Revue française d'informatique et de recherche opérationnelle. Série rouge

Claude Boucher

Fermeture de certaines classes de langages formels sous des permutations linguistiques

Revue française d'informatique et de recherche opérationnelle. Série rouge, tome 3, nº R2 (1969), p. 51-61

<http://www.numdam.org/item?id=M2AN_1969__3_2_51_0>

R.I.R.O.
(3e année, N° R-2/1969, p. 51-62)

FERMETURE DE CERTAINES CLASSES DE LANGAGES FORMELS SOUS DES PERMUTATIONS LINGUISTIQUES

par Claude BOUCHER (1)

Résumé. — *On démontre que la classe des langages de Kleene, des langages de Chomsky ou des langages dépendants du contexte sont chacune fermées sous la permutation des symboles d'un sous-alphabet de l'alphabet dont ils sont issus.*

On donnera le nom d'*alphabet* à un ensemble fini non vide $\Sigma = \{ \sigma_1, ..., \sigma_n \}$ dont les éléments seront appelés *symboles.* Σ^* désignera l'ensemble de toutes les suites finies de symboles de Σ, y compris la suite formée d'aucun symbole de Σ que l'on appellera *mot vide* et que l'on désignera par λ. Les éléments de Σ^* seront appelés des *mots.* Un sous-ensemble L quelconque de Σ^* recevra le nom de *langage.*

On peut définir une loi de composition qui, à deux langages L_1 et L_2, fait correspondre un nouveau langage désigné par $L_1 \cdot L_2$. Ce langage est formé de tous les mots que l'on obtient en écrivant à la suite les symboles d'un mot quelconque de L_1, puis ceux d'un mot quelconque de L_2. On appellera *concaténation* cette loi de composition. Dans le cas où l'un des langages ne comporterait qu'un seul mot x, on écrirait

$$L_1 \cdot x \quad \text{ou} \quad x \cdot L_2 \qquad \text{plutôt que} \qquad L_1 \cdot \{ x \} \quad \text{ou} \quad \{ x \} \cdot L_2.$$

Un *automate fini* $\mathcal{A}$ sera un 5-tuplet $\langle K, \Sigma, \delta, s_1, F \rangle$ où

K est un ensemble fini, l'*ensemble des états* de $\mathcal{A}$;

Σ un ensemble fini, l'*alphabet* de $\mathcal{A}$;

δ une application de $K \times \Sigma$ dans $\mathfrak{T}(K)$, la *fonction de transition* de $\mathcal{A}$;

s_1 un élément distingué de K, l'*état initial* de $\mathcal{A}$;

F un sous-ensemble de K, l'*ensemble des états terminaux* de $\mathcal{A}$.

(1) Département de mathématiques, Université de Sherbrooke, Sherbrooke, Canada.

Un mot quelconque du langage $K \cdot \Sigma^*$ sera appelé une *configuration* de l'automate $\mathcal{A}$. Soient α et β deux configurations de l'automate fini $\mathcal{A}$. On dira que α *induit* β *d'une manière immédiate* ssi

$$\alpha = s\sigma x$$
$$\beta = s'x$$
$$\text{et} \qquad s' \in \delta(s, \sigma)$$

avec $\sigma \in \Sigma$, $x \in \Sigma^*$, $s \in K$ et $s' \in K$. On écrit alors $\alpha \vdash^{*}_{\mathcal{A}} \beta$.

Soient encore α et β deux configurations de l'automate fini $\mathcal{A}$. On dira que α *induit* β ssi

1° $\alpha = \beta$

ou bien

2° il existe une suite $\alpha_1, ..., \alpha_p$ de configurations de $\mathcal{A}$ telles que

(i) $\alpha_1 = \alpha$,

(ii) $\alpha_p = \beta$

et (iii) $\alpha_i \vdash^{*}_{\mathcal{A}} \alpha_{i+1}$ pour $i = 1, ..., p - 1$.

On écrit alors $\alpha \vdash_{\mathcal{A}} \beta$.

Soit $x \in \Sigma^*$. On dira que x est *accepté* par l'automate fini $\mathcal{A}$ ssi il existe un état $s' \in F$ tel que $s_1 x \vdash_{\mathcal{A}} s'$. On désignera par $T(\mathcal{A})$ l'ensemble des mots acceptés par $\mathcal{A}$.

Soit $L \subseteq \Sigma^*$. On dira que L est un *langage de Kleene* ssi il existe un automate fini $\mathcal{A}$ tel que $T(\mathcal{A}) = L$.

Un *automate à mémoire empilée* $\mathcal{A}$ est un 7-tuplet $\langle K, \Sigma, \Gamma, \delta, s_1, Z_1, F \rangle$, où K, Σ, s_1 et F sont interprétés comme dans le cas précédent, mais Σ est appelé *alphabet d'entrée de* $\mathcal{A}$;

où Γ est un ensemble fini non vide, appelé *alphabet auxiliaire* de $\mathcal{A}$;

Z_1 est un élément distingué de Γ, appelé *symbole auxiliaire initial de* $\mathcal{A}$;

δ est une application de $K \times (\Sigma \cup \{ \lambda \}) \times \Gamma$ dans $\mathcal{PF}(K \times \Gamma^*)$, l'ensemble des parties finies de $K \times \Gamma^*$, encore appelée *fonction de transition de* $\mathcal{A}$.

Une *configuration* d'un tel automate sera un mot de $\Gamma^* \cdot K \cdot \Sigma^*$. Soient α et β deux configurations de l'automate à mémoire empilée $\mathcal{A}$. On dira que α *induit* β *d'une manière immédiate* ssi

ou bien

1°

$$\alpha = y\tau s\sigma x$$
$$\beta = yws'x$$
$$\text{et} \qquad \langle s', w \rangle \in \delta(s, \sigma, \tau)$$

pour $y \in \Gamma^*$, $w \in \Gamma^*$, $x \in \Sigma^*$, $\sigma \in \Sigma$ et $\tau \in \Gamma$;

ou bien

2°

$$\alpha = y\tau sx$$
$$\beta = yws'x$$

et
$$\langle s', w \rangle \in \delta(s, \lambda, \tau).$$

On écrit encore $\alpha \mid\frac{*}{\mathcal{A}} \beta$.

Le fait qu'une configuration α *induit* une configuration β est défini comme précédemment.

Soit $x \in \Sigma^*$. On dira que x est *accepté* par un automate à mémoire empilée $\mathcal{A}$ ssi il existe un mot $y \in \Gamma^*$ et un état $s' \in F$ tels que $Z_1 s_1 x \mid\frac{}{\mathcal{A}} ys'$. $T(\mathcal{A})$ désignera encore l'ensemble des mots acceptés par l'automate à mémoire empilée $\mathcal{A}$.

Soit $L \subseteq \Sigma^*$. On dira que L est un *langage de Chomsky* ssi il existe un automate à mémoire empilée $\mathcal{A}$ dont l'alphabet d'entrée est Σ et qui est tel que $T(\mathcal{A}) = L$.

Un *automate linéaire borné avec doubles marquants* $\mathcal{A}$ est un 5-tuplet $\langle K, \Sigma \cup \{ \# \}, \delta, s_1, F \rangle$, où K, Σ, s_1, F sont interprétés comme dans le cas des automates finis,

où $\#$ est un symbole qui n'apparaît pas dans Σ, appelé *marquant*,

où δ est une application de $K \times (\Sigma \cup \{ \# \})$ dans

$$\mathfrak{P}(K \times (\Sigma \cup \{ \# \}) \times \{ -1, 0, 1 \})$$

telle que l'une ou l'autre des expressions $\langle s', \sigma, p \rangle \in \delta(s, \#)$ ou $\langle s', \#, p \rangle \in \delta(s, \sigma)$ entraîne $\sigma = \#$. On l'appelle encore *fonction de transition.*

Une configuration d'un tel automate est un mot pris dans

$$(K \cdot \# \cdot \Sigma^* \cdot \#) \cup (\# \cdot \Sigma^* \cdot K \cdot \Sigma^* \cdot \#) \cup (\# \cdot \Sigma^* \cdot \# \cdot K).$$

Soient α et β deux configurations d'un tel automate. On dira que α *induit* β *d'une manière immédiate* ssi :

ou bien 1°
$$\alpha = y\tau s\sigma x$$
$$\beta = ys'\tau\sigma' x$$

et
$$\langle s', \sigma', -1 \rangle \in \delta(s, \sigma);$$

ou bien 2°
$$\alpha = ys\sigma x$$
$$\beta = ys'\sigma' x$$

et
$$\langle s', \sigma', 0 \rangle \in \delta(s, \sigma);$$

ou bien 3°
$$\alpha = ys\sigma x$$
$$\beta = y\sigma' s' x$$

et
$$\langle s', \sigma', 1 \rangle \in \delta(s, \sigma)$$

avec $x, y \in (\Sigma \cup \{ \# \})^*$ et $\sigma, \sigma', \tau \in \Sigma \cup \{ \# \}$. On écrit encore $\alpha \overset{*}{\vdash}_{\mathcal{A}} \beta$ et l'idée que α induit β se définit et se dénote comme précédemment.

Soit $x \in \Sigma^*$. On dira que x est *accepté* par l'automate linéaire borné avec doubles marquants $\mathcal{A}$ ssi il existe un $y \in \Sigma^*$ et un $s' \in F$ tels que

$$s_1 \# x \# \vdash_{\mathcal{A}} \# y \# s'.$$

$T(\mathcal{A})$ désigne encore l'ensemble des mots acceptés par $\mathcal{A}$.

Soit $L \subseteq \Sigma^*$. On dira que L est un *langage dépendant du contexte* ssi il existe un automate linéaire borné avec doubles marquants $\mathcal{A}$ tel que $T(\mathcal{A}) = L$.

Soient $\Sigma = \{ \sigma_1, ..., \sigma_n \}$ un alphabet, $\Sigma' = \{ \sigma_{j_1}, ..., \sigma_{j_p} \}$ un sous-ensemble de Σ, φ une permutation de Σ' et $L \subseteq \Sigma^*$. On appellera *φ-permutation* de L le langage L' [noté $P_\varphi(L)$] que l'on peut former à partir de L de la manière suivante :

1° si x est un mot de L qui ne contient pas au moins une occurrence de chaque symbole de Σ', x sera inclus dans L';

2° si x est un mot de L qui contient au moins une occurrence de chaque symbole de Σ', on inclura dans L' tous les mots formés à partir de x en y remplaçant une et une seule occurrence de chaque symbole σ_{j_i} de Σ' par son image $\varphi(\sigma_{j_i})$.

Par exemple, supposons que l'on ait $\Sigma = \{ a, b, c, d \}$, $\Sigma' = \{ a, b, c \}$ et que φ soit la permutation $\begin{pmatrix} a & b & c \\ c & a & b \end{pmatrix}$. Si le langage L contient un mot $x = adbaad$, on l'inclura dans $P_\varphi(L)$, puisque ce mot ne contient pas une occurrence de tous les symboles de Σ'. Par contre, si L contient un mot comme $x = a\,b\,a\,c\,b\,d\,a\,c$, on adjoindra à $P_\varphi(L)$ des mots comme $c\,a\,a\,b\,b\,d\,a\,c$, $c\,a\,a\,c\,b\,d\,a\,b$, $c\,b\,a\,b\,a\,d\,a\,c$, $c\,b\,a\,c\,a\,d\,a\,b$, $a\,a\,c\,b\,b\,d\,a\,c$, $a\,a\,c\,c\,b\,d\,a\,b$, etc.

Le mot $c\,a\,a\,b\,b\,d\,a\,c$ est obtenu à partir de x en remplaçant la 1^re^ occurrence de a, la 1^re^ occurrence de b et la 1^re^ occurrence de c par c, a et b respectivement. Le mot $c\,a\,a\,c\,b\,d\,a\,b$ est obtenu à partir de x en remplaçant la 1^re^ occurrence de a, la 1^re^ occurrence de b et la 2^e^ occurrence de c par c, a et b respectivement. Le mot $c\,b\,a\,b\,a\,d\,a\,c$ est obtenu en remplaçant la 1^re^ occurrence de a, la 2^e^ occurrence de b et la 1^re^ occurrence de c par c, a et b respectivement. Et l'on pourrait continuer ainsi. Comme on avait 3 occurrences de a, 2 de b et 2 de c, cela donne $3 \times 2 \times 2 = 12$ possibilités de substitution.

En général, si $o(\sigma_{j_i})$ indique le nombre d'occurrences du symbole σ_{j_i} dans le mot x, le nombre total de substitutions possibles sera

$$\prod_{i=1}^{p} o(\sigma_{j_i}),$$

où $p = \text{card}(\Sigma')$, pourvu que $o(\sigma_{j_i}) \neq 0$ pour $1 \leqslant i \leqslant p$.

Nous voulons démontrer que si L est un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte), il en est de même de $P_\varphi(L)$, pour toute permutation φ d'un sous-alphabet de l'alphabet dont L est issu.

Soient φ une permutation de $\Sigma' = \{ a, b, c \}$, sous-alphabet de $\Sigma = \{ a, b, c, d \}$ et $L \subseteq \Sigma^*$. On pourrait partager le langage $P_\varphi(L)$ en un certain nombre de sous-ensembles disjoints dont certains sont susceptibles d'être vides. D'une part, on aurait le sous-ensemble E_L^* de tous les mots de L qui ne contiennent pas au moins une occurrence de chaque symbole de Σ'. D'autre part, on aurait les sous-ensembles

$$E_L^\varphi(a, b, c),\ E_L^\varphi(a, c, b),\ E_L^\varphi(b, a, c),\ E_L^\varphi(b, c, a),\ E_L^\varphi(c, a, b) \text{ et } E_L^\varphi(c, b, a),$$

où, par exemple, $E_L^\varphi(a, b, c)$ contient tous les mots x' de $P_\varphi(L)$ obtenus à partir d'un mot x de L, tels que dans x l'occurrence de a remplacée par $\varphi(a)$ est située à gauche de l'occurrence de b remplacée par $\varphi(b)$, qui est elle-même située à gauche de l'occurrence de c remplacée par $\varphi(c)$, où $E_L^\varphi(a, c, b)$ contient les mots x' de $P_\varphi(L)$ obtenus par une substitution dans un mot x, où l'occurrence de a remplacée par $\varphi(a)$ est à gauche de celle de c remplacée par $\varphi(c)$, qui est elle-même à gauche de celle de b remplacée par $\varphi(b)$, etc. On peut donc écrire

$$P_\varphi(L) = E_L^* \cup \left(\bigcup_{\psi \in S} E_L^\varphi(\psi(a), \psi(b), \psi(c)) \right),$$

où S représente l'ensemble de toutes les permutations de $\{ a, b, c \}$.

Plus généralement, on peut énoncer le théorème suivant dont la véracité devrait paraître évidente à la lumière des remarques qui précèdent.

Théorème 1. Soient $\Sigma = \{ \sigma_1, \ldots, \sigma_n \}$ un alphabet, $\Sigma' = \{ \sigma_{j_1}, \ldots, \sigma_{j_p} \}$ un sous-alphabet de Σ, φ une permutation de Σ' et $L \subseteq \Sigma^*$.

Alors, on peut écrire

$$P_\varphi(L) = E_L^* \cup \left(\bigcup_{\psi \in S} E_L^\varphi(\psi(\sigma_{j_1}), \ldots, \psi(\sigma_{j_p})) \right),$$

où S représente l'ensemble des permutations de Σ' et où les ensembles

$$E_L^* \text{ et } E_L^\varphi(\psi(\sigma_{j_1}), \ldots, \psi(\sigma_{j_p}))$$

sont interprétés comme dans les remarques qui précèdent l'énoncé du théorème.

Nous allons maintenant tenter de démontrer que les ensembles

$$E_L^* \text{ et } E_L^\varphi(\psi(\sigma_{j_1}), \ldots, \psi(\sigma_{j_p}))$$

sont de même nature que le langage L dont on est parti.

Théorème 2. Si L est un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte), alors l'ensemble E_L^* est également un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte).

Démonstration

Si E_L^* était un ensemble vide, le théorème irait de soi, car, alors, il suffirait de prendre un automate fini, un automate à mémoire empilée ou un automate linéaire borné avec doubles marquants, selon le cas, pour lequel on aurait $F = \varnothing$. On peut donc supposer que $E_L^* \neq \varnothing$.

Si L est un langage de Kleene, il doit exister un automate fini

$$\mathcal{A} = \langle K, \Sigma, \delta, s_1, F \rangle$$

tel que $T(\mathcal{A}) = L$. Nous allons maintenant construire un automate fini

$$\mathcal{B} = \langle K', \Sigma, \delta', s_1, F' \rangle$$

tel que $T(\mathcal{B}) = E_L^*$.

Pour tout $s_m \in K$, formons des états $s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}$, où $1 \leqslant i \leqslant p - 1$ et où $\sigma_{j_1}, \ldots, \sigma_{j_i} \in \Sigma'$. Par exemple, si $\Sigma' = \{ a, b, c \}$, pour tout $s_m \in K$, on formera les états $s_m^a, s_m^b, s_m^c, s_m^{a,b}, s_m^{a,c}, s_m^{b,c}, s_m^{b,a}, s_m^{c,a}, s_m^{c,b}$. On considérera K' comme la réunion de K, de l'ensemble des états $s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}$ ainsi formés et de $\{ s^* \}$, où s^* est un état différent de tous ceux qui précèdent. On prendra pour F' la réunion de F et de l'ensemble des états $s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}$ pour lesquels $s_m \in F$. L'état initial de $\mathcal{B}$ et son alphabet seront les mêmes que ceux de $\mathcal{A}$. Enfin, la fonction de transition δ' sera définie comme suit :

si $s_n \in \delta(s_m, \sigma)$, on posera

1° $s_n^\sigma \in \delta'(s_m, \sigma)$ pour $\sigma \in \Sigma'$;

2° $s_n \in \delta'(s_m, \sigma)$ pour $\sigma \in \Sigma - \Sigma'$;

3° $s_n^{\sigma_{j_1}, \ldots, \sigma_{j_i}, \sigma} \in \delta'(s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma)$

pour

$$\sigma \in \Sigma' - \{ \sigma_{j_1}, \ldots, \sigma_{j_i} \} \qquad \text{et} \qquad \{ \sigma_{j_1}, \ldots, \sigma_{j_i}, \sigma \} \neq \Sigma' ;$$

4° $\delta'(s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma) = \{ s^* \}$

pour

$$\sigma \in \Sigma' - \{ \sigma_{j_1}, \ldots, \sigma_{j_i} \} \qquad \text{et} \qquad \{ \sigma_{j_1}, \ldots, \sigma_{j_i}, \sigma \} = \Sigma' ,$$

5° $s_n^{\sigma_{j_1}, \ldots, \sigma_{j_i}} \in \delta'(s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma)$

pour

$$\sigma \in \{ \sigma_{j_1}, \ldots, \sigma_{j_i} \} \cup (\Sigma - \Sigma') ;$$

enfin, on écrira $\delta'(s^*, \sigma) = \{ s^* \}$ pour tout $\sigma \in \Sigma$.

Dans le cas où L serait un langage de Chomsky, on pourra trouver un automate à mémoire empilée $\mathcal{A} = \langle K, \Sigma, \Gamma, \delta, s_1, Z_1, F \rangle$ tel que $T(\mathcal{A}) = L$. On construira un automate à mémoire empilée $\mathcal{B} = \langle K', \Sigma, \Gamma, \delta', s_1, Z_1, F' \rangle$ tel que $T(\mathcal{B}) = E_L^*$ d'une manière assez semblable à la précédente. En particulier, les ensembles K' et F' seront définis comme précédemment. La définition de la fonction δ' sera cependant légèrement différente.

Dans le cas où $\langle s_n, w \rangle \in \delta(s_m, \sigma, \tau)$, on écrira

1° $\langle s_n^\sigma, w \rangle \in \delta'(s_m, \sigma, \tau)$ pour $\sigma \in \Sigma'$;

2° $\langle s_n, w \rangle \in \delta'(s_m, \sigma, \tau)$ pour $\sigma \in \Sigma \cup \{ \lambda \} - \Sigma'$;

3° $\langle s_n^{\sigma_{j_1}, \ldots, \sigma_{j_i}, \sigma}, w \rangle \in \delta'(s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma, \tau)$

pour

$$\sigma \in \Sigma' - \{ \sigma_{j_1}, \ldots, \sigma_{j_i} \} \qquad \text{et} \qquad \{ \sigma_{j_1}, \ldots, \sigma_{j_i}, \sigma \} \neq \Sigma' ;$$

4° $\delta'(s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma, \tau) = \{ \langle s^*, w \rangle \}$

pour

$$\sigma \in \Sigma' - \{ \sigma_{j_1}, \ldots, \sigma_{j_i} \} \qquad \text{et} \qquad \{ \sigma_{j_1}, \ldots, \sigma_{j_i}, \sigma \} = \Sigma' ;$$

5° $\langle s_n^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, w \rangle \in \delta'(s_m^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma, \tau)$

pour

$$\sigma \in \{ \sigma_{j_1}, \ldots, \sigma_{j_i} \} \cup (\Sigma \cup \{ \lambda \} - \Sigma') ;$$

enfin, on écrira $\delta'(s^*, \sigma, \tau) = \{ \langle s^*, \lambda \rangle \}$ pour tout $\sigma \in \Sigma \cup \{ \lambda \}$ et tout $\tau \in \Gamma$.

En troisième lieu, si L est un langage dépendant du contexte, il existe alors un automate linéaire borné avec doubles marquants

$$\mathcal{A} = \langle K, \Sigma \cup \{ \# \}, \delta, s_1, F \rangle$$

tel que $T(\mathcal{A}) = L$. On formera un nouvel automate

$$\mathcal{B} = \langle K', \Sigma \cup \{ \# \}, \delta', t_1, F' \rangle$$

du même type, en procédant de la façon suivante. Pour obtenir K', on ajoutera à K des états s^*, t_1, t_2 et t_3, et tous les états de la forme $t_2^{\sigma_{j_1}, \ldots, \sigma_{j_i}}$, où $1 \leqslant i \leqslant p - 1$ et où les $\sigma_{j_1}, \ldots, \sigma_{j_i} \in \Sigma'$. On posera $F' = F$ et on prendra t_1 comme état initial de $\mathcal{B}$. Enfin, la fonction de transition δ' sera définie comme suit :

$$\delta'(t_1, \#) = \{ \langle t_2, \#, 1 \rangle \} ;$$

$$\delta'(t_2, \sigma) = \{ \langle t_2^\sigma, \sigma, 1 \rangle \} \qquad \text{pour} \qquad \sigma \in \Sigma' ;$$

$$\delta'(t_2, \sigma) = \{ \langle t_2, \sigma, 1 \rangle \} \qquad \text{pour} \qquad \sigma \in \Sigma - \Sigma' ;$$

$$\delta'(t_2^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma) = \{ \langle t_2^{\sigma_{j_1}, \ldots, \sigma_{j_i}, \sigma}, \sigma, 1 \rangle \}$$

pour

$$\sigma \in \Sigma' - \{ \sigma_{j1}, ..., \sigma_{j_i} \} \qquad \text{et} \qquad \{ \sigma_{j1}, ..., \sigma_{j_i}, \sigma \} \neq \Sigma' ;$$

$$\delta'(t_2^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma) = \{ \langle s^*, \sigma, 1 \rangle \}$$

pour

$$\sigma \in \Sigma' - \{ \sigma_{j1}, ..., \sigma_{j_i} \} \qquad \text{et} \qquad \{ \sigma_{j1}, ..., \sigma_{j_i}, \sigma \} = \Sigma' ;$$

$$\delta'(t_2^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma) = \{ \langle t_2^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \sigma, 1 \rangle \}$$

pour

$$\sigma \in \{ \sigma_{j1}, ..., \sigma_{j_i} \} \cup (\Sigma - \Sigma') ;$$

$$\delta'(t_2, \#) = \{ \langle t_3, \#, -1 \rangle \};$$

$$\delta'(t_2^{\sigma_{j_1}, \ldots, \sigma_{j_i}}, \#) = \{ \langle t_3, \#, -1 \rangle \};$$

$$\delta'(t_3, \sigma) = \{ \langle t_3, \sigma, -1 \rangle \} \qquad \text{pour } \sigma \in \Sigma;$$

$$\delta'(t_3, \#) = \{ \langle s_1, \#, 0 \rangle \};$$

$$\delta'(s^*, \sigma) = \{ \langle s^*, \sigma, 1 \rangle \} \qquad \text{pour } \sigma \in \Sigma \cup \{ \# \};$$

$$\delta'(s, \sigma) = \delta(s, \sigma) \qquad \text{pour tout } s \in K;$$

enfin, $\delta'(s, \sigma) = \varnothing$ pour toutes les autres possibilités.

Dans chacun des trois cas qui précèdent, il serait possible de démontrer que l'on a $T(\mathcal{A}) = E_L^*$. En fait, pour les cas des automates finis et des automates à mémoire empilée, la façon dont la fonction de transition δ' du second automate a été définie permet d'imiter le fonctionnement du premier automate tout en enregistrant à mesure qu'il les rencontre les symboles qui proviennent de Σ'. Si le second automate parvient à lire un mot x sans avoir rencontré tous les symboles issus de Σ', il entrera finalement dans un état de la forme s_k ou de la forme s_k^j. Si x peut être accepté par l'automate $\mathcal{A}$, on est susceptible d'avoir $s_k \in F$ et par conséquent $s_k \in F'$ ou $s_k^j \in F'$, d'où l'on tire que x sera accepté par $\mathcal{B}$. Si x ne peut être accepté par $\mathcal{A}$, on doit nécessairement avoir $s_k \notin F$, donc $s_k \notin F'$ et $s_k^j \notin F'$; alors, x ne peut donc pas être accepté par $\mathcal{B}$. Si, en lisant le mot x, le second automate réussit à rencontrer tous les symboles de Σ', il entrera dans l'état s^* dont il ne peut sortir et qui n'est pas terminal. Alors, le mot x ne saurait être accepté par $\mathcal{B}$.

Dans le cas d'un automate linéaire borné, l'automate $\mathcal{B}$ aura pour effet d'examiner le mot x pour voir si on y rencontre tous les symboles de Σ'. S'il n'en est pas ainsi, il reviendra scruter le marquant de gauche et entrera dans l'état s_1 qui était l'état initial du premier automate. Le mot x sera donc, dans cette situation, accepté par l'automate $\mathcal{B}$ si et seulement s'il l'était par l'automate $\mathcal{A}$. Dans le second cas où l'automate $\mathcal{B}$ aurait rencontré dans x tous les symboles de Σ', il serait entré dans l'état s^* dont il ne pourrait sortir. Alors, le mot x ne serait pas accepté par l'automate $\mathcal{B}$.

Rappelons que, si l'on a une permutation φ du sous-alphabet Σ' et que si $\psi(\sigma_{j_1}), ..., \psi(\sigma_{j_p})$ représente une liste des symboles de Σ' dans un ordre donné, $E_L^\varphi(\psi(\sigma_{j_1}), ..., \psi(\sigma_{j_p}))$ désigne l'ensemble des mots x' obtenus à partir des mots x de L, alors que dans x l'occurrence de $\psi(\sigma_{j_1})$ remplacée par $\varphi(\psi(\sigma_{j_1}))$ est située à gauche de l'occurrence de $\psi(\sigma_{j_2})$ remplacée par $\varphi(\psi(\sigma_{j_2}))$, qui est à gauche de l'occurrence de $\psi(\sigma_{j_3})$ remplacée par $\varphi(\psi(\sigma_{j_4}))$, etc. On peut à propos d'un tel ensemble démontrer le théorème suivant.

Théorème 3. — Soient φ une permutation du sous-alphabet Σ' de Σ, $\psi(\sigma_{j_1}), ..., \psi(\sigma_{j_p})$ une liste des symboles de Σ' dans un ordre donné, et L un langage dans Σ, alors, si L est un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte), l'ensemble $E_L^\varphi(\psi(\sigma_{j_1}), ..., \psi(\sigma_{j_p}))$ sera un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte).

Démonstration

Suivant les cas, on doit pouvoir trouver un automate fini

$$\mathcal{A} = \langle K, \Sigma, \delta, s_1, F\rangle,$$

un automate à mémoire empilée $\mathcal{A} = \langle K, \Sigma, \Gamma, \delta, s_1, Z_1, F\rangle$ ou un automate linéaire borné avec doubles marquants $\mathcal{A} = \langle K, \Sigma \cup \{ \# \}, \delta, s_1, F\rangle$ tels que $T(\mathcal{A}) = L$.

Dans le cas où $\mathcal{A}$ serait un automate fini, construisons un automate fini $\mathcal{B} = \langle K', \Sigma, \delta', s_1, F\rangle$ de la manière suivante. Pour tout $s_m \in K$, on formera un état s_m^k pour $1 \leqslant k \leqslant p$. K' sera obtenu par la réunion de K et de l'ensemble des états s_m^k. F' sera l'ensemble des états s_i^p pour lesquels $s_i \in F$. Enfin, δ' sera définie comme suit : si $s_n \in \delta(s_m, \sigma)$, on écrira $s_n \in \delta'(s_m, \sigma)$ et pour $1 \leqslant k \leqslant p$, $s_n^k \in \delta'(s_m^k, \sigma)$; on posera également $s_n^1 \in \delta'(s_m, \varphi(\sigma))$ si $\sigma = \psi(\sigma_{j_1})$ et $s_n^k \in \delta'(s_m^{(k-1)}, \varphi(\sigma))$ si $\sigma = \psi(\sigma_{j_k})$, pour $2 \leqslant k \leqslant p$.

Si $\mathcal{A}$ était un automate à mémoire empilée, la construction d'un automate à mémoire empilée $\mathcal{B} = \langle K', \Sigma, \Gamma, \delta', s_1, Z_1, F\rangle$ se ferait d'une manière très semblable. En particulier, les ensembles K' et F' seront définis de façon identique. La fonction δ' sera soumise aux clauses suivantes : si $\langle s_n, w\rangle \in \delta(s_m, \sigma, \tau)$, on écrira $\langle s_n, w\rangle \in \delta'(s_m, \sigma, \tau)$ et, pour tout $1 \leqslant k \leqslant p$, $\langle s_n^k, w \rangle \in \delta'(s_m^k, \sigma, \tau)$, ainsi que $\langle s_n^1, w \rangle \in \delta'(s_m, \varphi(\sigma), \tau)$ si $\sigma = \psi(\sigma_{j_1})$ et $\langle s_n^k, w \rangle \in \delta'(s_m^{(k-1)}, \varphi(\sigma), \tau)$ si $\sigma = \psi(\sigma_{j_k})$, pour $2 \leqslant k \leqslant p$.

Enfin, si $\mathcal{A}$ est un automate linéaire borné avec doubles marquants, la construction d'un automate $\mathcal{B} = \langle K', \Sigma \cup \{ \# \}, \delta', t_1, F'\rangle$ du même type se fera de la façon suivante. Étant donné une liste particulière $\psi(\sigma_{j_1}), ..., \psi(\sigma_{j_p})$ des symboles de Σ', on construira les états $t_1, t_2, ..., t_{j_p+3}$. K' sera formé par la réunion à K de l'ensemble de ces nouveaux états. On aura $F' = F$ et t_1 représentera l'état initial du nouvel automate. On définira la fonction δ' de la façon suivante :

$$\delta'(t_1, \#) = \{ \langle t_2, \#, 1 \rangle \};$$

$$\delta'(t_2, \#) = \{ \langle t_{j_p+3}, \#, -1 \rangle \};$$

$$\delta'(t_{i+1}, \sigma) = \{ \langle t_{i+1}, \sigma, 1 \rangle \}$$

pour

$$1 \leqslant i \leqslant j_p, \sigma \in \Sigma \qquad \text{et} \qquad \sigma \neq \varphi(\psi(\sigma_{j_i}));$$

$$\delta'(t_{i+1}, \varphi(\psi(\sigma_{j_i}))) = \{ \langle t_{i+1}, \varphi(\psi(\sigma_{j_i})), 1 \rangle, \langle t_{i+2}, \psi(\sigma_{j_i}), 1 \rangle \}$$

pour $1 \leqslant i \leqslant j_p$;

$$\delta'(t_{j_p+2}, \sigma) = \{ \langle t_{j_p+2}, \sigma, 1 \rangle \} \qquad \text{pour} \qquad \sigma \in \Sigma;$$

$$\delta'(t_{j_p+2}, \#) = \{ \langle t_{j_p+3}, \#, -1 \rangle \};$$

$$\delta'(t_{j_p+3}, \sigma) = \{ \langle t_{j_p+3}, \sigma, -1 \rangle \} \qquad \text{pour} \qquad \sigma \in \Sigma;$$

$$\delta'(t_{j_p+3}, \#) = \{ \langle s_1, \#, 0 \rangle \};$$

enfin, $\delta'(s, \sigma) = \varnothing$ pour les cas qui n'ont pas été mentionnés.

Dans chacune des situations étudiées précédemment, il serait possible de montrer que l'on a

$$T(\mathcal{B}) = E_L^{\varphi}(\psi(\sigma_{j_1}), ..., \psi(\sigma_{j_p})).$$

Dans le cas où $\mathcal{A}$ est un automate fini ou un automate à mémoire empilée, l'automate $\mathcal{B}$ correspondant a pour effet d'imiter le comportement de $\mathcal{A}$. Cependant, lorsqu'il rencontre un symbole σ susceptible d'être obtenu à partir d'une permutation φ des symboles de Σ', il peut alors ou bien se comporter comme $\mathcal{A}$, ou bien se comporter à l'égard du symbole σ comme $\mathcal{A}$ se comporterait à l'égard de $\varphi^{-1}(\sigma)$. De plus, dans ce cas, il enregistrera cette variation de comportement, en passant d'un état provenant de K à un état du type s_m^1 ou d'un état du type s_m^j à un état du type s_n^{j+1}. Supposons que x soit un mot de L qui contient au moins une occurrence de tous les symboles de Σ' et x' un mot qui provient de x au moyen d'une φ-permutation du langage L, la substitution se faisant dans l'ordre $\psi(\sigma_{j_1}), ..., \psi(\sigma_{j_p})$. Alors, puisque les seuls états terminaux de $\mathcal{B}$ sont les états de la forme s_n^p pour lesquels s_n est terminal dans $\mathcal{A}$, x' sera accepté par $\mathcal{B}$ dans le cas et seulement dans le cas où x est accepté par $\mathcal{A}$.

Si l'automate $\mathcal{A}$ est linéaire borné, l'automate $\mathcal{B}$ examinera d'abord le mot x' de manière à le ramener à la forme initiale x; il viendra alors scruter le marquant de gauche et entrera dans l'état s_1. A partir de ce point, il imitera scrupuleusement le comportement de $\mathcal{A}$. C'est donc dire que x' sera accepté par l'automate $\mathcal{B}$ si et seulement si x est accepté par $\mathcal{A}$.

Les théorèmes 1, 2 et 3, ainsi que le fait bien connu que la réunion d'une famille finie de langages de Kleene (resp. de langages de Chomsky ou de langages dépendants du contexte), est un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte), nous permettent d'énoncer le théorème suivant.

Théorème 4. — Si L est un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte) d'alphabet Σ, alors, pour toute permutation φ d'un sous-alphabet Σ' de Σ, $P_\varphi(L)$ sera un langage de Kleene (resp. un langage de Chomsky ou un langage dépendant du contexte).

***

Le concept de langage de Kleene fut introduit dans Kleene [1]; ceux de langage de Chomsky et de langage dépendant du contexte, dans Chomsky [1]. Les caractérisations que nous avons utilisées ici se retrouvent dans Rabin et Scott [1], Chomsky [2] et dans les travaux successifs de Myhill [1], Kuroda [1] et Landweber [1]. Les démonstrations du fait que la réunion d'une famille finie de langages de Kleene (resp. de langages de Chomsky ou de langages dépendants du contexte) est un langage du même type, apparaissent dans Rabin et Scott [1] et dans Bar-Hillel, Perles et Shamir [1].

BIBLIOGRAPHIE

[1] Y. Bar-Hillel, M. Perles et E. Shamir, On formal properties of simple phrase structure grammars. *Z. Phonetik, Sprach. Kommunikationsforsch.*, vol. 14, 1961, pp. 143-172.

[1] N. Chomsky, On certain formal properties of grammars. *Information and Control*, vol. 2, 1959, pp. 137-167.

[2] N. Chomsky, Context-free grammars and pushdown storage. *M.I.T. Res. Lab. Electron. Quant. Prog. Rept.* 65, 1962.

[1] S. C. Kleene, Representation of events in nerve nets, dans C. E. Shannon et J. McCarthy (ed.) « Automata Studies », pp. 3-40, Princeton University Press, Princeton, 1956.

[1] S. Y. Kuroda, Classes of languages and linear-bound automata. *Information and Control*, vol. 7, 1964, pp. 207-223.

[1] P. S. Landweber, Three theorems on phrase structure grammars of type 1. *Information and Control*, vol. 6, 1963, pp. 131-136.

[1] J. Myhill, Linear bounded automata. *Wright Air Development Division. Tech. Note* 60-165, 1960.

[1] M. O. Rabin et D. Scott, Finite automata and their decision problems. *IBM J. Res. Develop.*, vol. 3, 1959, pp. 114-125.