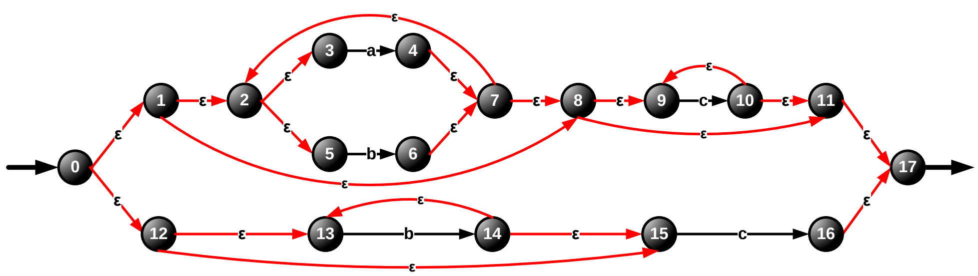
DE : Mathématiques pour l'informatique
Solutions

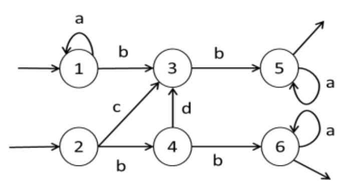
Exercice 1.

Construire, en respectant à la lettre les règles données en cours, l'automate asynchrone dont le langage est donné par l'expression rationnelle suivante : (a+b)*c* + b*c

Solution



Exercice 2. Déterminiser l'AF suivant :



Résultat sous forme de table de transition, avec identification des états de l'AFD en fonction de ceux de l'AF.

Solution

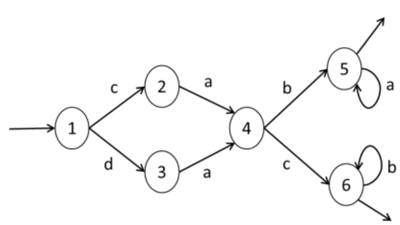
version non complétée

		a	b	c	d
E	1 2	1	3 4	3	--
	1	1	3	--	--
	3	--	5	--	--
	3 4	--	5 6	--	3
S	5	5	--	--	--
S	5 6	5 6	--	--	--

version complétée (cela n'a pas été demandé dans l'énoncé)

		a	b	c	d
E	1 2	1	3 4	3	P
	1	1	3	P	P
	3	P	5	P	P
	3 4	P	5 6	P	3
S	5	5	P	P	P
S	5 6	5 6	P	P	P
	P	P	P	P	P

Exercice 3. Minimiser l'AFD suivant :



Résultat : les partitions successives et les tables de transitions correspondantes doivent être explicitement données. L'automate minimal doit être fourni sous forme d'un schéma dans lequel les états sont nommés avec les parties (groupes) déterminées lors de la minimisation.

Solution

Il faut d'abord compléter l'automate :

Table de transitions

		a	b	c	d
E	1	--	--	2	3
	2	4	--	--	--
	3	4	--	--	--
	4	--	5	6	--
S	5	5	--	--	--
S	6	--	6	--	--

après complétion

		a	b	c	d
E	1	P	P	2	3
	2	4	P	P	P
	3	4	P	P	P
	4	P	5	6	P
S	5	5	P	P	P
S	6	P	6	P	P
	P	P	P	P	P

La partition initiale Θ_0={T, NT} où T={5,6}, NT={1,2,3,4,P}.

Itération 1 :

		a	b	c	d	sous Θ_0 a	b	c	d	
NT	1	P	P	2	3	NT	NT	NT	NT	
	2	4	P	P	P	NT	NT	NT	NT	
	3	4	P	P	P	NT	NT	NT	NT	
	4	P	5	6	P	NT	T	T	NT	séparation
	P	P	P	P	P	NT	NT	NT	NT	
T	5	5	P	P	P	T	NT	NT	NT	séparation
	6	P	6	P	P	NT	T	NT	NT	

Θ_1={A, (4), (5),(6)} où A={1,2,3,P} ; (4), (5) et (6) ne peuvent plus se séparer et ne participent plus à l'analyse.

Itération 2 :

		a	b	c	d	sous Θ_1 a	b	c	d	
A	1	P	P	2	3	A	A	A	A	
	2	4	P	P	P	(4)	A	A	A	séparation
	3	4	P	P	P	(4)	A	A	A	
	P	P	P	P	P	A	A	A	A	

Θ_2={(1,P), (2,3), (4), (5),(6)}

Itération 3 :

		a	b	c	d	sous Θ_2 a	b	c	d	
(1,P)	1	P	P	2	3	(1,P)	(1,P)	(2,3)	(2,3)	séparation
	P	P	P	P	P	(1,P)	(1,P)	(1,P)	(1,P)	
(2,3)	2	4	P	P	P	(4)	(1,P)	(1,P)	(1,P)	pas de séparation
	3	4	P	P	P	(4)	(1,P)	(1,P)	(1,P)	

Θ_3={(1), (P), (2,3), (4), (5), (6)}. Seul le groupe (2,3) consiste en plus d'un état. Mais il reste tel quel sous Θ_3 :

Itération 4 :

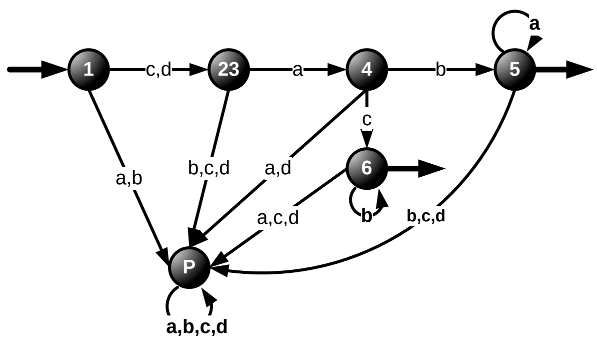
		a	b	c	d	sous Θ_3 a	b	c	d	
(2,3)	2	4	P	P	P	(4)	(P)	(P)	(P)	pas de séparation
	3	4	P	P	P	(4)	(P)	(P)	(P)	

Donc $\Theta_4 = \Theta_3 = \Theta_{fin} = \{(1), (P), (2,3), (4), (5), (6)\}$.

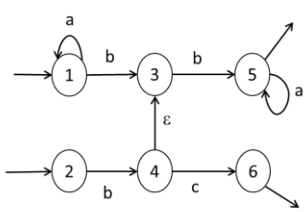
L'entrée est (1). Les sorties sont (5) et (6). La table de transitions :

		a	b	c	d
E	(1)	(P)	(P)	(2,3)	(2,3)
	(2,3)	(4)	(P)	(P)	(P)
	(4)	(4)	(5)	(6)	(P)
S	(5)	(5)	(P)	(P)	(P)
S	(6)	(P)	(6)	(P)	(P)
	(P)	(P)	(P)	(P)	(P)

Le schéma (où on a simplifié la notation des états en supprimant les parenthèses et les virgules) :



Exercice 4. Obtenir sous forme d'une expression rationnelle le langage de l'automate ci-contre :



Vous pouvez utiliser :

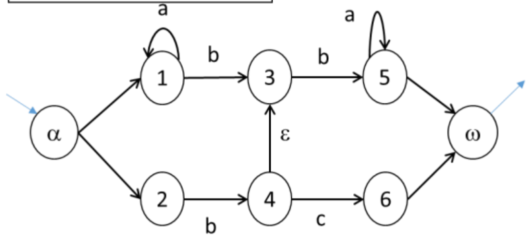
- Soit la méthode d'élimination des états,
- Soit la méthode de l'arrivée, avec les équations.

Quelle que soit la méthode utilisée, toutes les étapes intermédiaires doivent apparaître dans votre réponse.

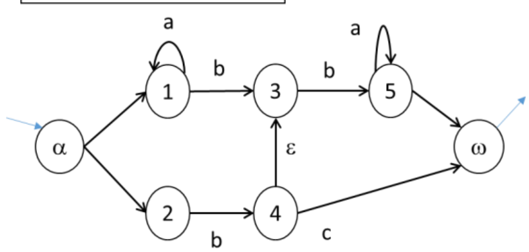
Solution (méthode d'élimination des états)

(*Dans les schémas ci-dessous, les transitions sans étiquettes sont des transitions epsilon*) :

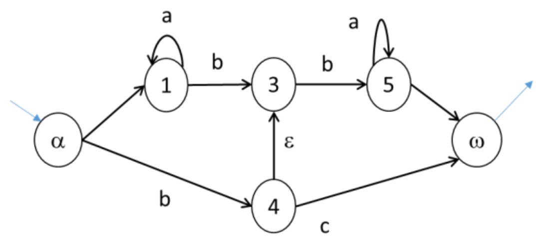
1. Ajout des états initial et terminal



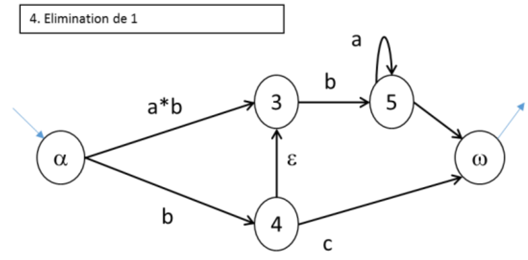
2. Elimination de 6



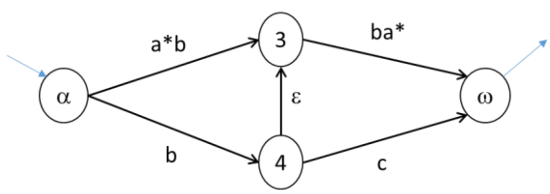
3.Elimination de 2



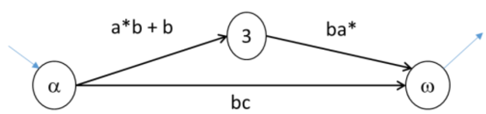
4. Elimination de 1



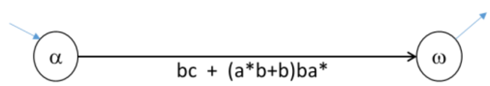
5. Elimination de 5



6. Elimination de 4



7. Elimination de 3



Donc, le langage L=bc+(a*b+b)ba* = bc+a*bba* (car b∈a*b, donc a*b+b=a*b)

ATTENTION ! Si vous avez donné une réponse correcte bc+a*bba* non pas parce que vous avez compris que a*b+b=a*b mais parce que vous avez oublié le terme bba* pendant le processus d'élimination d'états, vous n'obtenez pas tous les points ! C'est un cas assez fréquent.

Solution (méthode de l'arrivée) :

(on note 1 le langage reconnu par l'état 1, souvent noté X_1, etc.)

$$\begin{cases} 1=\varepsilon+1a=\varepsilon a^*=a^* \text{ (Lemme d'Arden)} \\ 2=\varepsilon \\ 3=1b+4\varepsilon=1b+4=a^*b+4 \text{ (en utilisant le 1 déjà trouvé)} \\ 4=2b=\varepsilon b=b \\ 5=3b+5a=3ba^* \text{ (Lemme d'Arden)} \\ 6=4c=bc \text{ (en utilisant le 4 déjà trouvé)} \end{cases}$$

Le langage de l'automate L=5+6. En utilisant 4=b, 3=a*b+b=a*b (car b∈a*b). Donc 5=a*bba*. Donc L=5+6=a*bba*+bc.

Exercice 5. Obtenir un AF dont le langage est le complément du langage de l'AF suivant :

		a	b
E/S	1	2	3
S	2	--	--
S	3	2	4
S	4	2	--

Résultat : sous forme d'une table de transitions avec les entrées et les sorties indiquées, ou sous forme d'un schéma avec les entrées et les sorties indiquées, ou les deux.

Solution Cet automate est déterministe mais non complet. On le complète :

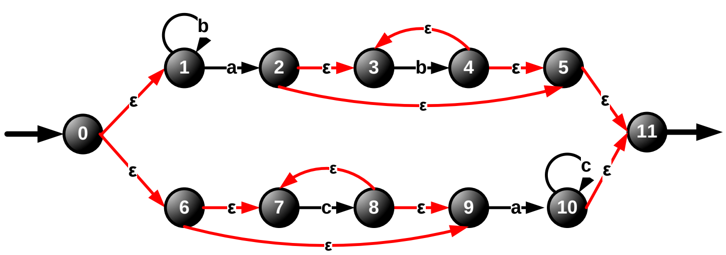
		a	b
E/S	1	2	3
S	2	P	P
S	3	2	4
S	4	2	P
	P	P	P

et on effectue l'opération T↔NT (tous les états terminaux deviennent non terminaux et vice versa) :

		a	b
E	1	2	3
	2	P	P
	3	2	4
	4	2	P
S	P	P	P

Cet automate reconnait le langage complémentaire à celui d'origine.

Exercice 6. Déterminiser l'automate asynchrone suivant :



Résultat : sous forme d'une table de transitions avec les entrées et les sorties indiquées, ou sous forme d'un schéma avec les entrées et les sorties indiquées, ou les deux. Vous devez absolument suivre scrupuleusement l'algorithme de déterminisation d'un automate asynchrone : il est interdit de faire une simplification plus ou moins intuitive de l'automate. Si vous utilisez des ε-clôtures, vous devez fournir une « table de traduction », où le contenu de chaque ε-clôture figurant dans la table de déterminisation est explicité. N'oubliez pas que tout automate où même une entrée ou sortie est incorrecte, est incorrect.

Solution

Les ε-clôtures figurant dans la déterminisation :

0'=(0 1 6 7 9); 1'=1; 2'=(2 3 5 11) (terminal); 4'=(3 4 5 11) (terminal); 8'=(7 8 9); 10'=(10 11) (terminal).

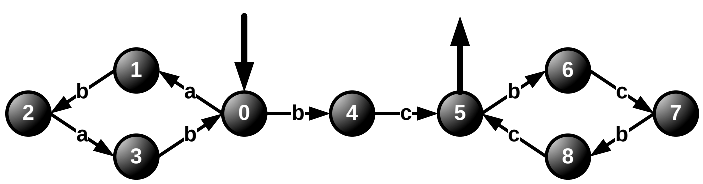
Avec ceci, la déterminisation :

a b c

E	0'	2'10'	1'	8'
S	2'10'	P	4'	10'
	1'	2'	1'	P
	8'	10'	P	8'
S	4'	P	4'	P
S	2'	P	4'	P
S	10'	P	P	10'
	P	P	P	P

Exercice 7. Construire un AF sur l'alphabet {*a*,*b*} reconnaissant le langage dont les mots commencent par un nombre paire (y compris 0) de '*ab*' suivis d'un nombre impaire de '*bc*'. Par exemple, les mots *'bc', 'bcbcbc', 'ababbc', 'ababbcbcbc'* appartiennent au langage, tandis que les mots *'bcbc', 'ab', 'abbc', 'bcabab'* etc., non.

Une solution :



ATTENTION Certains élèves ont produit un automate où les transitions sont marquées par 'ab', 'bc'. Ce n'est pas acceptable. De telles étiquettes ne sont utilisées que dans le cadre de la méthode d'élimination des états, où on introduit la notion d'un « automate fini généralisé » juste pour la durée de la méthode. Ici, il convient de construire un AF « normal », sur l'alphabet {a,b}.

<u>8. Questions de cours</u> :

a) Soit un automate déterministe complet non minimal A. Pour obtenir l'automate déterministe complet minimal reconnaissant le langage complémentaire à celui reconnu par A, doit-on (choisir la réponse correcte)

- Minimiser A d'abord et puis effectuer l'opération T↔NT (transformer chaque état terminal en état non terminal, et chaque état non terminal, en état terminal)
- Ou effectuer l'opération T↔NT sur A d'abord et minimiser l'automate obtenu
- Ou bien on obtient la même chose dans les deux cas ?

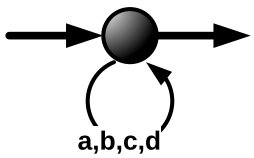
Réponse : on obtient la même chose dans les deux cas.
ATTENTION : dans le contexte de ces trois choix, nous ne considérons comme correcte que la troisième réponse, même si les deux manipulations décrites dans la première et la deuxième réponse sont tout à fait acceptables.

b) Quel est l'automate minimal équivalent à l'automate suivant :

	état	a	b	c	d
E/S	0	1	2	3	0
S	1	2	3	0	1
S	2	3	0	1	2
S	3	0	1	2	3

Donner une réponse immédiate, **<u>sans effectuer la procédure de minimisation par des partitions successives</u>** mais avec une explication !

Réponse : C'est un automate déterministe qui est complet, et dont tous les états sont terminaux. Donc il reconnait A*=(a+b+c+d)*. L'automate minimal reconnaissant A* consiste en un seul état :



c) Démontrer que si on a trois entiers relatifs non nuls, *a*, *b* et *c* tels que *a* divise le produit *bc* et que *a* et *b* sont premiers entre eux alors *a* divise *c*. (Théorème de Gauss)

Démonstration :

a divise bc donc $\exists$ k$\in\mathbb{Z}^*$tel que bc = ka. PGCD(a,b)=1 donc d'après Bézout $\exists$ (u,v)$\in\mathbb{Z}^{*2}$tel que au+bv=1.
On multiplie par c : auc+bvc =c, or bc=ka, donc auc+kva=c, c'est-à-dire a(uc+kv)=c : a divise c.