Université Lille 1 - Sciences et Technologies
Licence Informatique 2009–2010

UE AL - Automates et Langages

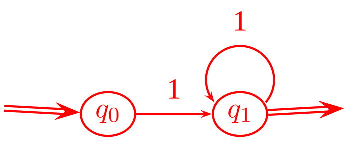
Correction d'exercices de TD

Automates finis

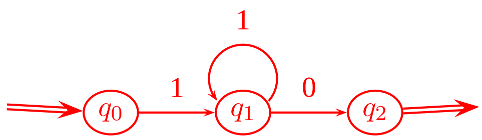
Exercice 6 :
On considère l'alphabet $X = \{0, 1\}$ et le langage $P = 1.(0 + 1)^*$ des mots qui commencent par 1. Chaque mot u de P peut être considéré comme l'écriture en binaire d'un entier strictement positif n. On notera alors $succ(u)$ l'écriture en binaire de $n + 1$. Par exemple, si $u = 11010011$ alors $succ(u) = 11010100$, si $u = 11$ alors $succ(u) = 100$.
On note $P' = \{u \in P||u| = |succ(u)|\}$ ($|u|$ désigne la longueur du mot u). Si l'on reprend les exemples précédents, 11010011 appartient à P', par contre 11 n'appartient pas à P' car $|11|$ vaut 2 et son successeur est de longueur 3.
Question 6.1 . Définir un automate fini déterministe M_1 tel que $L(M_1) = P'$.
Correction: Pour construire un tel automate, il faut dans un premier temps identifier plus précisément la nature du langage que l'on va représenter. En l'occurrence, la construction du langage $L(M_1)$ était le sujet de la question 1.3 de l'exercice 1 de la feuille d'exercices portant sur les *"Langages et expressions rationnels"*. Sa construction n'est donc pas détaillée ici, et seul reste le résultat :

$$L(M_1) = 1^+.0.(0 + 1)^*$$

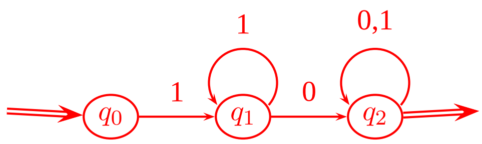
Ce langage est la concaténation de trois sous langages que nous appellerons L_1, L_2 et L_3, avec $L_1 = 1^+$, $L_2 = 0$ et $L_3 = (0 + 1)^*$. Nous allons construire progressivement M_1, en décrivant les automates reconnaissant les langages L_1, puis $L_1.L_2$, puis enfin $L1.L2.L3$.
L'automate reconnaissant le langage $L_1 = 1^+$ est :



L'automate reconnaissant le langage $L_1.L_2 = 1^+.0$ est :



Enfin, l'automate reconnaissant le langage $L_1.L_2.L_3 = L(M_1) = 1^+.0.(0 + 1)^*$ est :



Cet automate est déterministe, et répond donc à la question posée.

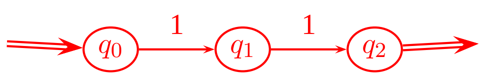
Soit le langage $E = \{h(u)|u \in P\}$ avec h la fonction qui "double" les lettres d'un mot. Plus formellement, h est définie par :
$h(0) = 00$, $h(1) = 11$, $h(u_1.u_2) = h(u_1).h(u_2)$.

Question 6.2 . Définir un automate fini déterministe M_2 tel que $L(M_2) = E$.

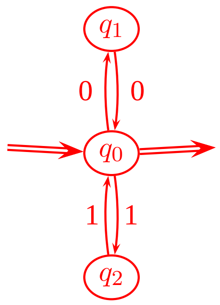
Correction: La première étape de cette question consiste à identifier le langage que l'on va reconnaitre avec l'automate M_2. Pour cela, nous commençons par écrire les premiers mots du langage E :

$h(1)$ = 11
$h(10)$ = 1100
$h(11)$ = 1111
$h(100)$ = 110000
$h(101)$ = 110011
$h(110)$ = 111100
$h(111)$ = 111111

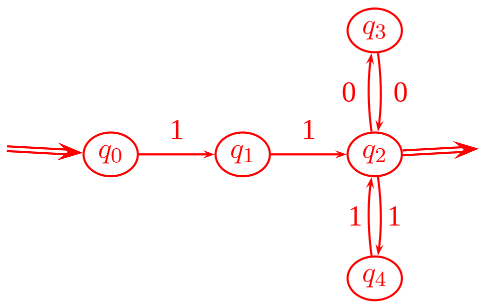
On peut remarquer que ces mots ne font que répéter un certain nombre de fois les couples 00 et 11. De plus, comme les mots de P commencent toujours par un 1, on peut en déduire que les mots de E commenceront toujours par 11. On peut donc en déduire que $E = 11.(00 + 11)^*$.
Comme pour la question précédente, nous décomposons la construction de l'automate en nous intéressant à des sous langages de E, en posant $E_1 = 11$, et $E_2 = (00 + 11)^*$. On a alors $E = E_1.E_2$
L'automate reconnaissant le langage $E_1 = 11$ est :



L'automate reconnaissant le langage $E_2 = (00 + 11)^*$ est :



Par conséquent, l'automate reconnaissant le langage $E = E_1.E_2 = 11.(00 + 11)^*$ est :



Cet automate est déterministe, et répond donc à la question posée.

Pour tout mot $u \in P'$, on note $s(u)$ le mot $a_1 b_1 a_2 b_2 \ldots a_k b_k$ tel que $u = a_1 a_2 \ldots a_k$ et $succ(u) = b_1 b_2 \ldots b_k$.
Par exemple, si $u = 1011$, alors $succ(u) = 1100$ et donc $s(u) = 11011010$.
On pose $S = \{s(u) | u \in P'\}$

Question 6.3 . Trouver deux langages L_1 et L_2 tels que $S = E.L1.L2$.

Correction: Pour trouver L_1 et L_2, nous commençons par écrire des exemples de mots dans S.

u	succ(u)	s(u)
10	11	1101
100	101	110001
101	110	110110
110	111	111101
1000	1001	11000001
1001	1010	11000110
1010	1011	11001101
1011	1100	11011010
1100	1101	11110001
1101	1110	11110110
1110	1111	11111101
1001010100	1001010101	11000011001100110001
10010100111	10010101000	1100001100110001101010

D'après l'énoncé, on sait que $S = E.L1.L2$. Cela signifie que tout mot de S peut être décomposé en trois parties : une partie provenant de E (le préfixe du mot), une partie provenant de $L1$, et une partie venant de $L2$. Écrit plus formellement, on sait que :

$$\forall u \in P', \exists e \in E, \exists u_1 \in L_1, \exists u_2 \in L_2 | s(u) = e.u_1.u_2$$

Pour caractériser $L1$ et $L2$, nous commençons par identifier le préfixe issu de E à tous les mots calculés dans le tableau ci-dessus. On obtient alors :

u	$s(u) = e.u_1.u_2$	
	e	$u_1.u_2$
10	11	01
100	1100	01
101	11	0110
110	1111	01
1000	110000	01
1001	1100	0110
1010	110011	01
1011	11	011010
1100	111100	01
1101	1111	0110
1110	111111	01
1001010100	110000110011001100	01
10010100111	11000011001100	01101010

En observant la colonne $u_1.u_2$, on peut remarquer que les mots de $L_1.L_2$ sont caractérisés par deux choses :

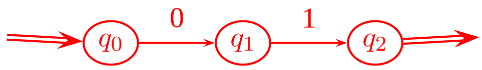
- Ils commencent toujours par le couple *01*;
- Les mots de $L_1.L_2$ sont soit *01*, soit *01* suivi d'une succession de *10*.

Les mots de la partie droite du tableau font donc partie du langage $0.1.(1.0)^*$. $L_1.L_2 = 0.1.(1.0)^*$ est donc une solution possible au problème que nous cherchons à résoudre. Par conséquent, en posant $L_1 = 0.1$, et $L_2 = (1.0)^*$, on a bien $S = E.L_1.L_2$.

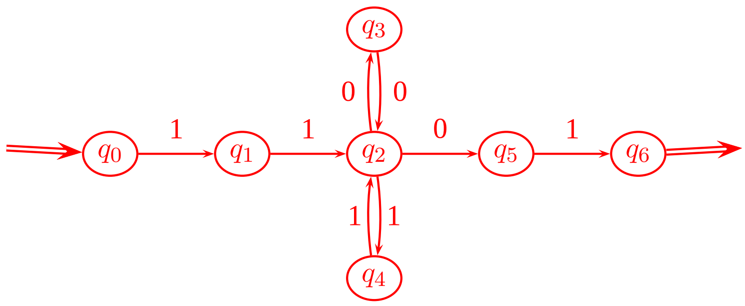
Question 6.4 . Définir un automate fini déterministe M_3 tel que $L(M_3) = S$.

Correction: Dans la question précédente, nous avons déterminé que $S = E.L_1.L_2$. D'après le cours, nous savons que l'automate reconnaissant la concaténation de plusieurs langages peut être construit simplement à partir des automates reconnaissant chaque élément de la concaténation. Nous commençons donc par construire les automates A_1, A_2 et A_3, tels que $L(A_1) = E$, $L(A_2) = L_1$ et $L(A_3) = L_1$. L'automate A_1 a été construit dans la question 6.2.

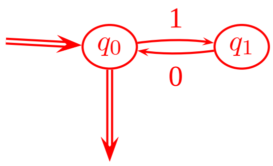
L'automate A_2 reconnaissant le langage $L_1 = 0.1$ est le suivant :



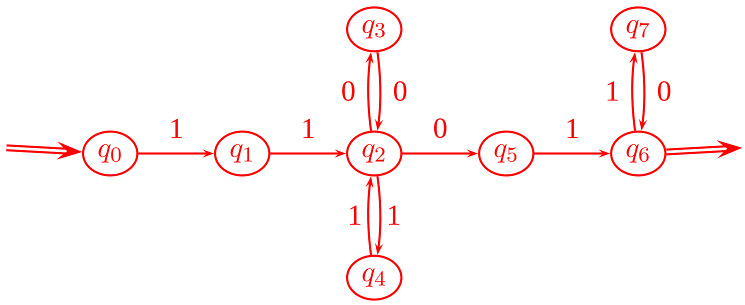
L'automate reconnaissant le langage $E.L_1 = 1.1.(0.0 + 1.1)^*.0.1$ est donc :



L'automate A_3 reconnaissant le langage $L_2 = (1.0)^*$ est le suivant :



L'automate M_3 reconnaissant le langage $S = 1.1.(0.0 + 1.1)^*.0.1.(1.0)^*$ est donc :



Cet automate n'est pas déterministe, à cause des deux transitions $q_2 \rightarrow^0 q_3$ et $q_2 \rightarrow^0 q_5$. Nous appliquons donc l'algorithme vu en cours permettant de construire une version déterministe de cet automate.

État	0	1
$\{q_0\}$	$-$	$\{q_1\}$
$\{q_1\}$	$-$	$\{q_2\}$
$\{q_2\}$	$\{q_3, q_5\}$	$\{q_4\}$
$\{q_3, q_5\}$	$\{q_2\}$	$\{q_6\}$
$\{q_4\}$	$-$	$\{q_2\}$
$\{q_6\}$	$-$	$\{q_7\}$
$\{q_7\}$	$\{q_6\}$	$-$

L'automate déterministe reconnaissant le langage S est donc :