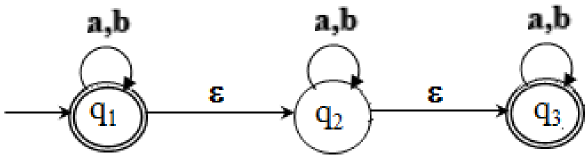
Université de Batna 2
Département d'informatique
$2^{ème}$ année Licence académique
Module : théorie des langages

Série N° 02 de la semaine:
L'automate d'états finis et la grammaire associée

Objectif du TD **: est de reprendre quelques algorithmes sur les automates d'états finis et d'apprendre le passage entre les deux représentations du langage: l'automate et la grammaire.**

Exercice: **……………………..** *(50 min au maximum)*

1. Supprimez convenablement les ε -transitions de cet automate d'états finis.



✓ à partir de l'automate les états concernés par l' ε -successeur sont *l'état* q_1 et *l'état* q_2

✓ à partir de l'automate on trouve: ε -successeur (q_1) = {$\boldsymbol{q_2}$, $\boldsymbol{q_3}$} et ε -successeur (q_2) = {$\boldsymbol{q_3}$}

✓ <u>Le statut des états concernés :</u>

✍ *L'état* q_1 est déjà final, donc ce n'est pas la peine cherche d'un état final dans son ε - successeur. .

✍ *L'état* q_2 devient final, puisque il existe un état final dans son ε -successeur

✓ <u>Les transitions ajoutées si les transitions n'existent pas dans l'automate:</u>

✍ Pour *l'état* q_1 :

On a cette transition non vide ∂ ($\boldsymbol{q_2}$, a)= q_2 C'est pour ça on ajoute la transition ∂ (q_1 , a)= q_2 dans l'automate (puisque elle n'existe pas).

On a cette transition non vide ∂ ($\boldsymbol{q_2}$, b)= q_2 C'est pour ça on ajoute la transition ∂ (q_1 , b)= q_2 dans l'automate (puisque elle n'existe pas).

On a cette transition non vide ∂ ($\boldsymbol{q_3}$, a)= q_3 C'est pour ça on ajoute la transition ∂ (q_1 , a)= q_3 dans l'automate (puisque elle n'existe pas).

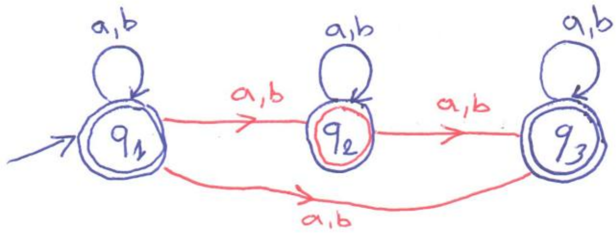
On a cette transition non vide ∂ ($\boldsymbol{q_3}$, b)= q_3 C'est pour ça on ajoute la transition ∂ (q_1 , b)= q_3 dans l'automate (puisque elle n'existe pas).

✍ Pour *l'état* q_2 :

On a cette transition non vide ∂ ($\boldsymbol{q_3}$, a)= q_3 C'est pour ça on ajoute la transition ∂ (q_2 , a)= q_3 dans l'automate (puisque elle n'existe pas).

On a cette transition non vide ∂ ($\boldsymbol{q_3}$, b)= q_3 C'est pour ça on ajoute la transition ∂ (q_2 , b)= q_3 dans l'automate (puisque elle n'existe pas).

✓ Maintenant, dans le graphe on supprime toutes les ε -transitions et on fait le changement nécessaire (l'ajout des nouvelles transitions et le changement du statut de q_2).

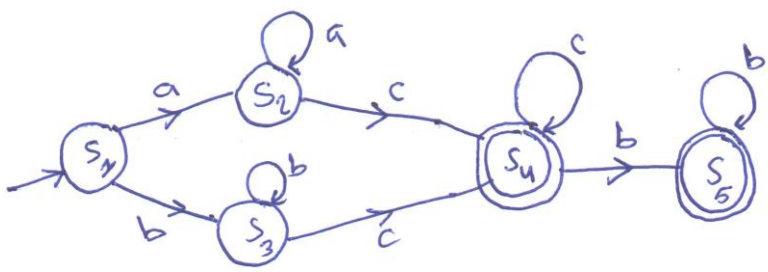


2. **Déterminisez cet automate d'états finis.**

✍ Donc, la table de transition de l'automate déterministe est:

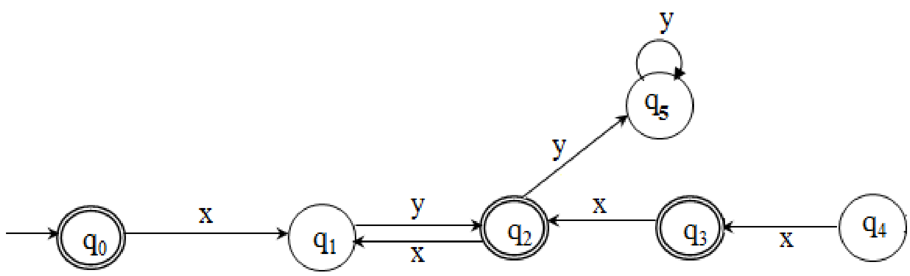
	a	b	c
→ $\{q_1, q_2\}$	$\{q_1, q_3\}$	$\{q_2, q_3\}$	-
$\{q_1, q_3\}$	$\{q_1, q_3\}$	-	$\{q_3, q_4\}$
$\{q_2, q_3\}$	-	$\{q_2, q_3\}$	$\{q_3, q_4\}$
$\{q_3, q_4\}$	-	q_4	$\{q_3, q_4\}$
q_4	-	q_4	-

✍ Le graphe de cet automate déterministe après le renommage des états est:



3.

3.1. Minimisez cet automate d'états finis.



✓ On élimine de l'automate tous les états inaccessibles et non co-accessibles avec leurs transitions.

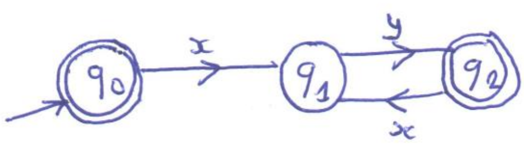
<u>Les états inaccessibles sont :</u>

L'état q_3 et par conséquent *l'état* q_4 ; on les supprime avec leurs transitions

<u>Les états non co-accessibles sont :</u>

L'état q_5 ; on le supprime avec ses transitions

✍ Le graphe de cet automate après la suppression :



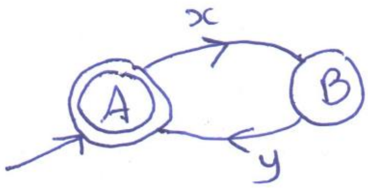
✓ On partitionne les états en deux classes, A = $\{q_0, q_2\}$ B = $\{q_1\}$

✓ On construit à chaque fois une table de transition dès qu'il existe une nouvelle classification.
La première table :

		Les symboles	
Classe obtenue	Ses états	x	y
→ A	q_0	**B**	—
	q_2	**B**	—
B	q_1	—	**A**

✎ On voit que la classe B contient un seul état, donc on ne peut pas l'éclater.
✎ On voit que la classe A contient deux états ayant tous le même résultat sur chaque symbole, donc on ne peut pas l'éclater.

✓ Comme il n'y ait plus d'éclatement, on s'arrête.
✓ La dernière table décrit l'automate minimal.
✓ Le graphe de l'automate minimal est représenté comme suit :



3.2. Trouvez son langage reconnu via le lemme d'Arden.

✓ Le système d'équations associé à notre automate minimal est :

$$\begin{cases} L(A) = x\,L(B) + \varepsilon & \textbf{..........(1)} \\ L(B) = y\,L(A) & \textbf{...........(2)} \end{cases}$$

✓ Le langage reconnu par notre automate est donnée par le langage engendré par l'état initial A.

C'est-à-dire $L(A)$.

✓ On cherche $L(A)$ en utilisant le **lemme d'Arden.**

☞ En remplaçant ***(2)*** dans ***(1)***, on trouve que :

$L(A) = x\,y\,L(A) + \varepsilon$ ***..........(3)***

☞ En appliquant maintenant le lemme sur ***(3)***, on trouve que:

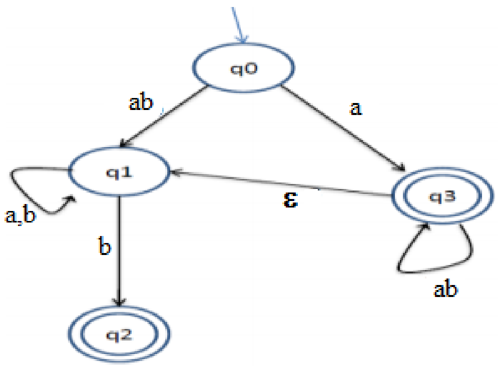
$L(A) = (xy)^{*}\varepsilon$ ***...............(4)*** D'après $\forall t : t\,\varepsilon = \varepsilon\,t = t$, la formule ***(4)*** devient :

$L(A) = (xy)^{*}$ ***...............(5)***

✍ Donc le langage reconnu par notre automate est l'expression suivante qui ne contient que des symboles de x et y.

C'est-à-dire $(xy)^{*}$

4. Soit l'AEF généralisé ci-dessous :



☞ **Donnez les règles de la grammaire associée.**

✓ Le principe de la constitution des **règles de la grammaire** associé à un automate est similaire à celui de constitution du système d'équations.
Un séparateur « / » au lieu de «+ ». Ils signifient «ou ».
Une affectation «→» au lieu de «= ». Ils signifient «est remplacé par».

$q_0 \to abq_1 \,/\, aq_3$

$q_1 \to aq_1 \,/\, bq_1 \,/\, bq_2$

$q_2 \to \varepsilon$

$q_3 \to abq_3 \,/\, q_1 \,/\, \varepsilon$

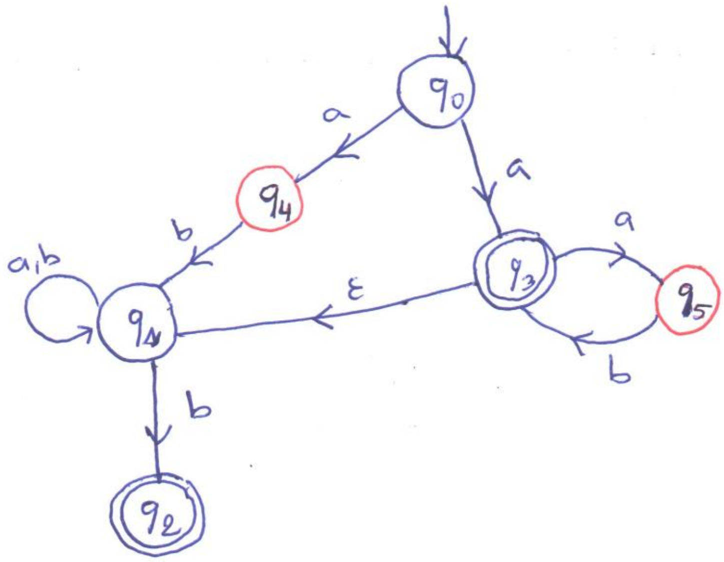
☞ **Citez les étapes pour le rendre simple.**

✓ éliminez les transitions qui portent un mot de longueur plus d'un symbole en ajoutant des états intermédiaires.

✍ Dans l'automate : on ajoute un autre état entre q_0 et q_1

on ajoute aussi un autre état dans la boucle entre q_3 et q_3

On obtient ce graphe :



✓ Ensuite éliminez les ε-transitions en suivant l'algorithme d'élimination.

☞ **Déduire son langage reconnu.**

✓ Le langage reconnu par un automate est l'ensemble des mots construits à partir de l'état initial jusqu'à l'état final.

✍ La formule globale du langage reconnu par cet automate est :

$$= \quad ab(a+b)^* b + a(ab)^* (a+b)^* b + a(ab)^*$$

Bon courage