ISIMA 1ère année

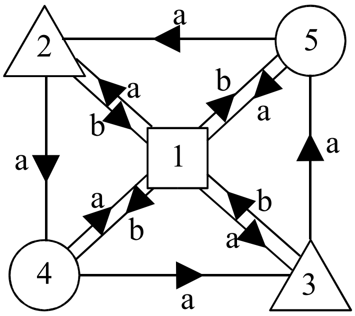
11 septembre 2000
Durée : 2 heures
Documents autorisés

AUTOMATES

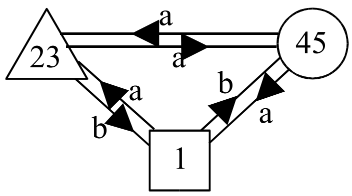
Exercice 1 :

1°) Les deux automates A1 et A2 ci-dessous sont-ils équivalents ?

A1 :



A2 :



Solution :
Il faut déterminiser chaque automate pour pouvoir ensuite les comparer.

A1 :

	a	b	
1	2,3	4,5	
2	4	1	f
3	5	1	f
4	1,3	-	
5	1,2	-	

⇒

	a	b	
1	23	45	
23	45	1	f
45	123	-	
123	2345	145	f
2345	12345	1	f
145	123	45	
12345	12345	145	f

A2 :

	a	b	
1	23	45	
23	45	1	f
45	1,2 3	-	

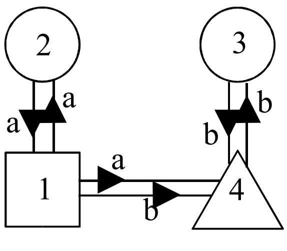
⇒

	a	b	
1	23	45	
23	45	1	f
45	123	-	
123	2345	145	f
2345	12345	1	f
145	123	45	
12345	12345	145	f

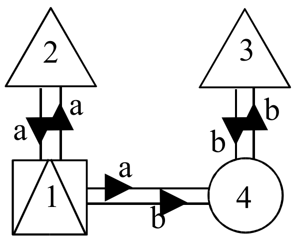
On observe que les deux automates A1 et A2 sont équivalents.

2°) Les langages acceptés par les deux automates A1 et A2 ci-dessous sont-ils complémentaires ? Déterminer ces langages.

A1 :



A2 :



Solution :

Il faut déterminiser ces automates, puis vérifier si leurs états sont complémentaires (c'est-à-dire que les états finaux de l'un sont les états non finaux de l'autre, et réciproquement).

A1 :

	a	b	
1	2,4	4	
2	1	-	
3	-	4	
4	-	3	f

⇒

	a	b	
1	24	4	
24	1	3	f
4	-	3	f
3	-	4	

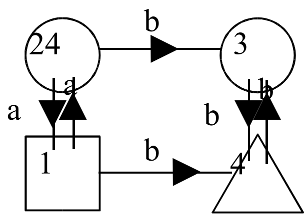
A2 :

	a	b	
1	2,4	4	f
2	1	-	f
3	-	4	f
4	-	3	

⇒

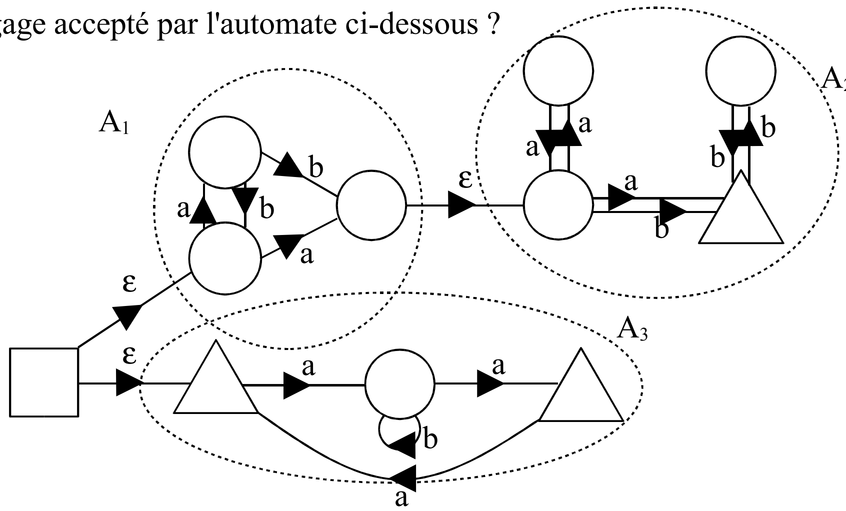
	a	b	
1	24	4	f
24	1	3	f
4	-	3	
3	-	4	f

On observe donc que ces langages ne sont pas complémentaires.
Le langage accepté pas A1 est <(aa)*(a+b)(bb)*> = {$a^n b^m$ avec $n+m$ impair}.
Pour déterminer le langage accepté par A2, comme celui-ci a trois états finaux, et un seul état non final, il est préférable de partir de l'automate complémentaire à l'automate déterministe équivalent à A2, représenté ci-dessous :



Le langage accepté par l'automate ci-dessus est {$a^n b^m$ avec $n+m$ impair et $m>0$} et le langage accepté par A2 est donc le complémentaire de ce langage (c'est-à-dire le langage des mots qui ne sont pas du type $a^n b^m$, ou qui sont de ce type, mais avec $n+m$ pair ou $m=0$) .

3°) Quel est le langage accepté par l'automate ci-dessous ?



Solution :
En désignant respectivement par L_1, L_2 et L_3 les langages acceptés par les automates A_1, A_2 et A_3 (où on considère comme état initial l'état par lequel on "entre" dans l'automate) représentés ci-dessus, le langage accepté par l'automate ci-dessus est $L = L_1L_2 + L_3^*$ avec $L_1 = \langle(ab)^*(ab+a)\rangle$, $L_2 = \langle(aa)^*(a+b)(bb)^*\rangle$ et $L_3 = \langle ab^*a\rangle$.

Exercice 2 :

1°) Montrer que le langage $L = \{a^n$ avec $n = 2^k, k \in \mathbb{N}\}$ n'est ni régulier, ni hors contexte.
Solution :
• Pour montrer que ce langage n'est pas régulier, on doit utiliser la contraposée du théorème du gonflement pour les langages réguliers : si $\forall K \in \mathbb{N}$, $\exists m \in L$ de longueur supérieure à K tel pour toute décomposition de m du type $m = xuy$, avec $x, u, y \in V^*$ (ici $V = \{a\}$) et $u \neq \varepsilon$, $\exists n \in \mathbb{N}$ tel que $xu^ny \notin L$, alors L n'est pas régulier.
Dans le cas du langage considéré, pour tout mot $m \in L$ de longueur 2^k, $k \in \mathbb{N}$ (et donc quelle que soit la longueur de m), alors pour toute décomposition de m du type $m = xuy$, avec $x, u, y \in V^*$ et $u \neq \varepsilon$, en notant respectivement p, q et r les longueurs des mots x, u et y, on a $p+q+r = 2^k$, et il est évident qu'il existe $n \in \mathbb{N}$ tel que $p+nq+r$ n'est pas une puissance entière de 2 (en effet, si $p+2q+r = 2^{k'} = 2^{k+l}$, alors $q = 2^{k+l} - 2^k$ et dans ce cas $2^{k+l} < p+3q+r < 2^{k+l+1}$).
• Pour montrer que ce langage n'est pas hors-contexte, on utilise de même la contraposée du théorème du gonflement pour les langages hors-contexte: si $\forall K \in \mathbb{N}$, $\exists m \in L$ de longueur supérieure à K tel pour toute décomposition de m du type $m = uvxyz$, avec $u, v, x, y, z \in V^*$, $v \neq \varepsilon$ et $y \neq \varepsilon$, $\exists n \in \mathbb{N}$ tel que $uv^nxy^nz \notin L$, alors L n'est pas hors-contexte.
Dans le cas du langage considéré, pour tout mot $m \in L$ de longueur 2^k, $k \in \mathbb{N}$ (et donc quelle que soit la longueur de m), alors pour toute décomposition de m du type $m = uvxyz$, avec $u, v, x, y, z \in V^*$, $v \neq \varepsilon$ et $y \neq \varepsilon$, en notant respectivement p, q, r, s et t les longueurs des mots u, v, x, y et z, on a $p+q+r+s+t = 2^k$, et il est évident qu'il existe $n \in \mathbb{N}$ tel que $p+nq+r+ns+t$ n'est pas une puissance entière de 2 (en effet, si $p+2q+r+2s+t = 2^{k'} = 2^{k+l}$, alors $q+s = 2^{k+l} - 2^k$ et dans ce cas $2^{k+l} < p+3q+r+3s+t < 2^{k+l+1}$).

2°) Déterminer une machine de Turing qui accepte ce langage.
Algorithme suggéré : on marque un premier "a" qui, une fois marqué, sert à en marquer un deuxième, puis ces deux "a" marqués sont à leur tour utilisés pour en marquer deux autres, …
On pourra supposer, si besoin est, l'existence d'un symbole ">" pour marquer le début du ruban.
Solution :

	a	A	α	>	#
q_0	$(q_1,A,\leftarrow)$			$\rightarrow$	
q_1		$(q_2,a,\rightarrow)$		$\rightarrow$	
q_2	$(q_3,\alpha,\leftarrow)$	$\rightarrow$	$\rightarrow$		
q_3	$(q_5,a,\rightarrow)$	$(q_4,A,\leftarrow)$	$\leftarrow$		
q_4	$(q_1,a,\rightarrow)$	$\leftarrow$			
q_5	$(q_6,a,\leftarrow)$		$\rightarrow$		$(q_F,\#,\leftarrow)$
q_6	$(q_6,A,\leftarrow)$		$(q_6,A,\leftarrow)$	$(q_1,>,\rightarrow)$	
q_F	$\leftarrow$		$(q_F,a,\leftarrow)$		

Les symboles "$\rightarrow$" et "$\leftarrow$" indiquent un simple déplacement à droite ou à gauche, sans modification de l'état du ruban, et sans changement d'état.

Algorithme :

q_0 : on marque le premier "a" en "A", on se repositionne en début de ruban, et on passe en q_1 ;

q_1 : pour marquer autant de "a" qu'il y a de "A", il faut marquer les "A". On peut ici réutiliser le symbole "a" : on marque donc le "A" en "a", et on passe en q_2 ;

q_2 : on va chercher le premier "a" que l'on trouve et on le marque, mais d'une manière différente pour pouvoir le distinguer des "A" : on le marque donc en "α", et on passe en q_3 ;

q_3 : on retourne en arrière sur les "α" : si après les avoir tous parcourus, on retombe sur un "A", c'est qu'on ne les a pas tous traités, on passe en q_4 ; si ar contre, on tombe directement sur un "a", c'est que l'on a fini de traiter tous les "A", et on passe en q_5 ;

q_4 : on recherche le "A" le plus à gauche et l'on se positione dessus en passant en q_1 où la boucle continue ;

q_5 : on doit vérifier si l'on a fini ou non : si l'on a fini, il n'y a plus de "a" à droite des "α". On parcourt donc tous les "α" : s'il n'y a plus de "a", on a fini, et on va en q_F ; s'il y a encore des "a", on passe en q_6 ;

q_6 : on marque tous les "α" et les "a" en "A" et on se positione sur le premier "A" en passant en q_1, où l'on recommence une autre boucle qui sera deux fois plus longue ;

q_F : o remet en forme le mot, et l'on se repositionne en début de ruban.