu'il existe deux mots $f = (h_1 h_2)^m$, $g = (h_2 h_1)^n$ et une dérivation dans $G : t \xrightarrow{*} f t g$. Considérons les homomorphismes θ et θ' définis sur $A = \{a_1, \dots, a_p\}$ par :

$$\forall i \in \{1, \dots, p\}, \quad \theta(a_i) = f_i; \quad \theta'(a_i) = g_i.$$

Alors, f et g peuvent s'écrire :

$$(3) \quad f = (h_1 h_2)^m = f^{(1)} a^{(2)} f^{(3)} a^{(4)} \dots a^{(4i)} f^{(4i+1)},$$

$$(4) \quad g = (h_2 h_1)^n = a^{(4i+1)} g^{(4i)} a^{(4i-1)} \dots g^{(2)} a^{(1)},$$

avec, y^R désignant l'image miroir du mot y :

$$\forall j \in \{1, \dots, 2i\}, \quad g^{(2j)} = \Theta'((a^{(2j)})^R) \in X^+, \quad a^{(2j)} \in A^+$$

et :

$$\forall j \in \{0, \dots, 2i\}, \quad f^{(2j+1)} = \Theta((a^{(2j+1)})^R) \in X^+.$$

REMARQUE : La notation « modulo 4 » dans les décompositions de f et g sert à mettre en évidence les parties engendrées par chacun des quatre non-terminaux t, u, v et w de G . Un examen simple des deux décompositions (3) et (4) ci-dessus montre que :

$$h_1 \in X^+ Z^* A^+, \quad h_2 \in A^+ Z^* X^+,$$

ce qui entraîne à son tour, parce que les alphabets X et A sont disjoints, que $n=m$, et de plus $i > 0$.

Puisque f et g doivent respectivement commencer et se terminer par le même facteur, h_1 , il existe j tel que :

$$h_1 = f^{(1)} a^{(2)} \dots f^{(2j+1)} \alpha = \gamma a^{(2j+1)} \dots g^{(2)} a^{(1)},$$

avec :

$$\alpha \in A^+ \quad \text{et} \quad \gamma \in X^+.$$

Si $j \geq 1$, on en déduit $a^{(2)} = a^{(2j+1)}$ et $f^{(2j+1)} = g^{(2)}$. Donc le mot $w = a^{(2)} = a^{(2j+1)}$ est tel que $\Theta(w) = \Theta'(w)$. On a donc bien trouvé une suite d'indices i_1, i_2, \dots, i_r telle que : $f_{i_1} f_{i_2} \dots f_{i_r} = g_{i_1} g_{i_2} \dots g_{i_r}$ et le problème de correspondance de Post a une solution.

Si $j=0$ on arrive à la même conclusion, en posant $a^{(2)} = \beta\beta'$, $g^{(2)} = \delta\delta'$ et en considérant l'égalité :

$$(h_2 h_1)^{n-1} h_2 = \alpha' f^{(3)} \dots a^{(4i)} f^{(4i+1)} = a^{(4i+1)} g^{(4i)} \dots a^{(3)} \gamma,$$

où $\alpha' \in A^+$ et $\gamma \in X^+$.

Réciproquement, si le problème de correspondance de Post a une solution, il existe $a \in A^+$ tel que $y = \Theta(a) = \Theta'(a)$ et nous avons, dans G , la dérivation : $t \xrightarrow{*} ya^R ya^R t ya^R ya^R$, ce qui implique $((ya^R)^4)^* \subseteq L_G(t)$.

Finalement, décider si un langage de la famille \mathcal{F} n'est pas IRS équivaut à décider si le problème de Post a une solution.

Le problème de savoir si un langage L de la famille \mathcal{F} (donné par G) n'est pas IRS est donc récursivement indécidable.

Puisque $\mathcal{F} \subset \text{Lin}$ et que G est une grammaire linéaire, il est indécidable de savoir si un langage linéaire (donné par une grammaire linéaire qui l'engendre) n'est pas IRS. \square

REMARQUE : Nous avons dit que l'alphabet X comprenait au moins deux lettres. Si $|X| = h$, le langage L est donc écrit sur un alphabet à $(h + p) = r$ lettres, soit (z_1, z_2, \dots, z_r) . Or, on s'aperçoit aisément que l'on peut coder cet alphabet sur un alphabet à deux lettres, x et y , par $\gamma(z_i) = xy^i x, \forall i \in [1, r]$ et reproduire la preuve précédente.

La proposition 1' reste donc vraie si l'on se borne à des langages sur des alphabets à deux lettres.

Nous considérons maintenant le même problème dans la famille des langages déterministes, et nous prouvons :

PROPOSITION 2 : *Si L est un langage déterministe, on peut décider si L est un langage IRS.*

Preuve : Si L est déterministe, reconnu par un automate à pile déterministe on peut associer à cet automate, par la triple construction [6], une grammaire algébrique engendrant L . Nous allons montrer que, pour cette grammaire, la condition (3) du lemme 1 ne peut jamais être satisfaite.

Dans la grammaire algébrique résultant de la triple construction, les non-terminaux sont les triplets $[q, y, q']$ où q et q' sont deux états de l'automate à pile et y un symbole de pile.

Écrire qu'il existe une dérivation $[q, y, q'] \xrightarrow{*} (f_1 f_2)^p f_1$ dans cette grammaire entraîne qu'il existe un calcul valide dans l'automate, que l'on peut représenter, en utilisant les notations de [6], de la manière suivante :

$$(q, (f_1 f_2)^p f_1, y) \vdash^* (q', \varepsilon, \varepsilon)$$

Écrire qu'il existe une dérivation $[q, y, q'] \xrightarrow{*} (f_1 f_2)^m [q, y, q'] (f_2 f_1)^n$ dans cette grammaire entraîne qu'il existe un calcul valide dans l'automate, représenté par :

$$(q, (f_1 f_2)^m, y) \vdash^* (q, \varepsilon, w y),$$

pour un certain mot de pile w (Le haut de la pile est à droite). Sans perte de généralité, on peut supposer $m > p$. Puisque le premier calcul, qui vide la pile, est un début du second, il apparaît alors que les deux calculs induisent deux comportements différents de l'automate à pile, placé dans l'état q , lisant le mot $(f_1 f_2)^p f_1$ avec y en sommet de pile.

Puisque l'automate à pile est supposé déterministe on arrive à une contradiction.

Par conséquent, pour une grammaire résultant de la triple construction à partir d'un automate à pile déterministe, la condition (3) du lemme 1 n'est jamais satisfaite.

Clairement, on peut décider si une telle grammaire satisfait ou non les conditions (1) ou (2) de ce même lemme 1, d'où la proposition 2. \square

Nous nous intéressons maintenant à un problème dual : peut-on décider si un langage IRS est linéaire

(Rappelons qu'il est indécidable de savoir si un langage algébrique est linéaire [8].) Nous n'apportons pas de réponse à ce problème. Néanmoins nous donnons une condition générale qui, si elle est satisfaite pour une famille de langages, entraîne la décidabilité de la linéarité pour cette famille.

Nous prouvons que cette condition n'est pas satisfaite par les langages IRS mais nous conjecturons qu'elle l'est pour les langages non ambigus IRS. Pour introduire cette condition, nous posons une nouvelle définition :

DÉFINITION 2 [11] : *Un langage L est premier (sur Alg), si et seulement si :*

$$\forall A, B \in \text{Alg}, \quad AB \subseteq L \Rightarrow A \text{ ou } B \text{ est fini.}$$

REMARQUE : Dans [11] où est apparue initialement cette définition, on ne dit pas L premier mais L (relativement à Alg) CIL (Concatenation Infinite Language).

LEMME 2 : *Un langage algébrique premier est linéaire.*

Preuve : On sait que tout langage algébrique infini peut être engendré par une grammaire algébrique réduite $G = \langle X, V, P \rangle$, satisfaisant la propriété suivante : pour tout non-terminal v de V , le langage $L_G(v)$ est infini.

Examinons une telle grammaire. Ou bien chaque membre droit de règle contient au plus un non-terminal et cette grammaire est linéaire, ou bien il existe un membre droit de règle contenant au moins deux non-terminaux et alors le langage qu'elle engendre n'est pas premier. \square

Notons qu'un langage peut fort bien être à la fois linéaire et non premier. C'est par exemple, le cas du langage :

$$L = \{ a^n b^n a^p \mid n, p \in \mathbb{N} \},$$

L est le produit $L_1 L_2$ du langage linéaire $L_1 = \{ a^n b^n \mid n \in \mathbb{N} \}$ et du langage rationnel $L_2 = a^*$.

Maintenant, excluons ce cas trivial en nous restreignant à la famille des langages IRS et demandons nous si, dans cette famille, les qualificatifs linéaires et non-premiers sont contradictoires.

Nous préciserons ce point en prouvant :

PROPOSITION 3 : *Il existe un langage linéaire IRS qui n'est pas premier.*

Preuve : Soit $X = \{a, b, c, d\}$. Considérons le langage $L_1 = L_{G_1}(v)$ où G_1 est la grammaire :

$$G_1 = \langle X, \{v, w\}, P_1 \rangle,$$

P_1 étant l'ensemble de règles :

$$\begin{aligned} v &\rightarrow avd + avd^2 + a^2vd + awd, \\ w &\rightarrow bwc + bwc^2 + b^2wc + bc, \end{aligned}$$

L_1 est un langage linéaire IRS qui contient le langage $\{a^n b^n c^p d^p \mid n < 2p \text{ et } p < 2n\}$.

Considérons d'autre part, le langage $L_2 = L_{G_2}(v)$ où G_2 est la grammaire :

$$G_2 = \langle X, \{v, w, u\}, P_2 \rangle,$$

P_2 étant l'ensemble de règles :

$$\begin{aligned} v &\rightarrow avd^2 + w + awcd, \\ w &\rightarrow awc^2 + u, \\ u &\rightarrow aub^2 + aub + 1, \end{aligned}$$

L_2 est un langage linéaire IRS qui contient le langage $\{a^n b^n c^p d^p \mid n \geq 2p\}$. Considérons enfin le langage $L_3 = L_{G_3}(v)$, où G_3 est la grammaire :

$$G_3 = \langle X, \{v, w, u\}, P_3 \rangle.$$

P_3 étant l'ensemble des règles :

$$\begin{aligned} v &\rightarrow a^2vd + w + abwd, \\ w &\rightarrow b^2wd + u, \\ u &\rightarrow c^2ud + cud + 1 \end{aligned}$$

L_3 est un langage IRS qui contient $\{a^n b^n c^p d^p \mid p \geq 2n\}$.

Par conséquent, le langage $L = L_1 \cup L_2 \cup L_3$ est un langage linéaire qui contient le langage $\{a^n b^n \mid n \geq 0\} \cdot \{c^p d^p \mid p \geq 0\}$. L n'est donc pas premier. On vérifie cependant sans peine que L est IRS. \square

En vertu du lemme 2, dans toute sous-classe des langages algébriques dans laquelle les langages linéaires sont disjoints des langages non-premiers, on peut décider dans cette sous-classe de la linéarité des langages. Nous venons de montrer que cette disjonction n'était pas réalisée pour la sous-classe des langages IRS.

Nous conjecturons qu'elle l'est pour les langages non ambigus IRS, ce qui résoudrait la :

CONJECTURE : *Si un langage est non ambigu IRS, on peut décider s'il est linéaire.*

Remarquons que l'hypothèse de non-ambiguïté est très contraignante puisque, sous l'hypothèse certes plus restrictive, mais voisine de déterminisme fort, il a été démontré [10], sans utiliser l'hypothèse d'être IRS, que la linéarité était décidable.

BIBLIOGRAPHIE

1. J. M. AUTEBERT, G. COUSINEAU et M. NIVAT, *Les Langages Algébriques*, Hermann, Paris (à paraître).
2. L. BOASSON, *Un Langage Algébrique Particulier*, R.A.I.R.O., Informatique théorique, vol. 13, 1979, p. 203-215.
4. E. FRIEDMANN, *The Inclusion Problem for Simple Languages*, Theoretical Computer Science, vol. 1, 1976, p. 297-316.
5. C. FROUGNY, *Langages très Simples Générateurs*, R.A.I.R.O., Informatique Théorique, vol. 13, 1979, p. 69-86.
6. S. GINSBURG, *The Mathematical Theory of Context-free Languages*, Mc Graw Hill, 1966.
7. S. GINSBURG et E. H. SPANIER, *Finite-Turn Pushdown Automata*, S.I.A.M. J. Control, vol. 4, 1966, p. 429-453.
8. S. A. GREIBACH, *The Unsolvability of the Recognition of Linear Context-Free Languages*, J. Assoc. Comp. Mach., vol. 13, 1966, p. 582-588.
9. S. A. GREIBACH, *One-Counter Languages and the IRS Condition*, J. Comput. System Sc., vol. 10, 1975, p. 237-247.
10. S. A. GREIBACH et E. FRIEDMANN, *Superdeterministics Pda's : a Subcase with a Decidable Equivalence Problem* (à paraître).
11. M. LATTEUX, *Cônes Rationnels Commutativement Clos*, R.A.I.R.O., Informatique Théorique, vol. 11, 1977, p. 29-51.
12. R. E. STEARNS, *A Regularity Test for Pushdown Machines*, Information and Control, vol. 11, 1967, p. 323-340.
13. L. G. VALIANT, *The Equivalence Problem for Deterministic Finite-Turn Pushdown Automata*, Information and Control, vol. 25, 1974, p. 123-133.
14. L. G. VALIANT, *Decision Procedures for Families of Deterministic Pushdown Automata*, Ph. D. Thesis, University of Warwick, 1973.