Université de M'hamedBouguerraBoumerdès
Faculté des sciences
Département d'Informatique

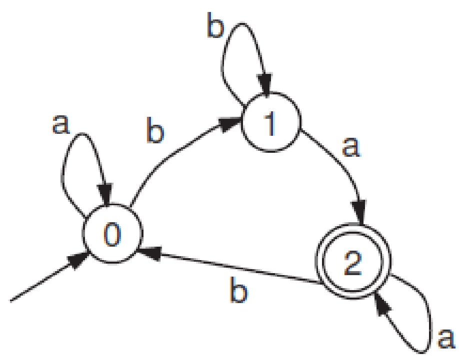
Module : Théorie des Langages. **Année : 2019-2020**
Filière : LI- S4 **Document : Série 4 (Corrigé)**

Chapitre 4 : Langage régulier

Objectif : Comprendre la relation entre les automates à états finis, grammaire régulière, et expression régulière.

Exercice1

a) Déterminer une expression régulière pour l'automate suivant :



On construit le système d'équations (on a 3 états donc 3 équations)

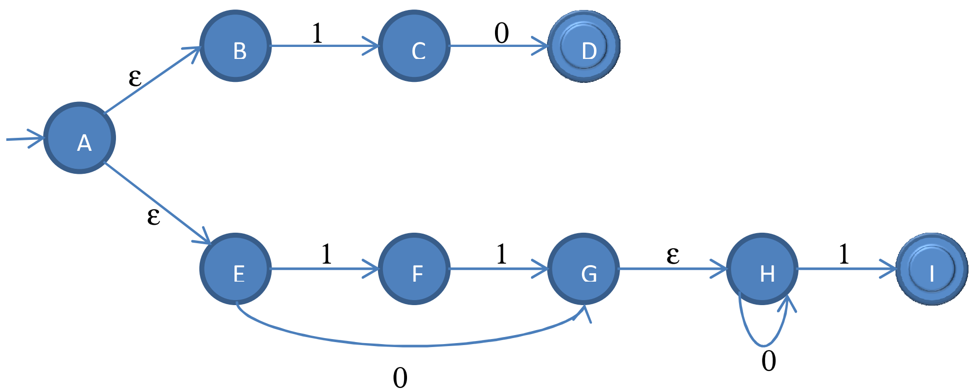
$$\begin{cases} L_0 = a\,L_0 + b\,L_1 \\ L_1 = b\,L_1 + a\,L_2 \\ L_2 = b\,L_0 + a\,L_2 + \varepsilon \end{cases}$$

⇨ $L_2 = a^* (b\,L_0 + \varepsilon) = (a^*\,b)\,L_0 + a^*$
⇨ $L_1 = (b^*\,a)\,L_2$
⇨ $L_1 = b^*\,a\,(a^*\,b\,L_0 + a^*)$
⇨ $L_1 = (b^*\,a\,a^*\,b)\,L_0 + (b^* a\,a^*)$
⇨ $L_0 = (a^*\,b)\,L_1$
⇨ $L_0 = (a^*\,bb^*\,aa^*\,b)\,L_0 + (a^*\,bb^* aa^*)$
⇨ $L_0 = (a^*\,bb^*\,aa^*\,b)^*\ \ a^*\,bb^*\,aa^*$
⇨ Ou $L_0 = (a^*\,b^+\,a^+\,b)^*\ \ a^*\,b^+\,a^+$

b) Construire les AEF correspondants aux expressions régulières suivantes :

Exp 1 ≡ 10+(0+11)0*1

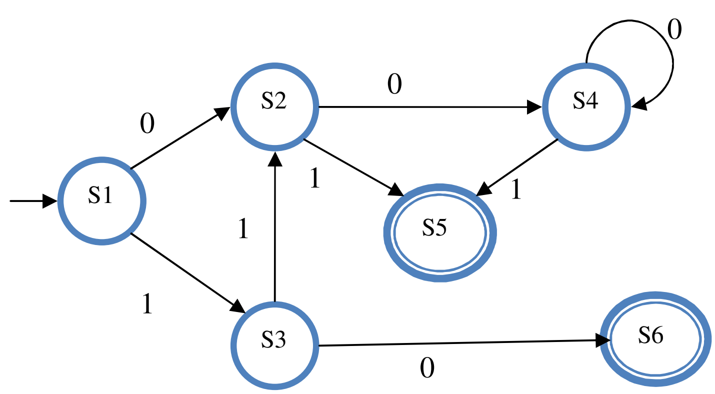
Etape 1 : construire l'automate indéterministe avec ε-transitions



Remarque : on peut avoir plusieurs automates avec ε-transitions selon votre expérience.
Dans cet exemple j'ai donné un automate pas trop détaillé mais vous pouvez détailler plus en rajoutant d'autres ε-transitions (pour 0* ou (0+11)) mais l'automate minimal sera le même quel que soit l'automate de départ (bien sûr s'il est juste)

Etape 2 : déterminiser l'automate

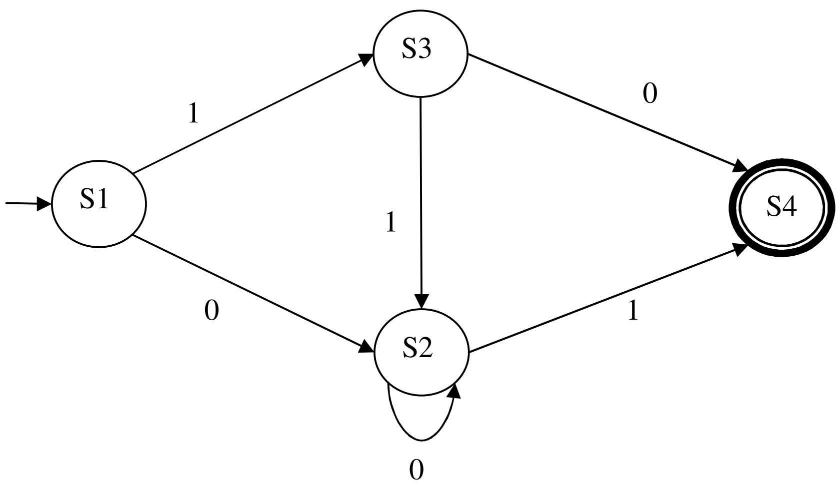
	0	1
ε-f {A}= { A, B, E} S1	ε-f {G}= { G, H} S2	ε-f {C, F}= { C, F} S3
{G, H } S2	ε-f {H}= { H } S4	ε-f {I}= { I } S5
{C, F } S3	ε-f {D}= { D} S6	ε-f {G}= { G, H } S2
{ H} S4	ε-f {H}= { H } S4	ε-f {I}= { I } S5
{ I} S5	/	/
{D}S6	/	/



Etape 3 : minimiser l’automate

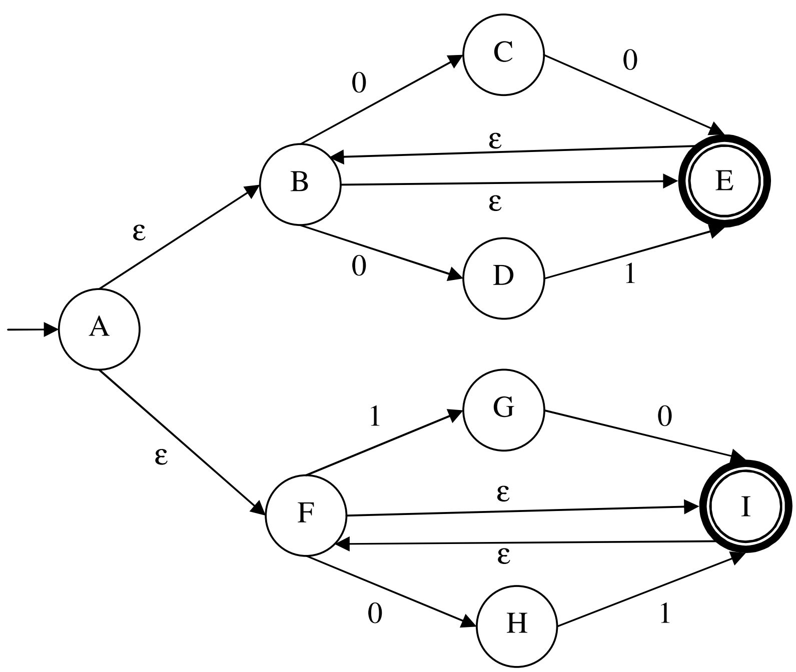
Pas d’états inaccessibles
β 0 ≡ ({S1, S2, S3, S4}, {S5, S6})
β 1 ≡ ({S1}, {S3}, {S2, S4} , {S5, S6})
β 2 ≡ β 1 , on s’arrête.

Donc l’automate d’états finis minimal qui reconnaît cette expression est



2. Exp 2 ≡ (00+01)* + (10+01)*

Etape 1 : construire l’automate indéterministe avec ε-transitions



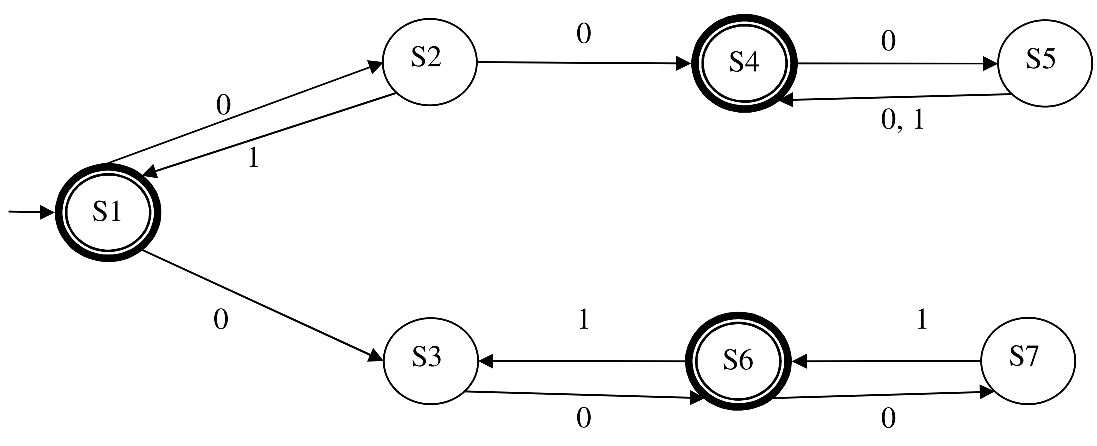
Etape 2 : déterminer l'automate

	0	1
ε-f {A}= { A, B, E, F, I } S1 initial et final	ε-f {C, D, H}= { C, D, H} S2	ε-f {G}= { G} S3
{ C, D, H} S2	ε-f {E}= { B, E } S4	ε-f {E, I}= {B, E, F,I } S5
{G} S3	ε-f {I}= { F, I } S6	/
{ B, E } S4 final	ε-f {C, D}= { C, D} S7	/
{B, E, F, I } S5 final	ε-f {C, D , H}= { C, D, H} S2	ε-f {G}= { G} S3
{ F, I } S6 final	/ε-f {H}= { H} S8	ε-f {G}= { G} S3
{ C, D} S7	ε-f {E}= { B, E } S4	ε-f {E}= { B, E } S4
{ H} S8	/	ε-f {I}= { F, I } S6

Etape 3 : minimiser l'automate

Pas d'états inaccessibles
β 0 ≡ ({S2, S3, S7, S8}, {S1, S4, S5, S6})
β 1 ≡ ({S2, S7}, {S3}, {S8}, {S4}, {S1, S5, S6})
β 2 ≡ ({S2} ,{ S7}, {S3}, {S8}, {S4}, {S1, S5}, {S6})
β 3 ≡ β 2 , on s'arrête.

Donc l'automate d'états finis minimal qui reconnaît cette expression est

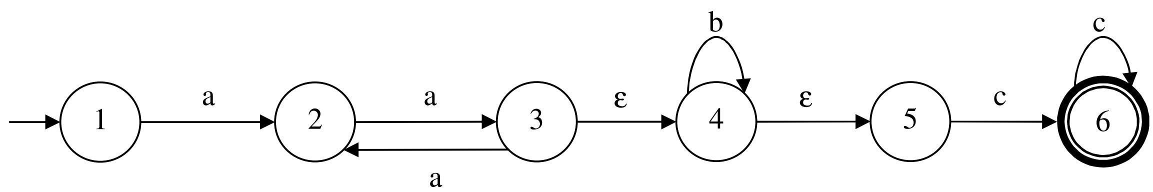


c) Proposer un automate à états finis et une expression régulière pour les langages suivants :

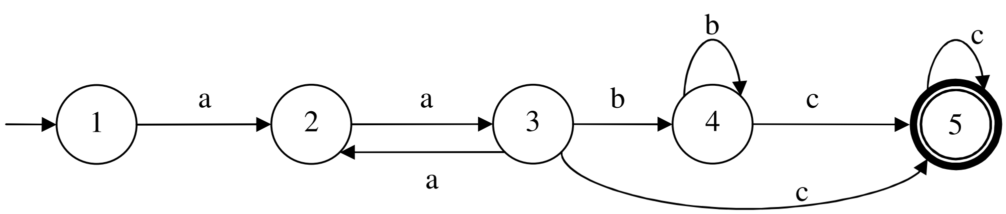
1. $L1 = \{ a^{2n+2} b^{p} c^{m+1} , \; n,m,p \geq 0 \}$

On remarque que L1 est la concaténation de 3 langages :

- $L_{1.1}$: le nombre de « a » est pair (2,4,6,…),
- $L_{1.2}$: le « b » supérieur ou égal à 0 et
- $L_{1.3}$: le « c » supérieur ou égal à 1.



Après on détermine et on minimise l'automate, on trouve l'automate suivant :



L'expression régulière :

$$\begin{cases} L_1 = a\,L_2 \\ L_2 = a\,L_3 \\ L_3 = a\,L_2 + b\,L_4 + c\,L_5 \\ L_4 = b\,L_4 + c\,L_5 \\ L_5 = c\,L_5 + \varepsilon \end{cases}$$

⇨ $L_5 = c^*$

⇨ $L_4 = b\,L_4 + cc^*$

⇨ $L_4 = b^*cc^*$

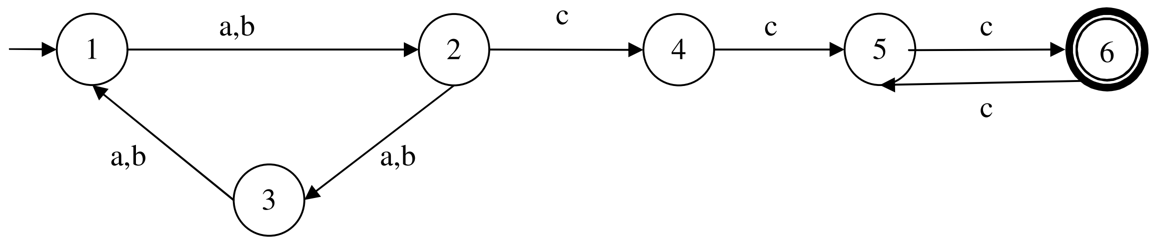
⇨ $L_3 = a\,L_2 + bb^*cc^* + cc^*$

⇨ $L_2 = aa\,L_2 + a\,(bb^*cc^*+cc^*)$

⇨ $L_2 = (aa)^*a(bb^*cc^*+cc^*)$

⇨ $L_1 = a(aa)^*\,a\,(bb^*cc^* + cc^*)$

2. $L2=\{w\, c^{2n+1},\ n \geq 1 \text{ et } w \in \{a, b\}^* \text{ et } |w| = 3m+1,\ m \geq 0\}$



L'expression régulière :

$$\begin{cases} L_1 = (a+b)\, L_2 \\ L_2 = (a+b)\, L_3 + c\, L_4 \\ L_3 = (a+b)\, L_1 \\ L_4 = c\, L_5 \\ L_5 = c\, L_6 \\ L_6 = c\, L_5 + \varepsilon \end{cases}$$

⇨ $L_6 = cc\, L_6 + \varepsilon$
⇨ $L_6 = (cc)^*$
⇨ $L_5 = c\, (cc)^*$
⇨ $L_4 = cc\, (cc)^*$
⇨ $L_2 = (a+b)^2\, L_1 + ccc\, (cc)^*$
⇨ $L_1 = (a+b)^3\, L_1 + (a+b)\, ccc\, (cc)^*$
⇨ $L_1 = (\, (a+b)^3\,)^*\, (a+b)\, ccc\, (cc)^*$

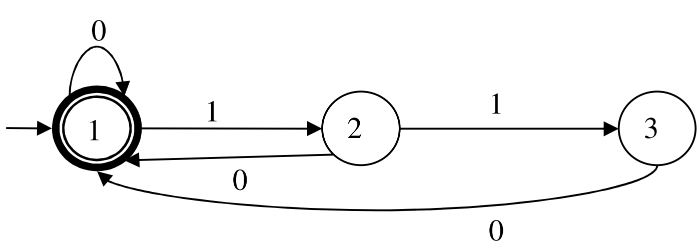
Exercice 2 (EXAMEN 2017)

Soient les deux langages suivants :

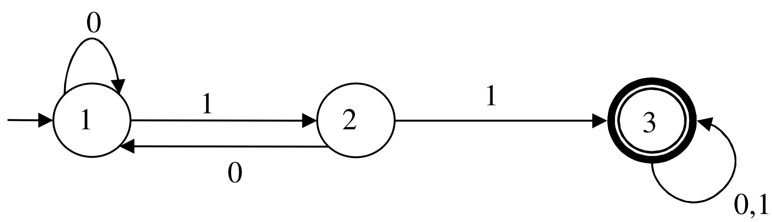
L1={w∈{0, 1}* tel que dans chaque mot w de L1, toute sous-chaîne « 11 » est immédiatement suivie par au moins un «0» }

L2= {w∈{0, 1}*tel que chaque w contient au moins la sous séquence « 11 »}

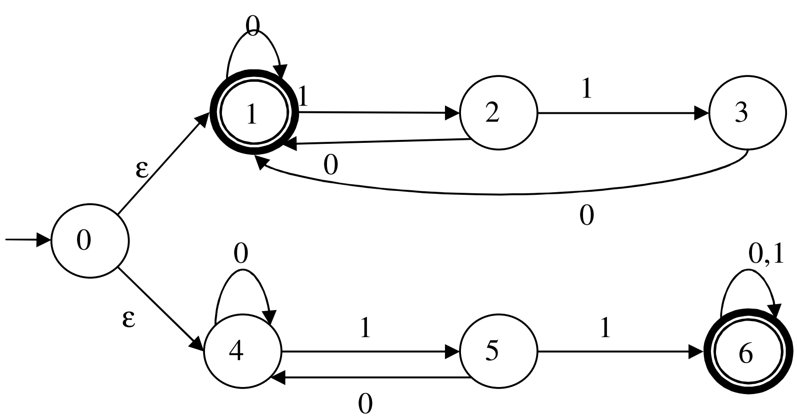
1. Déterminer un automate à états finis minimal qui accepte le langage L1.



2. Déterminer un automate à états finis minimal qui accepte le langage L2.



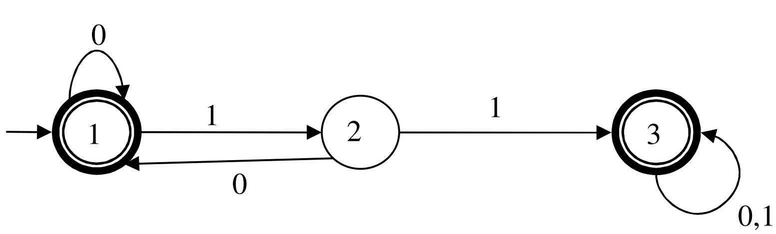
3. Déterminer un automate à états finis minimal qui accepte le langage $L1 \cup L2$.



	0	1
{0,1,4}q0	{1,4}	{2,5}
{1,4 }q1	{1,4}	{2,5}
{2,5 }q2	{1,4}	{3,6}
{3,6}q3	{1,6}	{6}
{6}q4	{6}	{6}
{1,6 }q5	{1,6}	{2,6}
{2,6}q6	{1,6}	{3,6}

Pas d'états inaccessibles

β 0 ≡ ({q2}, {q0,q1, q3, q4, q5, q6})
β 1 ≡ ({q2}, {q0, q1}, {q3, q4, q5, q6})
β 2 ≡ ({q2}, {q0, q1}, {q3, q4, q5, q6})
β 2 ≡ β 1, on s'arrête.



4. Donner une expression régulière qui dénote le langage $L1 \cup L2$.

$$\begin{cases} L_1 = 0\, L_1 + 1\, L_2 + \varepsilon \\ L_2 = 0\, L_1 + 1\, L_3 \\ L_3 = (0+1)\, L_3 + \varepsilon \end{cases}$$

$\Rightarrow L_3 = (0+1)^*$

$\Rightarrow L_2 = 0\, L_1 + 1(0+1)^*$

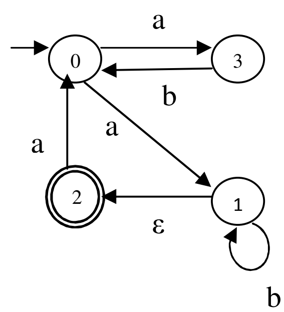
$\Rightarrow L_1 = 0\, L_1 + 1\, (0\, L_1 + 1(0+1)^*) + \varepsilon$

$\Rightarrow L_1 = (0+10)\, L_1 + 11(0+1)^* + \varepsilon$

$\Rightarrow L_1 = (0+10)^*\, 11(0+1)^* + (0+10)^*$

Exercice 3 (EXAMEN 2018)

1. Soit l'automate non déterministe M suivant :



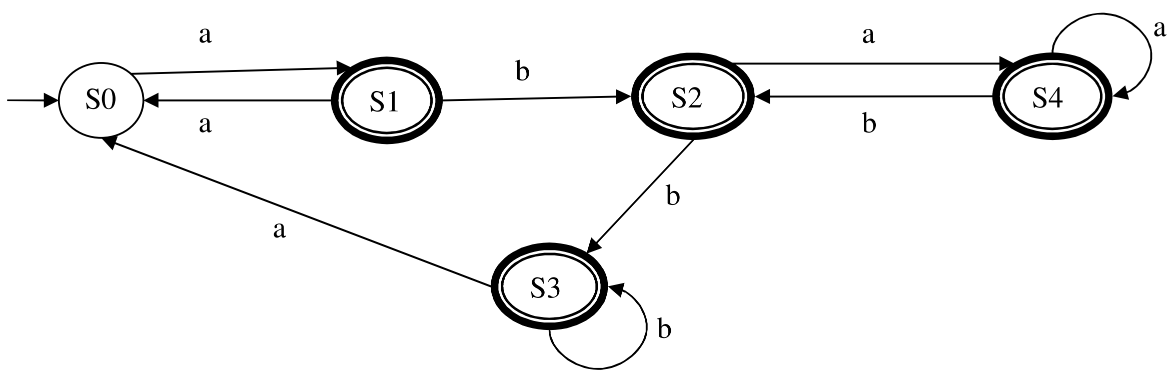
a) Construire un automate M' déterministe minimum équivalent à M.

	a	b
0 S0 initial	{1,2,3}	/
{1,2,3} S1 final	0	{0,1,2}
{0,1,2} S2 final	{0,1,2,3}	{1,2}
{0,1,2,3} S3 final	{0,1,2,3}	{0,1,2}
{1,2} S4 final	0	{1,2}

Minimisation :

Pas d'états inaccessibles

β 0 ≡ ({S0}, {S1, S2, S3, S4})
β 1 ≡ ({S0}, {S1,S4}, {S2, S3})
β 2 ≡ ({S0}, {S1}, {S4}, {S2}, {S3})
On s'arrête.



b) Déterminer une grammaire régulière à droite G1 qui engendre L(M).

G1= ({a, b}, {S, A, B, C, D}, S, P1)

P1 ≡ { S→aA, A→aS| bB| ε, B→aD| bC| ε, C→aS| bC| ε, D→aD| bB|ε }

2. Soit la grammaire G2= ({a,b}, {S,A,B}, S, R) avec R={S→abS/aA,
A→bA/B
B→aS/ε}

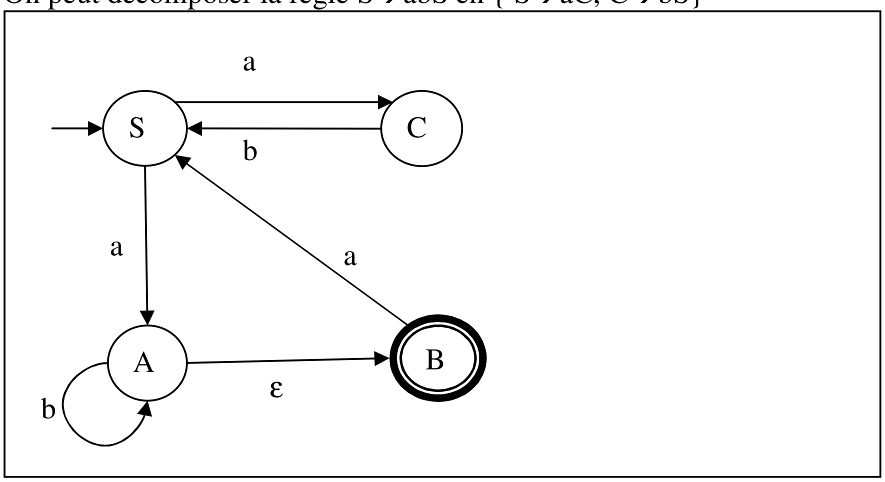
a) Quel est le type de G2 et L(G2)?

La grammaire G2 est de type 3 d'après le format de ses règles =>L(G2) est un langage régulier.

c) Comparer L(G1) et L(G2)

On construit l'automate M2 à partir de la grammaire G2

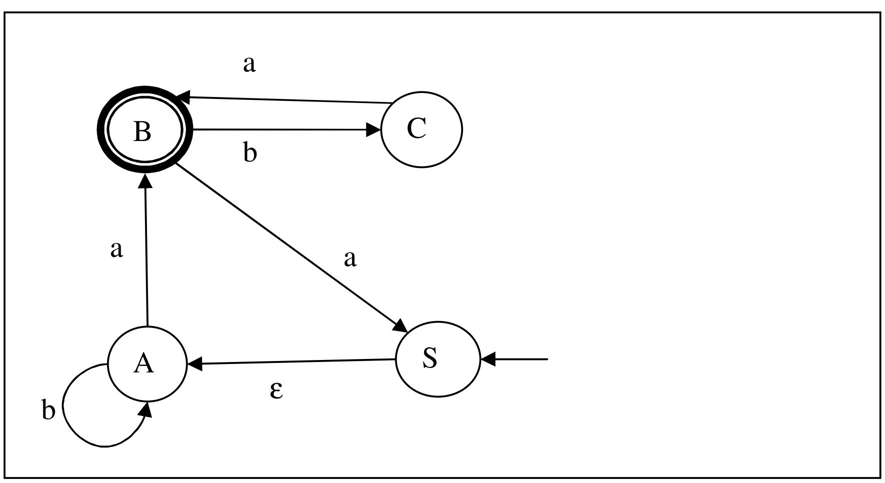
On peut décomposer la règle S→abS en { S→aC, C→bS}



On remarque qu'on retrouve l'automate M =>L(G1) =L(G2)

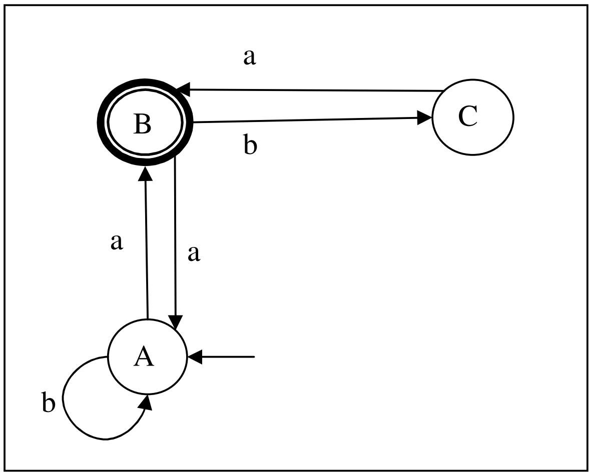
d) Déterminer une grammaire régulière à gauche G3 qui engendre L(M).

Etape 1 : construire l'automate transposé de M



Etape 2 : déterminer et minimiser

On obtient l'automate suivant :



Etape 3 : construire la grammaire régulière à droite

G3=({a,b}, {A, B, C}, A,{ A→bA| aB, B→ bC |a A| ε, C→aB})

Etape 4 : construire la grammaire régulière à gauche (en inversant les symboles)

G3=({a,b}, {A, B, C}, A,{ A→Ab| Ba, B→ Cb | Aa| ε, C→Ba})

Exercice 4

Pour démontrer qu'un langage infini est non régulier, il faut utiliser la contradiction du lemme de pompage.

Pour démontrer qu'un langage est régulier, il faut utiliser l'une des méthodes suivantes :

- Dessiner l'automate
- Donner une grammaire régulière
- Appliquer les opérations de fermeture sur les langages réguliers
- Donner une expression régulière

Parmi les langages suivants quels sont ceux qui sont réguliers ?

1. L_1= {s = a^n : n est un nombre premier}

L1 n'est pas régulier et la démonstration se fait par contradiction du lemme de pompage.

$\forall p \geq 0$, $\exists w \in \sum^* = \{a\}^*$, soit $w = a^i$ avec i nombre premier, $i \geq P$ et $|w| = i \geq P$;
et $\forall$ x; y; z $\in \sum^*$, w = xyz avec $|y| > 0$ et $|xy| \leq p$; (Il ne pourra exister qu'un seul découpage au niveau du a)
$w = a^{|x|}:a^{|y|}:a^{i-|x|-|y|}$.
Si on calcule w_2 en pompant au niveau de y on obtient :
$w_2 = w = a^{|x|}a^{2|y|}a^{i-|x|-|y|}$.
$w_2 = a^{i+1.|y|}$
Si on calcule w_{i+1} en pompant au niveau de y on obtient :

$W_{i+1} = a^{i+i|y|} = a^{i(1+|y|)}$, l'exposant de a n'est pas nombre premier
$\Rightarrow w_{i+1} \notin L$
$\Rightarrow L_1$ est non régulier

2. $L_2=\{a^n b^{2m}, n, m>=0\} \cup \{a^n, n>=0\}$

L₂ est régulier car c'est l'union de deux langage réguliers L_{21} et L_{22} avec $L_{21}= \{a^n b^{2m}\ n,m>=0\}$ et $L_{22}= \{a^n\ n>=0\}$ nous remarquons également que $L_{22} \subseteq L_{21}$

3. $L_3= \{s = a^n b^{2n}, n<=100\}$

L_3 est régulier car L_3 est un langage fini.

4. L'ensemble des mots ayant autant de zéros que de uns.

L_4 n'est pas régulier, il suffit de démontrer en utilisant lemme du pompage sur le mot $w=0^n1^n$ (même démonstration vue dans le cours).

5. L'ensemble des mots sur {0,1} n'ayant pas 3 zéros consécutifs.

L_5 est régulier, une des démonstrations possibles est l'écriture d'une grammaire G de type 3 qui génère ce langage.

G =<Vn,Vt,S,P> avec S axiome, Vn={S,T,V} , Vt={0,1} , P={S→1S/0T/ε
T→1T/ 0V/ε
V→ 1S/ ε }

Ou de donner l'automate complément de langage {tous les mots sur {0,1} ayant 3 zéros consécutifs} vu dans le chapitre 3

Exercice 5

Donnez une expression régulière étendue décrivant les langages suivants

1. tous les mots sur {a, b} où chaque a est précédé d'un b ;
2. tous les mots sur {a, b} contenant à la fois les facteurs aa et bb ;
3. tous les mots sur {a, b} contenant soit aa soit bb mais pas les deux à la fois ;
4. tous les mots sur {a, b} ne contenant pas deux a consécutifs ;
5. tous les mots sur {a, b, c} où le nombre de a est multiple de 2 ;
6. Tous les entiers (en base dix) multiples de 5.

Solution

1. (b|ba)*
2. [ab]{2,}
3. soit aa soit bb mais pas les deux ab*ab* | ba*ba* ou (ab*| ba*)*
4. pas deux a consécutif (b* a ?)*
5. a multiple de deux (aa|b|c)*
6. [0-9] *[05]

Devoir (EXAMEN 2017)

Soit la grammaire suivante G= ({a, b},{S, A, B, C}, S, R) avec R :
R={S→Sa/ Aa/Ca

$A \rightarrow Bb$
$B \rightarrow Ca/ Sa/ Aa$
$C \rightarrow \varepsilon$
}

1. Déterminer un automate d'état fini minimal qui accepte le langage engendré par cette grammaire
2. Donner une expression régulière (notée EXP1) qui dénote le langage engendré par cette grammaire (noté L(EXP1).
3. Soit l'expression régulière EXP2= $((a+ab)^*+(a+aa)^*)^*$. Le langage dénoté par cette expression est noté L(EXP2). Déterminer un automate d'état fini minimal qui accepte le langage dénoté par cette expression.
4. Comparer L(EXP1) et L(EXP2) : a-t-on **L(EXP1)=L(EXP2)** ou **L(EXP1) ⊃ L(EXP2)** ou **L(EXP1)⊂ L(EXP2)**? Justifier.