Année universitaire : 2015/2016
Université de Batna
Département d'informatique
2ème année licence informatique
Module : Théorie des langages

TD N° 03 : correction
Les expressions régulières, les langages et les automates d'états fins

Exercice 01 :

1) Proposer une grammaire pour chacun des langages représentés par les expressions régulières suivantes :

☞ $(a/b)^*$

$S \to aS/bS/\varepsilon$

☞ a/b^*c

$S \to a/Bc$
$B \to Bb/\varepsilon$

Ou

$S \to a/B$
$B \to bB/c$

☞ $(a/b)^*abb(a/b)^*$

$S \to TabbT$
$T \to aT/bT/\varepsilon$

Ou

$S \to aS/bS/abbT$
$T \to aT/bT/\varepsilon$

☞ $b^*ab^*ab^*ab^*$

$S \to BaBaBaB$
$B \to bB/\varepsilon$

Ou

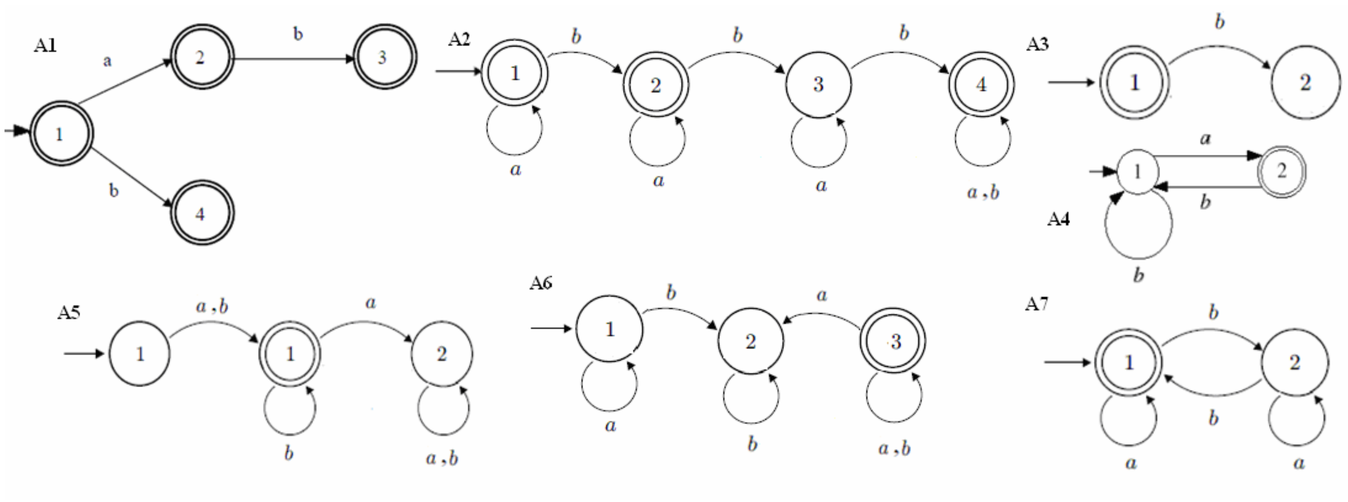
$S \to bS/aT$
$T \to bT/aU$
$U \to bU/aV$
$V \to bV/\varepsilon$

☞ $(abbc/baba)^+aa(cc/bb)^*$

$S \to TaaU$
$T \to abbcT/babaT/abbc/baba$
$U \to ccU/bbU/\varepsilon$

Exercice 02 :

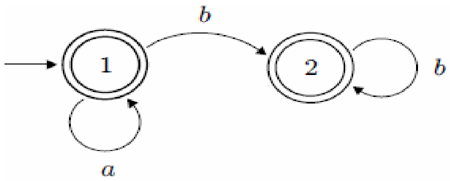
1) Donner l'expression régulière du langage reconnu par chaque automate d'états finis :



Expression(L(A1)) = ε + a + b + ab
Expression(L(A2)) = a* + a*ba* + a*ba*ba*b(a+b)*
Expression(L(A3)) = ε
Expression(L(A4)) = a(bb*a)*
Expression(L(A5)) = (a + b)b*
Expression(L(A6)) = ϕ
Expression(L(A7)) = (a + ba*b)*

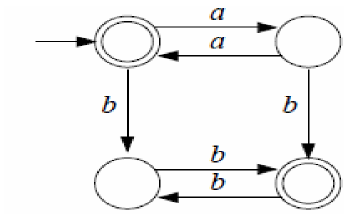
2) Proposer un automate d'états finis pour chacun des langages et expressions ci-dessous :

$L_1 = \{a^n b^m / n, m \geq 0\}$

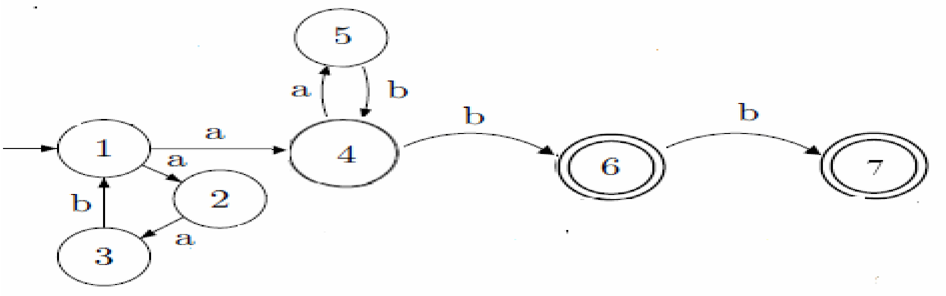


$L_2 = \{a^n b^n / n \geq 0\}$ on ne peut pas construire un automate d'états finis puisqu'il est de type 2.

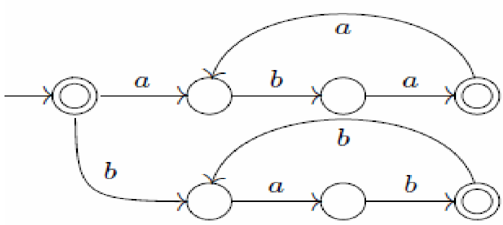
$L_3 = \{a^n b^m / n + m$ est pair$\}$



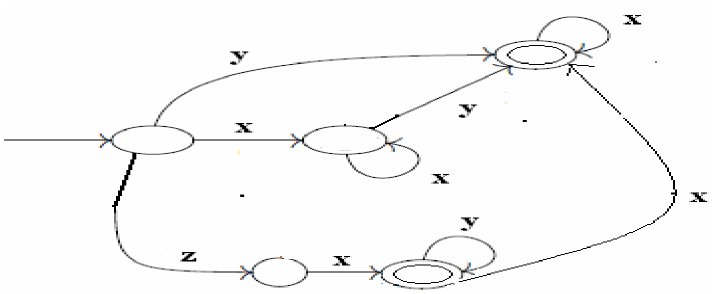
$E_1 = (aab)^* a(ab)^* (b + bb)$



$E_2 = (aba)^* + (bab)^*$



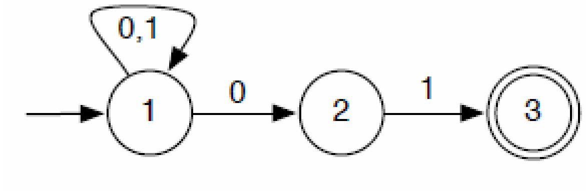
E_3 = zxy* + zxy*xx* + xx*yx* + yx*



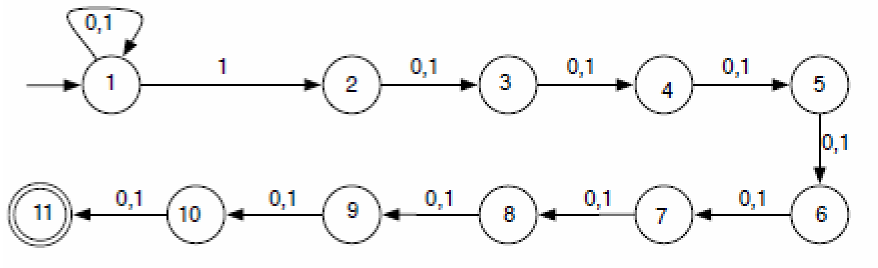
Exercice 03 :

Construire un automate d'états finis pour chacun des langages suivants définis sur X= {0, 1}

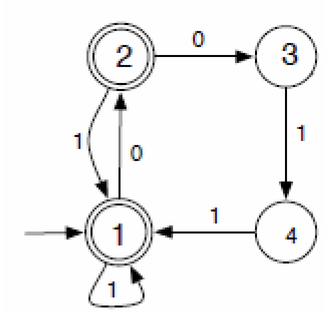
1) Toutes les chaînes qui se terminent par 01.



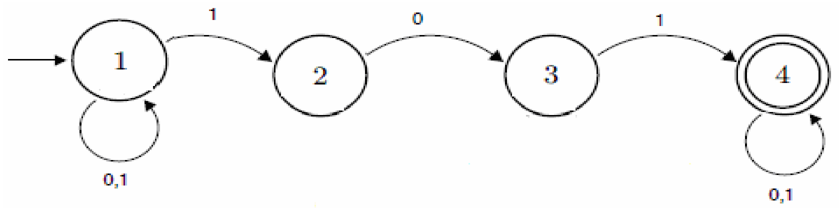
2) Toutes les chaînes dont le 10ème symbole, compté à partir de la fin de la chaîne, est un 1.



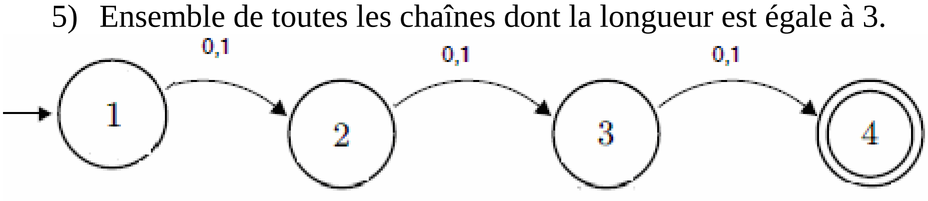
3) Ensemble de toutes les chaînes dans lesquelles chaque paire de 0 apparaît devant une paire de 1.



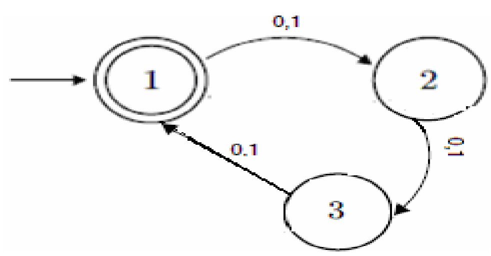
4) Ensemble de toutes les chaînes qui contiennent la sous chaîne 101 = Ensemble de toutes les chaînes qui contiennent au moins une sous chaîne 101



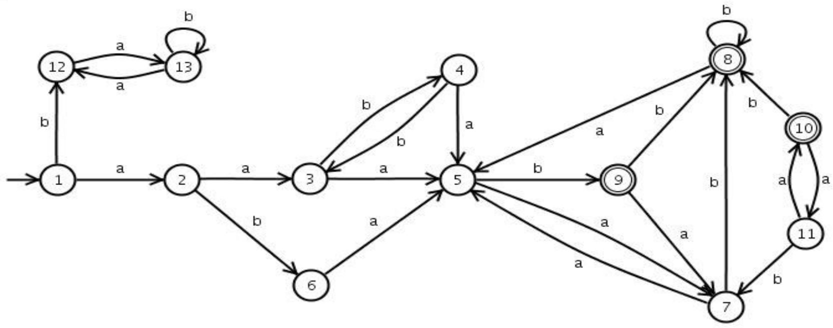
5) Ensemble de toutes les chaînes dont la longueur est égale à 3.



6) Ensemble de toutes les chaînes dont la longueur est divisible par 3.



Exercice 04 :



Pour minimiser un automate, il faut tout d'abord :
- S'assurer qu'il est déterministe.
- Éliminer les états qui ne sont pas accessibles à partir de l'état initial, ainsi que les transitions adjacentes à ces états.
- Éliminer les états à partir desquels on ne peut pas atteindre un état final, ainsi que les transitions adjacentes à ces états.
-Compléter l'automate en ajoutant un état puits non final qui boucle sur lui-même sur chaque symbole et vers lequel arriveront touts les transitions qui manquent.
- partitionner les états en deux classes. (classe des états finaux et une classe pour le reste)
- dans chaque classe obtenue d'une étape précédente, on regroupe les états équivalents (les états ayant une même classe sur chaque symbole) en éclatant ainsi cette classe en d'autres classes.
- répéter cette dernière étape jusqu' à ce qu'il n' y ait plus d'éclatement possible.

On minimiser cet automate :
- Dans notre cas, l'automate à minimiser est déjà déterministe.
- Les états 10 et 11 sont inaccessibles à partir de l'état initial, et les états 12 et 13 ne mènent à aucun état final. Nous pouvons donc éliminer tout de suite ces états.
- on complète l'automate c'est-à-dire au niveau de sa table de transition, cela consiste à boucher les cases vides par un état puits non final (nouveau état 10 par exemple) qui boucle sur lui-même sur chaque symbole et vers lequel arriveront touts les transitions qui manquent.
-L'étape suivante est de construire ce tableau, en regroupant les états finaux dans une classe B, et les autres états dans une seconde classe A :

Classe	état	a	b
A	1	A	A
A	2	A	A
A	3	A	A
A	4	A	A
A	5	A	B
A	6	A	A
A	7	A	B
A	10	A	A
B	8	A	B
B	9	A	B

Comme les états de la classe A n'ont pas tous une même classe d'arrivée sur chaque symbole, il faut éclater cette classe en 2 nouvelles classes. On continue ainsi jusqu'à ce qu'on obtienne un tableau où chaque classe respecte cette condition.

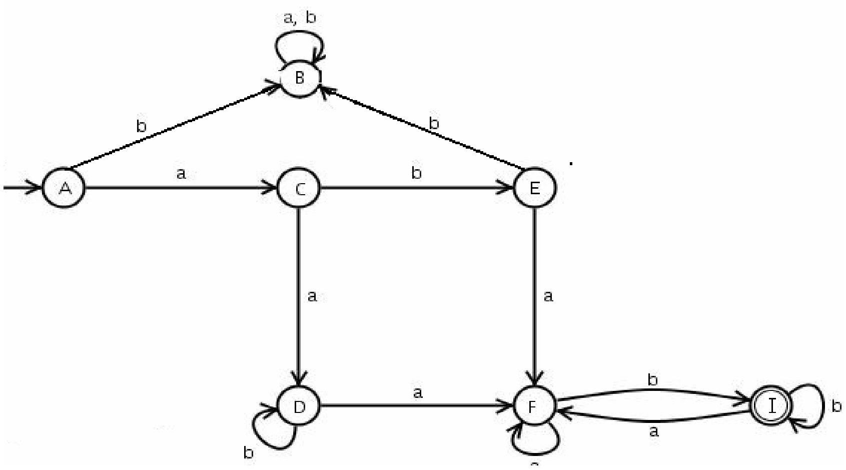
Classe	état	a	b
A	1	A	A
A	6	B	A
A	2	A	A
A	3	B	A
A	4	B	A
A	10	A	A
B	5	B	C
B	7	B	C
C	8	B	C
C	9	B	C

Classe	état	a	b
A	1	A	A
A	2	B	B
A	10	A	A
B	3	C	B
B	4	C	B
B	6	C	A
C	5	C	D
C	7	C	D
D	8	C	D
D	9	C	D

Classe	état	a	b
A	1	B	A
A	10	A	A
B	2	C	D
C	3	E	C
C	4	E	C
D	6	E	A
E	5	E	F
E	7	E	F
F	8	E	F
F	9	E	F

Classe	état	a	b
→A	1	C	B
B	10	B	B
C	2	D	E
D	3	F	D
D	4	F	D
E	6	F	B
F	5	F	I
F	7	F	I
I	8	F	I
I	9	F	I

Nous pouvons maintenant créer, à partir du dernier tableau, l'automate minimal acceptant le même langage que l'automate de l'énoncé. Chaque classe correspond à un état de l'automate minimal. L'état initial est la classe contenant l'état initial de l'automate de départ. L'état final est toute classe contenant au moins un état final de l'automate initial. La fonction de transition est bien définie à partir du dernier tableau.



Bonne suite