•

## Exercice 1 Déterminisation

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

| | état | a | b |
|---|---|---|---|
| E/S | 1 | 1, 5 | 2 |
| E | 2 | 1, 3 | - |
| | 3 | 2, 3, 5 | - |
| S | 4 | 4 | - |
| | 5 | 4 | 1, 3, 5 |

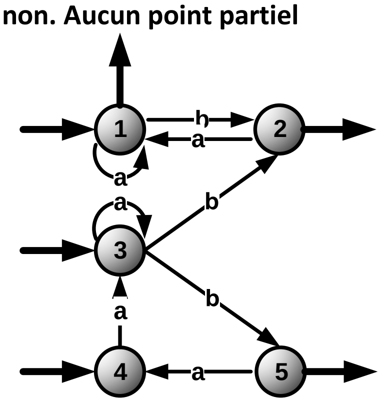
Construire un automate déterministe complet équivalent à cet automate. Le résultat peut être présenté sous forme de schéma ou sous forme de tableau de transitions ou les deux, <u>avec toutes les entrées/sorties indiquées</u>.

**Solution**

| | état | a | b |
|---|---|---|---|
| E/S | 12 | 135 | 2 |
| S | 135 | 12345 | 1235 |
| | 2 | 13 | P |
| S | 13 | 1235 | 2 |
| S | 1235 | 12345 | 1235 |
| S | 12345 | 12345 | 1235 |
| | P | P | P |

## Exercice 2 Standardisation (**Le résultat sera classifié comme correct ou non. Aucun point partiel pour cet exo).**

a) Obtenir un automate standard $A_2$ reconnaissant le langage reconnu par l'automate ci-contre :



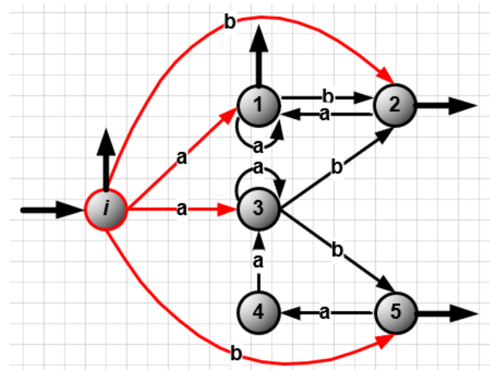

Le résultat doit être présenté sous forme de schéma avec toutes les entrées/sorties indiquées.

b) L'automate obtenu reconnait-il le mot vide ?

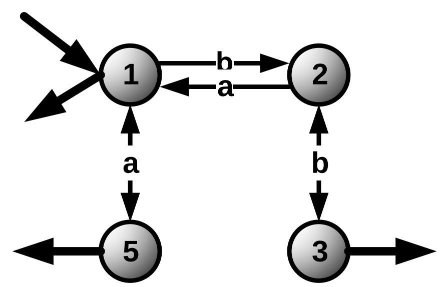
**Solution** **Les transitions**

| | | |
|---|---|---|
| 1a1 | | ia1 |
| 1b2 | | ib2 |
| 3a3 | engendrent | ia3 |
| 3b2 | | ib5 |
| 3b5 | | |
| 4a3 | | |

L'état *i* est terminal car l'état 1 est une E/S
Donc l'automate obtenu reconnait le mot vide




## Exercice 3 Langage complémentaire

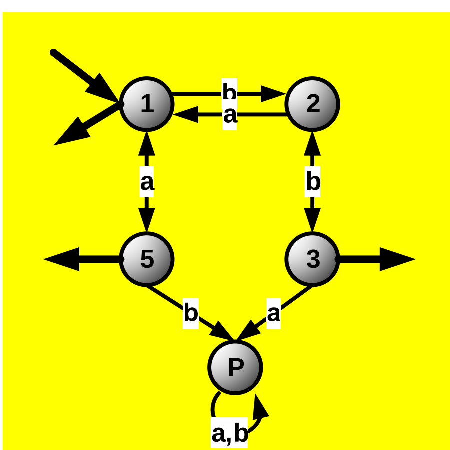



Construire un automate reconnaissant le langage complémentaire à celui reconnu par l'automate ci-contre.

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.

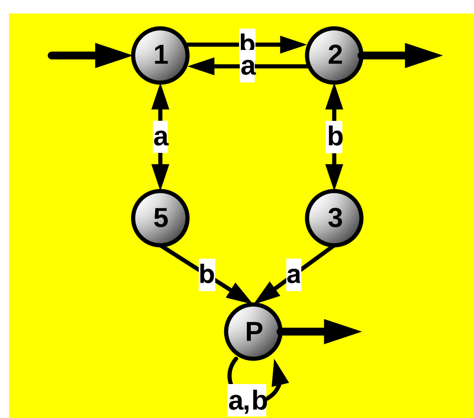
**Solution**

Cet automate est déterministe mais non complet. Complétons-le, et puis transformons tous les états terminaux en non terminaux et vice versa :




ADC

ADC complémentaire



## Exercice 4 Minimisation et le langage complémentaire

A = {a,b} est l'alphabet.

Pour l'automate défini par la table de transitions ci-dessous :

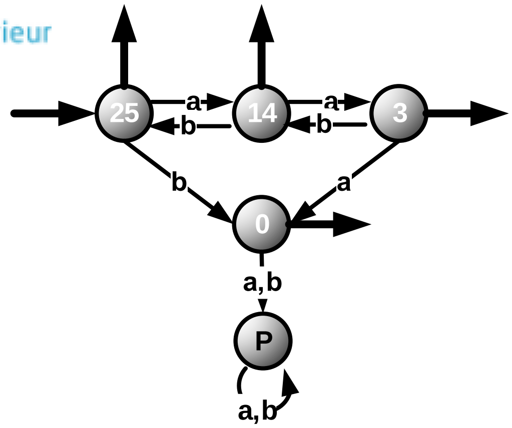
| | état | a | b |
|---|---|---|---|
| S | 0 | -- | -- |
| S | 1 | 3 | 5 |
| E/S | 2 | 4 | 0 |
| S | 3 | 0 | 1 |
| S | 4 | 3 | 5 |
| S | 5 | 4 | 0 |

a) Construire un automate <u>déterministe complet minimal</u> équivalent.

**Solution :**

| | | | | | |
|---|---|---|---|---|---|
| | $\Theta_0$={T, NT} où T={0,1,2,3,4,5}, NT={P} | | | | |
| | **Itération 1:** | | | sous $\Theta_0$ | |
| | état | a | b | a | b |
| T | 0 | P | P | NT | NT |
| T | 1 | 3 | 5 | T | T |
| T | 2 | 4 | 0 | T | T |
| T | 3 | 0 | 1 | T | T |
| T | 4 | 3 | 5 | T | T |
| T | 5 | 4 | 0 | T | T |
| NT | P | P | P | | |
| | $\Theta_1$={A, (0), (P)} où A={1,2,3,4,5} | | | | |
| | **Itération 2:** | | | sous $\Theta_1$ | |
| | état | a | b | a | b |
| A | 1 | 3 | 5 | A | A |
| A | 2 | 4 | 0 | A | 0 |
| A | 3 | 0 | 1 | 0 | A |
| A | 4 | 3 | 5 | A | A |
| A | 5 | 4 | 0 | A | 0 |
| | $\Theta_2$={(1,4), (2,5), (3), (0), (P)} | | | | |
| | **Itération 3:** | | | sous $\Theta_2$ | |
| | état | a | b | a | b |
| | 1 | 3 | 5 | 3 | 5 |
| | 4 | 3 | 5 | 3 | 5 |
| | 2 | 4 | 0 | 4 | 0 |
| | 5 | 4 | 0 | 4 | 0 |
| | $\Theta_3=\Theta_2=\Theta_{fin}$ | | | | |

| | | | |
|---|---|---|---|
| | Les états de l'AM: 14, 25, 0, 3, P | | |
| | L'entrée: | 25 | |
| | Les sorties: tous sauf P | | |
| | Les transitions: | | |
| | état | a | b |
| E/S | 25 | 14 | 0 |
| S | 14 | 3 | 25 |
| S | 0 | P | P |
| S | 3 | 0 | 14 |
| | P | P | P |

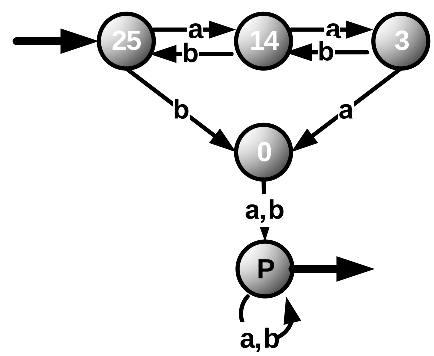



**REMARQUE** Un nombre d'élèves ont formé le « groupe » (0,P). C'est complètement impossible car 0 est un état terminal, et P ne l'est pas. Je vous ai dit plusieurs fois qu'il convient de nettement séparer les groupes à n'importe quelle itération les uns des autres pour NE JAMAIS les remélanger.

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

**Solution :**

| | état | a | b |
|---|---|---|---|
| E | 25 | 14 | 0 |
| | 14 | 3 | 25 |
| | 0 | P | P |
| | 3 | 0 | 14 |
| S | P | P | P |



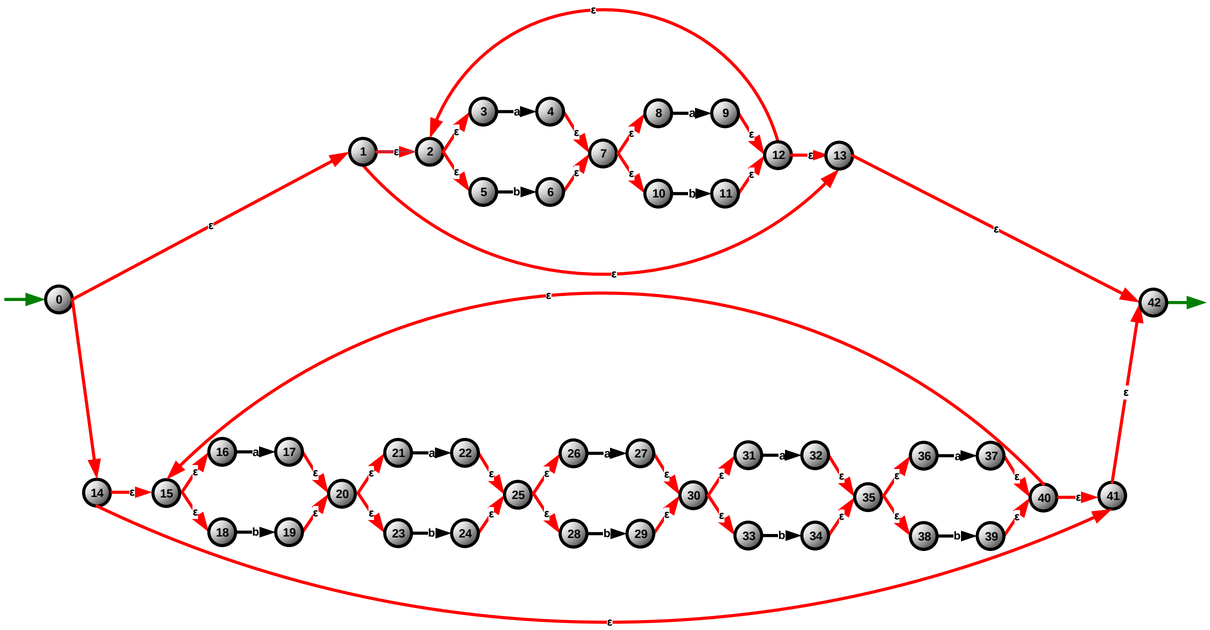

## Exercice 5 D'une expression rationnelle vers un automate fini

a) Construire un automate reconnaissant le langage

$$L = \big((a+b)(a+b)\big)^* + \big((a+b)(a+b)(a+b)(a+b)(a+b)\big)^*$$

<u>suivant les règles formelles données dans le cours</u>.

**Solution (a) :**



**REMARQUE**

Défauts constatés :

- Inversion des directions des flèches en ε directes et inverses
- Des flèches en ε placées entre les hexagones, ce qui viole la règle de concaténation
- Deux sorties au lieu d'une seule
- La remplacement des flèches 1- 13 et 14—41 par une seule flèche 0 – 42 ce qui viole la modularité

Puis, au choix,

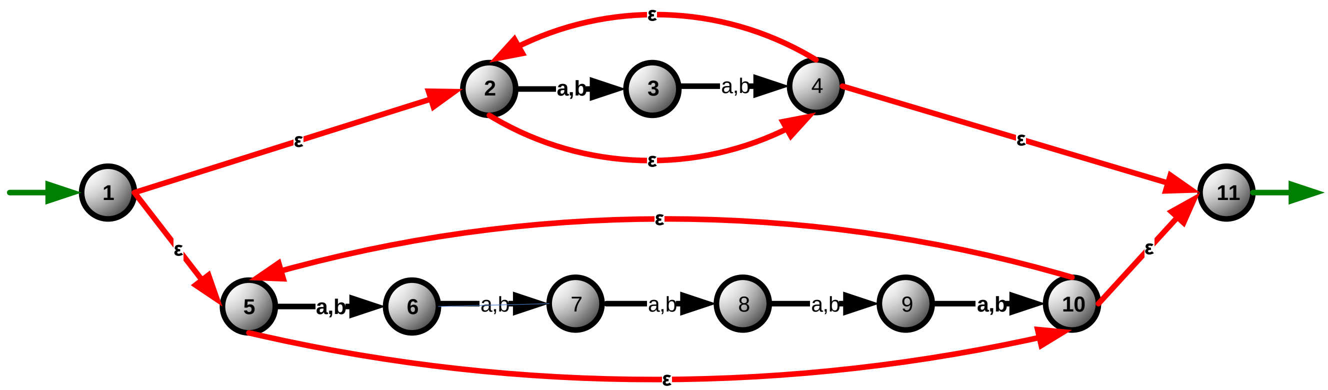
soit b) déterminiser et
c) minimiser } l'automate obtenu en (a) (c'est assez compliqué !),

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).

**Solution (b) :**

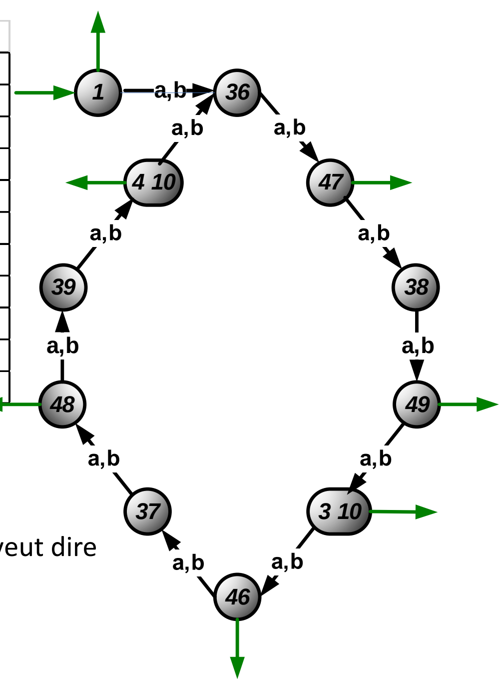
**Je choisis de travailler avec un automate graphiquement simplifié suivant :**




Déterminisons-le.

Les ε-clôtures à utiliser :
1' = 1 2 4 5 10 11 (terminal)
3'=3
4'=2 4 11 (terminal)
6'=6, 7'=7, 8' = 8, 9'=9
10' = 5 10 11 (terminal)

| | | a+b |
|---|---|---|
| E/S | 1' | 3'6' |
| | 3'6' | 4'7' |
| S | 4'7' | 3'8' |
| | 3'8' | 4'9' |
| S | 4'9' | 3'10' |
| S | 3'10' | 4'6' |
| S | 4'6' | 3'7' |
| | 3'7' | 4'8' |
| S | 4'8' | 3'9' |
| | 3'9' | 4'10' |
| S | 4'10' | 3'6' |




(Sur le dessin, je n'ai pas mis les primes, de façon que, par exemple, 4 10 veut dire 4'10')

## Solution (c) (minimisation) :

$\Theta_0$={T, NT} où T={1',4'7',4'9',3'10',4'6',4'8',4'10'}, NT={3'6',3'8',3'7',3'9'}

| | **itération 1** | | sous $\Theta_0$ | |
|---|---|---|---|---|
| | | a+b | a+b | |
| T | 1' | 3'6' | NT | |
| T | 4'7' | 3'8' | NT | |
| T | 4'9' | 3'10' | NT | |
| T | 3'10' | 4'6' | T | |
| T | 4'6' | 3'7' | NT | |
| T | 4'8' | 3'9' | NT | |
| T | 4'10' | 3'6' | NT | |
| NT | 3'6' | 4'7' | T | |
| NT | 3'8' | 4'9' | T | |
| NT | 3'7' | 4'8' | T | |
| NT | 3'9' | 4'10' | T | |

$\Theta_1$={A, (3'10'), NT} où A={1',4'7',4'9',4'6',4'8',4'10'}, NT={3'6',3'8',3'7',3'9'}

| | **itération 2** | | sous $\Theta_1$ | |
|---|---|---|---|---|
| | | a+b | | |
| A | 1' | 3'6' | NT | |
| A | 4'7' | 3'8' | NT | |
| A | 4'9' | 3'10' | 3'10' | |
| A | 4'6' | 3'7' | NT | |
| A | 4'8' | 3'9' | NT | |
| A | 4'10' | 3'6' | NT | |
| NT | 3'6' | 4'7' | A | |
| NT | 3'8' | 4'9' | A | |
| NT | 3'7' | 4'8' | A | |
| NT | 3'9' | 4'10' | A | |

$\Theta_2$={B, (3'10'), (4'9'), NT} où B={1',4'7',4'6',4'8',4'10'}, NT={3'6',3'8',3'7',3'9'}

| | **itération 3** | | sous $\Theta_2$ | |
|---|---|---|---|---|
| | | a+b | | |
| B | 1' | 3'6' | NT | |
| B | 4'7' | 3'8' | NT | |
| B | 4'6' | 3'7' | NT | |
| B | 4'8' | 3'9' | NT | |
| B | 4'10' | 3'6' | NT | |
| NT | 3'6' | 4'7' | B | |
| NT | 3'8' | 4'9' | 4'9' | |
| NT | 3'7' | 4'8' | B | |
| NT | 3'9' | 4'10' | B | |

$\Theta_3$={B, (3'10'), (4'9'), (3'8'), C} où B={1',4'7',4'6',4'8',4'10'}, C={3'6',3'7',3'9'}

| | **itération 4** | | sous $\Theta_3$ | |
|---|---|---|---|---|
| | | a+b | | |
| B | 1' | 3'6' | C | |
| B | 4'7' | 3'8' | 3'8' | |
| B | 4'6' | 3'7' | C | |
| B | 4'8' | 3'9' | C | |
| B | 4'10' | 3'6' | C | |
| C | 3'6' | 4'7' | B | |
| C | 3'7' | 4'8' | B | |
| C | 3'9' | 4'10' | B | |

$\Theta_4$={D, (3'10'), (4'9'), (3'8'), (4'7'), C} où D={1',4'6',4'8',4'10'}, C={3'6',3'7',3'9'}

| | **itération 5** | | sous $\Theta_4$ | |
|---|---|---|---|---|
| | | a+b | | |
| D | 1' | 3'6' | C | |
| D | 4'6' | 3'7' | C | |
| D | 4'8' | 3'9' | C | |
| D | 4'10' | 3'6' | C | |
| C | 3'6' | 4'7' | 4'7' | |
| C | 3'7' | 4'8' | D | |
| C | 3'9' | 4'10' | D | |

$\Theta_5$={D, (3'10'), (4'9'), (3'8'), (4'7'), (3'6'), E} où D={1',4'6',4'8',4'10'}, E={3'7',3'9'}

| | **itération 6** | | sous $\Theta_5$ |
|---|---|---|---|
| | | a+b | |
| D | 1' | 3'6' | 3'6' |
| D | 4'6' | 3'7' | E |
| D | 4'8' | 3'9' | E |
| D | 4'10' | 3'6' | 3'6' |
| E | 3'7' | 4'8' | D |
| E | 3'9' | 4'10' | D |

$\Theta_6$={(1',4'10'), (4'6', 4'8'), (3'10'), (4'9'), (3'8'), (4'7'), (3'6'), E} où E={3'7',3'9'}

| | **itération 7** | | sous $\Theta_6$ | |
|---|---|---|---|---|
| | | a+b | | |
| | 1' | 3'6' | 3'6' | |
| | 4'10' | 3'6' | 3'6' | |
| | 4'6' | 3'7' | E | |
| | 4'8' | 3'9' | E | |
| E | 3'7' | 4'8' | (4'6',4'8') | |
| E | 3'9' | 4'10' | (1',4'10') | |

$\Theta_7$={(1',4'10'), (4'6'), (4'8'), (3'10'), (4'9'), (3'8'), (4'7'), (3'6'), (3'7'),(3'9')}

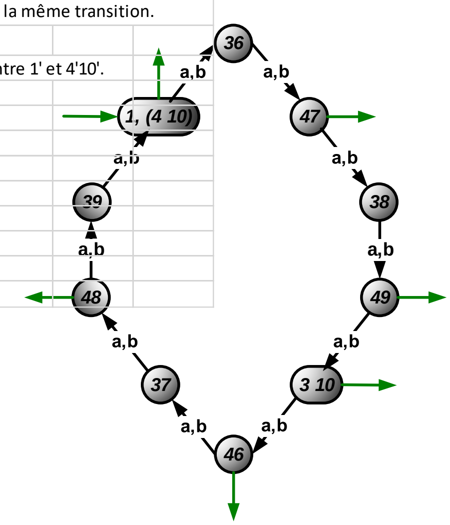
| | **itération 8** | | sous $\Theta_7$ | |
|---|---|---|---|---|
| | | a+b | | |
| | 1' | 3'6' | 3'6' | |
| | 4'10' | 3'6' | 3'6' | |
| | 4'6' | 3'7' | 3'7' | |
| | 4'8' | 3'9' | 3'9' | |

$\Theta_8$={(1',4'10'), (4'6', 4'8'), (3'10'), (4'9'), (3'8'), (4'7'), (3'6'), (3'7'),(3'9')}

Les états 1' et 4'10' ne peuvent pas se séparer car ils ont la même transition.

Donc $\Theta_9=\Theta_8=\Theta_{fin}$

Le seul effet de la minimisation consiste en la fusion entre 1' et 4'10'.

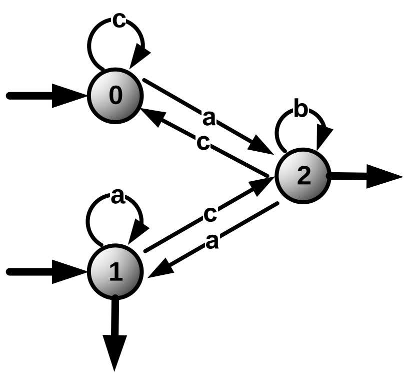


**Solution (b+c) immédiate, remplaçant la déterminisation et la minimisation précédentes :**

Nous avons une ER $E=(c^2)^*+(c^5)^*$ où $c=a+b$. Le langage consiste donc en tous les mots sur l'alphabet {a,b} de longueur multiple de 2 ou multiple de 5. Le cycle minimal accommodant de telles longueurs est le PPCM(2,5)=2*5=10. L'automate minimal consiste en 10 états. Les sorties sont situées après des longueurs multiples de 2 ou 5, donc en 0,2,4,5,6,8. On obtient l'automate figurant ci-dessus.

## Exercice 6 D'un automate fini vers une expression rationnelle

Obtenir, sous forme d'une expression rationnelle, le langage reconnu par l'automate suivant, soit par la méthode de l'arrivée (équations), soit par celle d'élimination d'états.




Vous devez indiquer clairement les résultats de chaque étape du traitement effectué.
Aucune réponse obtenue « intuitivement » ne sera considérée.

**Solution par la méthode de l'arrivée :**

$$\begin{cases} 0 = \varepsilon + 0c + 2c & \text{(eq 0)} \\ 1 = \varepsilon + 1a + 2a & \text{(eq 1)} \\ 2 = 0a + 2b + 1c & \text{(eq 2)} \end{cases}$$

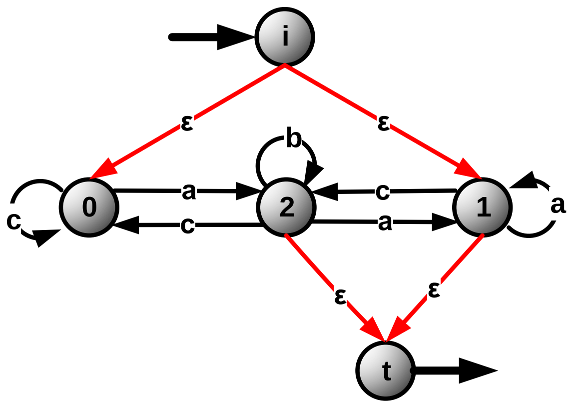
$L = 1 + 2$

Eq 0 : $0 = (\varepsilon + 2c) + 0c$ donc le Lemme d'Arden donne $0 = (\varepsilon + 2c)c^* = c^* + 2cc^*$
Eq 1 : $1 = (\varepsilon + 2a) + 1a$ donc le Lemme d'Arden donne $1 = (\varepsilon + 2a)a^* = a^* + 2aa^*$
Remplaçant ceci dans Eq 2 :
$2 = (c^* + 2cc^*)a + 2b + (a^* + 2aa^*)c = c^*a + a^*c + 2(cc^*a+b+aa^*c)$, et le Lemme d'Arden donne
$2= (c^*a + a^*c)(cc^*a+b+aa^*c)^*$
d'où $1 = a^* + 2aa^* = a^* + (c^*a + a^*c)(cc^*a+b+aa^*c)^*aa^*$

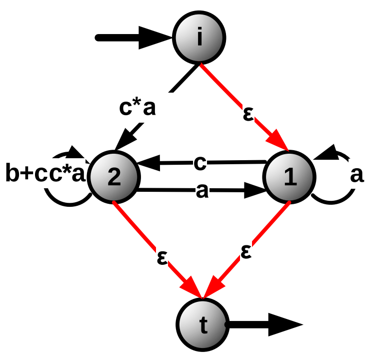
Donc $L = 1 + 2 = (c^*a + a^*c)(cc^*a+b+aa^*c)^*aa^* + a^* + (c^*a + a^*c)(cc^*a+b+aa^*c)^*$
$= a^* + (c^*a + a^*c)(cc^*a+b+aa^*c)^*(aa^* + \varepsilon) = a^* + (c^*a + a^*c)(cc^*a+b+aa^*c)^*a^*$

**Solution par la méthode d'élimination d'états**

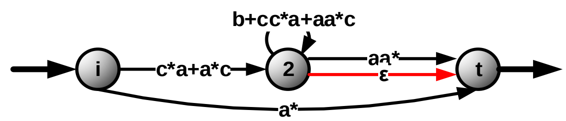
Initialisation :




Elimination de 0 : les chemins i02 et 202 donnent ia*a2 et 2cc*a2, donc la boucle sur 2 devient b+cc*a

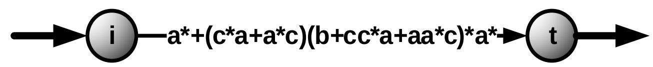



Elimination de 1 : les chemins i1t, i12, 11t, 212 donnent ia*t, ia*c2, iaa*t, 2aa*c2




(les deux flèches parallèles en aa* et ε sont équivalentes à une fleche en aa*+ ε, or aa*+ ε=a*, donc la fleche de 1 vers t est libellee a*)

Elimination de 2 : le seul chemin i2t donne (c*a+a*c)(b+cc*a+aa*c)*a*. La fleche déjà existante en a* s'y ajoute, et on obtient

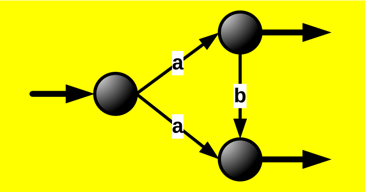



L=a*+(c*a + a*c) (cc*a+aa*c+b)*a*

## Questions de cours

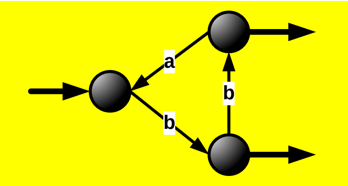
1. Un automate standard est-il obligatoirement déterministe ? Si la réponse est non, donner un exemple d'automate standard non déterministe. Si la réponse est oui, expliquer pourquoi. Sans l'exemple/explication, aucun point.

   Non, voici un exemple :


2. Un automate déterministe est-il obligatoirement standard ? Si la réponse est non, donner un exemple d'automate déterministe non standard. Si la réponse est oui, expliquer pourquoi. Sans l'exemple/explication, aucun point.

   Non, voici un exemple :

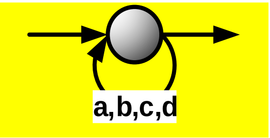

3. Un automate complet contient-il toujours un état poubelle ? **NON**
4. Peut-il y avoir plusieurs états initiaux dans un automate minimal ? **NON**
5. Peut-il y avoir plusieurs états terminaux dans un automate minimal ? **OUI**
6. Peut-il y avoir des états non accessibles dans un automate minimal ? **NON. S'il y a une état non accessible, il peut être toujours éliminé de façon à ce que l'automate reste équivalent, et l'automate deviendra plus petit.**
7. Peut-il y avoir des états non coaccessibles dans un automate minimal ? **OUI, l'état poubelle s'il est présent**

8. Quel est l'automate minimal équivalent à l'automate suivant :

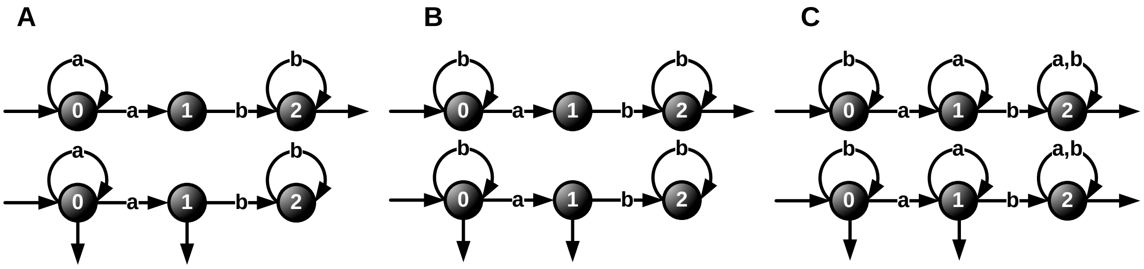
| | état | a | b | c | d |
|---|---|---|---|---|---|
| E/S | 0 | 1 | 2 | 3 | 0 |
| S | 1 | 2 | 3 | 0 | 1 |
| S | 2 | 3 | 0 | 1 | 2 |
| S | 3 | 0 | 1 | 2 | 3 |

Donner une réponse immédiate, <u>**sans effectuer la procédure de minimisation par des partitions successives,**</u> mais avec une explication (qui peut se tenir dans une ou deux phrases) !

**C'est un ADC dont tous les états sont terminaux. Donc il reconnait A*, et son minimal consiste en un seul état :**




9. En disant qu'un automate est minimal, on parle de quelle quantité ? **Du nombre d'états**
10. Pour un langage donné, il y a un seul automate fini le reconnaissant (Oui/Non**)** **NON**
11. Pour un langage donné, il y a un seul automate déterministe le reconnaissant (Oui/Non) **NON**
12. Pour un langage donné, il y a un seul automate déterministe complet le reconnaissant (Oui/Non) **NON**
13. Pour un langage donné, il y a un seul automate minimal le reconnaissant (Oui/Non) **OUI**
14. Pour chacun des couples d'automates A, B et C, dire si l'automate en haut reconnaît le langage complémentaire à celui reconnu par l'automate en bas. <u>Expliquer le oui et le non en phrases courtes</u>. Cette question vous donne des points uniquement si toutes les trois réponses sont bonnes.




A : non, car l'automate du haut n'est pas déterministe. Par exemple, l'automate du haut ne reconnait pas le mot 'b', mais l'automate du bas, non plus. Les deux langages ne sont pas complémentaires.

B : non, car l'automate du haut n'est pas complet. Par exemple, l'automate du haut ne reconnait pas le mot 'aa', mais l'automate du bas, non plus. Les deux langages ne sont pas complémentaires.

C : non, car 2 est sortie pour les deux automates. Par exemple, l'automate du haut reconnait 'ab', mais l'automate du bas, lui aussi le reconnait. Les deux langages ne sont pas complémentaires.