RAIRO. Informatique théorique

Laurent Chottin

Langages algébriques et systèmes de réécriture rationnels

RAIRO. Informatique théorique, tome 16, nº 2 (1982), p. 93-112

<http://www.numdam.org/item?id=ITA_1982__16_2_93_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 16, n° 2, 1982, p. 93 à 112)

LANGAGES ALGÉBRIQUES ET SYSTÈMES DE RÉÉCRITURE RATIONNELS (*)

par Laurent CHOTTIN [1]

Communiqué par la Rédaction

Résumé. — *Dans cet article, un nouveau mécanisme de description de langages est proposé. On se donne des règles de la forme : fug peut être réécrit fvg si f appartient à un langage rationnel* $R(u, v)$ *donné. On démontre que tout langage déterministe préfixe peut être engendré à partir d'un langage fini, par itération de ce type de règles. Réciproquement, sous certaines conditions sur les* $R(u, v)$, *le langage engendré par un système de ce type est algébrique.*

Abstract. — *In this paper, a new method for generating languages is introduced. It consists of the definition of a finite number of rules of the type: fug may be replaced by fvg if f belongs to a given regular language. It is proved that any deterministic language which is prefix can be generated starting from finite sets and iterating such rules. Conversely under suitable hypotheses on the* $R(u, v)$ *the language generated by such a system is algebraic and deterministic.*

Après les premiers résultats obtenus sur les langages définis par les classes de congruence [2, 3, 4, 5, 10, 11] on pouvait croire au développement d'une branche de la théorie des langages reposant sur des structures algébriques classiques et rendant compte des phénomènes observés dans l'étude des grammaires et des automates. De fait, il est particulièrement commode de présenter, par exemple, les langages de Dyck comme la classe du mot vide pour une certaine congruence, et il est surprenant de constater que bon nombre de langages classiques sont « congruentiels ». Malheureusement, les résultats obtenus dans cette direction se sont révélés fort rares; ainsi par exemple, il n'est

(*) Reçu novembre 1980, révisé juillet 1981.

[1] Université de Bordeaux-I.

Cet article a été rédigé après la disparition de l'auteur par Luc Boasson et Robert Cori à partir de notes laissées par L. Chottin et de l'exposé qu'il avait fait au Colloque international de Graz.

Toute correspondance à adresser à Robert Cori, Université de Bordeaux-I, 351, cours de la Libération, 33405 Talence Cedex, France.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1982/93/ $ 5.00

pas encore apporté de réponse satisfaisante aux deux questions suivantes : à quelle condition un langage algébrique est-il congruentiel, et réciproquement, à quelle condition un langage défini comme classe d'une certaine congruence peut-il être engendré par une grammaire algébrique? La lecture de l'exposé de J. Berstel [1], qui présente un résumé des résultats actuellement connus sur le sujet fournit une illustration du caractère difficile des questions énoncées plus haut. Les travaux plus récents de J. Sakarovitch [12, 13] confirment ce point de vue.

Dans le présent article, un mécanisme de description de langages est proposé; ce mécanisme intègre à la fois des notions de congruence, de grammaire et d'automate : il s'agit des systèmes de réécriture contrôlés par des langages rationnels. Plus précisément, on se donne des règles de la forme : le mot fug peut être réécrit fvg si f appartient à un langage rationnel $R(u, v)$ donné. Ainsi, le caractère orienté du remplacement et le fait que ce remplacement soit déterminé par la présence d'un facteur gauche dans un certain rationnel apparente notre mécanisme à une grammaire ou à un automate. Par contre, le fait que tous ces remplacements se passent sans utiliser d'alphabet auxiliaire, se rapproche plus de la définition de langages par congruence.

Les deux principaux résultats de cet article mettent en évidence l'intérêt des réécritures contrôlées par un rationnel.

(i) D'une part, on peut montrer (th. 1) que tout langage déterministe préfixe peut être défini par un système de réécritures rationnel fini. La preuve de ce résultat utilise un principe réciproque de celui de la preuve originelle du théorème de Parikh pour ces langages. Dans la preuve originelle de ce théorème, les mots d'un langage algébrique sont décomposés en facteurs élémentaires qui sont en nombre fini; ici, de façon quelque peu inverse, un nombre fini de « composants » permettent de reconstituer un langage déterministe préfixe donné.

(ii) D'autre part, on prouve que sous certaines conditions, le langage obtenu à partir d'un système de réécriture est algébrique (th. 2), et qu'il est déterministe si l'on restreint ces conditions (th. 3). L'analyse syntaxique d'un tel type de langage s'apparente alors beaucoup à l'analyse $LR(k)$ telle qu'elle est présentée par exemple dans (7).

La suite est articulée autour de ces deux résultats principaux. Après un paragraphe précisant la définition et donnant quelques exemples de systèmes de réécriture rationnels, nous donnons la preuve du théorème 1 (partie 2). La troisième partie est consacrée aux théorèmes 2 et 3 qui utilisent des hypothèses restrictives sur les systèmes de réécriture. Il se trouve qu'une partie de ces hypothèses est vérifiée par les systèmes construits à partir des automates déterministes dans la partie 2.

1. DÉFINITIONS, NOTATIONS ET EXEMPLES

Dans toute la suite X dénote un alphabet fini. Un *système de réécriture contrôlé par des langages rationnels*, ou *système de réécriture rationnel* est donné par une famille finie S de triples (R_i, g_i, d_i) $i=1, p$; où pour tout i, R_i est un langage rationnel inclus dans X^* et où g_i et d_i sont deux mots de X^*. Un mot f de X^* est dit obtenu par une *réécriture* à partir du mot g dans le système S, on notera $f \in \mathrm{Rempl}_S(g)$, si la condition suivante est vérifiée :

$$\exists\, i (1 \leqq i \leqq p), \qquad f=f' d_i f'', \qquad g=f' g_i f'', \qquad f' \in R_i.$$

Étant donné un langage A et un système S on notera $\mathrm{Rempl}_S(A)$ l'ensemble de tous les mots f obtenus par une réécriture à partir des mots de A dans le système S. Il est immédiat de constater que Rempl_S est une transduction rationnelle. Ainsi $\mathrm{Rempl}_S(A)$ est rationnel (resp. algébrique) lorsque A l'est. On définit aussi $\mathrm{Rempl}_S^k(A)$ par les relations :

$$\mathrm{Rempl}_S^0(A)=A; \qquad \mathrm{Rempl}_S^{k+1}(A)=\mathrm{Rempl}_S(\mathrm{Rempl}_S^k(A)).$$

Enfin :

$$\mathrm{Rempl}_S^*(\qquad \qquad \underset{n=0,\,\infty}{} \mathrm{Rempl}_S(A).$$

Dans le cas où $|g_i| < |d_i|$ quel que soit i, on vérifie aisément que $\mathrm{Rempl}_S^*(A)$ est la solution de l'équation :

$$L=A+\mathrm{Rempl}_S(L).$$

Notons que, dans le cas où tous les mots g_i sont vides, on retrouve les « systèmes d'insertion » étudiés par F. Kierszenbaum [8].

Donnons maintenant quelques exemples de systèmes de réécriture rationnels.

Exemple 1 : Le système défini par un seul triple $S=\langle (a^*, ab, a^2b^2) \rangle$ engendre de manière évidente $\{ a^n b^n \mid n>0 \}$ à partir de $A=\{ ab \}$.

Exemple 2 : Un système de réécriture rationnel peut engendrer un langage non algébrique comme le montre l'exemple suivant (où R dénote le langage $\{ a^n b^p \mid n+p \text{ impair} \}$ qui est évidemment rationnel).

$\langle (a^*, a, aa), (R, c, bcc) \rangle$; on trouve en effet si on choisit $A=\{ abc \}$:

$$\mathrm{Rempl}_S(A)=\{ a^2bc \}, \qquad \mathrm{Rempl}_S^2(A)=\{ a^2b^2c^2, a^3bc \},$$
$$\ldots \mathrm{Rempl}_S^n(A)=\{ a^i b^j c^j ; j \leqq i, \ i+j=n+2 \},$$

Et $\mathrm{Rempl}_S^*(A)=\{ a^n b^p c^p, p \leqq n \}$ qui n'est pas algébrique.

Exemple 3 : Montrons que, quel que soit le système de réécriture rationnel S et l'ensemble fini A, le langage $L = a^* \cup a^n b^n \mid n \geqq 0\}$ ne peut pas être égal à $\mathrm{Rempl}_S^*(A)$. Supposons que L est engendré par :

$S = \langle (R_i, g_i, d_i)\ i=1, p \rangle$ et nous montrons que cela conduit à une contradiction. Soit $N = \operatorname{Max}_{i=1,p} |g_i|$, $M = \operatorname{Max}_{i=1,p} |d_i|$ et soit P la longueur du plus long mot de A.

Considérons le mot a^n où n est un entier plus grand que $N+M$ et P et vérifions que a^n appartient alors à $\mathrm{Rempl}_S(a^*)$; en effet a^n ne peut être dans A ni dans $\mathrm{Rempl}_S(a^q b^q)$ sinon il existerait un triple $(R_i, a^{q-j} b^q, a^{n-j})$ tel que a^j appartienne à R_i et on aurait $n-j \leqq M$; $j \leqq q \leqq |a^{q-j} b^q| \leqq N$ d'où $n \leqq M+N$.

Du fait que a^n appartient à $\mathrm{Rempl}_S(a^*)$ on déduit l'existence d'un triple (R_k, a^q, a^r) dans S avec $r > q$ et $R_k \cap a^* \neq \emptyset$.

Soit a^m un mot de $R_k \cup a^*$, $a^{m+q} b^{m+q}$ étant un mot de L il devrait en être de même pour $a^{m+r} b^{m+q}$ qui s'obtient par un remplacement à partir de $a^{m+q} b^{m+q}$ d'où la contradiction cherchée.

2. SYSTÈME DE RÉÉCRITURE ASSOCIÉ À UN AUTOMATE DÉTERMINISTE

Après avoir précisé nos notations sur les automates déterministes, nous définissons le système de réécriture associé à un tel automate. Nous démontrons ensuite que ce système engendre un langage qui est exactement celui que l'automate reconnaît par pile vide. Enfin, quelques propriétés de « préfixité » des systèmes associés aux automates clôturent cette partie.

2.1. Notations sur les automates à pile déterministes

Soit $\mathscr{A} = \langle X, Q, Y, \delta, q_0, y_0 \rangle$ un automate à pile déterministe, il est dit normalisé si toute transition par mot vide fait décroître la taille de la pile soit si :

$$\delta(1, q, y) = (q', m) \quad \Rightarrow \quad m = 1.$$

Dans la suite, les automates seront supposés normalisés. Nous utiliserons les notions classiques de configuration et de calcul (6), une *configuration* est un triple (f, q, m) où $f \in X^*$, $q \in Q$, $m \in Y^*$; on notera $(f, q, m) \vdash (f', q', m')$ si $f = f'u$, $(u \in X \cup \{1\})$, $m = m_1 y$, $m' = m_1 m_2$ et $\delta(u, q, y) = (q', m_2)$.

Un *calcul* sera une suite de configurations (f_i, q_i, m_i), $i = 1, \ldots, n$ telles que $(f_i, q_i, m_i) \vdash (f_{i+1}, q_{i+1}, m_{i+1})$; et on notera $(f_1, q_1, m_1) \overset{*}{\vdash} (f_n, q_n, m_n)$. Un mot f est *reconnu* par l'automate par pile vide si $(f_0, q_0, y_0) \overset{*}{\vdash} (1, q, 1)$ on notera $L(A)$ l'ensemble des mots reconnus par pile vide. Un (q, y)-*calcul* sera un calcul dont le premier triple est de la forme (f, q, y). Étant donné un mot f quelconque, un état q et un mot m de Y^*, il peut y avoir plusieurs calculs commençant par la configuration (f, q, m) et se terminant par la lecture complète de f; tous ces calculs ne diffèrent que par le nombre de transitions par mot vide qui ont lieu après la lecture de f. Afin de réaliser une correspondance biunivoque entre mot lu et calcul, on introduit un nouveau symbole ε, on factorise tout calcul en $(f, q, m) \overset{*}{\vdash} (x, q_1, m_1) \vdash (1, q', m') \vdash (1, q'', m')$ et on dit que le mot associé à ce calcul est $f\varepsilon^k$.

Réciproquement, étant donné un mot $f\varepsilon^k \in X^* \varepsilon^k$, un état q et un mot m de Y^*, on peut construire au plus un (q, m)-calcul associé à $f\varepsilon^k$. Si $\varepsilon = 0$, on aura le *calcul minimal* associé à f, si à $f\varepsilon^{k+1}$ ne correspond aucun calcul et qu'il en correspond un à $f\varepsilon^k$ on dira qu'il s'agit du *calcul maximal* associé à f On remarquera que du fait que l'automate $\mathscr{A}$ est normalisé, si f a un calcul minimal, il a un nombre fini de calculs correspondant à un nombre fini de valeurs possibles pour k dans $f\varepsilon^k$.

2.2. Système de réécriture associé à un automate à pile déterministe $\mathscr{A}$

Un (q, y) calcul constitue un *facteur itérant* s'il s'écrit :

$$(f, q, y) \overset{*}{\vdash} (1, q, y), \qquad f \neq 1, \qquad y \in Y.$$

Du fait que $\mathscr{A}$ est un automate déterministe normalisé, un tel calcul est le (q, y) calcul maximal associé à f. Par abus de langage on dira que le mot $f\varepsilon^k$ associé à ce calcul est un facteur itérant. Un (q, y) calcul constitue une *paire itérante* s'il se factorise en :

$$(u, q, y) \overset{*}{\vdash} (1, q, my), \qquad y \in Y, \quad m \in YY^*,$$
$$(v, q, y) \overset{*}{\vdash} (1, q', 1),$$
$$(w', q', m) \overset{*}{\vdash} (1, q', 1).$$

A nouveau, par abus de langage, le mot $uvw\varepsilon^k$, ou plus précisément le triplet $(u, v, w\varepsilon^k)$, est dit former une (q, y, q')-paire itérante. Un (q, y) *calcul contient une paire itérante* (resp. un facteur itérant) s'il se factorise en $f_1 f_2 f_3$ avec $(f_1 \varepsilon^{n_1}, q, y) \overset{*}{\vdash} (1, q', \alpha y')$ et $f_2 \varepsilon^{n_2}$ est une (q', y', q'') paire itérante [resp. un (q', y') facteur itérant]. Un calcul contenant un facteur itérant ou une paire itérante est dite contenir un *élément itérant.*

Un (q, y) calcul ne contenant pas d'élément itérant est (q, y) *premier.*

Un (q, y) calcul qui n'est pas premier, mais dont tout facteur gauche propre est (q, y) premier est dit (q, y)-primaire. Il ne contient alors qu'un seul élément itérant : en effet s'il en contenant deux on vérifierait que ce ne peuvent être que deux facteurs itérants donnant lieu à un troisième dans un facteur gauche propre du calcul.

Considérons les ensembles D, P et F suivants; intuitivement D est constitué des mots reconnus par $\mathscr{A}$ et sans éléments itérants; au contraire P et F sont formés des éléments itérants primaires qui seront nécessaires pour « reconstruire » les mots de $L(\mathscr{A})$ qui ne sont pas dans D, plus précisément :

$$D=\{f\in L(\mathscr{A}) \mid \text{le } (q_0, y_0) \text{ calcul maximal associé à } f \text{ est premier}\},$$

$$P(q, y, q')=\{(u, v, w\varepsilon^k) \mid uvw\varepsilon^k \text{ est une } (q, y, q') \text{ paire itérante } (q, y) \text{ primaire}\},$$

$$F(q, y)=\{v\varepsilon^k \mid v\varepsilon^k \text{ est un } (q, y) \text{ facteur itérant } (q, y) \text{ primaire}\}.$$

Pour construire notre système de réécriture, nous avons besoin de définir les langages rationnels $R(q, y, u)$ formé des mots f tels que fu conduise à une configuration $(1, q, \alpha y)$ sans avoir utilisé d'élément itérant; ainsi, si f est long, α le sera aussi. Plus précisément : $R(q, y, u)$ est formé de tous les mots f tels que :

- le calcul maximal associé à f s'écrit $(f, q_0, y_0) \overset{*}{\vdash} (1, q, my)$;
- tout facteur gauche propre de fu est premier.

On construit alors le système de réécriture contrôlé $\mathscr{S}(\mathscr{A})$ qui contient $\langle R(q, y, v\varepsilon^k); 1, v\rangle$ pour tout mot $v\varepsilon^k$ de $F(q, y)$ et $(R(q, y, uvw\varepsilon^k); v; uvw)$ pour tout élément $(u, v, w\varepsilon^k)$ de $P(q, y, q')$.

Exemple : Considérons l'automate à pile donné par :

$$\mathscr{A}=\langle \{x, y, z, t, a, b\}, \{q_0, q, q'\}, \{\alpha, \beta, \gamma, y_0\}, \delta(q_0, y_0)\rangle,$$

avec :

$$\delta(z, q_0, y_0)=(q, \alpha); \qquad \delta(x, q, \alpha)=(q, \beta);$$

$$\delta(y, q, \beta)=(q, \delta),$$

$$\delta(a, q, \beta)=(q, \gamma_\beta); \qquad \delta(t, q, \alpha)=(q', 1);$$

$$\delta(b, q', \gamma)=(q', 1).$$

Cet automate reconnaît :

$$L=zt \cup x(xy)^* x(a(yx)^*)^n ytb^n.$$

et on trouve $D=\{zt\}$.

$$F(q, \alpha)=\{xy\}; \qquad P(q, \beta, q)=(a, yt, b); \qquad F(q, \beta)=\{yx\},$$

$$R(q, \alpha, xy)=\{z\}; \qquad R(q, \beta, yx)=zxa^t; \qquad R(q, \beta, qytb)=zxa^*.$$

Le lecteur pourra vérifier que L est aussi engendré par :

$$\langle z, 1, zy\rangle; \qquad \langle zxa^t, 1, yx\rangle; \qquad \langle zxa^*, yt, aytb\rangle,$$

à partir du langage réduit à $\{zt\}$.

2.3. Théorème principal

Le système $\mathscr{S}$ associé à $\mathscr{A}$ est un système de réécriture rationnel tel que : $\mathrm{Rempl}^*_{\mathscr{S}}(D)=L(\mathscr{A})$.

Pour démontrer ce théorème, nous vérifierons d'abord l'égalité $\mathrm{Rempl}^*_{\mathscr{S}}(D)=L(\mathscr{A})$; le fait que $\mathscr{S}$ est un système rationnel sera vérifié ensuite.

LEMME 1 : $L(\mathscr{A})\subseteq L=\mathrm{Rempl}^*_{\mathscr{S}}(D)$.

Preuve : On procède par induction sur la longueur du calcul de $\mathscr{A}$ reconnaissant un mot.

Considérons donc un calcul de longueur minimale et qui vide la pile de l'automate; il accepte un mot f. Il est bien clair que si ce calcul contenant un élément itérant on pourrait trouver un calcul plus court acceptant un mot g. Ainsi, le calcul de longueur minimale est-il premier et le mot qu'il reconnaît est dans D, donc dans $\mathrm{Rempl}^*_{\mathscr{S}}(D)$.

Considérons maintenant un calcul de longueur l reconnaissant le mot f , à nouveau si ce calcul est premier, f est dans D et donc dans L. Sinon, ce calcul a un facteur gauche maximal premier et il se décompose en deux étapes :

$$(1)\qquad \begin{cases} (f, q_0, y_0) \overset{*}{\vdash} (f_1, q_1, m_1) \text{ calcul premier} \\ (f_1, q_1, m_1) \overset{*}{\vdash} (1, q', 1) \text{ calcul de longueur non nulle.} \end{cases}$$

La configuration obtenue après une transition à partir de (f_1, q_1, m_1) est alors :

$$(2)\qquad \begin{cases} (f_1, q_1, m_1) \vdash (f_2, q_2, m_2) \text{ et le calcul,} \\ (f, q_0, y_0) \vdash f_2, q_2, m_2) \text{ n'est pas premier,} \end{cases}$$

il contient donc au moins un élément itérant. Comme (1) est un calcul premier, l'élément itérant doit utiliser (ou contenir) le dernier pas de calcul (2). On peut donc écrire $f=f'hf_2$ où h (qui appartient à $\chi^*\varepsilon^*$) correspond à l'élément itérant :

$$(f, q_0, y_0) \overset{*}{\vdash} (g, q, my), \qquad f=f'g,$$

$$(g, q, y) \overset{*}{\vdash} (f_2, q_2, p),$$

$$q_2=q, \qquad p=y \qquad \text{ou} \qquad p=1 \quad \text{avec} \quad g=hf_2,$$

h constitue un élément itérant qui doit être primaire [sinon, il contiendrait un élément itérant en contradiction avec le fait que (1) soit un calcul premier].

Comme tout facteur gauche propre de $f'\,h$ correspond à un calcul premier, on sait que $R(q, y, h)$ contient f'. Dans $\mathscr{S}$ on a donc une règle :

$$R(q, y, h);\ h',\ h$$

et le calcul de $\mathscr{A}$ pour le mot $f' h' f_2$ est strictement plus court que celui donné pour f. Par hypothèse d'induction $f' h' f_2 \in \mathrm{Rempl}^*_{\mathscr{S}}(D)$ et par inspection directe $f = f' h f_2 \in \mathrm{Rempl}_{\mathscr{S}}(f' h' f_2)$ ce qui termine la preuve du lemme.

LEMME 2 : *Le langage L est inclus dans* $L(\mathscr{A})$.

La preuve se fait par induction sur le nombre de réécritures de $\mathscr{S}$ utilisées. Si on ne procède à aucune réécriture, alors on a un mot de D et ce mot est dans $L(\mathscr{A})$ par définition.

Supposons maintenant que si le nombre de réécriture est inférieur ou égal à k, le mot obtenu soit dans $L(\mathscr{A})$ et considérons un mot f obtenu en $(k+1)$ réécritures. On peut donc construire à partir de D un mot f_1 en k réécritures, puis en une réécriture de $\mathscr{S}$ trouver f. Par hypothèse d'induction f_1 est dans $L(\mathscr{A})$ et l'on peut écrire :

$$f_1 = g \nu h, \qquad f = g \mu\nu\theta h$$

avec soit $\nu = 1$, $g \in R(q, y, v\varepsilon^k)$, $\mu\nu\theta = v$ et $v\varepsilon^k \in F(q, y)$; soit $g \in R(q, y, \mu\nu\theta\varepsilon^k)$ et $(\mu, \nu, \theta\varepsilon^k) \in P(q, y, q')$.

Dans l'un et l'autre cas, on peut reconstruire le calcul de $\mathscr{A}$ acceptant f_1 [avec en particulier $\delta(g, q_0, y_0) \overset{*}{\vdash} (q, \alpha y)$].

On en déduit facilement que le calcul de $\mathscr{A}$ sur f accepte le mot car il se termine comme celui de $\mathscr{A}$ sur f_1.

LEMME 3 : *Les ensembles* $F(q, y)$, $P(q, y, q')$ *et D sont finis.*

Preuve : Cette propriété résulte immédiatement des théorèmes d'itération (7). Chaque ensemble est partie d'un langage algébrique et tout mot assez long de ces langages admet une itération dans le calcul le reconnaissant. Ainsi, puisque les mots de D ont un calcul associé (q_0, y_0)-premier ces mots sont de longueur bornée. Pour les mots de $F(q, y)$ et $P(q, y, q')$ les facteurs gauches propres des calculs associés sont premiers et donc, à nouveau, ces ensembles sont finis puisque les calculs associés sont de longueur bornée.

LEMME 4 : *Les ensembles* $R(q, y, u)$ *sont rationnels*

Preuve : Nous allons établir que chaque ensemble $R(q, y, u)$ peut être décrit comme une intersection de langages rationnels. Nous commençons par montrer que tous les $R(q, y, u)$ sont contenus dans un même langage rationnel K.

On sait que, si au cours du calcul d'un automate à pile $\mathscr{A} = \langle X, Q, Y, \delta, q_0, y_0 \rangle$ on peut trouver deux configurations (q, m) et $(q' m')$ où la différence $|m'| - |m|$ dépasse $|Q|^2 \times |Y| = s$, alors on peut trouver une paire itérante dans le calcul allant de (q, m) à (q', m'). Or les mots de $R(q, y, u)$ sont premiers. On peut donc garantir que ces mots sont tous dans le rationnel reconnu par l'automate $\mathscr{F}$

décrit ci-dessous (qui n'est en fait que l'automate à pile $\mathscr{A}$ dont on ne retient que les s éléments supérieurs de la pile) :

$$\mathscr{F}=\langle P, X, \mathscr{F}, (q_0, y_0), P\rangle,$$
$$P=\{(q, m) \mid q\in Q, m\in Y^* \mid m \mid \leqq s\}.$$

Les transitions sont de deux types :

type 1 :

$$\lambda((q, my), x)=(q', mp),$$

si $\delta(q, y, x)=(q', p)$ et $|mp|\leqq s$;

type 2 :

$$\lambda((q, my), x)=(q', m_1),$$

si $\delta(q, y, x)=(q', p)$ et m_1 est le facteur droit de longueur s de mp.

Remarquons que, puisque $x\in X\cup\{\varepsilon\}$, cet automate fini a des ε-transitions. Cependant, le caractère déterministe de $\mathscr{A}$ fait de $\mathscr{F}$ un automate fini déterministe. On notera que, cependant, certains mots conduisent $\mathscr{F}$ à un blocage. Ce sont ceux qui emmènent $\mathscr{F}$ dans un état du type $(q, 1)$. Nous dirons qu'un mot est reconnu par $\mathscr{F}$ si l'automate fini décrit ici peut lire ce mot sans blocage, et noterons K l'ensemble des mots reconnus par $\mathscr{F}$ de cette manière.

On sait que :

$$(1) \qquad \forall q, y, u, \quad R(q, y, u)\subseteqq K.$$

Fixant maintenant q, y et u, on peut préciser les choses : un mot de $R(q, y, u)$, s'il doit être dans K, doit en outre satisfaire les deux conditions :

- il conduit $\mathscr{F}$ dans un état de la forme (q, my);
- prolongé par u, il ne provoque aucun blocage de $\mathscr{F}$.

Ainsi, on peut choisir dans $\mathscr{F}$ des états terminaux $T(q, y, u)$:

$$\{(q', m') \mid q'=qm' \text{ se termine par } y \text{ et } \lambda((q', m'), u) \text{ est défini}\}.$$

Désignant par $K(q, y, u)$ le rationnel reconnu par $\langle P, X, \mathscr{A}, T(q, y, u)\rangle$:

$$(2) \qquad \forall q, y, u, \quad R(q, y, u)\subseteqq K(q, y, u).$$

Il ne nous reste plus qu'à chasser les mots de $K(q, y, u)$ contenant un élément itérant. Or, un tel mot contenant un (q', y')-facteur itérant sera dans :

$$K(q', y', v\varepsilon^k).F(q', y')X^*$$

qui est rationnel. Il en est alors de même de l'union sur (q', y') de ces rationnels, désignés ici par FI.

Un tel mot contenant une (q', y') paire itérante sera lui dans :

$$K(q', y', uvw\varepsilon^k)P(q', y', q'')X^* \qquad \text{pour} \quad u, v, w, \varepsilon^k \quad \text{dans } P(q', y', q'').$$

A nouveau, il en est de même de l'union sur (q', y', q'') de ces langages, notée ici PI.

Le résultat annoncé vient alors de ce que, si $\overline{PI}$ et $\overline{FI}$ sont les complémentaires respectifs de PI et FI :

$$R(q, y, u)=K(q, y, u)\cap\overline{PI}\cap\overline{FI}.$$

2.4. Propriétés de préfixité de $\mathscr{S}$

PROPRIÉTÉ 2 : *$\langle R_i, g_i, d_i\rangle$ et $\langle R_j, g_j, d_j\rangle$ étant deux éléments de S, $r_i\in R_i$ et $r_j\in R_j$, $r_i d_i=r_j d_j f=i=j, f=1$* (2.2).

Preuve : On désigne par $d_i\varepsilon^k$ et $d_j\varepsilon^r$ les mots associés aux calculs de $\mathscr{A}$ qui sont à l'origine des règles de réécriture correspondantes de S :

a) si $f\neq 1$, alors $r_i d_i\varepsilon^k$ a un facteur gauche propre $r_j d_j$ dont le calcul maximal contient un élément itérant, à savoir $d_j\varepsilon^r$. Il en résulte que le calcul associé à $r_i d_i\varepsilon^{k-1}$ n'est pas premier et ceci est impossible par construction de $\mathscr{S}$: on aurait $r_i\notin R_i$;

b) on sait donc que $f=1$. Soit encore : $r_i d_i=r_j d_j$. Supposons donc $|r_i|>|r_j|$, on peut alors écrire :

$$r_j=r_i d_{j_1}, \qquad d_i=d_{j_2}, \qquad d_j=d_{j_1}d_{j_2}.$$

Si $r\geqq k$, alors $d_j\varepsilon^r=d_{j_1}d_{j_2}\varepsilon^r=d_{j_1}d_i\varepsilon^k\varepsilon^{r-k}$ et $d_j\varepsilon^r$ est un élément itérant contenant strictement un élément itérant (à savoir $d_i\varepsilon^k$); il n'est pas primaire et ne peut pas figurer dans $\mathscr{S}$.

Si $r\leqq k$, alors $r_j d_{j_1}d_{j_2}\varepsilon^r\varepsilon^{k-r}$ admet comme facteur gauche propre $r_j d_j\varepsilon^r$ et n'est pas premier ce qui, à nouveau, n'est pas possible dans $\mathscr{S}$. Le cas $|r_i|<|r_j|$ se traite de la même façon. Il en résulte que $r_i d_i=r_j d_j$ et $|r_i|=|r_j|$. Ainsi $r_i=r_j$ et $d_i=d_j$. Il en résulte immédiatement $i=j$.

PROPRIÉTÉ 1 : *$(R_i, \langle g_i, d_i\rangle)$ et $(R_j, \langle g_j, d_j\rangle)$ étant deux éléments de $\mathscr{S}$, $r_i\in R$ et $r_j\in R_j$, on a l'implication : $(r_i=r_j f;\ g_i=gh;\ d_j=fg;\ h\neq 1)\Rightarrow g_j=1$* (2.1).

Preuve : Nous procédons par l'absurde. Supposons donc :

$$r_j fgh=r_i g_i=r_j d_j h, \qquad h\neq 1 \quad \text{et} \quad g_j\neq 1.$$

Si l'on considère le calcul de $\mathscr{A}$ associé à $r_j d_j h$, on peut le factoriser en :

$$(r_j, q^0, y^0)\overset{*}{\vdash}(1, q_j, \alpha y_j),$$

$$(d_j, q_j, y_j)\overset{*}{\vdash}(1, q, 1)$$

car $g_j\neq 1$ implique que l'élément itérant est une paire itérante.

Comme $d_j=fg$, on peut écrire :

$$(f, q_j, y_j) \overset{*}{\vdash} (1, q_1, m_1 y) \text{ calcul maximal de } f,$$

$$(g, q_1, m_1 y) \overset{*}{\vdash} (1, q, 1).$$

Si $h \neq 1$ alors $g_i \neq 1$ et g_i est aussi associé à une paire itérante; on sait donc que si :

$$(r_i, q^0, y^0) \overset{*}{\vdash} (1, q_i, \beta y_i) \quad \text{calcul maximal},$$

$$(q_i, q_i, y_i) \overset{*}{\vdash} (1, q', 1).$$

Or, puisque $r_i=r_j f$ on sait que $q_i=q_1$ et $\alpha m_1 y=\beta y_i$.

Si l'on rapproche alors :

$$(g_i, q_i, y_i) \overset{*}{\vdash} (1, q', 1),$$

$$(g, q_i, m_i y_i) \overset{*}{\vdash} (1, q, 1)$$

et $g_i=gh$ on obtient une contradiction puisque $h \neq 1$ fait que le premier calcul doit se factoriser en :

$$(g, q_i, y_i) \overset{*}{\vdash} (1, q'', m), \qquad m \in Y^+$$

et donc :

$$(g, q_i, m_i y_i) \overset{*}{\vdash} (1, q'', m_1 m), \qquad m \in Y^+.$$

Remarque : En reprenant l'exemple donné à la fin du paragraphe 2.2, on constate que la situation décrite dans la propriété 1 est effectivement possible; en effet, en prenant :

$$r_i=zx, \qquad g_i=yt \qquad \text{et} \qquad r_j=z, \qquad d_j=xy, \qquad h=1$$

on a :

$$r_i g_i=r_j d_j h.$$

Bien entendu, dans ce cas, $h=1$.

3. CONDITIONS D'ALGÉBRICITÉ DE LA SOLUTION D'UN SYSTÈME DE RÉÉCRITURE

Dans cette partie, nous nous intéressons à l'équation :

$$(1) \qquad L=F+\mathrm{Rempl}\,(L),$$

où F est un langage fini, et où $|d_i|>|g_i|$ pour tout i.

Nous nous proposons de donner des conditions suffisantes pour que le langage solution de cette équation soit algébrique et déterministe; nous démontrons plus précisément les théorèmes suivants :

THÉORÈME 2 : *Soit une famille* $\langle R_i, g_i, d_i \rangle$ *définissant un système de réécriture rationnel satisfaisant à la condition :*

Pour tout $r_i \in R_i$ *et* $r_j \in R_j$,

(3.1) $r_i g_i = r_j d_j f$ *n'a lieu que si* f *est le mot vide.*

La solution de l'équation (1) *est alors un langage algébrique.*

THÉORÈME 3 : *Si la famille* $\langle R_i, g_i, d_i \rangle$ *vérifie de plus la condition :*

Pour tout $r_i \in R_i$ *et* $r_j \in R_j$ *:*

(3.2) $r_i d_i r_j d_j f$ *n'a lieu que si* f *est vide et si* $d_i = d_j$.

La solution de l'équation (1) *est alors un langage algébrique déterministe.*

Remarquons tout de suite que le théorème 3 ne constitue pas exactement la réciproque du théorème de la section 2. En effet, si nous avons prouvé que les langages déterministes préfixes sont solutions d'équations du type (1), nous n'avons obtenu comme propriété de ces équations que 2.1 (qui est exactement 3.2) et 2.2 qui est une condition plus faible que 3.1. La question de savoir si 2.1 et 2.2 suffisent à garantir le fait que la solution de (1) soit un langage algébrique déterministe est un problème ouvert. Les raisons qui font que notre preuve ne fonctionne pas dans ce cas, seront mis en lumière à l'aide d'un exemple que nous donnerons à la fin de cette partie, et, bien que nous n'ayons pas de contre-exemple, il semble raisonnable de penser que l'on puisse trouver un langage non algébrique solution d'une équation vérifiant 2.1 et 2.2.

Avant d'entamer la preuve du théorème, nous proposons deux exemples qui montrent d'une certaine manière l'optimalité des résultats que nous avons obtenus; le premier est celui d'un système vérifiant 3.2 mais pas 3.1 dont la solution est un langage qui n'est pas algébrique, le second est un système qui vérifie 3.1 et pas 3.2 et dont la solution est un langage algébrique non déterministe.

Exemple 1 (dû à Serge Dulucq) sur l'alphabet $\{a, b, c\}$:

$$R_1 = a^*, \qquad g_1 = b, \qquad d_1 = ab^2 c,$$
$$R_2 = a^+ b^+, \qquad g_2 = cb, \qquad d_2 = bbc.$$

On peut vérifier facilement (par récurrence sur la longueur des mots) que la solution L de l'équation associée en prenant $F = \{abc\}$ est telle que $L \cap a^* b^* c^* = \{a^{n+1} b^{2^n} c^{n+1}\}$.

On peut aussi s'assurer que les R_i, g_i, d_i vérifient la condition 3.2.

Exemple 2 :

$$\begin{array}{lll} R_1=a^*, & g_1=c, & d_1=acb, \\ R_2=a^*, & g_2=d, & d_2=adbb, \\ R_3=a^*, & g_3=c, & d_3=acdb, \\ R_4=a^*, & g_4=d, & d_4=acdbb, \end{array}$$

On peut vérifier simplement que le système satisfait 3.1 mais pas 3.2 et la solution du système en prenant $F=\{c, d\}$ vérifie :

$$L \cap a^* cdb^* = \{a^n cdb^n\} \cup \{a^n cdb^{2n}\}.$$

En effet, en partant de c on construit $a^n cb^n$ en utilisant (R_1, g_1, d_1) puis $a^n cdb^n$ en utilisant R_3, g_3, d_3 tout autre utilisation de règle faisant sortir du langage $a^* cdb^*$ sans possibilité de retour. De même, partant de d on construit $a^n cdb^{2n}$ par (R_2, g_2, d_2) et (R_4, g_4, d_4).

3.1. Équation propre

L'équation (1) est dite *propre* s'il n'existe pas de sous-ensemble F' de F tel que l'équation :

$$(1)' \qquad L=F'+\mathrm{Rempl}(L)$$

admette même solution que (1).

On peut prouver facilement que pour toute équation, il existe une équation propre ayant même solution, il suffit pour cela de choisir le plus petit F_0 inclus dans F (fini) qui donne le même résultat pour l'équation (1).

Le fait que $|g_i| < |d_i|$ implique aussi qu'il existe un algorithme très simple permettant de réaliser ce choix.

Dans la suite, nous ne nous intéresserons plus qu'aux équations propres, cette hypothèse sera utile dans le théorème 3, pour la construction d'un automate déterministe.

3.2. Union de langages rationnels

PROPRIÉTÉ : Étant donnée une famille $\mathscr{R}=(R_1, R_2, \ldots, R_p)$ de langages rationnels il existe un automate fini $\mathscr{A}$ ayant pour ensemble d'états $S=S_1 \cup S_2 \cup \ldots \cup S_p \cup S'$ (les S_i n'étant pas nécessairement disjoints) et tel que $\delta(s_0, f)$ appartient à S_i si et seulement si f appartient à R_i.

La preuve de ce résultat n'est autre que celle qui permet de prouver que l'union d'une famille de langages reconnaissables est reconnaissable, il suffit de vérifier que l'automate ainsi construit satisfait aux conditions de la propriété. Plus précisément, soit : $\mathscr{A}_i=(Q_i, \delta_i, q_0^i, F_i)$, l'automate qui reconnaît R_i, l'automate $\mathscr{A}$ a alors pour ensemble d'états le produit cartésien des Q_i et pour fonction de transition :

$$\delta(q_1, \ldots, q_p), x=(\delta_1(q_1, x), \delta_2(q_2, x), \ldots, \delta_p(q_p, x)),$$

il vérifie les conditions voulues en choisissant :

$$S_i=Q_1 \times Q_2 \times \ldots \times Q_{i-1} \times F_i \times Q_{i+1} \times \ldots \times Q_p.$$

3.3. Mots alternés

Pour décrire le calcul effectué par un automate fini lorsqu'il lit un mot, nous allons associer un mot qui contient à la fois les lettres lues et la suite des états atteints. Plus précisément, soit $\mathscr{A}$ un automate ayant $\mathscr{X}$ comme alphabet d'entrée et δ comme fonction de transition, on note Alt$(\mathscr{A})$ le langage inclus dans $(QX)^* Q$ formé des mots :

$$f=q_1 x_1 q_2 x_2, \ldots, q_i x_i q_{i+1}, \ldots, x_{n-1} q_n$$

tels que $\delta(q_i, x)=q_{i+1}$. Si q_0 est l'état initial de $\mathscr{A}$ on note $\mathrm{Alt}_0(\mathscr{A})$ l'intersection de $q_0(XQ)^*$ avec Alt$(\mathscr{A})$.

Soit φ le morphisme de $(Q \cup X)^*$ dans X donné par $\varphi(x)=x$ si $x \in X$, $\varphi(q)=1$ pour $q \in Q$. Pour tout mot w de Alt$(\mathscr{A})$ et tout mot f de X^*, il existe un mot unique w' tel que $\varphi(w')=\varphi(w)f$; $w' \in \mathrm{Alt}(\mathscr{A})$ et w' facteur gauche de w, on notera $w'=w \circ f$.

Soit T un système de réécriture donné par la famille $\langle R_i, g_i, d_i \rangle$.

La propriété du paragraphe précédent assure l'existence d'un automate fini $\mathscr{A}_T$ (associé au système T) reconnaissant chacun des R_i; S_i étant l'ensemble des états de $\mathscr{A}_T$ reconnaissant R_i.

Un mot w de Alt$(\mathscr{A}_T)$ est dit *réductible* s'il se décompose en $w=uqv$ de telle façon que q appartienne à un S_i et que $\varphi(v)$ soit égal au d_i correspondant, un *mot réduit* de w est alors $uq \circ g_i$. On notera Red(w) l'ensemble des mots réduits de w.

REMARQUES : 1) Si w appartient à $\mathrm{Alt}_0(A_T)$ et si v appartient à Red(w) on a : $\varphi(w) \in \mathrm{Rempl}_S(\varphi(uv))$.

2) Si uw appartient à $\mathrm{Alt}_0(A_T)$ et si v appartient à Red(w) alors uv est un élément de $\mathrm{Alt}_0(A_T)$ tel que $\varphi(uw) \in \mathrm{Rempl}_S(\varphi(uv))$.

3) S'il existe u tel que uw appartienne à $\mathrm{Alt}_0(A_T)$ et si le système vérifie la condition (3.2) alors Red(w) contient au plus un élément.

3.4. Automate à pile associé à T

Nous nous proposons de construire ici un automate à pile P_T qui reconnaisse les mots engendrés par le système T. Afin de vérifier si un mot appartient à $\mathrm{Rempl}_T^*(F)$ l'automate P_T réduit f au fur et à mesure de sa lecture et conserve le mot réduit dans la pile. Plus exactement, le mot réduit est égal au produit mot de pile, état (l'ensemble des états étant constitué par une partie de X^*). Ce type de transcription étant choisi pour permettre à l'automate de tenir compte des k derniers symboles de pile pour opérer une réduction. L'hypothèse 3.1 est évidemment cruciale pour assurer que les réductions s'effectuent toujours en fin de mot (c'est-à-dire en sommet de pile).

Choisissons un nouveau symbole $\#$ n'appartenant pas à X et pour tout système de remplacement T, étendons l'automate A_T de façon à ce que $\delta(\#, q)=q$ pour tout état q de Q. On note encore A_T ce nouvel automate, et on étend φ de façon que $\varphi(\#)=1$. Le rôle du symbole $\#$ est de maintenir constamment supérieure à k la longueur du mot pile-état.

Pour le système T, soit n la longueur maximale des mots de $D \cup F$ (où D désigne l'ensemble des mots d_i). Posons $k=2n+3$. L'automate à pile associé à T, $\mathscr{P}_T$ a pour ensemble d'états les mots écrits sur l'alphabet $Y=Q \cup X \cup \#$ de longueur au plus k. L'état initial est $(q_0 \#)^{k+1} q_0=s_0$ et la pile contient initialement $q_0 \#$, l'alphabet de pile étant $Y=\{\#\} \cup Q \cup X$.

La fonction de transition Δ est définie par les règles :

1) *normalisation :* quel que soit le mot w tel que $|w|<k$:

$$\Delta(1, w, y)=\{(yw, 1)\}.$$

2) *réduction :* pour tout w tel que $|w|=k$ et pour tout x de $X \cup \{\#\}$:

$$\Delta(1, w, x)=\{(v, x) \mid v \in \mathrm{Red}(w)\};$$

3) *lecture recopie :* pour tout q_1 dans Q; pour tous x, x_1, x' dans $X \cup \{\#\}$; et pour tout w dans $(Q \cup X)^*$ tel que $|w|=k-2$:

$$\Delta(x, q_1 x_1 w, x')=(w_0 x, x' q_1 x_1).$$

La première règle permet de retrouver des états correspondant à des mots de longueur k, la deuxième effectue des réductions chaque fois que cela est possible, enfin la troisième lit les lettres successives du mot f d'entrée en les recopiant dans le mot constitué par le produit du mot de pile par l'état.

Dans la suite on note $(w, \alpha) \underset{i}{\dashv} (w', \alpha')$ pour indiquer la règle appliquée ($i=1$, 2 ou 3).

Ainsi l'état de l'automate est toujours un élément de $Q((X \cup \#).Q)^*$ et la pile est un mot de $(Q(X \cup \#))^*$. Cet automate fonctionne comme un « réducteur » de mots comme le montrent les lemmes suivants :

LEMME 1 : *Si* (w, α) *est une configuration de l'automate alors* αw *appartient à* $(q_0 \#)^{n+2} \mathrm{Alt}_0(\mathscr{A}_T)$.

L'état initial de l'automate étant $(q_0 \#)^{n+1} q_0$, le lemme est vérifié par la configuration initiale. D'autre part, si (w, α) vérifie la propriété on se convainc facilement qu'il en est de même pour (w', α') tel que $(w, \alpha) \underset{i}{\dashv} (w', \alpha')$ quand $i=1$ (car $\alpha w = \alpha' w'$) quand $i=2$ [car $\alpha v \in \mathrm{Red}(\alpha w)$ et en utilisant la remarque 2], enfin quand $i=3$ (car $\alpha' w' = \alpha w_0 x$).

LEMME 2 : *Si* (w, α) *est une configuration atteinte après lecture de* f *alors* $\varphi(\alpha w)$ *vérifie* $f \in \mathrm{Rempl}_T^*(\varphi w)$.

La preuve s'effectue par récurrence sur le nombre de transitions qui ont eu lieu pour obtenir (w, α). S'il n'y en a eu qu'une seule, le mot lu est composé d'une seule lettre x, et $\varphi(w) = x$. Si $(w, \alpha) \underset{1}{\dashv} (w', \alpha')$, le mot f lu pour obtenir (w, α) est le même que celui lu pour obtenir (w', α'). Comme $\alpha w = \alpha' w'$ on a $f \in \mathrm{Rempl}_T \varphi(\alpha' w')$ si $f \in \mathrm{Rempl}_T(\varphi(\alpha w))$. Si $(w, \alpha) \underset{2}{\dashv} (w', \alpha')$ le mot f lu est le même pour les deux configurations et on a $\alpha = \alpha'$; d'après le lemme précédent et les remarques du paragraphe 3.3, on a $\varphi(\alpha' w') \in \mathrm{Rempl}_T(\varphi(w))$ ainsi $\varphi(\alpha' w')$ appartient à $\mathrm{Rempl}_T(f)$. Si $(w, \alpha) \underset{3}{\dashv} (w', \alpha')$ les mots lus sont f et fx, et de plus $\varphi(\alpha' w') = \varphi(\alpha w).x$; on se convainc alors que $\varphi(\alpha w) \in \mathrm{Rempl}_T^*(F)$ implique $\varphi(\alpha w).x \in \mathrm{Rempl}_T(fx)$.

3.5. Systèmes de remplacement préfixes

Remarquons que les lemmes précédents ont été établis sans hypothèse particulière sur le système T; supposons maintenant qu'il satisfait (3.1) c'est-à-dire que pour tout $r_i \in R_i$ et $r_j \in R_j$, $r_i g_i = r_j d_j f$ implique que f est le mot vide. On a alors :

LEMME 3.3 : *Si le mot* f *appartient à* $\mathrm{Rempl}_T^*(u)$ *il existe un calcul de l'automate* $\mathscr{P}$ *tel que la configuration* (w, α) *obtenue après avoir lu* f *vérifie :* $\varphi(\alpha w) = u$.

La preuve de ce lemme s'effectue par récurrence sur l'entier p tel que $f \in \mathrm{Rempl}_T^p(u)$.

Si $n=0$, u est alors égal à f, et le calcul composé de transition de type 3 (c'est-à-dire de lecture recopie) conduit de manière évidente à une configuration (w, α) telle que $\varphi(\alpha w)=f$.

Supposons que f appartienne à $\mathrm{Rempl}_T^{p+1}(u)$ alors il existe g appartenant à $\mathrm{Rempl}_T^p(u)$ tel que :

$$f=f_1 d_i f_2, \qquad g=f_1 g_i f_2, \qquad f_1 \in R_i.$$

L'hypothèse de récurrence assure alors l'existence d'un calcul conduisant après lecture de g à la configuration (w, α) telle que $\varphi(\alpha w)=u$. D'après la condition 3.1 ce calcul ne peut contenir de réduction qu'après la lecture complète de $f_1 g_i$ ainsi :

$$(f_1 g_i f_2, (q_0\#)^{k+1} q_0, q_0\#) \underset{3}{\overset{*}{\dashv}} (g_i f_2, w_1, \alpha_1) \underset{3}{\overset{*}{\dashv}} (f_2, w_2, \alpha_2) \overset{*}{\dashv} (1, w, \alpha).$$

Les deux premiers $\dashv$ ne contenant pas de réductions, w_1 se termine par l'état $\delta(s_0, f_1)$.

On peut construire alors :

$$(f_1 d_i f_2, (q_0\#)^k q_0, q_0\#) \underset{3,4}{\dashv^*} (d_i f_2, w_1, d_j) \underset{3,4}{\dashv^*} (f_2, w_3, \alpha_3)$$

où $\varphi(\alpha_3 w_3)=f_1 d_i$ et $\alpha_3 w_3=\alpha_1 w_1 \circ d_i$. Ainsi, puisque w_1 se termine par un q_i de S_i, $(f_1 \in R_i)$, on a $(f_2, w_3, \alpha_3) \underset{2}{\dashv} (f_2, w_2, \alpha_2)$ et le calcul peut être complété comme celui de g pour obtenir $(1, w, \alpha)$.

Ces lemmes étant prouvés, on est maintenant en mesure de démontrer le théorème 2 :

Soit l'automate $\mathscr{P}$ associé au système de remplacement T_{R_i, g_i, d_i}; considérons comme états terminaux l'ensemble $\hat{F}$ formé des mots u de $(q_0\#)^+ \mathrm{Alt}_0(\mathscr{A}_T)$ tels que $\varphi(u)$ appartienne à F. Si f est un mot du langage solution de l'équation, c'est qu'il existe f_0 dans F tel que $f \in \mathrm{Rempl}_T^*(f_0)$; d'après le lemme précédent, il existe un calcul qui conduit par lecture de f à la configuration (w, α) telle que $\varphi(\alpha w)=f_0$; f_0 étant de longueur inférieure à $(k-3)/2$, on a $\varphi(w)=f_0$ et $w \in \mathrm{Alt}_0(A_T)$ d'après le lemme 3.1, f est donc reconnu par l'automate.

Réciproquement, si f est reconnu par l'automate, l'état contenant une lettre $\#$, la pile ne peut contenir de lettres de X, et $\varphi(\alpha w)=\varphi(w)$ appartient à F. D'après le lemme 3.2, f peut être obtenu à partir de $\varphi(w)$ par remplacements successifs contrôlés et il appartient au langage solution de (1).

3.6. Preuve du théorème 3

Nous effectuons cette preuve en construisant un automate à pile déterministe du même type que celui construit plus haut. Nous supposons de plus qu'aucun mot de F n'admet de facteur gauche dans $R_i d_i$. Pour traiter du cas général, il faudrait construire un automate à pile qui ne commence les réductions qu'une fois qu'il a vérifié que le mot d'entrée n'est pas dans F, ceci peut être fait sans trop de problèmes du fait que la longueur des mots de F est majorée; nous laissons le soin au lecteur de s'en convaincre.

Automate à pile déterministe $\mathscr{DP}$ associé à un système réécriture T vérifiant 2.1 *et* 2.2.

Il a mêmes états, symboles de piles que l'automate $\mathscr{P}_T$. Ses transitions Δ' sont données par :

(1') $\Delta'(1, w, y)=(yw, 1)$ pour tout mot w de longueur plus petite que k.

$(2')_a$ Si w est réductible et de longueur k alors :

$$\Delta'(1, w, x')=(v, x') \qquad \text{où} \quad v \in \text{Red}(w), \qquad x' \in X.$$

$(2')_b$ Si w est un mot irréductible et de longueur k et si $w \circ x$ est réductible, alors :

$$\Delta'(x, w, x')=(v, x') \qquad \text{où} \quad v \in \text{Red}(w \circ x) \quad \text{et} \quad x' \in X.$$

(3') Si $q_1 x_1 w_1$ est un mot irréductible de longueur k et si $q_1 x_1 w_1 \circ x$ est aussi irréductible :

$$\Delta'(x, q_1 x_1 w_1, x')=(w_1 \circ x, x' q_1 x_1).$$

Il est aisé de vérifier que les lemmes 1 et 2 sont satisfaits par l'automate $\mathscr{DP}$ dont les règles de transition sont plus restrictives que celles de $\mathscr{P}$ (toute transition de $\mathscr{DP}$ peut-être simulée par une ou deux transitions de $\mathscr{P}$).

Le fait que l'automate $\mathscr{DP}$ est déterministe apparaît clairement dans sa construction et résulte de ce que $\text{Red}(w)$ contient au plus un élément si $\varphi(w) \in R_i d_i$.

Il reste donc à vérifier que quel que soit le mot f de $\text{Rempl}_T(F)$ il existe une suite de transitions de $\mathscr{DP}$ conduisant à une configuration (w, α) telle que $\varphi(\alpha w)$ appartienne à F. Or ceci résulte d'un lemme analogue au lemme 3 et dont la preuve est en tous points analogue; il n'est pas besoin de la reproduire ici.

3.7. Un système satisfaisant 2.1 et 2.2

Considérons le système composé des trois triples suivants :

$$((abcde)^* a, cde, bcded), ((abcde)^*, 1, ac), (de^*, ed, e^2 d^2).$$

On peut aisément vérifier que ce système vérifie 2.1 et 2.2 (mais pas 3.1), en effet $(abcde)^n acde$ appartient à $R_1 g_1$ et $R_2 d_2 X^*$ mais g_2 est bien égal au mot vide.

Le mot $(abcde)^n d^{n+1}$ appartient à Rempl* (ded) et la suite des réductions obtenues en lisant le mot de gauche à droite est la suivante :

(i) lecture de $(abcde)^{n-1} abcde$ sans réduction;

(ii) lecture de d et réduction à $(abcde)^{n-1} acde$;

(iii) réductions successives *sans lecture* aux mots suivants :

$$(abcde)^{n-1} de, (abcde)^{n-2} acde^2, (abcde)^{n-2} de^2, \ldots,$$
$$(abcde)^{j+1} de^{n-j-1}, (abcde)^j acde^{n-j}, (abcde)^j de^{n-j}, \ldots, de^n;$$

(iv) lectures successives des lettres d et réduction de $de^n d^2$ en $de^{n-1} d$ jusqu'à obtention de ded.

En examinant de près la phase (iii), on s'aperçoit que de nombreuses réductions sont effectuées sans lecture de nouveau symbole; de plus ces réductions occurrent en laissant un facteur droit non vide et de longueur *non bornée*. Ce fait interdit la construction d'un automate à pile fonctionnant de la même façon que l'automate $\mathscr{P}$ défini plus haut. Remarquons toutefois que le système ainsi construit engendre le langage $(ac)^* (abcde (ac)^*)^n de^p d^{n+p}$ à partir de $A = \{ded\}$; de plus, par un A quelconque, il n'est pas très difficile de montrer (bien que fastidieux !) que le langage obtenu est algébrique.

BIBLIOGRAPHIE

1. J. BERSTEL, *Congruences plus que parfaites et langages algébriques*, Séminaire d'Informatique théorique (75-76-77), Paris-VI, p. 123-147.
2. P. BUTZBACH, *Une famille de congruences de Thue pour lesquelles le problème de l'équivalence est décidable* in M. NIVAT, éd. Automata, Languages and Programming, North Holland 1973, p. 3-12.
3. P. BUTZBACH, *Sur l'équivalence des grammaires simples* in J. P. CRESTIN et M. NIVAT, éd., Langages algébriques, Actes des Journées d'Informatique théorique de Bouascre, E.N.S.T.A., Paris, 1978.
4. Y. COCHET, *Sur l'algébricité de certaines congruences définies sur le monoïde libre*, Thèse 3e cycle Rennes, 1971.
5. Y. COCHET et M. NIVAT, *Une généralisation des ensembles de Dyck*, Israël J. of Math., vol. 9, 1971, p. 389-395.
6. S. GINSBURG, *The Mathematical Theory of Context Free Languages*, McGraw Hill, New York, 1966.
7. M. HARRISON, *Introduction to Formal Languages Theory*, Addison Wesley, 1978.
8. J. HOPCROFT et J. ULLMAN, *Formal Languages and their Relation to Automata*, Addison Wesley, Reading 1969.
9. F. KIERSZENBAUM, *Les langages à opérateurs d'insertion*, Thèse 3e cycle, Université de Bordeaux-I, 1979.
10. M. NIVAT, *On Some Families of Languages Related to the Dyck Language*, 7th Ann. Symp. on Switching and Automata Theory, Berkeley, 1966, p. 36-46.

11. M. NIVAT, *Congruences de Thue et t-Langages*, Studia Sc. Math. Hungarica, vol. 6, 1971, p. 243-249.

12. J. SAKAROVITCH, *Un théorème de transversale rationnelle pour les automates à pile déterministe*, Proc. 4th G.I. Conf. on Theoretical Computer Science, K. WEIRAUCH, éd., Springer-Verlag (à paraître).

13. J. SAKAROVITCH, *Syntaxe des langages de Chomsky. Essai sur le déterminisme*, Thèse d'État de Math., Université de Paris-VII, Paris, 1979.

14. A. SALOMAA, *Formal languages*, Academic Press, New York, 1973.