Université Mouloud MAMMERI de Tizi-Ouzou
Faculté de génie électrique et informatique
Département d'informatique

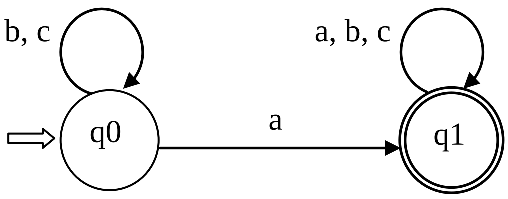
Année universitaire : 2017/2018
2ième année Licence-Informatique
module : Théorie des langages

CORRIGÉ ABREGÉ DE LA SÉRIE D'EXERCICES n° 2 de ThL

par : **M.S. Habet, C. Cherifi, N. Otmani, F. Bouhatem**

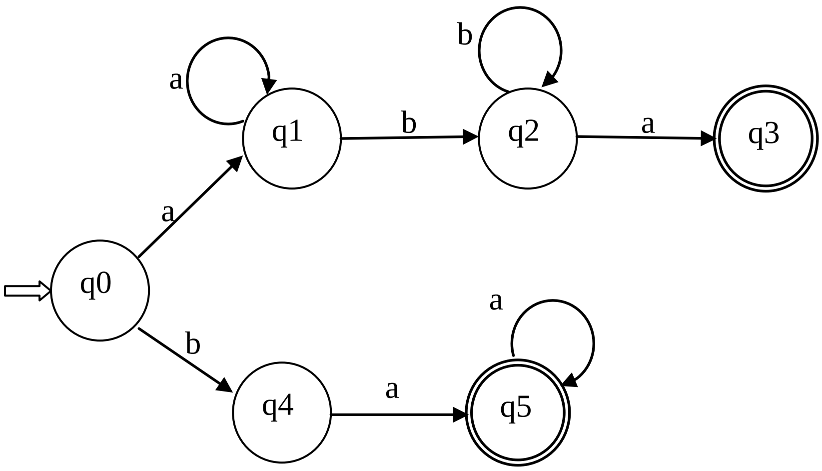
EXERCICE **1** :

a) $L_1 = \{ w \in \{a, b, c\}^* / w$ contient au moins une occurrence de la lettre 'a' } :



L'automate de L_1 est construit de telle sorte que pour arriver à l'état final, il faut effectuer une transition étiquetée par 'a' : cela assure que la chaîne reconnue contient au moins un 'a'. Evidement, avant le 'a' on peut avoir d'autres caractères et après aussi.

b) $L_2 = \{ w \in \{a, b\}^* / w = a^n b^m a$ ou $w = ba^n$; $n, m \geq 1 \}$:



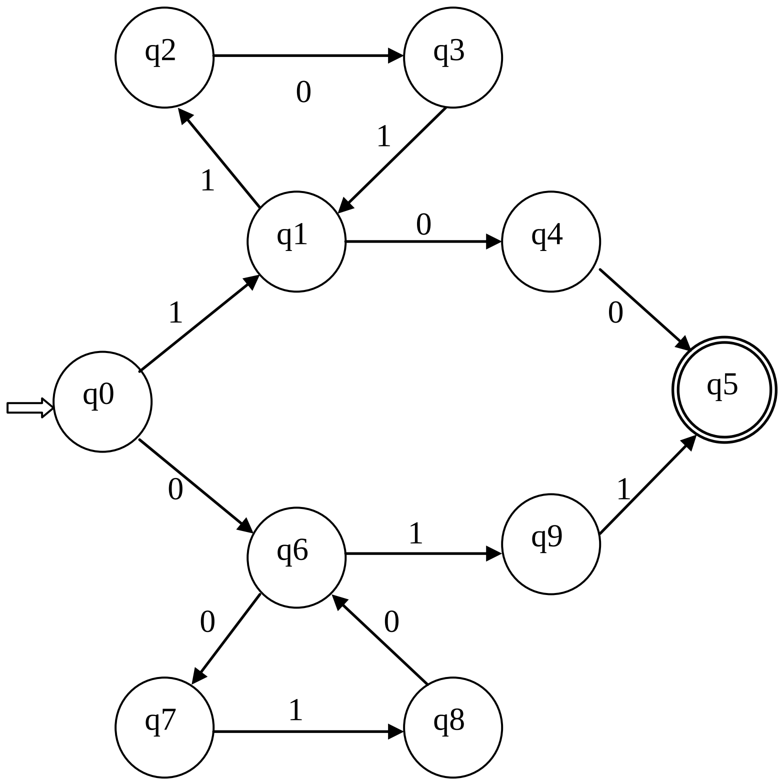
L'automate de L_2 comporte deux "branches" : une (q0, q1, q2 ,q3) pour reconnaitre les mots de la forme : $a^n.b^m.a$ et l'autre (q0, q4, q5) pour les mots de la forme : $b.a^n$.

Il faut bien noter que si la boucle avec a sur q1 était mise dans q0, la première branche accepterait des mots $a^n.b^m.a$; mais il y aurait un problème avec la deuxième branche : l'automate accepterait des mots de la forme $a^k.b.a^n$ qui n'appartiennent pas à L_2 (lorsque $k > 0$ et $n > 1$).

Question :

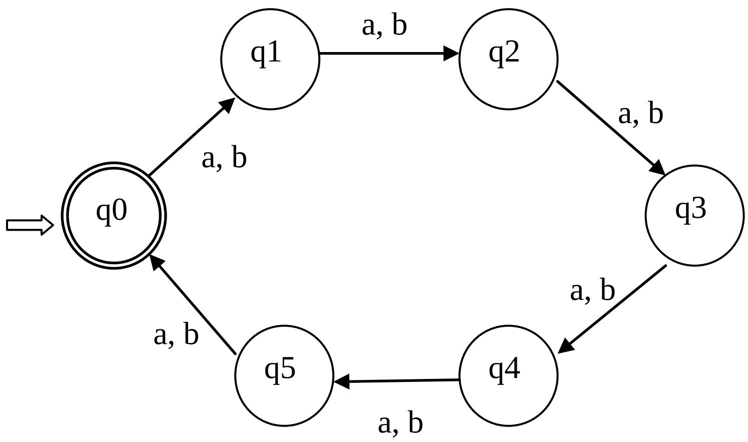
Construire l'automate du langage obtenu en modifiant les conditions sur n et m dans la définition de L_2 : on exige seulement que $n, m \geq 0$.

c) $L_3 = \{ w \in \{0, 1\}^* / w = 1(101)^n00$ ou $w = 0(010)^n11, n \geq 0 \}$:



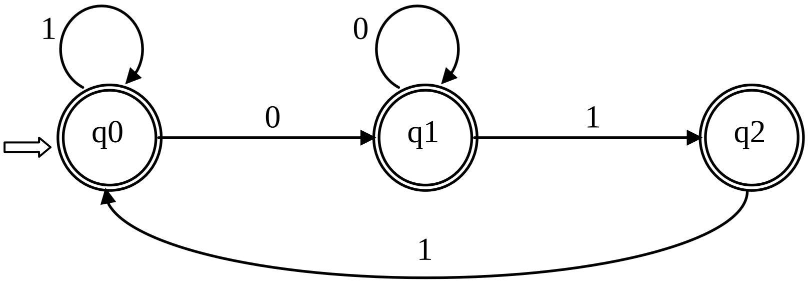
Là aussi, l'automate contient deux sous-graphes, l'un (basé sur les états q0, q1, q2 , q3, q4, q5) pour reconnaitre les mots de la forme : $1.(1.0.1)^n.0.0$ et l'autre (q0, q6, q7, q8, q9,q5) pour : $0.(0.1.0)^n.1.1$.

d) $L_4 = \{ w \in \{a, b\}^* / |w| \equiv 0 \ [6] \}$:



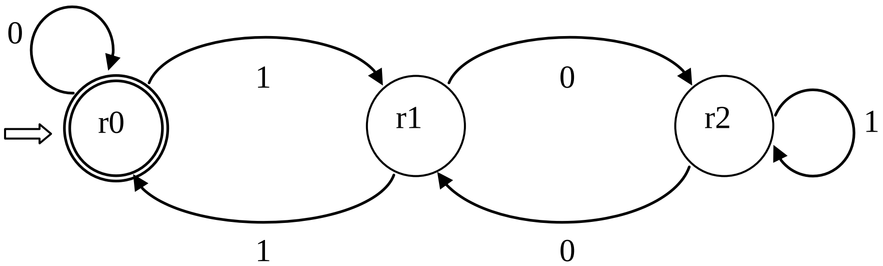
Le graphe de cet automate de L_4 consiste en un circuit (en fait plusieurs du fait que les étiquettes des transitions sont multiples) de longueur 6 (6 arcs). A chaque fois qu'on parcoure le circuit on aura ajouté un mot de longueur 6 composé de a et de b. Ce qui fait que, en commençant par q0 et en terminant dans ce même état on aura un mot de longueur multiple de 6.

e) $L_5 = \{ w \in \{0, 1\}^* / w$ ne contient pas la sous chaîne «010» } :



L'automate de L_5 est construit de telle sorte à autoriser toutes les transitions possibles sauf celles qui peuvent mener à la formation de la sous-chaîne «010».

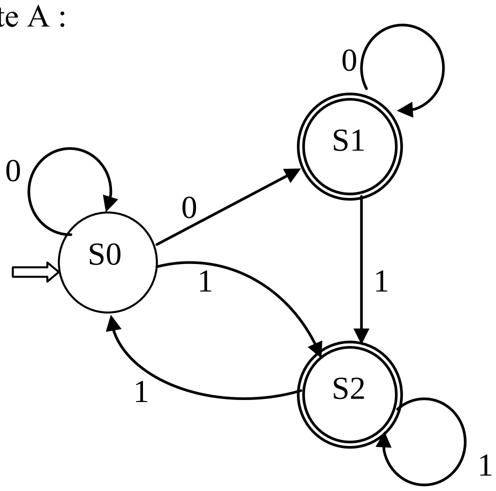
f) L_6 = ensemble des mots de $\{0, 1\}^*$ représentant les nombres divisibles par 3 (dans le système de numération binaire naturel) :



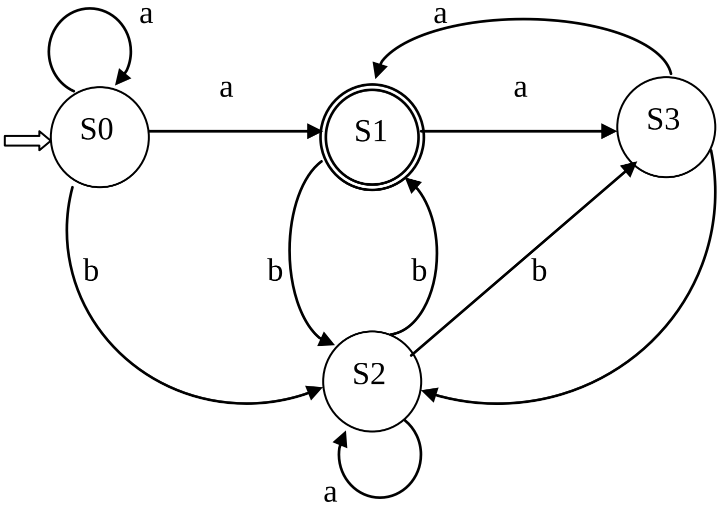
Pour construire cet automate de L_6, on a considéré trois états r0, r1 et r2 qui représentent le fait que si on est dans l'état ri, i=0,1,2 alors le reste de la division du nombre (représenté par la chaîne de 0, 1 lue par le dispositif de l'automate) est égal à i (bien sûr, si on considère les restes possibles de la division d'un entier par 3, il y en a trois : 0, 1 ou 2). C'est de là qu'on a déduit les transitions entre les états.

EXERCICE **2** :

1) L'automate A :



L'automate B :



2) Grammaire régulière à droite équivalente à A :

G_A = ({0, 1}, {S, A, B}, S, P_A), où P_A contient les règles :

S → 0S | 0A | 1B ; A → 0A | 1B | ε ; B → 1S | 1B | ε

Grammaire régulière à droite équivalente à B :

G_B = ({a, b}, {S, A, B, C}, S, P_B), où P_B contient les règles :

S → aS | aA | bB ; A → bB | aC | ε ; B → bA | aB | bC ; C → aA | bB

3) Table de transition de l'automate déterministe équivalent à A : (les états soulignés sont des états finaux)

	0	1
<S0> = q0	<S0,S1>	<S2>
<S0,S1> = q1	<S0,S1>	<S2>
<S2> = q2	/	<S0,S2>
<S0,S2> = q3	<S0,S1>	<S0,S2>

L'automate déterministe équivalent à A :

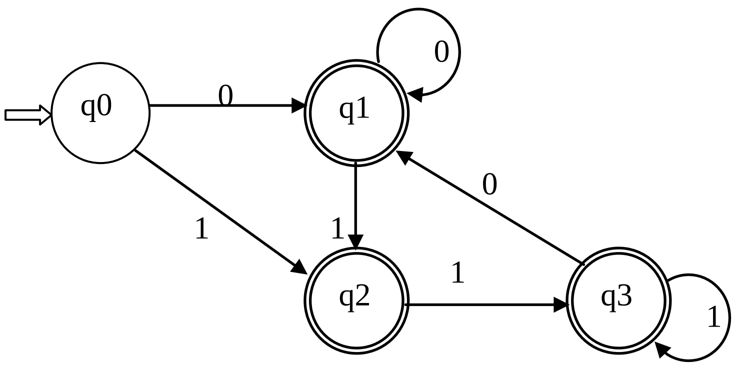
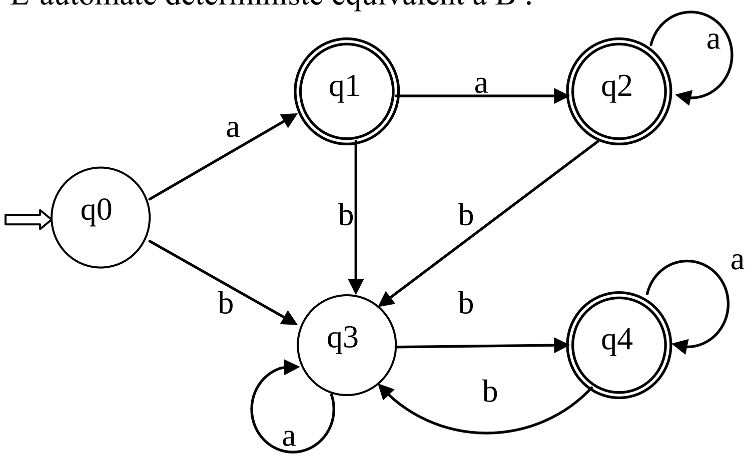


Table de transition de l’automate déterministe équivalent à B :

	a	b
<S0> = q0	<S0,S1>	<S2>
<S0,S1> = q1	<S0,S1,S3>	<S2>
<S0,S1,S3> = q2	<S0,S1,S3>	<S2>
<S2> = q3	<S2>	<S1,S3>
<S1,S3> = q4	<S1,S3>	<S2>

L’automate déterministe équivalent à B :



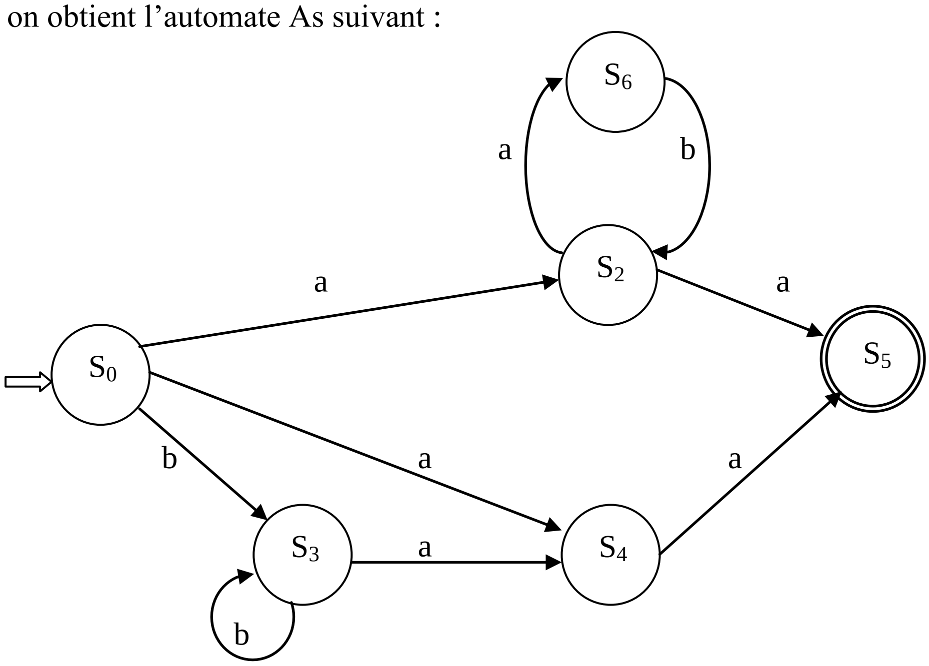
EXERCICE **3** :

D’abord on transforme l’automate généralisé A en automate partiellement généralisé Ap en décomposant les transitions qui se font sur des mots. C’est le cas, pour A, de la transition (ab , S_2 , S_2) ; pour la décomposer on ajoute un nouvel état S_6 et on la remplace par les transitions (a , S_2 , S_6) et (b , S_6 , S_2).

Ensuite, à partir de l’automate partiellement généralisé Ap, on construit l’automate simple As en éliminant les ε-transitions (qu’on appelle aussi les transitions spontanées) suivant les règles :

- si (ε , S_i , S_j) ∈ I et S_j est un état final alors S_i deviendra lui aussi état final ;
- si (ε , S_i , S_j) et (a , S_j , S_k) ∈ I alors ajouter la transition : (a , S_i , S_k) à I. À la fin des ajouts, éliminer la transition (ε , S_i , S_j) de I.

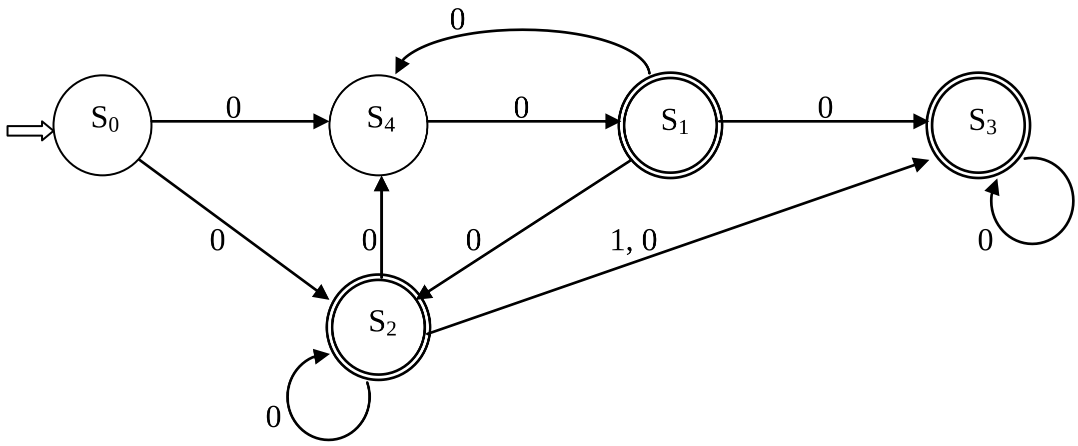
Ainsi, on obtient l’automate As suivant :



Remarque : On n'a pas représenté l'état S_1 car il est devenu inaccessible (c'est-à-dire qu'on ne peut pas l'atteindre à partir de l'état initial S_0) ; on l'a donc supprimé de As.

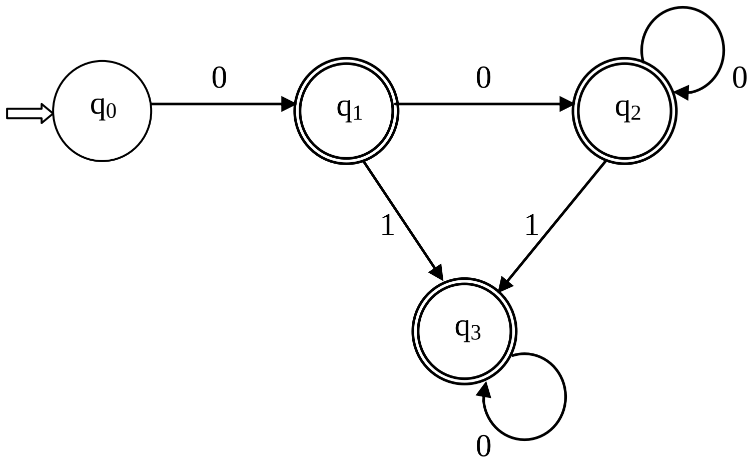
EXERCICE **4** :

1) On construit d'abord l'automate simple As équivalent à A, en procédant de la même manière que dans l'exercice 3 précédent. On obtient l'automate simple (non déterministe) As suivant :



Ensuite, on transforme As en automate déterministe, en procédant de la même manière que dans l'exercice 2.

On obtient l'automate déterministe Ad suivant :



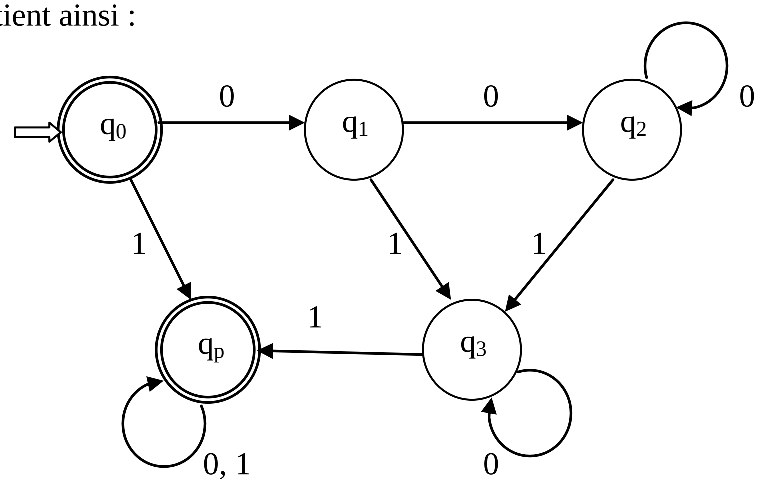
2) Pour obtenir l'automate du complémentaire de L(A) (c'est-à-dire du complémentaire de L(Ad), car A et Ad sont équivalents), on procède comme suit :

I) on prend l'automate simple déterministe Ad, obtenu en 1) ;

II) on complète Ad : on ajoute un état puits – non final – q_p, qu'on raccorde aux états qui on des transitions manquantes (ici, le raccordement se fait en ajoutant les transitions $(1, q_0 , q_p)$ et $(1 , q_3 , q_p)$ – ne pas oublier les transitions $(0, q_p , q_p)$ et $(1, q_p , q_p)$) ;

III) on inverse les états finaux et non finaux : les états non finaux vont devenir finaux et vice versa.

On obtient ainsi :



EXERCICE **5** :

On va traiter les deux questions 1) et 2) pour les deux grammaires g1 et g2.

Pour construire un automate équivalent à une grammaire régulière, on procède comme suit :

i-) on associe à chaque non terminal de la grammaire, un état de l'automate. L'état associé à l'axiome sera l'état initial de l'automate ;

ii-) à toute règle de production $A \rightarrow \varepsilon$ de la grammaire, l'état associé au non terminal A sera final ;

iii-) à toute règle $A \rightarrow wB$, où w est un mot, on ajoutera la transition (w , E_A , E_B) dans l'automate (E_A et E_B sont les états associés à A et B respectivement) ;

iv-) à toute règle $A \rightarrow w$, où w est un mot, on ajoutera la transition (w , E_A , F) dans l'automate (E_A est l'état associé à A, F est un nouvel état final).

Ainsi, en appliquant les étapes *i*-)..*iv*-), avec la grammaire g1 :

$g1 = <\pi, N_1, S, P_1>$ où $\pi = \{a, b, c\}$; $N_1 = \{S, A, B\}$;

$P_1 = \{ S \rightarrow abS \mid bA \mid \varepsilon ;\ A \rightarrow bB \mid B ; B \rightarrow aB \mid b \mid cA \}$;

on obtient l'automate A1 suivant, qui est équivalent à g1 :

$A1 = < V , S , F , S_0 , I >$ où $V = \{a, b, c\}$; $S = \{S_0 , S_1 , S_2, S_3\}$; $F = \{S_0, S_3\}$; S_0 état initial

$I = \{ (ab , S_0 , S_0) ; (b , S_0 , S_1) ; (b , S_1 , S_2) ; (\varepsilon , S_1 , S_2) ; (a , S_2 , S_2) ; (b , S_2 , S_3) ; (c , S_2 , S_1) \}$;

On transforme ensuite cet automate généralisé A1 en automate simple :

- automate partiellement généralisé :

$A1p = < V , S' , F , S_0 , I >$ où $V = \{a, b, c\}$; $S' = \{S_0 , S_1 , S_2, S_3, S_4\}$; $F = \{S_0, S_3\}$; S_0 état initial

$I = \{ (a , S_0 , S_4) ; (b , S_4 , S_0) ; (b , S_0 , S_1) ; (b , S_1 , S_2) ; (\varepsilon , S_1 , S_2) ; (a , S_2 , S_2) ; (b , S_2 , S_3) ;$
$(c , S_2 , S_1) \}$;

- automate simple :

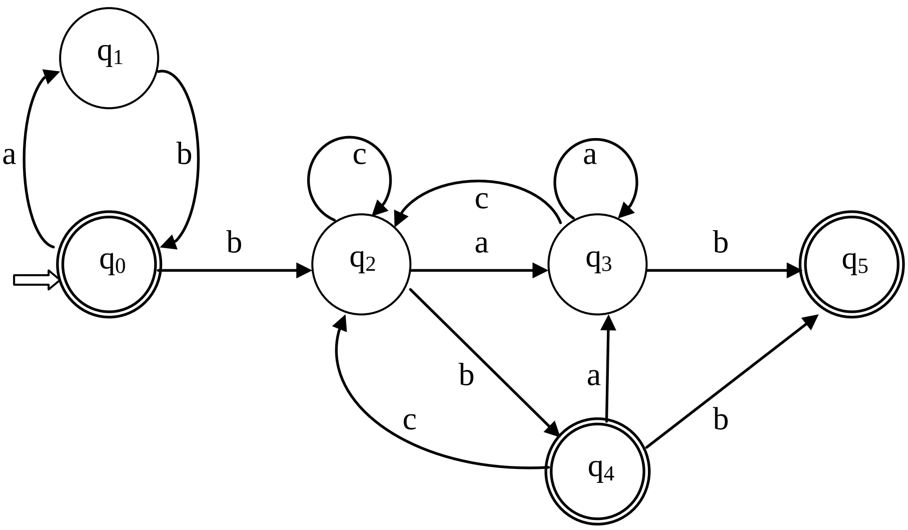
$A1s = < V , S' , F , S_0 , I >$ où $V = \{a, b, c\}$; $S' = \{S_0 , S_1 , S_2, S_3, S_4\}$; $F = \{S_0, S_3\}$; S_0 état initial

$I = \{ (a , S_0 , S_4) ; (b , S_4 , S_0) ; (b , S_0 , S_1) ; (b , S_1 , S_2) ; (a , S_2 , S_2) ; (b , S_2 , S_3) ;$
$(c , S_2 , S_1) ; (a , S_1 , S_2) ; (b , S_1 , S_3) ; (c , S_1 , S_1) \}$;

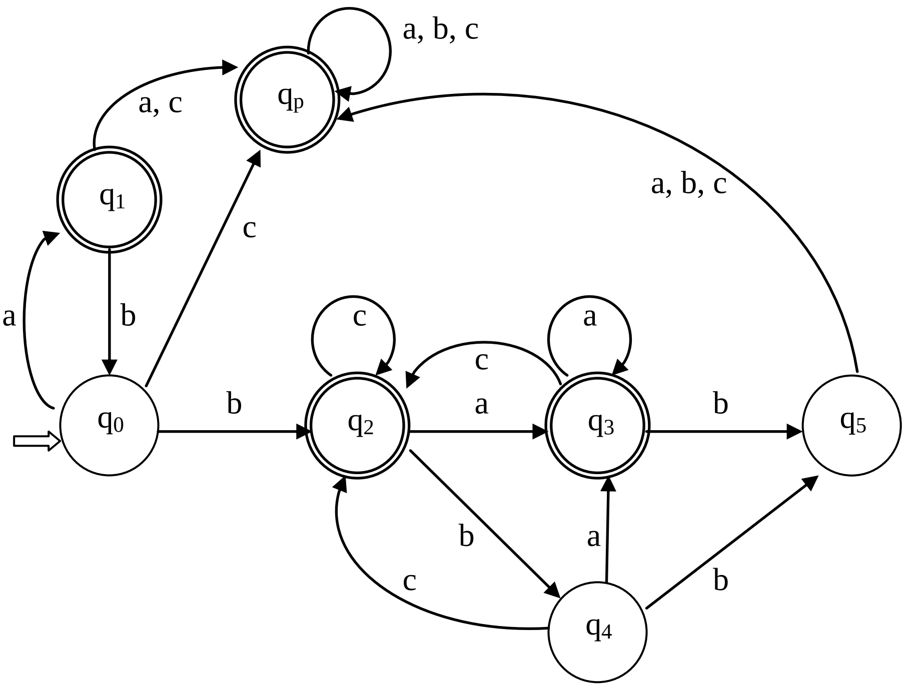
- table de transition de l’automate simple déterministe :

	a	b	c
<S0> = q0	<S4>	<S1>	/
<S4> = q1	/	<S0>	/
<S1> = q2	<S2>	<S2,S3>	<S1>
<S2> = q3	<S2>	<S3>	<S1>
<S2,S3> = q4	<S2>	<S3>	<S1>
<S3> = q5	/	/	/

- automate simple déterministe A1d :



On procède comme c’est fait dans l’exercice 4, pour obtenir l’automate du complémentaire, c’est à dire :
I) on prend l’automate simple déterministe A1d, obtenu précédemment ;
II) on complète A1d ;
III) on inverse les états finaux et non finaux. On obtient :



Avec la grammaire g2 :

$g2 = <\pi, N_2, S, P_2>$ où $\pi = \{a, b, c\}$; $N_2 = \{S, A\}$;

$P_2 = \{ S \rightarrow aS \mid bS \mid cS \mid abcA \; ; \; A \rightarrow aA \mid bA \mid cA \mid \varepsilon \}$.

Remarquons d'abord que L(g2) = ensemble des mots de $\{a, b, c\}^*$ qui **contiennent** la sous-chaîne «**abc**».

Ainsi son complémentaire sera :

l'ensemble des mots de $\{a, b, c\}^*$ qui **ne contiennent pas** la sous-chaîne «**abc**».

En appliquant les étapes *i*-)..*iv*-), avec g2, on obtient l'automate A2 suivant, qui est équivalent à g2 :

$A2 = < V , S , F , S_0 , I >$ où $V = \{a, b, c\}$; $S = \{S_0 , S_1\}$; $F = \{S_1\}$; S_0 état initial

$I = \{ (a , S_0 , S_0) ; (b , S_0 , S_0) ; (c , S_0 , S_0) ; (abc , S_0 , S_1) ; (a , S_1 , S_1) ; (b , S_1 , S_1) ; (c , S_1 , S_1) \}$;

On transforme ensuite cet automate généralisé A2 en automate simple :

- automate partiellement généralisé :

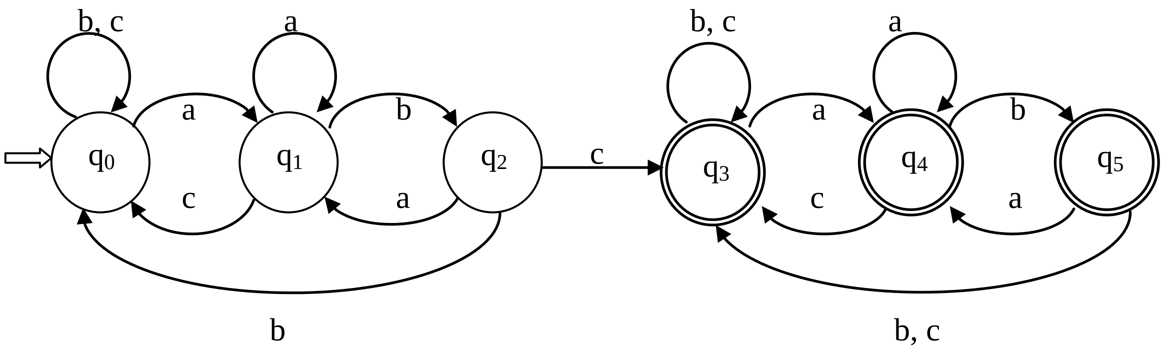
$A2p = < V , S , F , S_0 , I >$ où $V = \{a, b, c\}$; $S = \{S_0 , S_1, S_2, S_3\}$; $F = \{S_1\}$; S_0 état initial

$I = \{ (a , S_0 , S_0) ; (b , S_0 , S_0) ; (c , S_0 , S_0) ; (a , S_0 , S_2) ; (b , S_2 , S_3) ; (c , S_3 , S_1) ; (a , S_1 , S_1) ;$
$(b , S_1 , S_1) ; (c , S_1 , S_1) \}$

Cet automate A2p est simple, car l'automate initial A2 ne contient pas de transitions spontanées. Mais il n'est pas déterministe. Construisons la table de transitions de l'automate déterministe équivalent :

	a	b	c
<S0> = q0	<S0,S2>	<S0>	<S0>
<S0,S2> = q1	<S0,S2>	<S0,S3>	<S0>
<S0,S3> = q2	<S0,S2>	<S0>	<S0,S1>
<S0,S1> = q3	<S0,S2,S1>	<S0,S1>	<S0,S1>
<S0,S2,S1> = q4	<S0,S2,S1>	<S0,S3,S1>	<S0,S1>
<S0,S3,S1> = q5	<S0,S2,S1>	<S0,S1>	<S0,S1>

L'automate déterministe A2d est le suivant :



On procède comme précédemment, pour obtenir l'automate du complémentaire, c'est à dire :

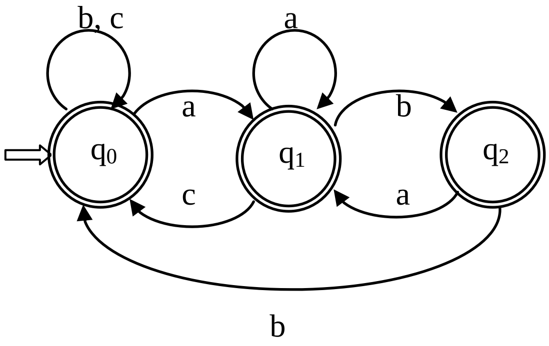
I) on prend l'automate simple déterministe A2d ;

II) on complète A2d (remarquons que A2d est déjà complet) ;

III) on inverse les états finaux et non finaux.

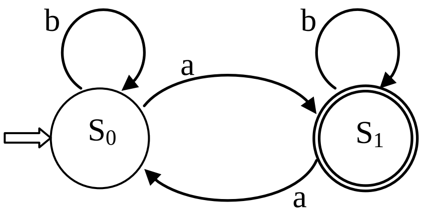
Dans la représentation graphique de l'automate du complémentaire de A2d, on ne représentera pas les états q3, q4 et q5 car ils sont devenus, après l'étape iii), improductifs (il n'y a aucun chemin qui mène de q3, q4 ou q5 vers un des états finaux de l'automate du complémentaire).

On aura donc :

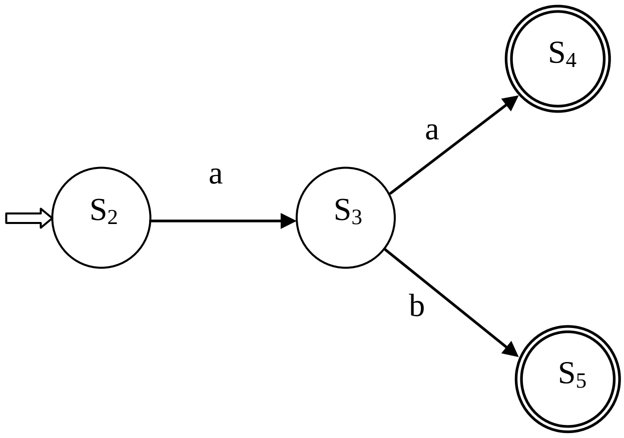


EXERCICE **6** :

1) Automate pour L_1 :



2) Automate pour L_2 :

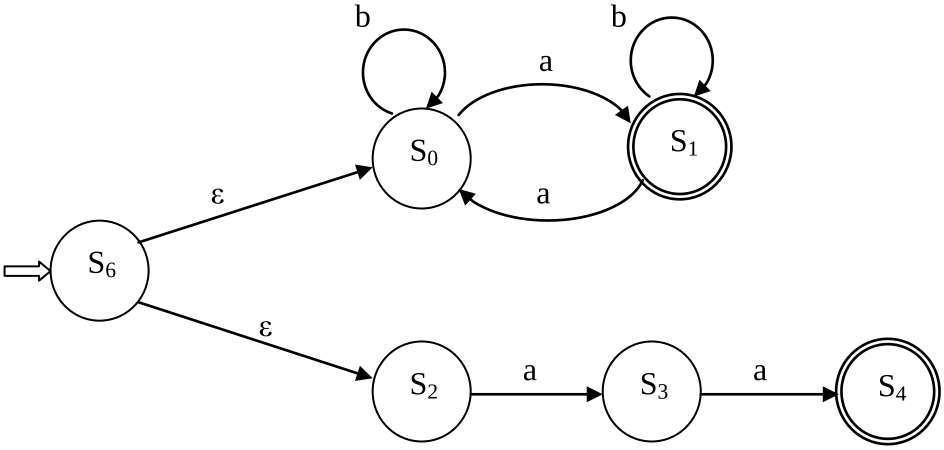


3) Automate pour $L_1 \cup L_2$:

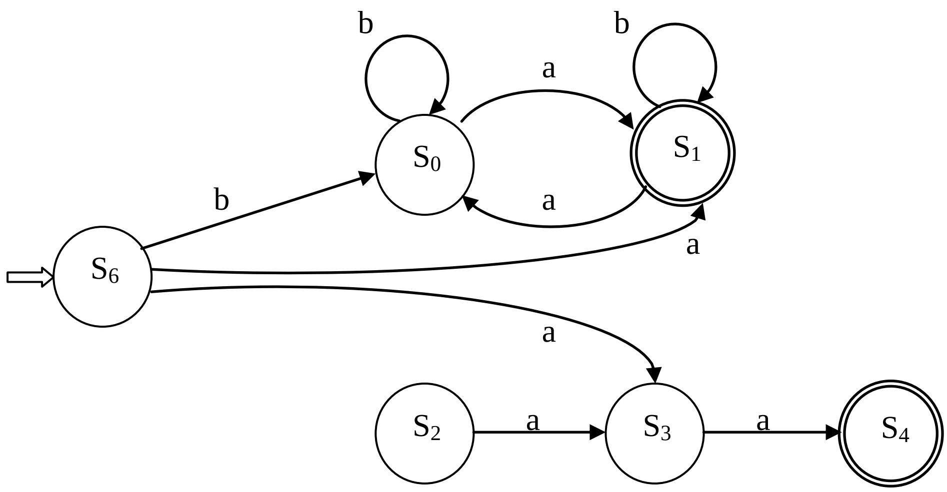
Pour construire l'automate simple de l'union, il suffit d'ajouter un nouvel état initial S_6, qui sera relié aux états initiaux des automates de L_1 et L_2 par des transitions spontanées et de le rendre ensuite simple.

Mais remarquons d'abord que ab $\in L_1$, ce qui fait que $L_1 \cup L_2 = L_1 \cup \{aa\}$.

- Automate partiellement généralisé de l'union :



- Automate simple de l'union (après élimination des ε-règles) :



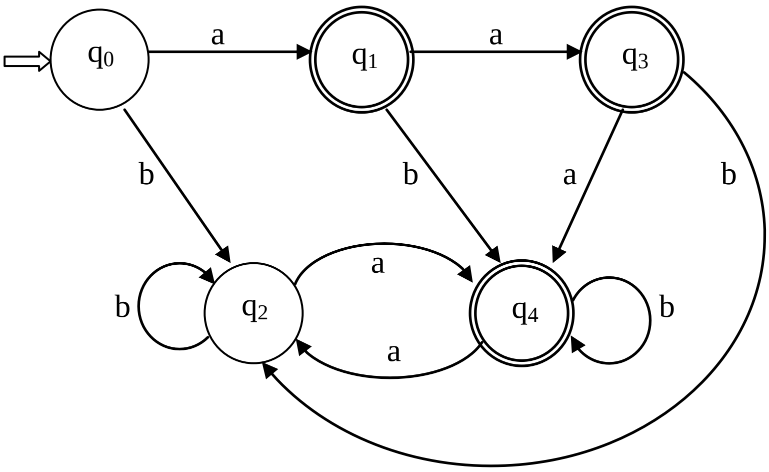
Remarque : l'état S_2 est devenu inaccessible, on peut le supprimer.

4) L'automate de 3) est non-déterministe. Construisons la table de transition de l'automate déterministe :

	a	b
<S_6> = q_0	<S_1,S_3>	<S_0>
<u><S_1,S_3></u> = q_1	<S_0,S_4>	<S_1>
<S_0> = q_2	<S_1>	<S_0>
<u><S_0,S_4></u> = q_3	<S_1>	<S_0>
<u><S_1></u> = q_4	<S_0>	<S_1>

Les états soulignés sont des états finaux.

Automate déterministe de l'union :



EXERCICE **7** :

1) Automate généralisé équivalent à g :

Ag = < V , S , F , S_0 , I > où V = {a, b, c} ; S = {S_0, S_1, S_2, S_3} ; F = {S_2, S_3} ; S_0 état initial

I = { (aa , S_0 , S_1) ; (ab , S_0 , S_2) ; (a , S_1 , S_2) ; (a , S_1 , S_3) ; (b , S_2 , S_2) }.

À partir de Ag, on construit l'automate généralisé qui accepte $L(g)^+$: on ajoute des ε-transitions des états finaux de Ag (S_2 et S_3) vers l'état initial de celui-ci.

Puis, on construit l'automate partiellement généralisé :

Apg = < V , S , F , S_0 , I > où V = {a, b, c} ; S = {S_0, S_1, S_2, S_3, S_4, S_5} ; F = {S_2, S_3} ; S_0 état initial

I = { (a , S_0 , S_4) ; (a , S_4 , S_1) ; (a , S_0 , S_5) ; (b , S_5 , S_2) ; (a , S_1 , S_2) ; (a , S_1 , S_3) ; (b , S_2 , S_2) ;
(ε , S_2 , S_0) ; (ε , S_3, S_0) }.

On procède ensuite à la construction de l'automate simple équivalent :

As = < V , S , F , S_0 , I > où V = {a, b, c} ; S = {S_0, S_1, S_2, S_3, S_4, S_5} ; F = {S_2, S_3} ; S_0 état initial

I = { (a , S_0 , S_4) ; (a , S_4 , S_1) ; (a , S_0 , S_5) ; (b , S_5 , S_2) ; (a , S_1 , S_2) ; (a , S_1 , S_3) ; (b , S_2 , S_2) ;
(a , S_2 , S_4) ; (a , S_2 , S_5) ; (a , S_3 , S_4) ; (a , S_3 , S_5) }.

Puis on fait la déterminisation :

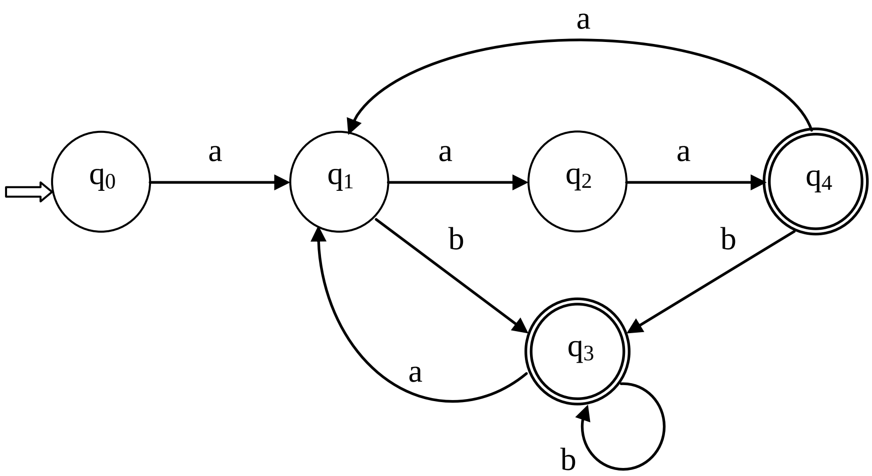
	a	b
<S0> = q0	<S4,S5>	/
<S4,S5> = q1	<S1>	<S2>
<S1> = q2	<S2,S3>	/
<S2> = q3	<S4,S5>	<S2>
<S2,S3> = q4	<S4,S5>	<S2>

L'automate déterministe :

Ad = < V , S , F , S_0 , I > où V = {a, b, c} ; S = {q_0, q_1, q_2, q_3, q_4} ; F = {q_3, q_4} ; q_0 état initial

I = { (a , q_0 , q_1) ; (a , q_1 , q_2) ; (a , q_2 , q_4) ; (b , q_1 , q_3) ; (a , q_3 , q_1) ; (a , q_4 , q_1) ; (b , q_3 , q_3) ;
(b , q_4 , q_3) }.

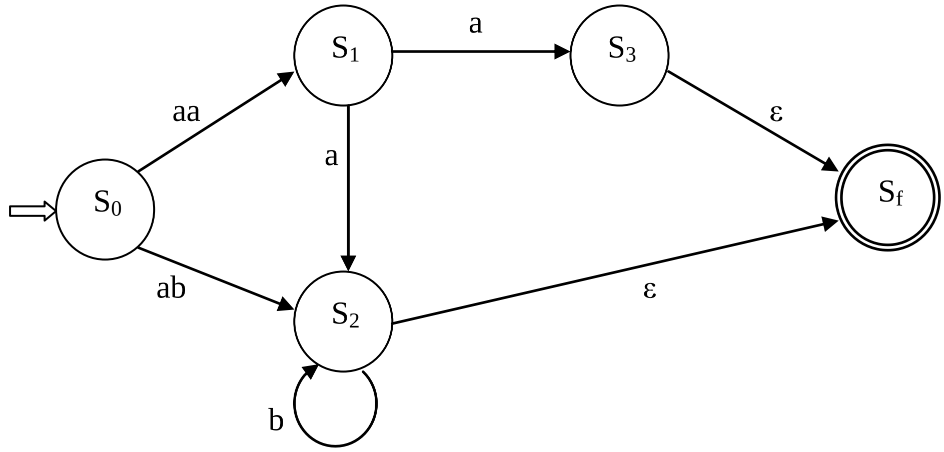
Représentation graphique de Ad :



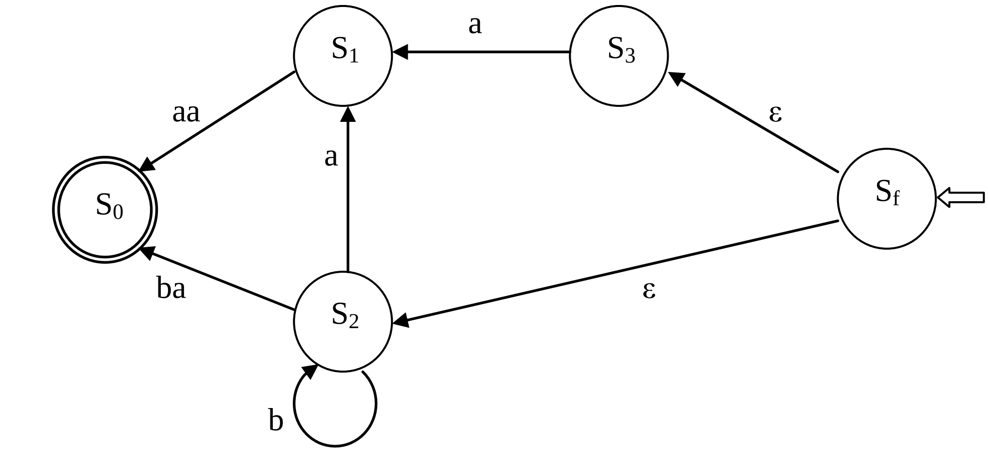
2) Pour trouver un automate (pas forcément simple et déterministe) Ar qui accepte le reflet miroir de L(g) (c'est-à-dire : $L(g)^R$) :

- on prend un automate acceptant L(g), par exemple Ag (de la question précédente) ;
- on le rend avec un seul état final : on ajoute le nouvel état final S_f et on ajoute aussi des ε-transitions des anciens états finaux de Ag vers S_f, typiquement les transitions : (ε , S_2 , Sf) et (ε , S_3 , S_f) ;
- chaque transition (w, Si , Sj) de Ag devient une transition (w^R , Sj , Si) de Ar (cela revient à inverser le sens des flèches, ainsi que les étiquettes, de Ag) ;
- l'état initial de Ar est S_f, l'état final sera S_0 (l'état initial de Ag).

Ag avec un seul état final :



Ar :



<u>EXERCICE **8** :</u>

1) Grammaire régulière à droite g1 pour L_1 :

Terminaux : a, b.

Non-terminaux : S, A

axiome : S

Règles de production : $S \rightarrow bS \mid aA \mid \varepsilon$; $A \rightarrow aA \mid \varepsilon$

Grammaire régulière à droite g2 pour L_2 :

Terminaux : a, b.

Non-terminaux : S

axiome : S

Règles de production : $S \rightarrow baS \mid b$

2) Automate A1 pour L(g1) :
A1 = < V , S , F , S_0 , I > où V = {a, b} ; S = {S_0, S_1} ; F = {S_0, S_1} ; S_0 état initial
I = { (b , S_0 , S_0) ; (a , S_0 , S_1) ; (a , S_1 , S_1) }.

Automate A2 pour L(g2) :
A2 = < V , S , F , S_2 , I > où V = {a, b} ; S = {S_2, S_3} ; F = {S_3} ; S_2 état initial
I = { (ba , S_2 , S_2) ; (b , S_2 , S_3) }.

3) Pour construire l'automate de $L_1.L_2^*$, on va d'abord construire l'automate de L_2^*. Celui-ci s'obtient à partir de A2 en :

- ajoutant des ε-transitions des états finaux de A2 vers l'état initial S_2 ;
- en ajoutant un nouvel état S_4 qui sera le nouvel état initial, il sera aussi final (pour accepter ε) ;
- S_4 sera relié à l'ancien état initial S_2 par une ε-transition.

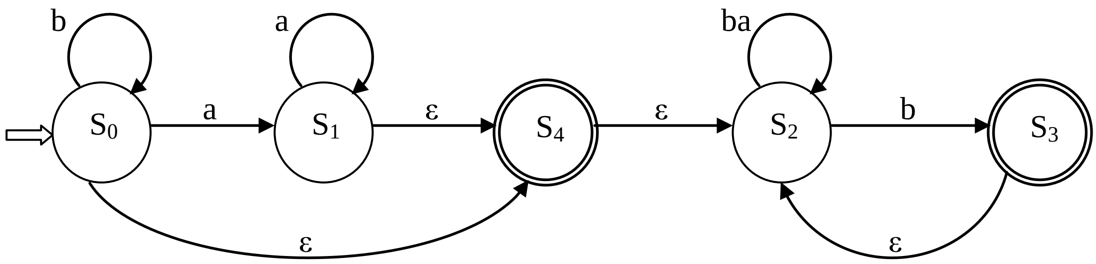
Soit A2it l'automate de l'itération obtenu :
A2it = < V , S , F , S_2 , I > où V = {a, b} ; S = {S_2, S_3, S_4} ; F = {S_4, S_3} ; S_4 état initial
I = { (ε , S_4 , S_2) ; (ba , S_2 , S_2) ; (b , S_2 , S_3) ; (ε , S_3 , S_2) }.

Pour construire l'automate Ac de la concaténation, on procède comme suit :

- on prend les deux automates A1 et A2it ;
- l'état initial de Ac sera l'état initial de A1 (c'est-à-dire S_0) ;
- les états finaux de Ac seront ceux de A2it (c'est-à-dire S_3 et S_4) ;
- on ajoute des ε-transitions des états finaux de A1 (S_0, S_1) vers l'état initial de A2it (S_4).

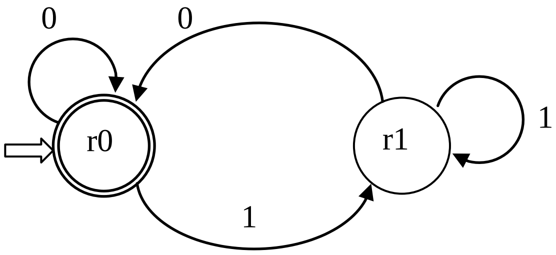
On obtient donc :
Ac = < V , S , F , S_0 , I > où V = {a, b} ; S = {S_0, S_1, S_2, S_3, S_4} ; F = {S_4, S_2} ; S_0 état initial
I = { (b , S_0 , S_0) ; (a , S_0 , S_1) ; (a , S_1 , S_1) ; (ε , S_4 , S_2) ; (ba , S_2 , S_2) ; (b , S_2 , S_3) ; (ε , S_3 , S_2) ;
(ε , S_0 , S_4) ; (ε , S_1 , S_4) }.

Graphiquement :

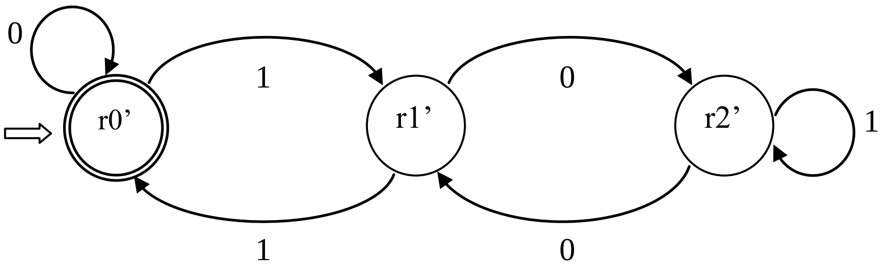


EXERCICE **9** :

D'abord l'automate A de L :



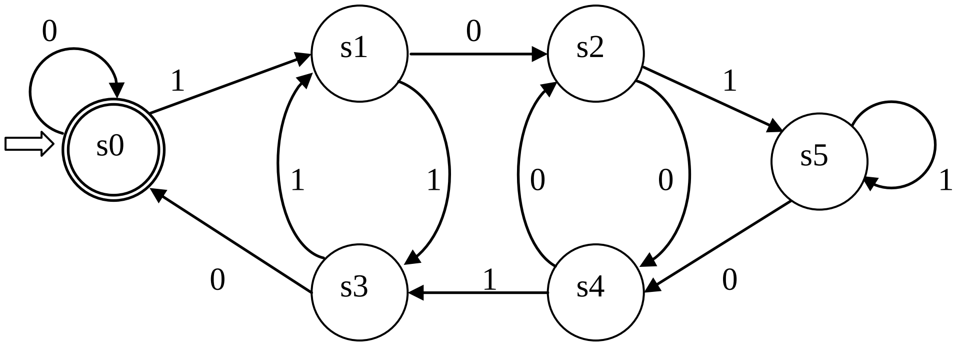
Puis l'automate B de L' :



A partir des automates de L et L', on procède comme indiqué dans l'énoncé de l'exercice pour obtenir la table de transition de l'intersection L'':

	0	1
(r0,r0') = s0	(r0,r0')	(r1,r1')
(r1,r0') = s3	(r0,r0')	(r1,r1')
(r0,r1') = s4	(r0,r2')	(r1,r0')
(r1,r1') = s1	(r0,r2')	(r1,r0')
(r0,r2') = s2	(r0,r1')	(r1,r2')
(r1,r2') = s5	(r0,r1')	(r1,r2')

D'où le graphe de l'automate :



Remarque : L'automate obtenu représente les chaînes de 0 et 1 qui symbolisent les entiers naturels divisibles par six (6).

-------------------------- Fin du corrigé de la série 2 de ThL --------------------------