Chazal
Baptiste

Maths pour l'info 1/5

Automate déterministe :

- une seule entrée
- aucun état d'où sort plus d'une flèche libellée par le même caractère

Automate standard :

- une seule entrée à laquelle aucune transition n'aboutit

Standardiser :

- on liste les transitions partant des anciennes entrées (0a0, 0b2, 1a3, etc...)
- on ajoute une entrée (i) qui sera la seule entrée (terminale si l'automate reconnait ε)
- les transitions de la première étape doivent désormais partir de (i) (ia0, ib2, ia3, etc...)

A^* : ensemble de tous les mots sur l'alphabet A, plus le mot vide.
A est un alphabet, il consiste en un nombre de caractères finis.
A^* est un ensemble infini de mots.

$\bar{L}$: langage complémentaire du langage L, soit tous les mots sur l'alphabet A non appartenant à L.
$\bar{L} = A^* \setminus L$.

Automate déterministe complet :

- automate complet
- il ne doit pas manquer de transition dans le tableau de transitions. S'il en manque, on rajoute un état poubelle.

		a	b
E	1	2	4
	2	2	3
S	3	6	/
	4	5	/
	5	/	6
S	6	/	/

$\xrightarrow{\text{complétion}}$

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

1. Un état est accessible s'il y a un chemin menant vers cet état depuis une entrée.

2. Un état est coaccessible s'il y a un chemin menant depuis cet état vers une sortie.

3. Un automate est accessible si tous les états sont accessibles.

4. Un automate est coaccessible si tous les états sont coaccessibles.

5. Accessible + Coaccessible = émondé

Complémentarisation :

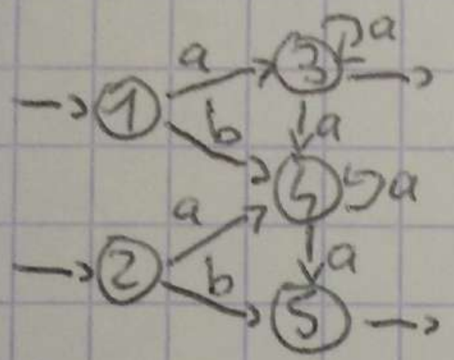
⚠ L'automate doit être un ADC !

Si un automate A_1 reconnait le langage L et qu'il s'agit d'un ADC, alors on peut créer l'automate A_2 reconnaissant le langage $\overline{L}$.

Il suffit de transformer les états terminaux en non terminaux, et vice-versa.

Déterminisation :

1. On a un AND :



2. On établit le tableau de transitions :

		a	b
E	1	3	4
E	2	4	5
S	3	3,4	/
	4	4,5	/
S	5	/	/

3. On regroupe les entrées :

		a	b
E	12	34	45

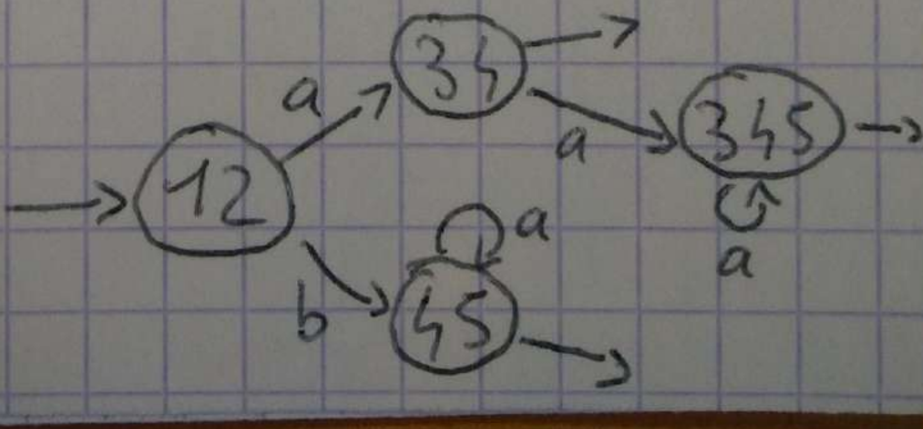
4. Chaque nouvel état apparent est une nouvelle ligne du nouveau tableau :

		a	b
E	12	34	45
	34	345	/
	45	45	/
	345	345	/

5. Les états héritants d'états terminaux (i.e. 3 et 5) sont terminaux :

		a	b
E	12	34	45
S	34	345	/
S	45	45	/
S	345	345	/

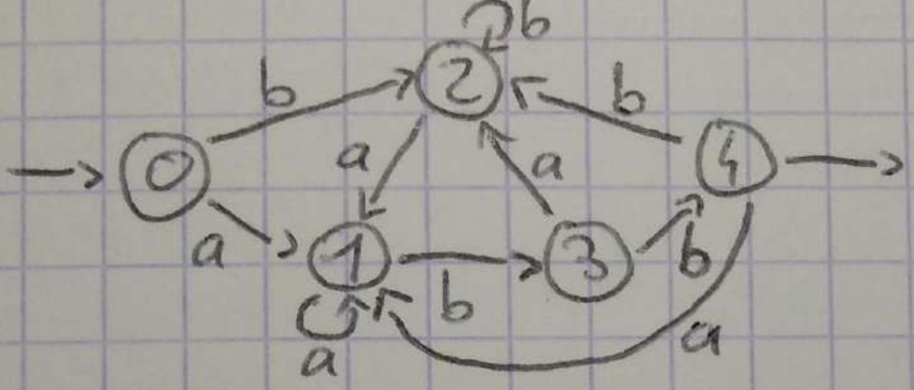
6. On obtient un AD :



Maths pour l'info 3/5

Minimisation :

1. L'automate à minimiser doit être un ADC :



2. On partitionne tous les états en terminaux et nonterminaux :
$\Theta_0 = \{T = \{4\}, NT = \{0, 1, 2, 3\}\}$.

3. Pour les états encore groupés $(0, 1, 2, 3)$, on dresse un tableau de transitions. On les sépare en sous-groupes en fonction de leurs transitions (si l'état d'arrivée est T ou NT) :

	a	b			
0	1	2	NT	NT	sous-groupe
1	1	3	NT	NT	
2	1	2	NT	NT	
3	2	4	NT	T	sous-groupe

$$\Theta_1 = \{(0, 1, 2), (3), (4)\}$$

4. On réitère l'étape 3 tant qu'il le faut :

	a	b	en termes de Θ_1	
0	1	2	012	012
1	1	3	012	3
2	1	2	012	012

Pour les transitions de chaque tableau, on utilise les groupes du Θ précédent (à partir de la recherche de Θ_2 et après)

$\Theta_2 = \{(02), (1), (3), (4)\}$.

5. On atteint Θ_{fin} lorsqu'il n'y a plus de séparation.

	a	b	en termes de Θ_2	
0	1	2	1	02
2	1	2	1	02

} pas de séparation

$\Theta_3 = \Theta_2 = \Theta_{fin}$

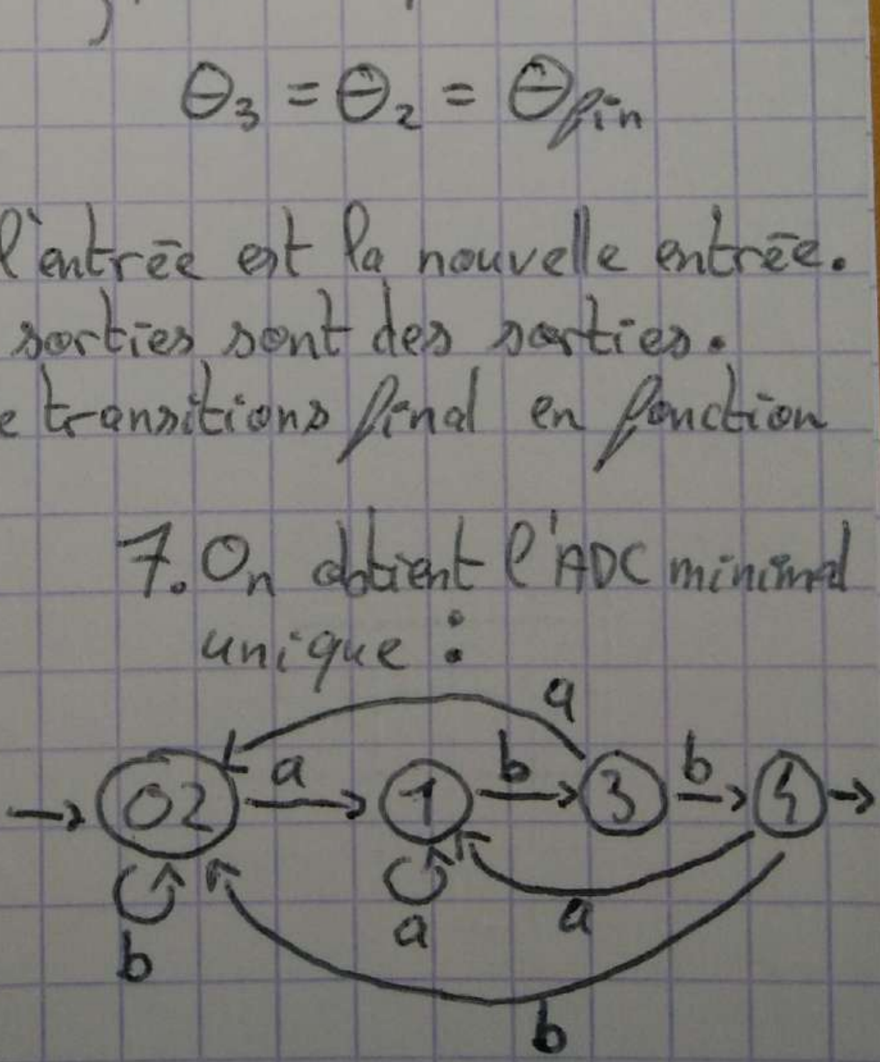
6. L'état héritant de l'entrée est la nouvelle entrée. Les états héritant de sorties sont des sorties. On dresse le tableau de transitions final en fonction de celui d'origine :

		a	b
E	0	1	2
	1	1	3
	2	1	2
	3	2	4
S	4	1	2

$\rightarrow$

	a	b
02	1	02
1	1	3
02	1	02
3	02	4
4	1	02

7. On obtient l'ADC minimal unique :



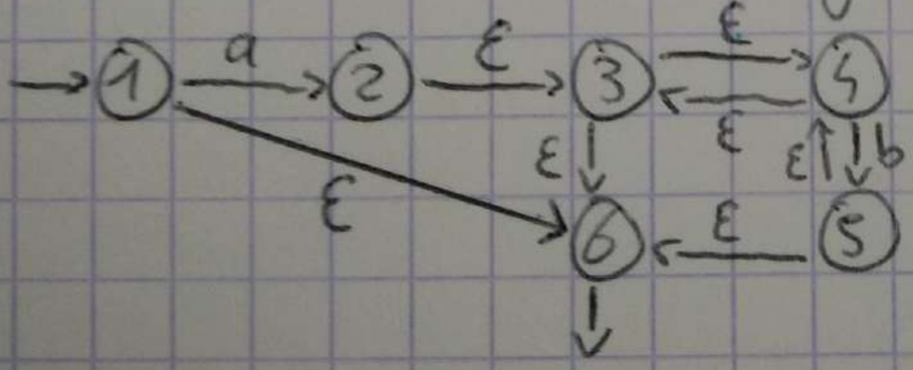
Automate asynchrone :

- un automate asynchrone est un automate possédant des ε-transitions, des transitions étiquetées par le mot vide.

- tout automate asynchrone est équivalent à un AD

Déterminisation d'un automate asynchrone :

1. On a un automate asynchrone :



2. On identifie l'état initial, soit l'état que l'on atteint en ne lisant que le mot vide :

On a (1) et (6).
Ils formeront l'état composé (1,6).

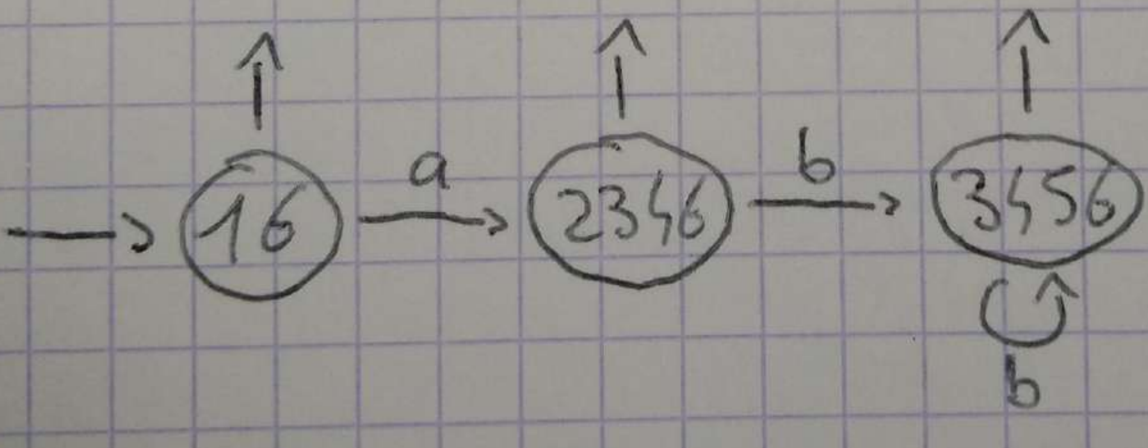
3. On détermine les transitions à partir de cet état initial :

En lisant b : /
En lisant a : 2, 3, 4, 6
(puisque $a = \varepsilon a = \varepsilon\varepsilon a = \dots$)

4. On procède ensuite comme toute déterminisation :

		a	b
E/S	16	2346	/
S	2346	/	3456
S	3456	/	3456

5. On obtient un AD :



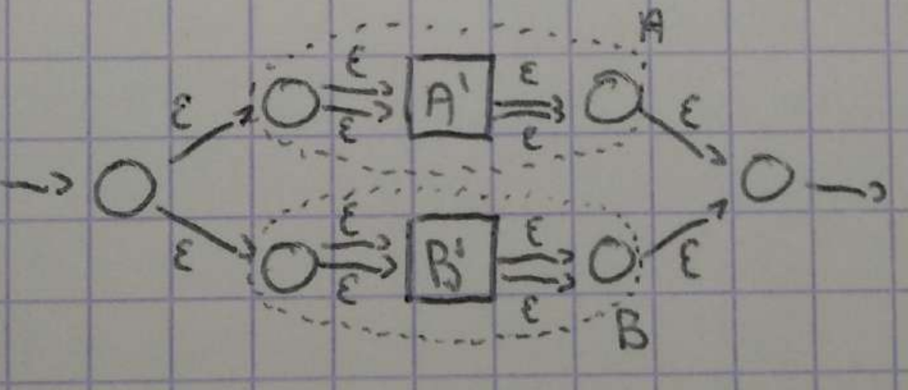
Maths pour l'info 5/5

Expression rationnelle :

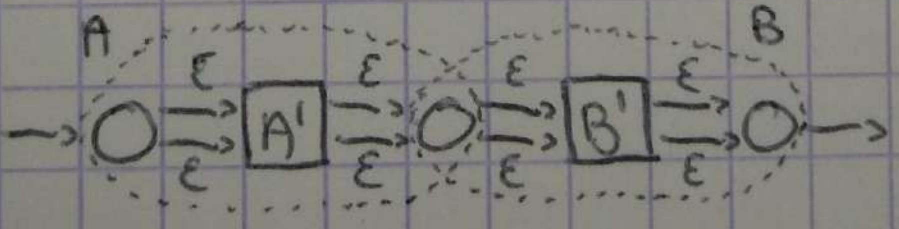
Théorème de Kleene : Un langage est reconnaissable par un AF ssi il peut être décrit par une expression rationnelle.

1. Dans tout état entrent deux flèches maximum.
2. De tout état sortent deux flèches maximum.
3. Tout automate a un unique état initial.
4. Tout automate a un unique état terminal.

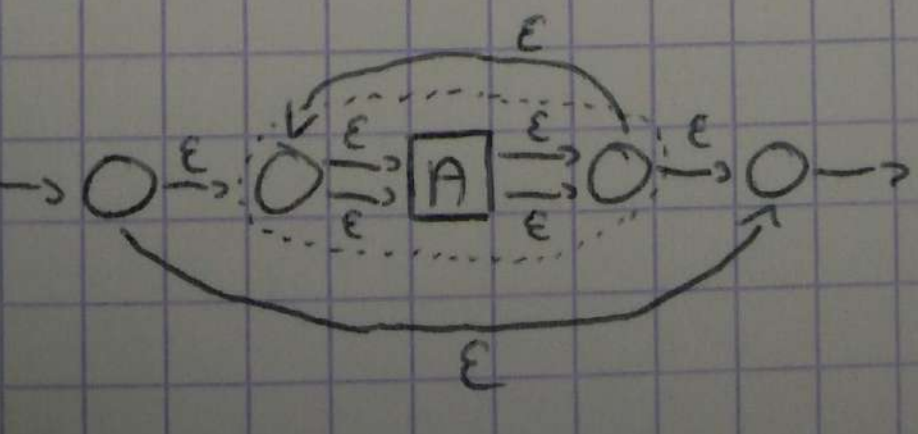
X+Y :



XY :



X* :



Par exemple $X = (ab + ba)^*$ donne :