AUTOMATES A PILE SUR DES ALPHABETS INFINIS

Jeanne IDT
Faculté des Sciences Economiques
47 X Centre de tri
38040 GRENOBLE Cédex - FRANCE

RESUME

Soit Σ un alphabet infini dénombrable. Une grammaire $G = <\Sigma,V,P>$ est Σ-algébrique si V est un ensemble fini de non-terminaux et si $P = \{(\alpha\to m) \ / \ \alpha\in V,\ m\in(\Sigma\cup V)^*\}$ est tel que la longueur de m soit bornée. On peut voir que des langages engendrés par de telles grammaires ne peuvent être caractérisés par des automates à pile classiques. C'est pourquoi il a été nécessaire de construire une nouvelle sorte d'automates que j'ai appelé "automates à pile et à tiroirs". La construction et les propriétés de ces automates sont exposés dans cet article.

I - INTRODUCTION

Un langage, d'après la définition de N. CHOMSKY, est un ensemble de mots finis de X^*, où X est un alphabet fini, et les recherches effectuées en théorie classique des langages ont pour but de donner une représentation finie de langages infinis.

Une première manière de définir des langages est d'utiliser un processus fini pouvant les engendrer, appelé grammaire [6].

Une grammaire G est un triplet $G = <X,V,P>$ où V est un alphabet fini non vide, dit alphabet non terminal, où X est un alphabet disjoint de V fini non vide, dit alphabet terminal, et où P est un ensemble fini de règles de la forme $\alpha\to\beta$ où α appartient à $(V\cup X)^*\ V(V\cup X)^*$ et β à $(V\cup X)^*$.

Parmi les plus étudiées, citons les grammaires algébriques engendrant la famille des langages algébriques, et les grammaires linéaires droites engendrant la famille des langages rationnels. En imposant ainsi des conditions à α et β, il est possible de former différentes familles de langages dont l'ensemble forme la hiérarchie de CHOMSKY.

La seconde manière de définir des langages est d'utiliser un processus fini pouvant les reconnaître, ou automate, [6]. Parmi ces processus classiques de reconnaissance citons :

- les automates finis qui sont des 5-uplets (Q,X,δ,q_o,F) où Q est un ensemble fini, dit ensemble d'états, X l'alphabet fini d'entrée, δ une fonction, dite de transistion, de $Q\times X$ dans Q, $q_o\in Q$ un état initial et $F\subseteq Q$ un ensemble d'états dits terminaux. La famille des langages reconnus par ces automates coïncide avec la famille des langages rationnels.

- les automates à pile qui sont des 7-uplets $(Q,X,\Gamma,\delta,q_o,Z_o,F)$ où Q est un ensemble fini d'états, X l'alphabet d'entrée, Γ un alphabet fini de symboles de pile, q_o un état initial, Z_o un symbole initial de pile, $F \subseteq Q$ un ensemble d'états terminaux et δ une fonction de $Qx(Xx\{\varepsilon\})x\Gamma$ dans l'ensemble des sous ensembles fini de $Qx\Gamma^*$. Ces automates reconnaissent exactement la famille des langages algébriques.

La notion de grammaire s'applique particulièrement bien aux langages de programmation dont la syntaxe est définie en effet par une grammaire, essentiellement algébrique, par exemple dans le cas d'ALGOL. Le problème est de pouvoir dire en un temps fini si un programme donné c'est à dire un mot du langage, est syntaxiquement correct, donc de construire un automate à pile reconnaissant ce langage. Mais pour qu'un tel automate puisse fonctionner rapidement il faut que chacun de ses mouvements soit entièrement déterminé par l'état, la lettre lue et le symbole de sommet de pile. C'est ce qui conduit à la notion d'automate déterministe :

un automate à pile $(Q,X,\Gamma,\delta,q_o,Z_o,F)$ est dit déterministe si δ vérifie :
1) $\forall q\in Q$, $\forall a\in X$, $\forall Z\in\Gamma$, $|\delta(q,a,Z)|\leq 1$
2) si $\delta(q,\varepsilon,Z) \neq \emptyset$ alors $\delta(q,a,Z) = \emptyset$, $\forall a\in X$.

On peut alors définir la famille des langages reconnus par un automate à pile déterministe.

La théorie classique des langages s'est d'abord limitée à l'étude des sous ensembles de X^*, où X est un alphabet fini, puis certaines modifications de ces hypothèses ont été apportées par exemple par Mc.NAUGHTON [7] et M. NIVAT [8] qui ont étudié des mots infinis et enfin par J.M. AUTEBERT, J. BEAUQUIER et L. BOASSON qui ont étudié des langages sur des alphabets infinis [1]. A priori, il apparaît en effet que certains problèmes importants peuvent s'exprimer à l'aide de langages sur des alphabets infinis. C'est le cas, par exemple, des systèmes de sécurité dans lesquels la création de nouveaux objets est autorisée et qui peuvent être considérés comme des langages sur l'alphabet ΣxM où Σ est l'ensemble des sujets et des objets du système et M l'ensemble des modalités d'accès des sujets aux objets [4,5]. Or, parmi ces objets figurent les programmes écrits par les utilisateurs qui sont donc en nombre a priori non borné.

C'est encore le cas, par exemple, des calculs réalisés à l'aide de procédures récursives. L'ensemble des calculs possibles peut être considéré comme un langage. Si les procédures récursives contiennent des variables locales, à chaque appel de la procédure correspond une lettre différente. Il s'agit donc, là encore, d'un langage sur un alphabet non borné.

A posteriori, l'étude des langages sur des alphabets infinis se justifie puisqu'elle a déjà permis de caractériser des cas où la limite d'une suite croissante de langages algébriques était algébrique [3].

Rappelons les principales définitions et propriétés qui ont été données dans l'article [1].

II - RAPPELS

Soit Σ un alphabet infini dénombrable.

Définition 1 : Une grammaire $G = <\Sigma,V,P>$ est une p-grammaire si V et P sont infinis dénombrables et si les membres droits des règles de P sont de longueur bornée.

Définition 2 : Une p-grammaire $G = <\Sigma,V,P>$ est Σ-algébrique si, de plus V est fini. On dira alors que G est une Σ-grammaire.

Définition 3 : Une p-grammaire $G = <\Sigma,V,P>$ est Σ-rationnelle si elle est Σ-algébrique et si, pour toute règle $(v\to m)$ de P, m est dans $\Sigma^*V\cup\{\varepsilon\}$ (autrement dit, si G est linéaire droite).

Définition 4 : Un langage L de Σ^* est Σ-algébrique (resp. Σ-rationnel) s'il est engendré par une p-grammaire Σ-algébrique (resp. Σ-rationnelle).

Exemple : L'extension du langage de DYCK à l'alphabet infini dénombrable $\Sigma = \{a_i,\bar{a}_i \ / \ i\in N\}$ est engendré par la Σ-grammaire $G = <\Sigma,V,P>$ où $V = \{S\}$ et $P = \{S\to a_iS\bar{a}_iS \ / \ i\in N\} \cup \{S\to\varepsilon\}$. Il est donc Σ-algébrique.

Définition 5 : (Rappel) Soit M un monoïde quelconque. Une partie A de M est un sous ensemble reconnaissable s'il existe un monoïde fini N, un morphisme $\alpha:M\to N$ et un sous ensemble $P\subset N$ tels que $A = \alpha^{-1}(P)$. On note $Rec(M)$ cette famille.

Rappelons maintenant quelques propriétés vérifiées par ces langages.

Proposition 1 : L'ensemble des langages reconnaissables de Σ^* est strictement contenu dans l'ensemble des langages Σ-rationnels. $Rec(\Sigma^*) \subsetneq \Sigma\text{-Rat}$.

Proposition 2 : Il existe des langages Σ-algébriques qui ne sont pas engendrables par des Σ-grammaires sous forme normale de GREIBACH (i.e. les règles $(v\to m)$ de P sont telles que m appartient à ΣV^*).

Proposition 3 : Tout langage Σ-algébrique peut être engendré par une Σ-grammaire sous forme faible de GREIBACH (i.e. les règles $(v\to m)$ de P sont telles que m appartient à $\Sigma(\Sigma\cup V)^*$).

Proposition 4 : La famille des langages Σ-algébriques (resp. Σ-rationnels) est fermée par union, produit, étoile, homomorphisme alphabétique, homomorphisme alphabétique inverse, intersection reconnaissable.

Proposition 5 : La famille des langages Σ-algébriques (resp. Σ-rationnels) n'est pas fermée par intersection Σ-rationnelle (resp. par intersection).

Proposition 6 : Les langages Σ-algébriques vérifient le lemme d'OGDEN.

Exemple de langage non Σ-algébrique : Soient R_1 et R_2 les deux langages Σ-rationnels suivants :

$R_1 = \{a_{2i}a_{2i+1} \ / \ i\in N\}$ engendré par $P = \{S\to a_{2i}a_{2i+1} S/i\in N\} \cup \{S\to\varepsilon\}$

$R_2 = a_o\{a_{2i+1}a_{2i+2} \ / \ i\in N\}^* \ \{a_{2i+1} \ / \ i\in N\}$ engendré par

$P = \{S\to a_o T\} \cup \{T\to a_{2i+1}a_{2i+2}T, \ U\to a_{2i+1} \ / \ i\in N\} \cup \{T\to U\}$

Le langage $R_1 \cap R_2 = \{a_o a_1 a_2 \ldots a_{2i+1} \ / \ i\in N\}$, ne vérifiant pas le lemme d'OGDEN, n'est pas Σ-algébrique.

III - PROBLEME DE LA RECONNAISSANCE DES LANGAGES Σ-ALGEBRIQUES PAR DES AUTOMATES A PILE CLASSIQUES

Soit $L = L_G(S)$ un langage Σ-algébrique engendré par la Σ-grammaire $G = <\Sigma,V,P>$. Il est facile de vérifier qu'il est reconnu par un automate à pile classique, mais avec un alphabet de pile infini, soit $A = (Q,\Sigma,\Gamma,\delta,q_o,S_o,F)$ où $Q = \{q_o,q\}$, où $\Gamma = \Sigma\cup V\cup\{S_o\}$ et où δ vérifie :

1) $\delta(q_o,\varepsilon,S_o) = \{(q,m) \ / \ m\in(\Sigma\cup V)^*$ et $(S\to m)\in P\}$
2) $\forall a\in\Sigma$, $\delta(q,a,a) = (q,\varepsilon)$
3) $\forall T\in V$, $\delta(a,\varepsilon,T) = \{(q,m) \ / \ m\in(\Sigma \cup V)^*$, $(T\to m)\in P\}$

Mais réciproquement il existe des automates à pile classiques, dont l'alphabet de pile est infini, et reconnaissant des langages qui ne sont pas Σ-algébriques. Par exemple, l'automate $A = (Q,\Sigma,\Gamma,\delta,q_o,\bar{a}_o,F)$ où $\Sigma = \{a_i \ / \ i\in N\}$, $\Gamma = \{\bar{a}_i \ / \ i\in N\}$, $Q=\{q_o,q\}$ et où δ est donnée par

1) $\delta(q_o,a_o,\bar{a}_o) = (q,\bar{a}_1)$
2) $\forall i\in N^*$, $\delta(q,a_i,\bar{a}_i) = (q,\bar{a}_{i+1})$
3) $\forall i\in N^*$, $\delta(q,\varepsilon,\bar{a}_{2i}) = (q,\varepsilon)$.

Cet automate reconnait $R = \{a_o a_1 a_2 \ldots a_{2i+1} \ / \ i\in N\}$ qui n'est pas Σ-algébrique.

Les procédés classiques de reconnaissance ne convenant pas, il devenait donc nécessaire d'en définir de nouveaux. C'est pourquoi j'ai construit des automates dits à pile et à tiroirs et caractérisant les langages Σ-algébriques.

IV - EXEMPLE DE FONCTIONNEMENT D'UN AUTOMATE A PILE ET A TIROIRS

Le fonctionnement d'un automate à pile et à tiroirs étant assez complexe, nous allons commencer par étudier un exemple.

Etant donné l'alphabet infini $\Sigma = \{d\} \cup \{a_i,b_i \ / \ i\in N\}$, on considère le langage L engendré par la p-grammaire Σ-algébrique $G = <\Sigma,V,P>$ où $V = \{S\}$ et $P = \{S\to d\} \cup \{S\to a_i S b_i \ / \ i\in N\}$. On utilisera de façon fondamentale les deux nombres :

N = le nombre de variables = $|V|$

p = la longueur maximale des membres droits des règles.

Ici N = 1 et p = 3. On rebaptisera les variables en les notant de 1 (1 pour l'axiome S) à N. Soit x une nouvelle lettre ($x\notin V$). On forme l'ensemble MOT = $\{u \ (\{x\}\cup V)^*, |u|\leq p\}$. Ici MOT = $\{\varepsilon$,x,xx,xxx,xx1,x1,x1x,x11,1,1x,1xx,1x1,11,11x,111$\}$ et $|MOT| = 15$. Soit W = $\{1,\ldots,15\}$. Soit l'application m de W dans MOT qui à un nombre i associe le

mot de rang i de MOT, écrit dans l'ordre lexicographique, $m : W \rightarrow MOT$. Par exemple $m(1) = \varepsilon$, $m(2) = x$, $m(5) = xx1$. Soit z une nouvelle lettre, $z \neq x$ et $z \notin V$. On transforme les mots m(j) en mots de longueur p+1, appelés type (j), en les complétant à droite avec des z. Par exmple type (3) = xxzz, type (4) = xxxz. On appelera <u>tiroir</u> une fonction t qui, au couple (i,j) de VxW tel que $m(j) = x^{n_1} i_1 x^{n_2} i_2 \dots i_p x^{n_{p+1}}$, fait correspondre une partie de $\{i\} \times \Sigma^{n_1} i_1 \Sigma^{n_2} i_2 \dots i_p \Sigma^{n_{p+1}}$. Par exemple, si $V = \{1\}$ et $m(2) = x$, on pourra prendre $t(1,2) = \{(1,d)\}$ ou $t(1,2) = \{(1,a_{2i}) / i \in N\}$ ou $t(1,2) = \emptyset$
Autre exemple : si $V = \{1,2\}$ et $m(5) = xx1$ on pourra prendre $t(2,5) = \{(2, a_{2i} a_{2i+1} 1) / i \in N\}$. On peut alors définir la fonction <u>début</u>, notée déb, qui à une lettre y de $\Sigma \cup V$ associe l'ensemble des couple (i,j) dont le tiroir t(i,j) contient un membre droit au moins commençant par y. Ainsi supposons les tiroirs définis par :

$t(1,2) = \{(1,d)\}$, $t(1,7) = \{(1, a_i 1 b_i) / i \in N\}$

T $t(1,j) = \emptyset$ si $j \neq 2$ et $j \neq 7$.

Alors $déb(d) = \{(1,2)\}$, $déb(a_i) = \{(1,7)\}$, $déb(b_i) = \emptyset$

La pile du nouvel automate est constituée de trois parties liées mais distinctes :

- TIR où sera indiqué le nom du tiroir contenant la règle en cours de reconnaissance
- P qui contiendra des mots de longueur p+1 à savoir les mots type(j) dans lesquels les lettres x seront remplacées par les lettres de Σ au fur et à mesure de la lecture du mot à reconnaître,
- C qui est un compteur indiquant à quelle composante du vecteur P on en est arrivé.

TIR	P	C

On suppose que les tiroirs sont définis par l'ensemble des relations (T) et l'on cherche à reconnaître le mot $a_1 a_2 d\ b_2 b_1$ de L. Au départ la pile contient le symbole $(\emptyset, \emptyset, \emptyset)$:

1) <u>On lit a_1</u> On cherche parmi les tiroirs t(1,j) s'il en existe possédant des facteurs droits commençant par a_1. On constate, dans les relations (T), que le tiroir t(1,7) appartient à $déb(a_1)$. On empile (1,7) dans TIR, type (7)=x1xz dans P. Puis on remplace P(1,1) = x par $p(1,1) = a_1$. Enfin on met 2 dans C(1) ce qui indique qu'à la prochaine lecture il faudra regarder ce qu'indique la $2^{ème}$ lettre de P c'est à dire P(1,C(1)) = P(1,2).

1,7	a_1	1	x	z	2

2) <u>P(1,C(1)) = P(1,2) = I</u> (c'est à dire une variable) Puisque l'on trouve une variable, cela signifie que l'on attend l'ouverture d'un nouveau tiroir (i,j) où i est égal à la variable indiquée, ici i = 1. On lit alors dans le mot à reconnaître la lettre suivante c'est à dire a_2. On constate que le tiroir (1,7) appartient à $déb(a_2)$. Alors on augmente C(1) de 1, d'où C(1) = 3, ce qui

signifie que, lorsque cette ligne se trouvera au sommet de pile, il faudra lire la troisième lettre de P, P(1,3). On empile (1,7) dans TIR, type (7)=x1xz dans P, 2 dans C et on remplace P(2,1) = x par P(2,1) = a_2.

2	1,7	a_2	1	x	z	2
1	1,7	a_1	1	x	z	3

3) P(2,C(2)) = P(2,2) = 1 (c'est à dire une variable) Comme en 2), on attend l'ouverture d'un tiroir (1,j). On lit alors dans le mot à reconnaître la lettre d et l'on constate que le tiroir (1,2) appartient à déb(d). On augmente C(2) de 1. On empile (1,2) dans TIR, type (2) = xzzz dans P, 2 dans C et on remplace P(3,1) par d.

3	1,2	d	z	z	z	2
2	1,7	a_2	1	x	z	3
1	1,7	a_1	1	x	z	3

4) P(3,C(3)) = P(3,2) = z On trouve un z, ce qui signifie que l'on doit dépiler la ligne 3 car une règle du tiroir (1,2) vient d'être reconnue entièrement.

2	1,7	a_2	1	x	z	3
1	1,7	a_1	1	x	z	3

5) P(2,C(2)) = P(2,3) = x On lit dans le mot à reconnaître la lettre suivante, soit b_2 et l'on remplace P(2,3) = x par P(2,3) = b_2. On vérifie si le facteur gauche de longueur C(2) = 3, ici $a_2 1 b_2$, du mot écrit dans P, ici $a_2 1 b_2$ z, est un facteur gauche possible d'un facteur droit du tiroir indiqué dans TIR, ici (1,7). D'après les relations (T) on a bien (1,$a_2 1 b_2$) dans t(1,7). On peut alors augmenter C(2) de 1, C(2) = 4.

2	1,7	a_2	1	b_2	z	4
1	1,7	a_1	1	x	z	3

6) P(2,C(2)) = P(2,4) = z Comme en 4), on dépile la ligne 2 puisqu'une règle du tiroir (1,7) vient d'être reconnue entièrement.

7) P(1,C(1)) = P(1,3) = x On procède comme ne 5) et on lit dans le mot à reconnaître la lettre b_1. On constate que $a_1 1 b_1$ est un facteur droit de t(1,7). Donc la ligne 1 devient :

1	1,7	a_1	1	b_1	z	4

8) P(1,C(1)) = P(1,4) = z On dépile. La pile est vide et le mot a été reconnu par pile vide par cet automate à pile et à tiroirs.

V - DEFINITION D'UN AUTOMATE A PILE ET A TIROIRS

Un automate à pile et à tiroirs est donné par :

1) (N,p) un couple d'entiers

Ces 2 entiers déterminent les ensembles :

- $V = \{1,\dots,N\}$
- $MOT = \{e\} \bigcup_{i=1}^{p} (V \cup \{x\})^i$, où e représente le mot vide et où x est une lettre hors de $V \cup \Sigma$.
- $w = \{1,\dots,M\}$ où $M = |MOT|$

et les applications :

- $m : W \to MOT$, qui est la numérotation des éléments de MOT par ordre lexicographique
- $type : W \to MOT\ (\bigcup_{q=1}^{p} z^q)$ où z est une lettre hors de $\Sigma \cup V \cup \{x\}$ et où $type(j) = m(j)\, z^{p+1-|m(j)|}$

2) La fonction tiroir t

$$t : V \times N \longrightarrow \mathcal{P}(V \times \bigcup_{i=0}^{p} (\Sigma \cup V)^i)$$

$$* \ j \neq 1,\ (i,j) \longmapsto t(i,j) \subset \{i\} \times \Sigma^{n_1} i_1\, \Sigma^{n_2} i_2 \dots \Sigma^{n_{q+1}}$$

$$\text{si } m(j) = x^{n_1} i_1\, x^{n_2} i_2 \dots i_q\, x^{n_{q+1}} \quad (\text{où } i_j \in V)$$

$$* \ j = 1,\ (i,1) \longmapsto t(i,1) \subset \{(i,e)\}$$

On définit de plus la fonction début :

$$\text{déb} : \Sigma \cup V \longrightarrow \mathcal{P}(V \times W)$$

$$\text{déb}(y) = \{(i,j) \ /\ \exists W \in (\Sigma \cup V)^*,\ (i,yw) \in t(i,j)\}$$

La pile de l'automate est formée de mots de longueur $2 + (p+1) + 1$:

$$\underbrace{\boxed{\quad}}_{\substack{2 \\ TIR}}\ \underbrace{\boxed{\qquad\qquad\qquad}}_{\substack{p+1 \\ P}}\ \underbrace{\boxed{\quad}}_{\substack{1 \\ C}}$$

- Dans TIR, seront placés des éléments (i,j) de $V \times W$
- Dans P seront placés d'abord des mots type(j) faisant place petit à petit à des mots de $(\Sigma \cup V)^*$
- Dans C sera placé un pointeur valant de 1 à p+1 indiquant à quelle case de P on en est arrêté dans le remplissage de P.

La fonction de transistion λ indique comment varie l'état de la pile (TIR, P, C) en fonction de la lettre lue.

$$\lambda : \underbrace{\{\emptyset\} \cup V \times W}_{TIR} \times \underbrace{\{\emptyset\} \cup (\{x\} \cup V \cup \{z\} \cup \Sigma)^p z}_{P} \times \underbrace{\{\emptyset\} \cup N}_{C} \times \underbrace{\Sigma \cup \{\varepsilon\}}_{\text{lettre lue}} \longrightarrow$$

$$\mathcal{P}((V \times W) \cup x \cup (\{x\} \cup V \cup \{z\} \cup \Sigma)^p z \times N)$$

Au départ la pile convient le symbole de fond de pile $(\emptyset\, \emptyset\, \emptyset)$ et on peut lire une lettre a ou le mot vide.

1 Début

1-a. On lit une lettre a de Σ

$$\lambda(\emptyset, \emptyset, \emptyset, a) = \{((1,j),\ type(j),\ 2) \ /\ (1,j) \in \text{Déb}(a)\}$$

$$P(1,1) = a$$

1-b. On lit le mot vide ε

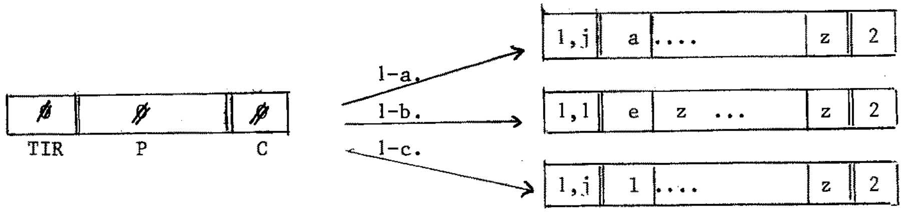
$\lambda(\not\emptyset, \not\emptyset, \not\emptyset, \varepsilon) = \{((1,1), ezz...z, 2) \ / \ t(1,1) \neq \emptyset\}$

1-c. On lit le mot vide ε

$\lambda(\not\emptyset, \not\emptyset, \not\emptyset, \varepsilon) = \{((1,j), type(j), 2) \ / \ (1,j) \in Déb(1)\}$

$P(1,1) = 1$

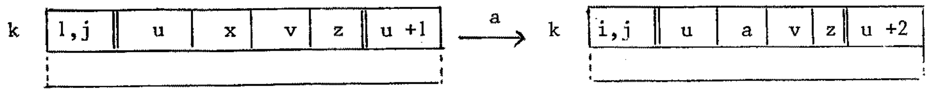
$1 \in V$



2 On attend la lecture d'une lettre de Σ : $P(k,C(k)) = x$

$\lambda((i,j), u\,x\,vz, |u|+1, a) = ((i,j), uavz, |u|+2)$

Si ua est facteur gauche d'un élément de $t(i,j)$.



3 On trouve une variable 1 dans $P(k,C(k))$ et on attend la lecture d'un membre droit de tiroir

3-a. On lit une lettre a de Σ

$\lambda((i,j), ulvz, |u|+1, a) = \{((1,m), type(m), C(k) = C(k)+1) \ / \ (1,m) \in Déb(a)\}$

$P(k+1,1)=a \quad C(k+1) = 2$

3-b. On lit le mot vide ε

$\lambda((i,j), ulvz, |u|+1, \varepsilon) = \{((1,1), ezz..z, C(k) = C(k) + 1) \ / \ t(1,1) \neq \emptyset\}$

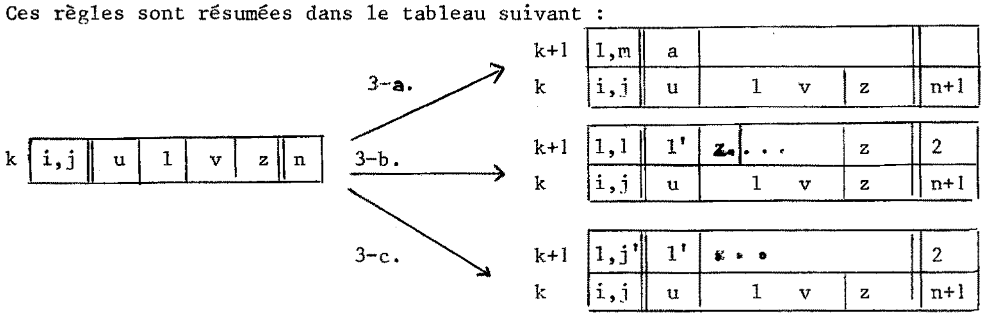
$C(k+1) = 2$

3-c. On lit le mot vide ε

$\lambda((i,j), ulvz, |u|+1, \varepsilon) = \{((1,j'), type(j') \quad , C(k) = C(k)+1 \ / \ (1,j') \in Déb(1')\}$

$P(k+1,1)=1' \quad C(k+1) = 1$

Ces règles sont résumées dans le tableau suivant :



4 On trouve le critère de fin de ligne $P(k,C(k)) = z$

Cela signifie que, au sommet de la pile, dans $P(k)$ est écrit un membre droit (tout entier) du tiroir $t(i,j)$. On dépile alors la ligne k sans rien lire :

$$\lambda((i,j),\ uz\ldots z,\ |u|+1,\ \varepsilon) = (\emptyset,\ \emptyset,\ \emptyset)$$

Remarque sur les ε-move

Les ε-move 1-b. et 3-b donnent la configuration de pile suivante :

$$\begin{array}{|c||c|c|c||c|}\hline i,j & u & 1 & v & n \\ \hline\end{array} \xrightarrow{\varepsilon} \begin{array}{|c||c|c|c|c||c|}\hline 1,j' & e & z & \ldots & z & 2 \\ \hline i,j & & u & 1 & v & n+1 \\ \hline\end{array} \longrightarrow$$

$$\longrightarrow \begin{array}{|c||c|c||c|}\hline i,j & u & 1 & n+1 \\ \hline\end{array}$$

VI - LANGAGES RECONNUS PAR UN AUTOMATE A PILE ET A TIROIRS

Proposition 7 Un langage Σ-algébrique est reconnu par un automate à pile et à tiroirs par pile vide.

Preuve a) Soit L un langage de Σ^* engendré par la Σ-grammaire $G = \langle\Sigma, V, P\rangle$. On peut alors construire l'automate à pile (N,p,t) où $N = |V|$, où p est la longueur maximale des membres droits des règles de P et où la fonction tiroir t est donnée par :

$$t(i,j) = \{(i,u)/i\in V,\ (i\to u)\in P, m(j) = x^{n_1} i_1\ x^{n_2} i_2 \ldots\ i_q\ x^{n_{q+1}},\ u\in \Sigma^{n_1} i_1\ \Sigma^{n_2} i_2 \ldots i_q \Sigma^{n_{q+1}}\}$$

soit w un mot de $L = L_G(i)$, où i est dans V. On montre qu'il est reconnu par pile vide par cet automate, par récurrence sur la longueur n des dérivations donnant naissance à w.

- si $n = 1$, soit w est vide et $(i\to\varepsilon)$ est dans P. La pile aura la configuration $\begin{array}{|c||c|c|c|c||c|}\hline i,1 & e & z & \ldots & z & 1 \\ \hline\end{array}$ puis se videra,

  soit w est différent de ε et $(i\to w)$ est dans P. Après la lecture de w la pile aura la configuration $\begin{array}{|c||c|c|c|c||c|}\hline i,j & w & z & \ldots & z & |w|+1 \\ \hline\end{array}$ puis se videra.
- par hypothèse de récurrence, on suppose que, pour tout i de V, tout mot de L_G (i) de longueur de dérivation inférieure ou égale à n est reconnu par pile vide par l'automate à pile ainsi défini.
- soit w un mot de $L_G(i)$ engendré par les dérivations $i \to u_1\ i_1\ u_2\ i_2 \ldots i_q\ u_{q+1} \xrightarrow{n} w$

  Comme dans le cas classique, il existe $v_1, \ldots, v_q$ de Σ^* tels que

  $$i_j \xrightarrow{n_j} v_j,\ 1\leq j\leq q,\ n_j\leq n$$

  $$w = u_1\ v_1\ \ldots\ v_q\ u_{q+1}$$

  Par hypothèse de récurrence, les v_j sont reconnus par pile vide par l'automate construit précédemment, pour lequel l'axiome est i_j. La pile aura la configuration $\begin{array}{|c||c|c|c|c|c||c|}\hline i,j & u_1 & i_1 & u_2 & \ldots & z & |w|+1 \\ \hline\end{array}$ puis se videra.

b) Réciproquement ; soit w un mot reconnu par pile vide par l'automate construit ci-dessus, le premier tiroir ouvert étant $t(i,j)$; montrons qu'alors w est dans $L_G(i)$. On fait une récurrence sur le nombre de tiroirs ouverts pour reconnaître le mot (nombre qui représente la hauteur maximale de la pile).

- si $n = 1$, alors (i,w) se trouve dans $t(i,j)$, donc $(i \to w)$ est une règle de P et w est dans $L_G(i)$.
- soit n le nombre de tiroirs ouverts pour reconnaître w et $t(i,j)$ le premier. Supposons que $\text{type}(j) = x^{n_1} i_1 \ldots$; alors w s'écrit $w = u_1\ v_1\ u_2\ v_2 \ldots$ avec $(i, u_1\ i_1\ u_2\ i_2 \ldots) \in t(i,j)$, v_k reconnu par pile vide, à partir de l'ouverture du tiroir $t(i_k, j_k)$ donc après l'ouverture de moins de n tiroirs.
  Par hypothèse de récurrence $v_k \in L_G(i_k)$ et finalement $i \to u_1 i_1 u_2 i_2 \ldots \xrightarrow{*} u_1 v_1 u_2 v_2 \ldots = w$
  Donc w est dans $L_G(i)$.

Proposition 8 Un automate à pile et à tiroirs reconnaît par pile vide un langage Σ-algébrique.

Preuve Soient N, p et $t(i,j)$ les caractéristiques de l'automate. On construit la Σ-grammaire $G = \langle \Sigma, V, P \rangle$ de la façon suivante :

$V = \{1, \ldots, N\}$, l'ensemble des variables

$P = \{(i \to m)\ /\ i \in V,\ \exists\ j \in N,\ (i,m) \in t(i,j)\}$ l'ensemble des règles.

Cette p-grammaire est Σ-algébrique puisque les variables sont en nombre fini et que les membres droits des règles sont de longueur bornée par p. D'après la proposition 7, l'automate reconnaissant les mots engendrés par la grammaire qui vient d'être construite est exactement l'automate initial.

Proposition 9 Un automate à pile et tiroirs reconnaît, par non blocage de la pile, les langages Σ-algébriques.

Preuve Etant donné un automate à pile et à tiroirs, on peut construire une Σ-grammaire $G = \langle \Sigma, V, P \rangle$ comme il a été indiqué dans la proposition 8. Puis on forme une nouvelle grammaire $G' = \langle \Sigma, V', P' \rangle$ où $V' = V \cup \overline{V}$, avec $\overline{V} = \{\overline{i}\ /\ i \in V\}$, et $P' = P \cup \overline{P}$ avec $\overline{P} = \{\overline{i}\ u_1 i_1 \ldots i_p\ u_{p+1}\ /\ i \in V,\ (i \to u_1\ i_1 \ldots u_{p+1}\ i_{p+1} \ldots i_n\ u_{n+1}) \in P$ et $\overline{u}_{p+1}$ facteur gauche de $u_{p+1}\}$ $\cup$
$\{\overline{i} \to u_1\ i_1 \ldots \overline{i}_p / i \in V,\ (i \to u_1\ i_1 \ldots i_p\ u_{p+1} \ldots\ i_n\ u_{n+1}) \in P\}$

Alors l'automate à pile et à tiroirs reconnaît par non blocage de la pile le langage L_G, (T). La démonstration est laissée au lecteur.

Proposition 10 Soit X un alphabet fini. Un langage L de X^* est algébrique si et seulement si il est reconnu par un automate à pile et a tiroirs.
Preuve évident laissée au lecteur.

VII - AUTOMATE A PILE ET A TIROIRS AVEC ETATS

Remarquons que, dans la définition de la fonction de transition des automates à pile et à tiroirs, le choix suivant a été fait : l'action à effectuer en présence d'une variable T dans la pile P dépend de :

- cette variable T
- des tiroirs $t(T,j')$
- de la lettre lue dans le mot à reconnaître

S,j	u	T	v	z	n+1
TIR		P			C

On peut alors se demander quel résultat on obteindrait en faisant, de plus, dépendre cette action du facteur gauche u qui vient d'être reconnu.

C'est cette nouvelle définition de la fonction de transition qui a permis l'introduction de la notion d'automate à pile, tiroirs et états. Mais ces automates reconnaissant des langages qui ne sont pas forcément Σ-algébriques, il a été nécessaire de poser des conditions supplémentaires pour qu'ils ne reconnaissent que des langages Σ-algébriques.

On dira qu'une boîte est un ensemble de facteurs gauches de la pile P

$$\begin{array}{l} \bar{u} = u_1\ v_1\ u_2\ v_2 \dots\ u_n \\ \bar{u}' = u'_1\ v'_1\ u'_2\ v'_2\ \dots\ u'_n \end{array} \quad \text{où} \quad \begin{cases} u_i, u'_i, \dots, \in \Sigma^+ \\ v_i, v'_i, \dots, \in V \end{cases}$$

tels que, pour toute lettre lue, a_i ou ε, les piles

i,j	u	v	u_2	$v_2 \dots$	1	...	c
i',j'	u'	v'	u'_2	$v'_2 \dots$	1	...	c'

aient le même comportement.

Proposition 11 Un automate à pile, tiroirs et états ayant un nombre fini de boîtes reconnaît par pile vide et état quelconque les langages Σ-algébriques.

VIII - AUTOMATES DETERMINISTES

Par analogie avec le cas classique, on définit des automates à pile et à tiroirs avec ou sans état, déterministes et l'on obtient les familles de langages suivantes :

les langages reconnus par des automates à pile et à tiroirs déterministes
- sans état, par pile vide, soit $D\mathcal{L}_1$
- sans états, par non blocage de la pile, soit $D'\mathcal{L}_1$
- avec états et boîtes finies par pile vide, soit, $D\mathcal{L}_2$
- avec états et boîtes finies par non blocage de la pile, soit $D'\mathcal{L}_2$.

Donnons sans démonstration les propriétés vérifiées par ces familles de langages.

Proposition 12 Les familles de langages déterministes $D\mathcal{L}_1$, $D'\mathcal{L}_1$, $D\mathcal{L}_2$, $D'\mathcal{L}_2$ vérifient les inclusions suivantes :

$D\mathcal{L}_1 \subset D\mathcal{L}_2 \subset D'\mathcal{L}_2$ et $D'\mathcal{L}_1 \subset D'\mathcal{L}_2$.

<u>Proposition 13</u> Soit $L \subset X^*$ un langage déterministe au sens de $D'\mathcal{L}_2$ (X étant fini) alors L est déterministe au sens classique.

<u>Proposition 14</u> Soit $L \subset X^*$ (où X est fini) un langage reconnu par un automate à pile classique déterministe, alors il est déterministe au sens de $D'\mathcal{L}_2$.

<u>Proposition 15</u> Soit L_1 un langage reconnu par état final par un automate à pile, tiroirs et états déterministe, c'est à dire un langage de $D'\mathcal{L}_2$. Soit L_2 un langage reconnaissable de Σ^*. Alors $L_1 \cap L_2$ est dans $D'\mathcal{L}_2$.

<u>Proposition 16</u> Un langage reconnu par pile vide par un automate à pile et tiroirs avec ou sans état déterministe est préfixe. Autrement dit, les langages de $D\mathcal{L}_1$ et de $D\mathcal{L}_2$ sont préfixes.

<u>Proposition 17</u> Les langages reconnus par des automates déterministes à boîtes finies ne sont pas fermés par :

- concaténation
- étoile
- union.

<u>Proposition 18</u> Les langages déterministes ne sont pas fermés par morphisme strictement alphabétique.

<u>Proposition 19</u> Soit ϕ un morphisme strictement alphabétique de Σ dans Σ'. Soit L' un langage déterministe de Σ'. Alors $\phi^{-1}(L')$ est déterministe.

IX - CONCLUSION

Dans cet article, certains types d'automates à pile classiquement utilisés n'ont pas été abordés. C'est le cas, notamment, des automates quasi-réaltime, realtime et finite turn. Leurs définitions s'étendent sans problème au cas des automates à pile, tiroirs et états. Il reste à étudier si les propriétés vérifiées dans le cas classique le sont toujours pour de telles extensions aux alphabets infinis.

Le problème des langages à compteur est plus complexe. En effet, par analogie avec le cas des automates à pile à un compteur, qui n'ont qu'un symbole de pile, on peut définir des automates à pile et à deux tiroirs (un seul tiroir engendrerait des langages finis sur tout alphabet fini donc peu intéressants). Ils reconnaissent des langages engendrés par des règles de deux types : $S \rightarrow v$ et $S \rightarrow u_1\ S\ u_2\ S\ u_3 \dots$ Ces langages constituent une famille qu'il est intéressant d'étudier mais qui, cependant rapportée au cas d'un alphabet fini, n'a pas grand chose à voir avec celle des langages à compteur.

On peut par ailleurs se pencher sur les problèmes de décidabilité soulevés dans le cas classique.

Certains sont résolus très facilement. Tout d'abord, étant donnée une p-grammaire -algébrique et un mot $w = x_1 \dots x_n$ de Σ^*, il existe un algorithme déterminant si w se trouve dans L(G) puisque l'on peut se ramener au cas de l'alphabet fini

$X = \{x_1, \ldots, x_n\}$ en prenant la trace de la grammaire G sur X.

Ensuite, il est possible de dire si un langage engendré par $G = \langle\Sigma, V, P\rangle$ est vide : en effet, comme on suppose toujours que les règles de la grammaire sont constituées d'ensembles récursivement énumérables, il est possible de former les ensembles :

$W_1 = \{T \in V \ / \exists\ w \in \Sigma^* \ , \ (T \to w) \in P\}$

$W_{k+1} = W_k \cup \{T \in V \ / \ \exists \alpha \in (\Sigma \cup W_k)^+, \ (T \to \alpha) \in P\}$, (cf. [6]), et le langage est non vide si l'axiome S appartient à W_n (où $n = |V|$).

Enfin, on peut dire si un langage Σ-algébrique est infini ou non. En effet, après avoir réduit la grammaire, deux cas se présentent : ou bien la grammaire est alors réduite à une grammaire sur un alphabet fini et il existe un algorithme décidant si le langage est infini ou non.

ou bien une des règles au moins utilise une infinité de lettres. On est alors certain d'obtenir un langage infini.

Dans ce domaine il reste à étudier la décidabilité de l'égalité de deux langages Σ-algébriques, de leur inclusion.

Maintenant, des problèmes, qui ne sont plus calqués sur ceux de la théorie classique, surgissent. Nous en donnons deux exemples :

- Dans [2], les auteurs ont appelé I-cône la famille des langages de Σ^*fermée par morphismes alphabétiques direct et inverse et par intersection N-rationnelle. Ils ont appelé ombre d'un langage L de la famille des langages images de L dans un morphisme de type fini. Puis ils ont montré que tout cône rationnel fermé par union de langages algébriques est l'ombre d'un I-cône principal admettant comme générateur un langage Σ-algébrique. et ils ont qualifié d'honnête un processus fini de génération de mots dont la Σ-extension vérifie une proposition analogue. On peut alors se demander quelles familles de langages sont engendrées par des processus honnêtes.

- Le deuxième exemple porte sur l'ambiguité. Les notions de grammaires et de langages ambigus s'étendent facilement au cas d'un alphabet infini et la vérification des propriétés classiques ne présente pas de difficultés. Par contre se posent des problèmes d'existence, aux réponses non évidentes ; par exemple :

Conjoncture Il existe un langage Σ-algébrique inhéremment ambigu et non ambigu sur tout partie finie.

BIBLIOGRAPHIE

[1] J.M. AUTEBERT, J. BEAUQUIER, L. BOASSON, "Langages sur des alphabets infinis" Discrete Math.2, 1980, p1-20.

[2] J.M. AUTEBERT, J. BEAUQUIER, L. BOASSON, "Formes de langages et de grammaires", Acta Informatica 7, 1982, p 193-219.

[3] J.M. AUTEBERT, J. BEAUQUIER, L. BOASSON, "Limites de langages algébriques", C.R. Acad. Sc. PARIS, t.29, 1980.

[4] J. BEAUQUIER, "Comptabilité des systèmes de sécurité", In program Transformation B. ROBINET Dunod-1978, p 109-124.

[5] J. BEAUQUIER, M. NIVAT, "Application of formal languages theory to problems of security and synchronization", In Formal language theory, 1980, Academic Press.

[6] M. HARRISON, "Introduction to formal languages theory", Addison Wesley, 1978.

[7] R. McNAUGHTON, "Testing and generating infinite sequences by a finite automaton" Inf. and Control. Vol 9, 1966, p 521-530.

[8] M. NIVAT, "Sur les ensembles de mots infinis engendrés par une grammaire algébrique" RAIRO Info. Théorique, 12, 1978, p 259 270.