Langages formels

11. Automates à pile (déterministes)

— correction —

30 avril 2007

Exercice 1

1. Montrer que le langage de Dyck D_n^* est reconnu par état final par un automate déterministe sans ε-transitions mais par aucun automate déterministe simple.

Correction D+TR : prendre comme alphabet de pile un symbole de fond $\perp$ et pour chaque parenthèse a_i deux symboles a_i et a_i', utiliser a_i' uniquement comme symbole de fond de pile (par dessus $\perp$), et s'en servir pour mettre dans l'état un bit qui indique si la pile est vide.
Pas D+S : simple et état final implique que le langage est clos par préfixe, ce n'est pas le cas de D_n^*.

2. Montrer qu'il est reconnu par sommet de pile par un automate déterministe, simple et sans ε-transition.

Correction On se contente d'empiler et dépiler, et on accepte si le sommet est $\perp$.

Exercice 2 – *Déterministes et non ambigus*

1. Montrer que tout langage déterministe est non ambigu.

Correction On considère un automate à pile déterministe $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, z_0, F\rangle$, acceptant par état final, qui reconnait un langage L. Soit L_q le langage accepté par $\mathcal{A}$ avec q comme unique état final. Le déterminisme de $\mathcal{A}$ entraine que les L_q sont disjoints, il suffit donc de montrer que les L_q pour $q \in F$ sont non-ambigus.
On suppose donc qu'il y a un unique état acceptant $q \in F$. Soit $\mathcal{A}'$ l'automate non déterministe obtenu à partir de $\mathcal{A}$ en ajoutant un nouvel état q', des transitions $(q, z, \varepsilon, q_f, \varepsilon)$ et $(q_f, z, \varepsilon, q_f, \varepsilon)$ pour chaque $z \in \Gamma$. On pose que $\mathcal{A}'$ accepte par état final q_f et par pile vide.
On construit une grammaire à partir de $\mathcal{A}'$ en prenant pour ensemble de variables $V = Q' \times \Gamma \times Q'$ avec $S = (q_0, z_0, q_f)$ et les règles

$$(q, z, q') \to a(q'', z_1, q_1)(q_1, z_2, q_2)\dots(q_{k-1}, z_k, q')$$

pour $(q, z, a, q'', z_1 \dots z_k) \in \delta'$ et $q_1, \dots, q_{k-1} \in Q'$. Par induction sur la longueur des dérivations, on montre que pour chaque $(q, z, q') \in V$ le langage engendré par (q, z, q') est l'ensemble des mots w tels qu'il existe dans $\mathcal{A}'$ une exécution $(q, z) \xrightarrow{w} (q', \varepsilon)$, donc S engendre bien le langage reconnu par $\mathcal{A}$.
Par induction sur la hauteur des dérivations, on montre qu'à toute dérivation $(q, z, q') \to^* w \in \Sigma^*$ correspond une unique exécution $(q, z) \xrightarrow{w} (q', \varepsilon)$. Réciproquement, par induction sur la longueur des exécutions, on montre qu'à chaque exécution $(q, z) \xrightarrow{w} (q, \varepsilon)$ correspond une unique dérivation $(q, z, q') \to^* w$. D'une part, ceci prouve que la grammaire considérée engendre bien le langage reconnu par $\mathcal{A}'$. D'autre part, il est facile de voir que $\mathcal{A}'$ a au plus une exécution acceptante par mot de Σ^*, donc ceci prouve également que chaque mot peut être dérivé d'au plus une façon, ce qui prouve la non-ambiguïté de la grammaire, et donc du langage.

2. Montrer que l'inclusion est stricte en considérant le langage
$L := \{ a^n b^n \mid n > 0 \} \cup \{ a^n b^{2n} \mid n > 0 \}$.

Correction Le langage L est non ambigu car il est union de deux langages clairement non ambigus et disjoints. Supposons qu'il existe un automate déterministe $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, z_0, F\rangle$ reconnaissant L.
Soit n un entier non nul. Notons $h(n)$ la hauteur maximale de la pile le long de l'exécution de $\mathcal{A}$ sur le préfixe $a^n b$. Puisque l'ensemble des résiduels de L par $a^n b$ est infini, l'ensemble des configurations obtenues après avoir parcouru $a^n b$ est infini, donc h n'est pas bornée.
Soit t^i l'instant de l'exécution de $\mathcal{A}$ sur $a^n b^n$ qui précède la lecture de la i-ème occurence de a. Définissons la suite $(t_1, \ldots, t_k)$ telle t_k soit le premier t^i où h atteint son maximum et telle que, pour tout j, t_{j-1} soit le plus grand t^i tel que $t_{j-1} < t_j$. À chaque t_j correspond une configuration (q_j, ν_j) de sorte que la suite des ν_j soit strictement croissante pour l'ordre préfixe et telle que l'exécution $(q_j, \nu_j) \xrightarrow{a^p} (q_{j+1}, \nu_{j+1})$ ne consomme pas ν_j. À chaque t_j on peut associer un élément de $d_j \in Q \times Z \times (Q \cup \{*\})$ de sorte que
–$d_j = (q, z, q')$ si la configuration au temps t_j s'écrit $(q, \nu z)$ et l'exécution finit par consommer la lettre z en aboutissant à l'état q',
–$d_j = (q, z, *)$ si la configuration au temps t_j s'écrit $(q, \nu z)$ et l'exécution finit par accepter sans consommer la lettre z.
Comme h n'est pas bornée, il existe un n tel que la suite $(t_1, \ldots, t_k)$ soit de longueur strictement supérieure à $|Q \times Z \times (Q \cup \{*\})|$, il existe alors $j < j'$ tels que $d_j = d'_j$. Deux cas sont possibles pour l'exécution à partir de la configuration (q_j, ν_j) au temps t_j :
–Si $d_j = (q, z, *)$ alors ν_j reste sur la pile jusqu'à l'acceptation, auquel cas il en est de même pour l'exécution à partir de $(q_{j'}, \nu_{j'})$, et comme $(q_j, z_j) = (q_{j'}, z_{j'})$ il est donc possible d'itérer le facteur a^q en atteignant toujours l'acceptation. On a alors $\{ a^{n+(i-1)q} b^n \mid i \in \mathbb{N} \} \subseteq L$, ce qui est contradictoire.
–Si $d_j = (q, z, q')$, alors la lettre z_j finit par être consommée en arrivant dans l'état q', et dans ce cas l'exécution à partir de $(q_{j'}, \nu_{j'})$ fait de même. Notons $a^n b^n = \alpha u \beta \nu \gamma$ où u est la partie du mot qui passe de t_j à $t_{j'}$ et ν est la partie du mot qui fait repasser la pile de $\nu_{j'}$ à ν_j en restant dans l'état q'. Pour tout i on a alors $\alpha u^i \beta \nu^j \gamma \in L$, ce qui prouve que u et ν sont des puissances de a et b de même longueur p.
La configuration atteinte après la lecture de $\alpha u^i \beta \nu^i$ est toujours la même, puisque l'automate est déterministe. À partir de cette configuration, l'automate accepte γ par hypothèse, or $a^n b^n = \alpha u \beta \nu \gamma$ et ν est une puissance non nulle de b donc il existe un $r \in \mathbb{N}$ tel que $\gamma = b^n$. Or l'automate est censé accepter L, donc à partir de la même configuration il doit accepter le mot b^{r+n} puisque $a^n b^2 n = \alpha u \beta \nu b^{r+n} \in L$. On en déduit que, pour tout $i \in \mathbb{N}$, $a^{n+(i-1)p} b^{2n+(i-1)p} \in L$, ce qui est contradictoire.
Il ne peut donc exister aucun automate à pile déterministe reconnaissant L.

Exercice 3 – *Lemme d'itération déterministe*

Montrer que pour tout langage déterministe $L \subseteq \Sigma^*$, il existe un entier k tel que pour tout $f \in L$ avec au moins k lettres marquées, il existe une décomposition $f = \alpha u \beta \nu \gamma$ telle que

1. pour tout $n \in \mathbb{N}$, $\alpha u^n \beta \nu^n \gamma \in L$;
2. soit α, u et β, soit β, ν et γ contiennent chacun au moins une lettre marquée ;
3. $u \beta \nu$ contient au plus k lettres marquées ;
4. pour tout mot $w \in \Sigma^*$, s'il existe $p \in \mathbb{N}$ tel que $\alpha u^p \beta \nu^p w \in L$ alors pour tout $p \in \mathbb{N}$ on a $\alpha u^p \beta \nu^p w \in L$.

Exercice 4 – *Variantes d'automates déterministes*

1. Soit $\mathcal{A} = \langle Q, \Sigma, Z, T, q_0, z_0, K\rangle$ un automate à pile déterministe reconnaissant par sommet de pile et état final (une configuration $(q, \alpha z)$ est acceptante si (q, z) est élément d'un certain ensemble $K \subseteq Q \times Z$). Montrer que l'on peut effectivement construire un automate à pile déterministe équivalent reconnaissant par état final.

   Correction Coder le sommet de la pile dans l'état, en ajoutant un symbole de fond de pile.

2. Soit $\mathcal{A}$ un automate à pile déterministe. Montrer que l'on peut effectivement construire un automate à pile déterministe qui reconnaît le même langage et dont les ε-transitions sont uniquement effaçantes : $(p, x) \xrightarrow{\varepsilon} (q, \varepsilon)$.

   Correction On commence par éliminer les cycles d'ε-transitions. Une suite d'ε-transitions contient des conditions sur la pile et des écritures sur la pile. On peut donc voir ces transitions comme une relation sur $Q \times \Gamma^* \times Q \times \Gamma^*$ où chaque état apparaît au plus un fois à gauche ou à droite. Si on se retrouve avec un transition $q \xrightarrow{\varepsilon,u/vz} q'$, on crée un état frais q'', on remplace cette transition par $q \xrightarrow{\varepsilon,u/\varepsilon} q''$, et pour chaque transition $q' \xrightarrow{a,z/w} q'''$ on ajoute une transition $q'' \xrightarrow{a,*/vw} q'''$, indépendante du sommet de pile.

Exercice 5 – *Complémentaires*

Soit $\mathcal{A}$ un automate à pile déterministe reconnaissant un certain langage L.

1. Rappeler ce qu'est un blocage dans $\mathcal{A}$. Montrer qu'il est possible de construire effectivement un automate déterministe $\mathcal{A}'$ reconnaissant L sans blocage.

   Correction Utiliser les ensembles calculables de la dernière fois pour détecter les cycles d'ε-transitions. Remplacer les états de ce cycle par un état puits, acceptant si le cycle contenait un état acceptant.

2. Montrer qu'il est possible de construire effectivement un automate dont aucun état final n'a d'ε-transition.

   Correction On commence par ajouter un symbole de fond de pile $\perp$. Supposons qu'on ait une transition $q \xrightarrow{\varepsilon,z/u} q'$ avec q acceptant. On introduit deux états frais q_1 et q_2, avec q_1 acceptant et ayant les mêmes transitions que q', et avec q_2 acceptant et sans transition. On remplace alors la transition ci-dessus par $q \xrightarrow{\varepsilon,z/u} q_1$ et $q \xrightarrow{\varepsilon,\perp/\perp} q_2$, et on rend q non acceptant. On vérifie que l'automate obtenu reconnaît le même langage, et il a une ε-transition de moins partant d'un état acceptant.

3. En déduire que la classe des langages algébriques déterministes est close par complémentaire.

   Correction Dans un automate vérifiant les conditions précédentes, il suffit de passer au complémentaire dans les états acceptants pour passer au complémentaire dans le langage. Mais il faut quand même le prouver.