Département Informatique Année 2019-2020
FEI - USTHB
L2-ACAD (Section A & C)

Théorie des Langages

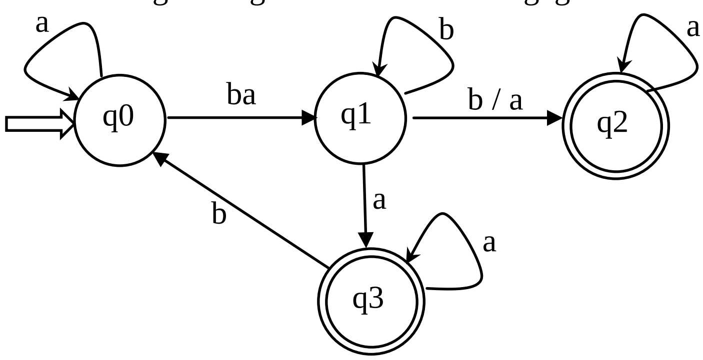
Série 2

LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

Exercice 3

Pour chacun des automates d'états finis suivants :

1. Trouver un automate déterministe équivalent
2. Donner une grammaire régulière générant le même langage

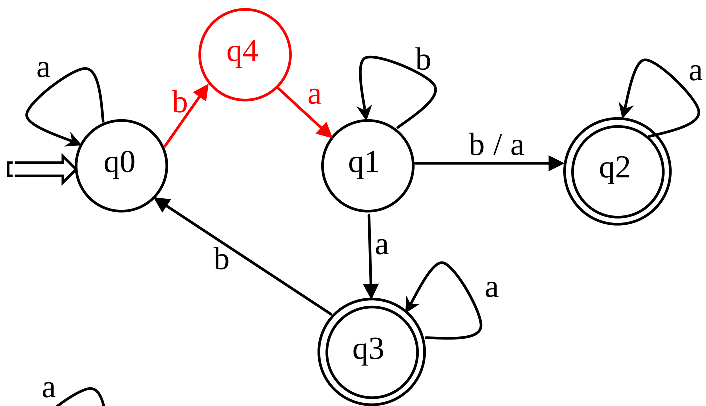


Cet automate est généralisé, car il existe une transition étiquetée par un mot (ba) qui va de q0 à q1. Donc il faut le rendre tout d'abord simple.

Pour ce faire, il suffit d'éclater la transition de q0 à q1 en deux transitions : la première étiquetée par **b** et la deuxième étiquetée par **a** et ajouter un nouvel état q4.
On obtient un automate simple non déterministe qu'il faut rendre déterministe.

Pour le rendre déterministe, on utilise un tableau (forme matricielle) et on procède à un regroupement des états comme illustré dans la figure ci-dessous (voir tableau de déterminisation). Rappelons que pour alléger les notations, les accolades sont omises et ainsi q2q3 correspond à l'ensemble {q2, q3}.

Automates Simple Non Déterministe



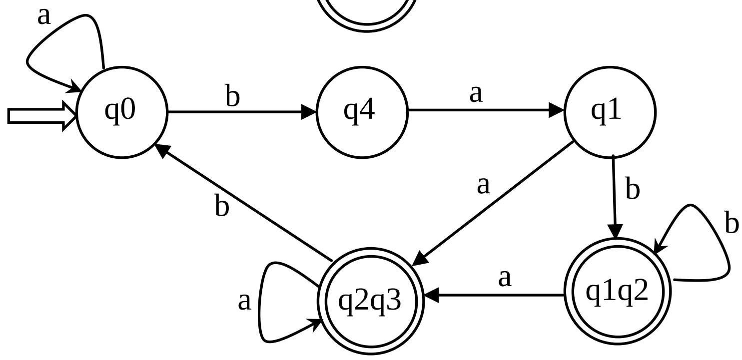
Représentation Matricielle

	a	**b**
q0	q0	q4
q1	q2q3	q1q2
q2	q2	-
q3	q3	q0
q4	q1	-

Déterminisation

	a	**b**
q0	q0	q4
q4	q1	-
q1	q2q3	q1q2
q2 q3	q2q3	q0
q1q2	q2q3	q1q2

Représentation Graphique



Automates Simple Déterministe

Département Informatique Année 2019-2020
FEI - USTHB
L2-ACAD (Section A & C) **Théorie des Langages**

Série 2

LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

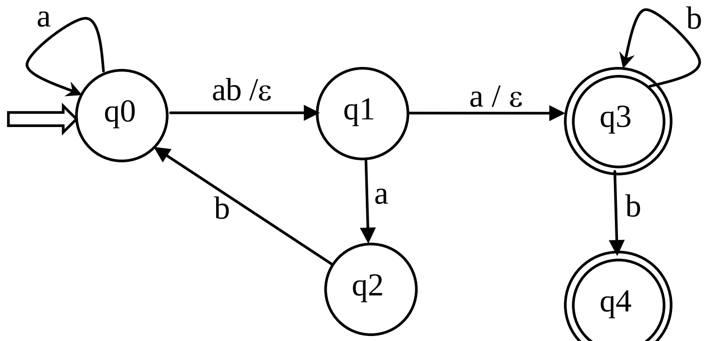
Rappelons qu'un état de l'automate déterministe est final si et seulement si il contient un état final de l'automate non-déterministe (initial). Ainsi, les états q2q3 et q1q2 sont finaux.

Pour trouver la grammaire régulière droite qui génère le langage reconnu par l'automate, on procède tout d'abord à un renommage des états : {q0} par S, {q4} par A, {q1} par B, {q2q3} par C, {q1q2} par D. Ainsi, la grammaire régulière droite équivalente est :
G=<{a,b}, {S,A,B,C,D}, S, P> où P est défini par :

$S \rightarrow aS$ / bA
$A \rightarrow aB$
$B \rightarrow aC$ / bD
$C \rightarrow aC$ / bS / ε
$D \rightarrow aC$ / bD / ε

Remarque : On peut obtenir une grammaire régulière droite à partir de n'importe quel type d'automates d'états finis (pas nécessairement un automate simple non déterministe).

2) Le deuxième automate :

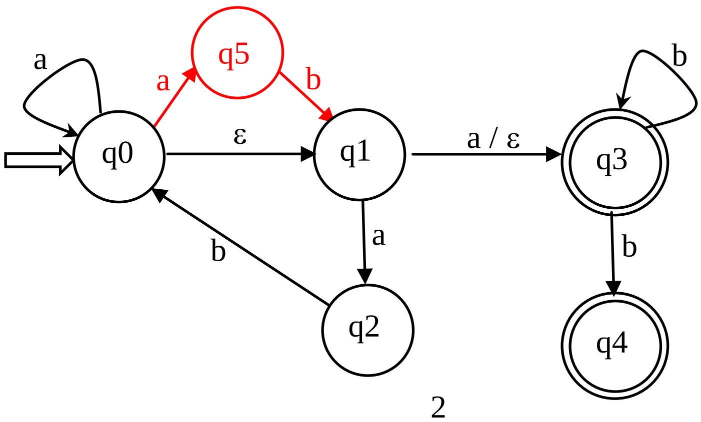


Cet automate est généralisé, car il existe une transition étiquetée par un mot (ab) qui va de q0 à q1. Donc il faut le rendre tout d'abord simple.

Pour ce faire, il suffit d'éclater la transition de q0 à q1 en deux transitions : la première étiquetée par **a** et la deuxième étiquetée par **b** et ajouter un nouvel état q5.

On obtient un automate partiellement généralisé (il existe des transitions spontanées). Donc, il faut éliminer ces transitions en passant par le tableau.

Automate Partiellement Généralisé



Département Informatique Année 2019-2020
FEI - USTHB
L2-ACAD (Section A & C)

Théorie des Langages

Série 2

LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

	a	b	ε
q0	q0q5	-	q1
q1	q2q3	-	q3
q2	-	q0	-
q3	-	q3q4	-
q4	-	-	-
q5	-	q1	-

Eliminer ε-transition de q1 vers q3

	a	b	ε
q0	q0q5	-	q1
q1	q2q3	q3q4	~~q3~~
q2	-	q0	-
q3	-	q3q4	-
q4	-	-	-
q5	-	q1	-

Eliminer ε-transition de q0 vers q1

	a	b	ε
q0	q0q2q3q5	q3q4	~~q1~~
q1	q2q3	q3q4	~~q3~~
q2	-	q0	-
q3	-	q3q4	-
q4	-	-	-
q5	-	q1	-

L'obtention de l'automate simple s'est faite en éliminant les transitions spontanées comme suit :

On commence par éliminer la transition spontanée de q1 à q3 et tout ce qui était accessible à partir de q3 (q3 par b et q4 par b), devient accessible à partir de q1 (par les mêmes lettres). Après suppression de la ε-transition, l'état q1 devient final car l'état q3 est final. Pour cela, il a suffi de translater la ligne du tableau étiquetée par q3 et de la fusionner avec la ligne étiquetée par q1.

Dans une deuxième étape, on élimine la transition spontanée allant de q0 à q1 par le même principe et l'état q0 devient final.

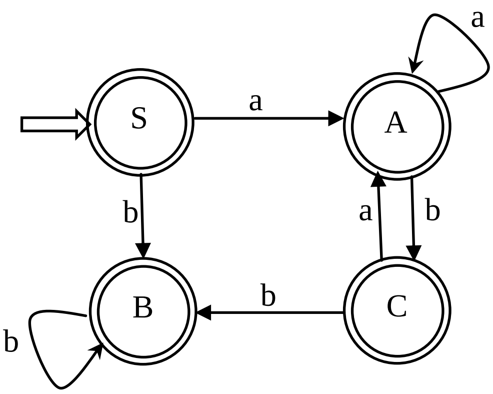
Apres l'obtention de l'automate simple non déterministe représenté par le dernier tableau (il n y a plus de transitions spontanées), on procède à sa déterminisation, par un regroupement des états.

Déterminisation

	a	b
q0	q0q2q3q5	q3q4
q0q2q3q5	q0q2q3q5	q0q1q3q4
q3q4	-	q3q4
q0q1q3q4	q0q2q3q5	q3q4

On procède à un renommage des états : {q0} par S, {q0q2q3q5} par A, {q3q4} par B, {q0q1q3q4} par C.

Ainsi, l'automate simple déterministe équivalent est le suivant :



Automate Simple Déterministe

Théorie des Langages
Série 2
LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

La grammaire régulière droite équivalente est :
G=<{a,b}, {S,A,B,C}, S, P> où P est défini par :

S→aA / bB / ε
A→aA / bC / ε
B→ bB / ε
C→aA / bB / ε

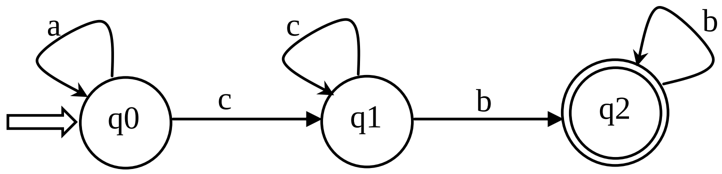
Rappels : la grammaire est obtenue par les règles suivantes : les états correspondent aux non terminaux, l'état initial à l'axiome, les transitions aux productions. Chaque état final est représenté par une production d'arrêt avec epsilon.

Exercice 4 :

L'objectif de cet exercice est de construire un automate d'états finis d'un langage en se basant sur les automates d'états finis des langages le composant.

Donner les automates d'états finis des langages suivants :

1) $L_1= \{a^nc^pb^m / \ n \geq 0 \text{ et } p, m\geq 1\}$

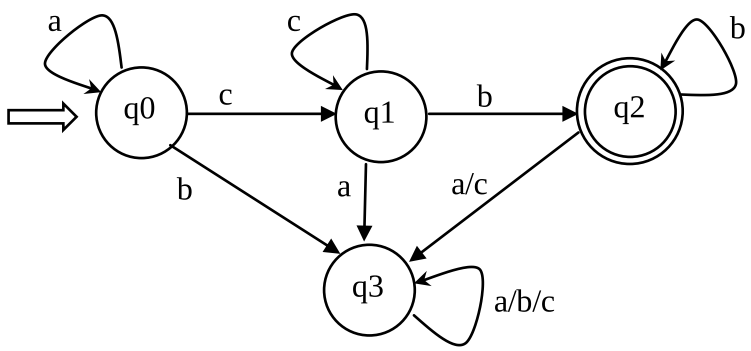


Les mots de L_1 sont composés d'une suite de **a**, suivie d'une suite de **c** suivie d'une suite de **b** avec au moins un **c** et au moins un **b**.

2) $L_2=\overline{L_1}$

L_2 est le complément de L_1 ainsi L_2 contient tous les mots qui appartiennent à X* et qui n'appartiennent pas à L_1. Donc, on pourra construire l'automate de L_2 à partir de celui de L_1.

Soit A_1 l'automate reconnaissant L_1 obtenu dans la question précédente. Notons que A_1 est déterministe mais non complet. Donc, on commence par rendre A_1 complet en ajoutant un état puits q3 et toutes les transitions manquantes vers cet état; on obtient un automate A'_1 équivalent à A_1 (voir figure ci-dessous). Rappelons qu'un automate déterministe complet permet de **lire tout mot de X***. Donc, les mots de L_2 sont ceux lus par cet automate sans être reconnus.

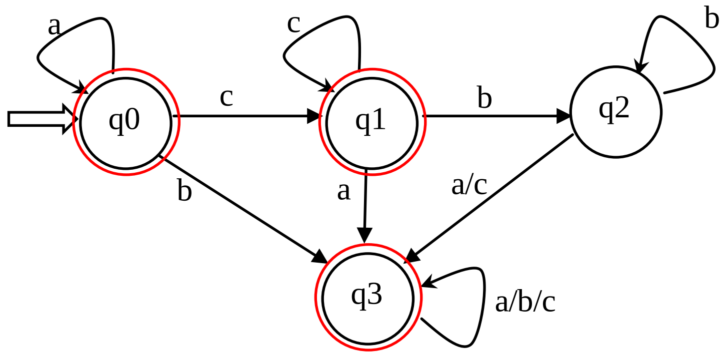


Théorie des Langages

Série 2

LES AUTOMATES A ETATS FINIS & LANGAGES REGULIERS

Pour reconnaitre les mots qui n'appartiennent pas à L_1, il faut rendre l'état final de A'_1 (qui est q2) non final et les états non finaux (q0, q1, q3), il faut les rendre finaux. L'état initial ne change pas.



D'une manière générale, soit l'automate déterministe et complet $A_1=(X_1, Q_1, q0_1, \delta_1, F_1)$ reconnaissant un langage L_1 (si un automate déterministe donné n'est pas complet, il faut le rendre tout d'abord complet). L'automate A=(X, Q, q0, δ, F) reconnaissant le langage complément de L_1 est défini comme suit :

$X=X_1$ /* L'alphabet de A est celui de A_1*/
$Q=Q_1$ /* Les états de A sont ceux de A_1 (ni ajout, ni suppression d'états)*/
$q0=q0_1$ /* L'état initial de l'automate A est celui de A_1*/
$F=Q\backslash F_1$ /* Les états finaux de l'automate A sont les états qui n'étaient pas finaux dans A_1*/
$\delta=\delta_1$ /* Maintenir toutes les transitions */