ISIMA 1ère année

25 juin 2001
Durée : 2 heures
Documents autorisés

AUTOMATES

1°) Soit L_1 le langage représenté par l'expression régulière ($(a + aa)\ (b + bb)$)*.

a) Déterminer la grammaire qui engendre L_1.
b) Déterminer l'AFD minimal qui accepte L_1.

2°) Soit L_2 le langage engendré par la grammaire $G = (\ V_N,\ V_T,\ S,\ R\ )$, avec $V_N = \{S\}$, $V_T = \{a, b\}$, et $R = \{\ S \rightarrow abS\ ;\ S \rightarrow a^2b^2S\ ;\ S \rightarrow \varepsilon\ \}$.

a) Expliciter le langage L_2.
b) L_2 est-il un langage régulier ?

3°) Soit $L_3 = \{\ a^{n_1}b^{n_1} \ldots a^{n_k}b^{n_k}\ ,\ k \in \mathbb{N},\ \forall\ i \in \{1 \ldots k\}\ n_i \in \mathbb{N}^*\ \}$.

a) Déterminer la grammaire qui engendre L_3.
b) Déterminer l'APND qui accepte L_3.
c) Montrer que :

$\forall\ m \in L_3$ tel que $|m| \geq 2$: ($\exists\ x, u, y$, tels que $m = xuy$, $u \neq \varepsilon$, et $\forall\ n \in \mathbb{N}^*\ xu^ny \in L_3$).

Que peut-on en déduire ?

4°) Soit $L = \{\ a^{2^n}\ ,\ n \in \mathbb{N}\ \}$.

a) Montrer que L n'est pas un langage hors contexte.
b) Déterminer la machine de Turing qui accepte le langage $L' = \{>\}L$.
c) En déduire la machine de Turing qui accepte le langage $L'' = \{>\}\{\ a^{2^n}b^{2^n}\ ,\ n \in \mathbb{N}\ \}$.
d) En déduire la machine de Turing qui accepte le langage :

$$L_4 = \{>\}\{\ a^{2^{n_1}}b^{2^{n_1}} \cdots\ a^{2^{n_k}}b^{2^{n_k}}\ ,\ k \in \mathbb{N}\ ,\ \forall i \in \{1 \ldots k\}\ n_i \in \mathbb{N}\ \}.$$

Solution :

4°) a) Principe général : répéter une boucle consistant à chaque itération à diviser le nombre de a par deux. Pour faire ceci, on prend le premier a non marqué, on le marque (q_1), et on va chercher à supprimer un a à la fin du mot (q_2 et q_3). Si à un moment, pour un a que l'on vient de marquer, on ne trouve pas un autre a en fin de mot (on tombe directement sur un # en q_2), c'est que le nombre de a était impair. Si le nombre de a est pair, on arrive avec cette méthode à diviser le nombre de a par deux. Dans ce dernier cas, on démarque tous les A (q_6), et on recommence. Si le nombre de a n'est pas une puissance de deux, alors on finira forcément avec un nombre de A impair supérieur à 1. Par contre si le nombre de a est une puissance de deux, alors on finira avec un seul A. C'est q_7 et q_8 qui contrôle s'il n' y a qu'un seul A et qui acceptent le mot dans ce seul cas.

Algorithme détaillé :

q_0 : aller sur le premier a et passer en q_1 ;

Boucle q_1 … q_5 avec sortie en q_6 si le nombre est pair et a été divisé par deux, et sortie en q_7 s'il est impair (et donc que le dernier A marqué n'a pas eu de a correspondant) :

q_1 : marquer le premier a et passer en q_2 ; s'il n'y a plus de a, le nombre de a était pair, et a donc été divisé par deux, passer alors en q_6 ;

q_2 : si derrière le A que l'on vient de marquer il n'y a plus rien, c'est que le nombre de a était impair, passer alors en q_7 pour contrôler si ce nombre de a est 1 ou un entier impair supérieur à 1 ; sinon passer en q_3 ;

q_3 : aller à la fin du mot, et passer en q_4 ;

q_4 : effacer le dernier a du mot et passer en q_5 ;

q_5 : retourner jusqu'au dernier A marqué, et passer en q_1 ;

q_6 : retourner au début en démarquant les A, et passer en q_1 pour recommencer une nouvelle division par 2 ;

q_7 et q_8 : pour que le mot soit accepté, il doit être de longueur 1 (résultat de n divisions successives de 2^n par 2).

	>	a	A	#
q_0	$(q_1, >, \rightarrow)$			
q_1		$(q_2, A, \rightarrow)$		$(q_6, \#, \leftarrow)$
q_2		$(q_3, a, \rightarrow)$		$(q_7, \#, \leftarrow)$
q_3		$\rightarrow$		$(q_4, \#, \leftarrow)$
q_4		$(q_5, \#, \leftarrow)$		
q_5		$\leftarrow$	$(q_1, A, \rightarrow)$	
q_6	$(q_1, >, \rightarrow)$		$(q_6, a, \leftarrow)$	
q_7			$(q_8, A, \leftarrow)$	
q_8	$(f, >, \rightarrow)$			