DE: Mathématiques pour l'informatique, solutions

Exercice 1 Déterminisation.

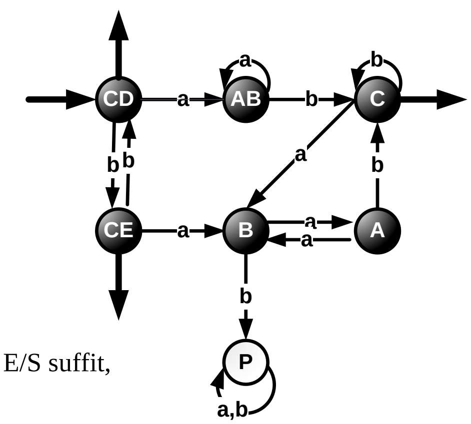
Soit l'automate sur l'alphabet {a,b} :

	état	a	b
	A	B	C
	B	A	--
E/S	C	B	C
E	D	A	E
	E	B	D

Construire un automate déterministe complet A_1 équivalent à cet automate.

Solution

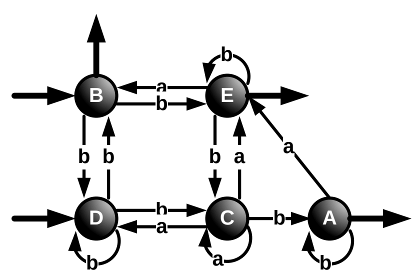
	déterminisation		
	état	a	b
E/S	CD	AB	CE
	AB	AB	C
S	CE	B	CD
S	C	B	C
	B	A	P
	A	B	C
	P	P	P



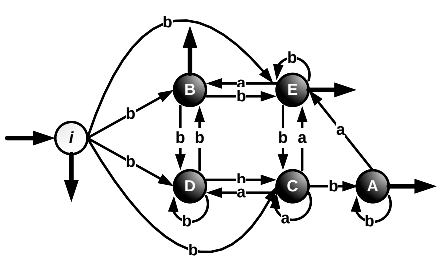
Remarque : la table de transitions avec les E/S suffit,
Il n'est pas nécessaire de donner le dessin

Exercice 2 Standardisation

a) Obtenir un automate standard A_2 reconnaissant le même langage que l'automate ci-contre :



Solution

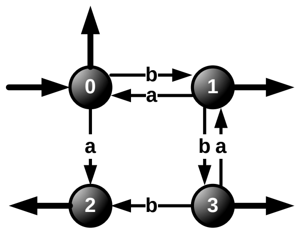


b) Obtenir un automate A_3 reconnaissant le même langage à l'exception du mot vide.

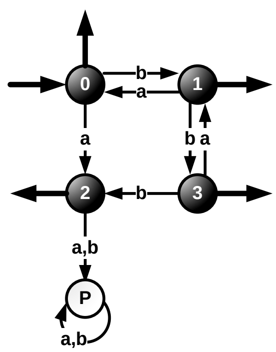
Solution
Il suffit d'enlever la flèche de sortie sur l'état ***i***.

Exercice 3 Langage complémentaire.

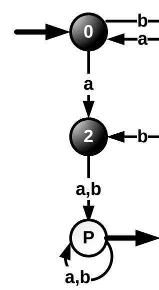
Construire un automate A_4 reconnaissant le langage complémentaire à celui reconnu par l'automate ci-dessous.



Solution
Cet automate est déterministe mais non complet.
L'automate déterministe complet équivalent est :



Donc, l'automate désiré est :



Exercice 4 Minimisation et le langage complémentaire

A = {a,b} est l'alphabet.
Pour l'automate défini par la table de transitions ci-dessous :

	état	a	b
E/S	A	B	C
S	B	--	--
S	C	D	E
S	D	B	C
S	E	F	B
S	F	D	E

a) Construire un automate <u>déterministe complet minimal</u> A_5 équivalent.
Solution
L'automate donné ci-dessus n'est pas complet. Complété, il devient :

	état	a	b
E/S	A	B	C
S	B	P	P
S	C	D	E
S	D	B	C
S	E	F	B
S	F	D	E
	P	P	P

Minimisation :

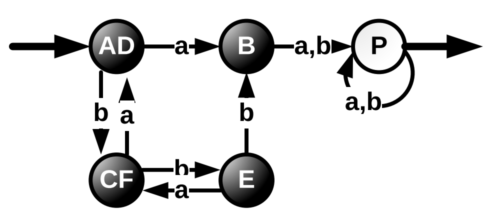
	Initialisation:				
	Θ_0={T, NT} où T={A,B,C,D,E,F} et NT={P}				
	Itération 1			Sous Θ_0	
		a	b	a	b
	A	B	C	T	T
	B	P	P	NT	NT
NT	C	D	E	T	T
	D	B	C	T	T
	E	F	B	T	T
	F	D	E	T	T
T	P	P	P		
	Θ_1={I, (B), (P)} où I={A,C,D,E,F}				
	Itération 2			Sous Θ_1	
		a	b	a	b
	A	B	C	B	I
	C	D	E	I	I
I	D	B	C	B	I
	E	F	B	I	B
	F	D	E	I	I
	Θ_2={(A,D), (C,F),(E), (B), (P)}				
	Itération 3			Sous Θ_2	
		a	b	a	b
	A	B	C	B	CF
	D	B	C	B	CF
	C	D	E	AD	E
	F	D	E	AD	E
	Θ_3=Θ_2 =Θ_{fin}				
	L'automate minimal:				
		a	b		
E/S	AD	B	CF		
S	B	P	P		
S	CF	AD	E		
S	E	CF	B		
	P	P	P		

b) Construire un automate <u>déterministe complet minimal</u> A_6 reconnaissant le <u>complémentaire</u> du langage reconnu par l'automate défini par la table ci-dessus.

Solution :

Le minimal du complémentaire est le complémentaire du minimal. Donc, on obtient

	L'automate complémentaire minimal:		
		a	b
E	AD	B	CF
	B	P	P
	CF	AD	E
	E	CF	B
S	P	P	P



Exercice 5 D'une expression rationnelle vers un automate fini

a) Construire un automate reconnaissant le langage

$$L = \big((0+1)(0+1)(0+1)\big)^* + \big((0+1)(0+1)(0+1)(0+1)\big)^*$$

suivant les règles formelles données dans le cours.

Puis, au choix,

soit

b) déterminiser et
c) minimiser

l'automate obtenu en (a) (c'est assez compliqué !),

soit (c'est plus facile à réaliser)

b) simplifier graphiquement et déterminiser, puis
c) minimiser

l'automate simplifié graphiquement.

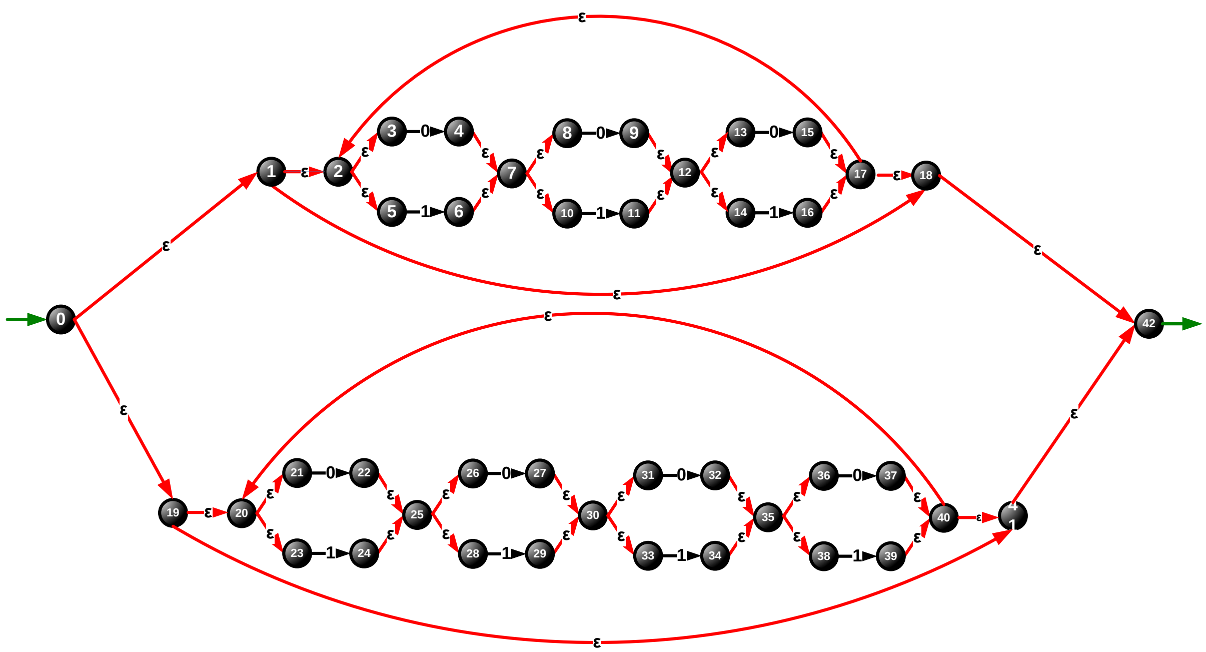
soit

(b+c) produire directement l'automate déterministe complet minimal que vous devriez obtenir en (c), si vous savez le faire *et si vous savez l'expliquer en quelques phrases courtes*. Dans ce cas, vous n'êtes pas tenus de faire le (b), mais vous devez toujours faire le (a).

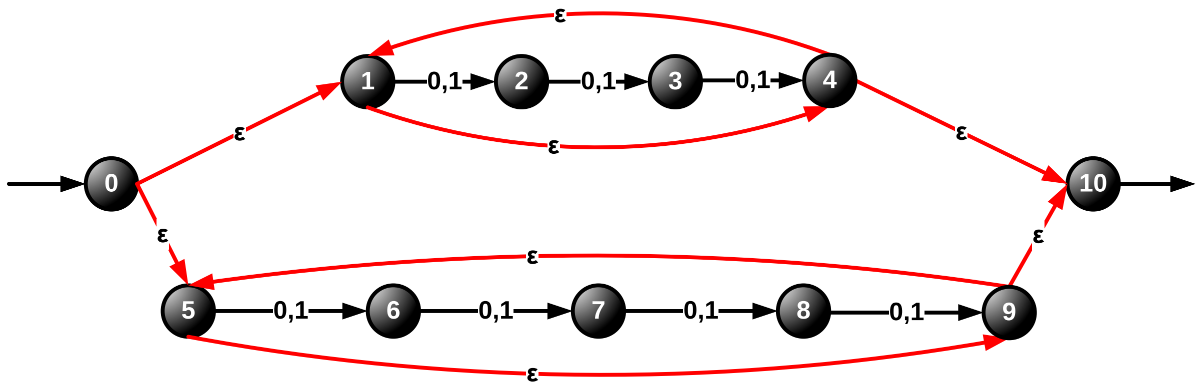
Notation pour cette exo : a) 33.3%, b) 33.3%, c) 33.3%.

Solution :

a)



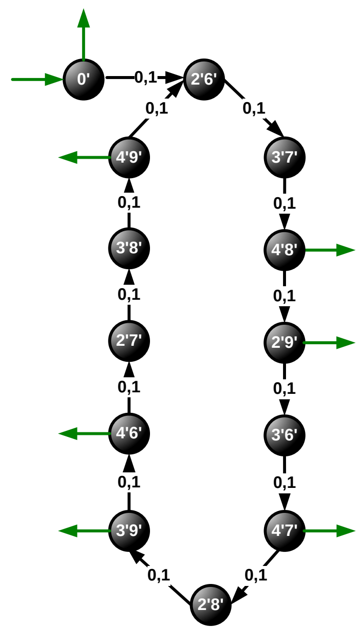
b) il est bien plus facile de déterminiser une version graphiquement simplifiée de cet automate :



Alors la déterminisation se fait, en utilisant la notation « prime » pour les ε-clôtures, ainsi :

	état	0 ou 1
E/S	0'	2'6'
	2'6'	3'7'
	3'7'	4'8'
S	4'8'	2'9'
S	2'9'	3'6'
	3'6'	4'7'
S	4'7'	2'8'
	2'8'	3'9'
S	3'9'	4'6'
S	4'6'	2'7'
	2'7'	3'8'
	3'8'	4'9'
S	4'9'	2'6'

(sont terminaux les états « prime » contenant 4' et/ou 9', ainsi que 0')



Minimisation : Θ_0={T, NT} où T={0' ,4'8',2'9',4'7',3'9',4'6',4'9'}, NT= {2'6',3'7',3'6',2'8',2'7',2'7'}

	état	0 ou 1	sous Θ_0
T	0'	2'6'	NT
T	4'8'	2'9'	T
T	2'9'	3'6'	NT
T	4'7'	2'8'	NT
T	3'9'	4'6'	T
T	4'6'	2'7'	NT
T	4'9'	2'6'	NT

NT	2'6'	3'7'	NT
NT	3'7'	4'8'	T
NT	3'6'	4'7'	T
NT	2'8'	3'9'	T
NT	2'7'	3'8'	NT
NT	3'8'	4'9'	T

Θ_1={A,B,C,D} où A={0',2'9',4'7',4'6',4'9'}, B={4'8',3'9'}, C={2'6',2'7'}, D={3'7',3'6',2'8',3'8'}

Itération 2 :

	état	0 ou 1	sous Θ_1
A	0'	2'6'	C
	2'9'	3'6'	D
	4'7'	2'8'	D
	4'6'	2'7'	C
	4'9'	2'6'	C
B	4'8'	2'9'	A
	3'9'	4'6'	A
C	2'6'	3'7'	D
	2'7'	3'8'	D
D	3'7'	4'8'	B
	3'6'	4'7'	A
	2'8'	3'9'	B
	3'8'	4'9'	A

Θ_2={E,F,B,C,G,H} où B et C sont les mêmes qu'auparavant, E={0',4'6',4'9'}, F={2'9',4'7'}, G={3'7',2'8'}, H={3'6',3'8'}

Itération 3 :

	état	0 ou 1	sous Θ_2
E	0'	2'6'	C
	4'6'	2'7'	C
	4'9'	2'6'	C
F	2'9'	3'6'	H
	4'7'	2'8'	G
B	4'8'	2'9'	F
	3'9'	4'6'	E
C	2'6'	3'7'	G
	2'7'	3'8'	H
G	3'7'	4'8'	B
	2'8'	3'9'	B
H	3'6'	4'7'	F
	3'8'	4'9'	E

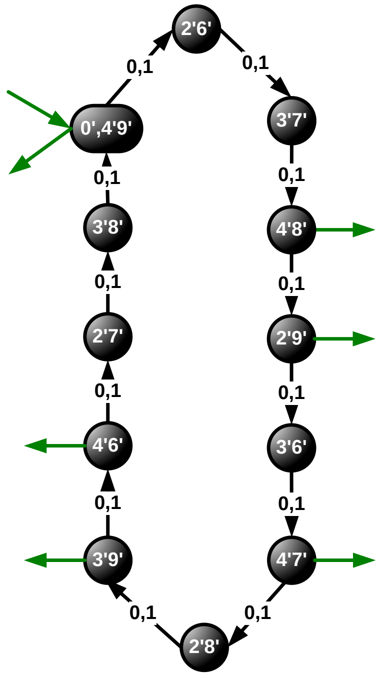
Θ_3={E, G,(2'9'),(4'7'),(4'8'),(3'9'),(2'6'),(2'7'),(3'6'),(3'8')} où E et G sont les mêmes qu'auparavant.

Itération 4, ne traitant que deux groupes de plus d'un état chacun :

	état	0 ou 1	sous Θ_3
E	0'	2'6'	2'6'
	4'6'	2'7'	2'7'
	4'9'	2'6'	2'6'
G	3'7'	4'8'	4'8'
	2'8'	3'9'	3'9'

Θ_4={(0',4'9'), (4'6'),(3'7'),(2'8'),(2'9'),(4'7'),(4'8'),(3'9'),(2'6'),(2'7'),(3'6'),(3'8')}

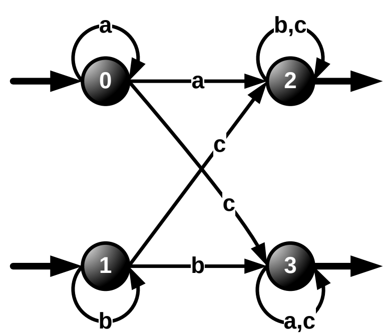
Les états 0' et 4'9' ne pourront jamais se séparer car ils ont la même transition au sein du même groupe.
Donc $\Theta_4 = \Theta_{fin}$. L'automate qui en résulte s'obtient à partir du résultats de la déterminisation en fusionnant les états 0' et 4'9' :



Comment obtenir ce résultat directement à partir de l'ER d'origine ?
$L=(c^3)^*+(c^4)^*$ où c=0+1
La longueur du cycle minimal divisible et par 3, et par 4 est 12= PPCM(3,4). (Si le nombre plus grand était multiple du nombre plus petit, cette règle serait modifiée, mais ce n'est pas le cas). Il y a donc 12 états dans l'automate minimal. Les sorties sont en positions qui sont multiples de 3 et/ou 4 : 0,3,4,6,8,9. On obtient l'automate présenté ci-dessus sans passer par la construction d'un automate asynchrone et sans déterminisation et minimisation.

Exercice 6 D'un automate fini vers une expression rationnelle

Trouver, soit par la méthode de l'arrivée, soit par celle d'élimination d'états, l'expression rationnelle représentant le langage reconnu par l'automate ci-contre :



Aucune solution basée sur « l'intuition » ne sera aceptée.

Solution

Méthode de l'arrivée :

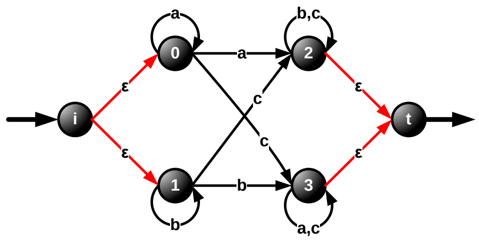
0 = ε + 0a =0* (Lemme d'Arden)
1 = ε + 1b =b* (Lemme d'Arden)
2 = 0a + 1c + 2(b+c)
3 = 0c + 1b + 3(a+c)
L=2+3

En mettant la solution pour 0 et pour 1 dans les équations pour 2 et pour 3, on obtient
2 = a*a +b*c + 2(b+c) = (a*a +b*c)(b+c)* (Lemme d'Arden)
3 = a*c + b*b + 3(a+c) =(a*c + b*b)(a+c)* (Lemme d'Arden)

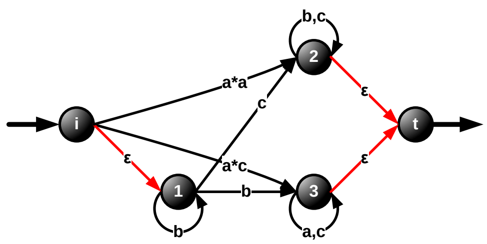
En résultat, L = (a*a +b*c)(b+c)* +(a*c + b*b)(a+c)*

Methode d'élimination d'états

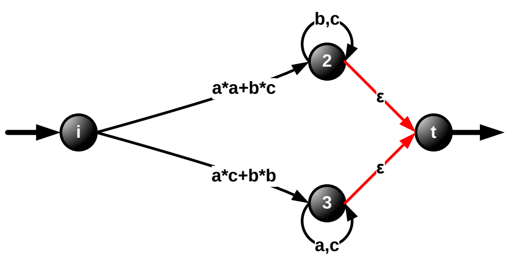
Initialisation **:**



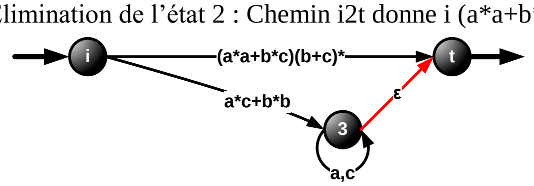
Elimination de l'état 0 : Chemins : i02, i03 donnent i εa*a2=ia*a2, i ε a*c3=ia*c3



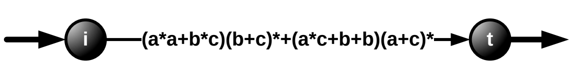
Elimination de l'état 1 : Chemins : i12, i13 donnent i εb*c2=ib*c2, i ε b*b3=ib*b3



Elimination de l'état 2 : Chemin i2t donne i (a*a+b*c)(b+c)* εt = i (a*a+b*c)(b+c)* t



Elimination de l'état 3 : Chemin i3t donne i (a*c+b*b)(a+c)* εt = i (a*c+b*b)(a+c)* t qui s'ajoute à l'expression déjà figurant sur la flèche de i à t :



Nous obtenons le même résultat qu'auparavant : L = (a*a +b*c)(b+c)* +(a*c + b*b)(a+c)*

6 : Questions de cours

1. Quel langage reconnaît un automate fini déterministe complet dont tous les états sont des états terminaux ?

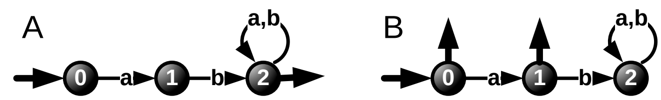
L=A* où A : l'alphabet.

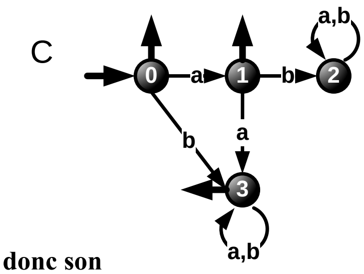
2. Quelle erreur risque-t-on de commettre si on ne vérifie pas qu'un automate déterministe soit complet avant de le transformer en un automate reconnaissant le complémentaire du langage ? Formuler la réponse en donnant des exemples des cas où une telle erreur se produirait et en décrivant l'erreur.

On perdra tous les mots qui, pour l'automate d'origine, aboutissaient dans la poubelle et qui, pour l'automate complémentaire, doivent faire partie du langage.

Par exemple, dans l’exo 9 du TD, si on oublie d’ajouter la poubelle et qu’on fait de tous les états terminaux, non terminaux et vice versa, on obtiendra un automate qui ne reconnaitra pas des mots à 5, 6 etc. de zéros.

3. Pour un langage donné, il y a un seul automate fini le reconnaissant (**Oui/Non)** **NON**
4. Pour un langage donné, il y a un seul automate déterministe le reconnaissant (**Oui/Non)** **NON**
5. Pour un langage donné, il y a un seul automate déterministe complet le reconnaissant (**Oui/Non)** **NON**
6. Pour un langage donné, il y a un seul automate minimal le reconnaissant (**Oui/Non**) **OUI**
7. Soit l’automate A.
   a) L’automate B est-il l’automate reconnaissant le complément $\overline{L}$ du langage L reconnu par A ? Expliquer votre réponse (pas de points s’il n’y a pas d’explication correcte). Si la réponse est « non », essayez de trouver un mot qui fait partie de $\overline{L}$ mais qui n’est pas reconnu par l’automate B, ou un mot qui ne fait pas partie de $\overline{L}$ mais qui est reconnu par B.





   Non, car A n’est pas complet. Par exemple, A ne reconnait pas b, donc son complémentaire dois le reconnaitre, mais l’automate B ne le reconnait pas.

   b) Même question pour l’automate C.

   Oui, car C est l’automate complémentaire à celui qu’on obtient en complétant A.