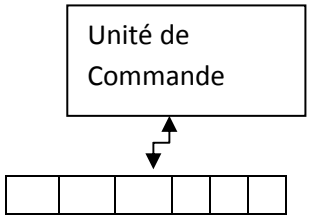
# *Chapitre IV : Les automates d'états finis et les langages réguliers*

I- ***Introduction :*** un automate d'états fini est la machine reconnaissant des langages réguliers de type 3, il est caractérisé par :

- Un ruban en entrée fini.
- La tête est une tête de lecture seulement qui se déplace uniquement vers la droite d'une case à la fois.




## *II- Définitions :*

### *1- Définition formelle :*

Un automate à états fini est une machine abstraite définie par le quintuplet **(V,Q, q0, F,$\delta$)** tel que :

- V est l'ensemble des symboles formant les mots en entrée (l'alphabet du mot à reconnaître);
- Q est l'ensemble des états possibles ;
- q0 est l'état initial ;
- F est l'ensemble des états finaux ($F \neq \emptyset$ , $F \subseteq Q$). F représente l'ensemble des états d'acceptation;
- $\delta$ est une fonction de transition qui permet de passer d'un état à un autre selon l'entrée en cours :
  - $\delta : Q \times (X \cup \{\varepsilon\}) \rightarrow 2^Q$
  - $\delta$ (qi, a) = {qj1 , ..., qjk} ou $\emptyset$ ($\emptyset$ signifie que la configuration n'est pas prise en charge)

### 2- *Représentation d'un AEF* :

Il existe 3 manières pour représenter un AEF :

- Par sa définition formelle en précisant les 5 paramètres et en particulier la fonction $\delta$
- Par la table des valeurs de $\delta$ représentée par une matrice où les lignes sont les états et les colonnes sont les caractères de V. En précisant l'état initial et les états finaux.
- Par un graphe de transitions où les sommets représentent les états, les flèches représentent les transitions étiquetées par les caractères lus. On

précise l'état initial par le symbole « ⟶» entrant et les états finaux par 2 cercles.

**Exemple :**

1- Représentation par la définition formelle :

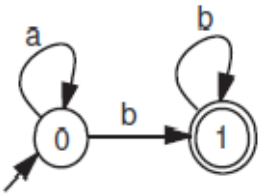
A=({a,b},{0,1},0,{1}, $\delta$) avec :

$\delta$(0,a)=0 $\delta(0,b)=1$ $\delta$ (1,b)=1

2- Représentation par table des valeurs :

| Caractère / Etat | a | b |
|---|---|---|
| ⟶0 | 0 | 1 |
| [1] | / | 1 |

3- Représentation par graphe :




### 3- ***Langage reconnu par un AEF :***

- Un mot est accepté par un AEF si, après avoir lu tout le mot, l'automate se trouve dans un état final **($q_f \in F$).** En d'autres termes, un mot est rejeté par un AEF dans deux cas :
  - L'automate est dans l'état **$q_i$**, l'entrée courante étant a et la transition **$\delta(q_i, a)$** n'existe pas (on dit que l'automate est *bloqué*) :
  - L'automate arrive à lire tout le mot mais l'état de *sortie* n'est pas un état final.
- Le langage reconnu par un AEF « A » est l'ensemble des mots reconnus par A : L(A) = {w / $\delta(q0, w) \in$ F}
- Le langage L est régulier s'il est accepté par un automate d'états fini.

## *III- Automate d'états fini et grammaire régulière :*

Théorème : un langage reconnu par un AEF est engendré par une grammaire de type 3 et réciproquement.

1- ***Passage d'un AEF à une grammaire de type 3 :*** soit A= (V,Q, q0, F,$\delta$) il existe une grammaire de type 3 tel que : L(A)=L(G), G est déterminée comme suit :

G=(V,Q,q0,R) donc $V_T$=V et $V_N$=Q et S=q0 et R est défini comme suit :

- Si $\delta(q,x) = q'$ $alors$ $q \longrightarrow xq' \in R$
- $Si\ q \in F$ $alors$ $q \longrightarrow \varepsilon \in R$
- $Si\ \varepsilon \in L(A)$ $alors\ on\ ajoute$ $q0 \longrightarrow \varepsilon \in R$

2- **Passage d'une grammaire de type 3 à un automate d'états fini :** soit G=($V_T$,$V_N$,S,R) de type 3, il existe un automate d'états fini A tel que L(G)=L(A). A est déterminé comme suit :

- V = $V_T$
- Q = $V_N$ ∪ F
- q0 = S
- $\delta\ est\ définie\ par$ :
  - Si A → $aB \in R\ alors\ \delta(A,a) = B$
  - Si A → $a \in R\ alors\ \delta(A,a) = X\ et\ X \in F$ (un nouvel état final)
- F = {X / $\delta(A,a)$=X}
- Si $\varepsilon$ ∈ L(G) alors l'état initial est aussi final.

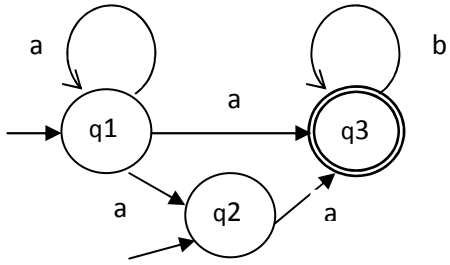
## IV- *Automate fini non déterministe :*

1- ***Définition :*** un automate fini non déterministe AFN est le quintuplet (V,Q,I,F, $\delta$) tel que :

– V est l'ensemble des symboles formant les mots en entrée (l'alphabet du mot à reconnaître);
– I ⊆ Q est l'ensemble des états initiaux ;
– F est l'ensemble des états finaux
– $\delta$ est une fonction de transition qui permet de passer d'un état à un autre selon l'entrée en cours :
  - $\delta$ : Q × (V ∪ {ε}) → P(Q)
  - $\delta$ (qi, a) = {qj1 , ..., qjk}

***Exemple1 :***

| | A | b |
|---|---|---|
| →q1 | q1,q2,q3 | / |
| →q2 | q3 | / |
| q3 | / | q3 |




***2- Fonctionnement et langage reconnu par un AFN :***

- Soit A=(V,Q,I,F, $\delta$) on définit la fonction de transition itérée sur cet automate : $\delta *: P(Q) \times$ V* → P(Q) par :

$$\delta*(X,w) = \begin{cases} X & si\ w = \varepsilon \\ \bigcup_{q \in \delta*(X,w0)} \delta(q,x) & pour\ w = w0x \end{cases}$$

- Le langage reconnu par un AFN est définit par :

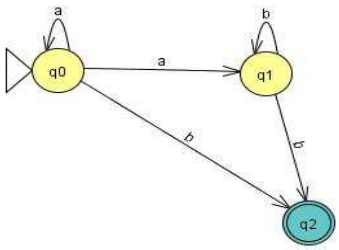
$$L(A) = \{w \ / \ w \in V* \ \ et\ (q,w) \vdash X\ avec\ q \in I\ et\ X \cap F \neq \emptyset\}$$

***Exemple :* fonctionnement de l'automate précédent sur le mot ab à partir de q1 :**

$$\delta*(q1,ab) = \delta*(\delta(q1,a),b) = \delta*(\{q1,q2,q3\},b)$$
$$= \delta*(\delta(q1,b) \cup \delta(q2,b) \cup \delta(q3,b),\varepsilon) = \delta*(\emptyset \cup \emptyset \cup \{q3\},\varepsilon) = \{q3\}$$

le mot a été lu entièrement et on termine avec un ensemble qui contient un état final donc le mot ab est reconnu par l'automate.

***Exemple 2 :* soit l'AFN suivant, est-ce que les mots aab, aa, aba sont reconnus par cet AFN ?**




1- **aab**

$$\delta*(q0,aab) = \delta*(\delta(q0,a),ab) = \delta*(\{q0,q1\},ab) = \delta*(\delta(\{q0,q1\}a),b)$$
$$= \delta*(\{q0,q1\},b) = \delta*(\delta(q0,b) \cup \delta(q1,b),\varepsilon) = \{q2,q3\}$$

le mot a été lu entièrement et on termine avec un ensemble qui contient un état final donc le mot aab est reconnu par l'automate.

***2-* aa**

$$\delta*(q0,aa) = \delta*(\delta(q0,a),a) = \delta*(\{q0,q1\},a) = \delta*(\{q0,q1\},a) = \{q0,q1\}$$

le mot a été lu entièrement et on termine avec un ensemble qui ne contient pas un état final donc le mot aa n'est pas reconnu par l'automate.

***3-* aba**

$$\delta*(q0,aba) = \delta*(\delta(q0,a),ba) = \delta*(\{q0,q1\},ba) = \delta*(\delta(\{q0,q1\}b),a)$$
$$= \delta*(\{q2,q1\},a) = automate\ bloqué$$

le mot n'a pas été lu entièrement l'automate est bloqué

l□mot aba n'□st pas r□connu par l'automat□.

***Théorème :*** à tout automate non déterministe correspond un automate déterministe équivalent.

***3- <u>Algorithme de déterminisation :</u>***

Données : A=(V,Q,I,F, $\delta$) non déterministe.

Résultat : A'=(V,Q',q0,F', $\delta'$) déterministe équivalent.

Méthode : construire une suite croissante de fonctions de transition $\delta'_k$ sur des ensembles d'états $Q'_k$ inclus dans P(Q) en partant de :

$$\delta'_0 = \emptyset \; et \; Q'_0 = \{I\}$$ **puis en posant :**

$$\delta'_{k+1} = \delta'_k \cup \{ (X , x, \delta (X, x)) ; X \in Q \; et \; x \in V \}$$

$$Q'_{k+1} = Q'_k \cup \cup_{x \in V, X \in Q'_k} \{\delta(X, x)\}$$

$$\text{Où } \delta (X, x) = \cup_{q \in X} \delta(q, x)$$

- Arrêter construction lorsque $Q'_{k+1} = Q'_k$
- L'automate déterministe est :

$$A'=(V, Q'_k, I , F', \delta'_{k+1}) \text{ tel que :}$$

$$F' = \{ X \in Q'_k / X \cap F \neq \emptyset \}$$

- **A la fin on renomme les états.**

***Exemple 3 :*** construire l'AFD correspondant à l'AFN de l'exemple 1.

$$\delta'_0 = \emptyset \; et \; Q'_0 = \{\{q1, q2\}\}$$

$$\delta'_1 = \emptyset \cup \{ (\{q1,q2\} , a, \delta (\{q1, q2\}, a)), (\{q1, q2\} , b, \delta (\{q1, q2\}, b))\}$$

$$\delta'_1 = \{ \; (\{q1,q2\} , a, \{q1, q2, q3\}) \; , \; (\{q1, q2\} , b, \emptyset) \}$$

$$Q'_1 = Q'_0 \cup \{ \emptyset, \{q1, q2, q3\}\}$$

$$\delta'_2 = \delta'_1 \cup \{ \; (\{q1,q2,q3\} , a , \{q1,q2,q3\}),$$
$$(\{q1, q2, q3\}, b, \{q3\}) , (\emptyset, a, \emptyset) , (\emptyset, b, \emptyset)\}.$$

$$Q'_2 = Q'_1 \cup \{ \{q3\} \}$$

$\delta'_3 = \delta'_1 \cup \{ (\{q3\}, a, \emptyset), (\{q3\}, b, \{q3\}) \}$.

$Q'_3 = Q'_2 \cup \emptyset$

- L'automate déterministe est :

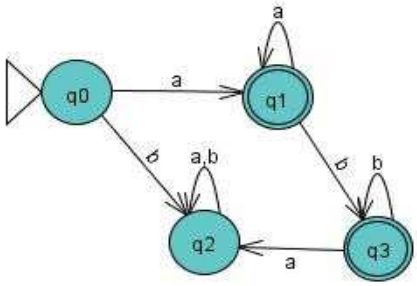
**A'=(V, $Q'_2$, {q1,q2} , F', $\delta'_3$ ) tel que :**

**F' = { {q1,q2,q3}, {q3} }**

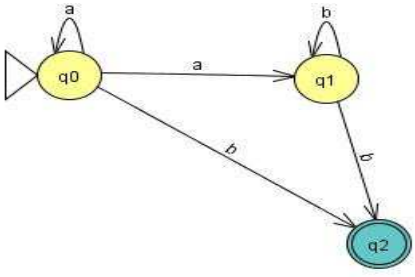
- **On renomme les états :**

**{q1,q2}** devient q0 **{q1,q2,q3}** devient q1

**{q3}** devient q3 ∅ devient q2 cet état est appelé **état puits**




***Exemple 4 :*** construire l'AFD correspondant à l'AFN suivant :




$\delta'_0 = \emptyset$ *et* $Q'_0 = \{\{q0\}\}$

$\delta'_1 = \{ (\{q0\}, a, \{q0, q1\}) , (\{q0\}, b, \{q2\}) \}$

$Q'_1 = Q'_0 \cup \{ \{q0, q1\}, \{q2\}\}$

$\delta'_2 = \delta'_1 \cup \{ (\{q0,q1\}, a, \{q0,q1\}),$
$(\{q0,q1\}, b, \{q1,q2\}), (\{q2\}, a, \emptyset), (\{q2\}, b, \emptyset)\}.$

$Q'_2 = Q'_1 \cup \{ \{q1,q2\}, \emptyset \}$

$\delta'_3 = \delta'_1 \cup \{ (\{q1,q2\}, a, \emptyset), (\{q1,q2\}, b, \{q1,q2\}) \}.$

$Q'_3 = Q'_2 \cup \emptyset$

- L'automate déterministe est :

**A'=(V, $Q'_2$, {q0}, F', $\delta'_3$ ) tel que : F' = { {q1,q2}, {q2} }**

- **On renomme les états :**

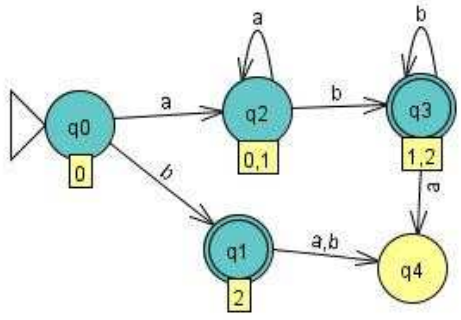
{q0} devient q0 {q0,q1 } devient q2

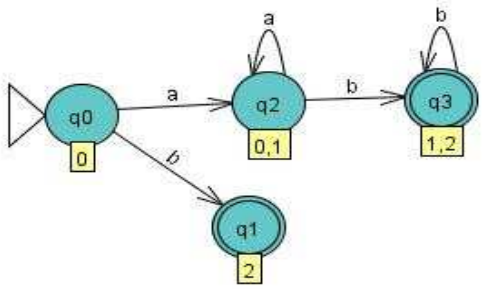
{q2} devient q1 $\{q1,q2\}$ devient q3

$\emptyset$ devient q4 etat puits qu'on peut supprimer.

**On peut aussi représenter l'exécution de l'algorithme par un tableau :**

| | | a | b |
|---|---|---|---|
| Etape1 | {q0} | $\{q0,q1\}$ | $\{q2\}$ |
| Etape 2 | $\{q0,q1\}$ | {q0,q1} | $\{q1,q2\}$ |
| Etape 3 | $\{q2\}$ | $\emptyset$ | $\emptyset$ |
| Etape 4 | $\{q1,q2\}$ | $\emptyset$ | $\{q1,q2\}$ |
| Etape 5 arrêt il n'y a pas de nouveaux états | $\emptyset$ | $\emptyset$ | $\emptyset$ |






## *V- Minimisation d'un AEF déterministe*

D'une manière générale, moins un automate contient d'états, moins il prendra du temps à reconnaître un mot. De plus, il prendra moins d'espace en mémoire s'il est question de le sauvegarder. Il est donc logique de vouloir minimiser ce temps en essayant de réduire le nombre d'états. La minimisation s'effectue en éliminant les états dits inaccessibles et en fusionnant les états reconnaissant le même langage.

### *1- Les états inaccessibles*

**Définition** : Un état est dit inaccessible s'il n'existe aucun chemin permettant de l'atteindre à partir de l'état initial. Les états inaccessibles sont improductifs, c'est-à-dire qu'ils ne participeront jamais à la reconnaissance d'un mot. Ainsi, la première étape de minimisation d'un AEF consiste à éliminer ces états.

**Algorithme de suppression des états inaccessibles :**

Soit : A=(V,Q, $q_0$,F, $\delta$) , construire une suite d'états $Q_0$, $Q_1$ ,… en partant de $Q_0=\{q_0\}$ et en posant : $Q_{k+1}=Q_k \cup \{\delta(q,a)/\ q \in Q_k$ et a∈V } arrêter lorsque $Q_{k+1}=Q_k$ le résultat est A'=(V,$Q_k$,$q_0$,F', $\delta'$).

### *2- Minimiser un AEF*

La méthode de réduction d'un AEF est la suivante :

1. l'automate doit être déterministe.
2. *Nettoyer* l'automate en éliminant les états inaccessibles ;
3. Regrouper les états congruents (appartenant à la même classe d'équivalence).

**Algorithme : Regroupement des états équivalents**

Dans cet algorithme, chaque classe d'équivalence représentera un état dans l'automate minimal. **Cet algorithme ne peut s'appliquer que si l'automate est déterministe.**

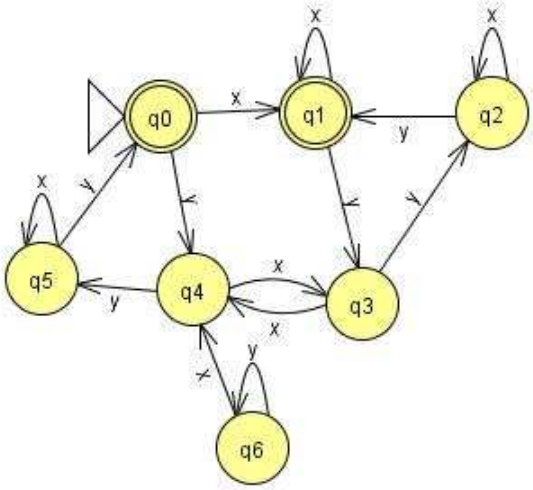
1- Faire deux classes : A contenant les états finaux et B contenant les états non finaux ;
2- S'il existe un symbole a et deux états qi et qj d'une même classe tel que $\delta$(qi,a) et $\delta$(qj,a) n'appartiennent pas à la même classe, alors créer une nouvelle classe et séparer qi et qj. On laisse dans la même classe tous les états qui donnent un état d'arrivée dans la même classe ;

3- Recommencer l'étape 2 jusqu'à ce qu'il n'y ait plus de classes à séparer ;

A la fin de l'algorithme les paramètres de l'automate minimal sont les suivants :

- Chaque classe d'équivalence est un état de l'automate minimal ;
- La classe qui contient l'ancien état initial devient l'état initial de l'automate minimal ;
- Toute classe contenant un état final devient un état final ;
- La fonction de transition est définie comme suit : soient A une classe d'équivalence obtenue, a un symbole de l'alphabet et qi un état ∈ A tel que $\delta$(qi, a) est définie. La transition $\delta$ (A, a) est égale à la classe B qui contient l'état qj tel que $\delta$ (qi, a) = qj.

***Exemple :***

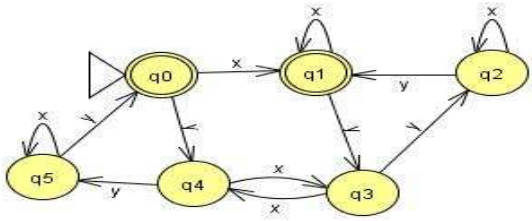



***1- L'automate est déterministe :***

$$\forall x \in V \text{ et } \forall q \in Q \; |\delta(q, x)| = 1$$

***2- Les états inaccessibles :***

- $Q_0$={$q_0$}
- $Q_1$=$Q_0 \cup \{\delta(q_0, x), \delta(q_0, y)\}$ = {q0 ,q1 ,q4 }
- $Q_2 = Q_1 \cup \{(q_1, x), \delta(q_1, y), \delta(q_4, x), \delta(q_4, y)\} = \{q0, q1, q4, q3,$ q5 }
- $Q_3 = Q_2 \cup \{(q_3, x), \delta(q_3, y), \delta(q_5, x), \delta(q_5, y)\} = \{q0, q1, q4, q3,$q5, q2 }
- $Q_4 = Q_3 \cup \{(q_2, x), \delta(q2, y)\} = \{q0, q1, q4, q3,$q5, q2 }= $Q_3$ ***Arrêt***
- **L'état q6 est inaccessible il sera supprimé *de l'automate***




**3- Calcul de l'automate minimal $A_{min}$ :**

a. Faire deux classes : A contenant les états finaux et B contenant les états non finaux **A= [q0 , q1 ] B= [q2,q3,q4,q5 ]**

**b. On teste la classe A**

$\delta(q_0, x) = q1$ $\delta(q1, x) = q1$ *résultat même état*

$\delta(q_0, y) = q4$ $\delta(q_1, y) = q3$

***résultat des états qui sont de la même classe***

**Donc la classe A n'est pas modifiée.**

- **On teste la classe B**

$\delta(q_2, x) = q2$ $\delta(q3, x) = q4$

$\delta(q_4, x) = q3$ $\delta(q_5, x) = q5$ *résultats même classe*

$\delta(q_2, y) = q1$ $\delta(q3, x) = q2$

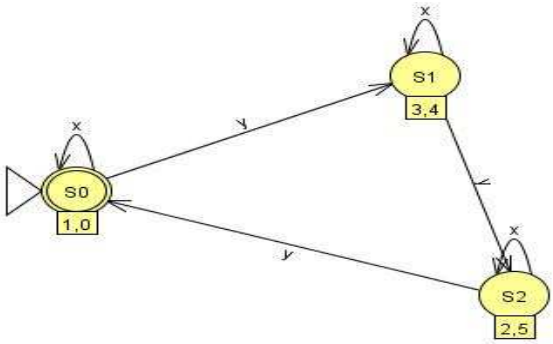
$\delta(q_4, x) = q5$ $\delta(q_5, x) = q0$

*résultats classes différentes*

*la classe B sera divisée en 2 classes* **[q2, q5 ] [q3,q4]**

On répète avec les 3 classes obtenues et on trouvera aucun changement alors l'algorithme s'arrête.

On obtient un automate minimal avec 3 états (les classes d'équivalence) , l'état initial est aussi l'état final.



***Propriété de clôture (fermeture) :***

Soit L1 un langage reconnu par A1=(V,Q1, $q_{01}$,F1, $\delta 1$)

et soit L2 un langage reconnu par A2=(V,Q2, $q_{02}$,F2, $\delta 2$) :

1. L1+L2 est reconnu par A=(V, Q1× Q2, ($q_{01}$, $q_{02}$), F1×Q2 ∪ Q1×F2, $\delta$) tel que :

   $\delta = \delta 1 \times \delta 2$

   $\delta : (Q1 \times Q2) \times V \rightarrow (Q1 \times Q2)$

   $\delta((q1, q2), x) = (\delta 1(q1, x), \delta 2(q2, x))$

2. L1∩L2 est reconnu par A=(V, Q1× Q2, ($q_{01}$, $q_{02}$), F1× F2, $\delta 1 \times \delta 2$)

3. $\overline{L1}$ $(complément)$ est reconnu par A=(V, Q1, $q_{01}$, Q1-F1, $\delta 1$)