Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Informatique

# Cours Théorie des langages (THL)

## Chapitre 4 : Automates d'états finis (AEF)

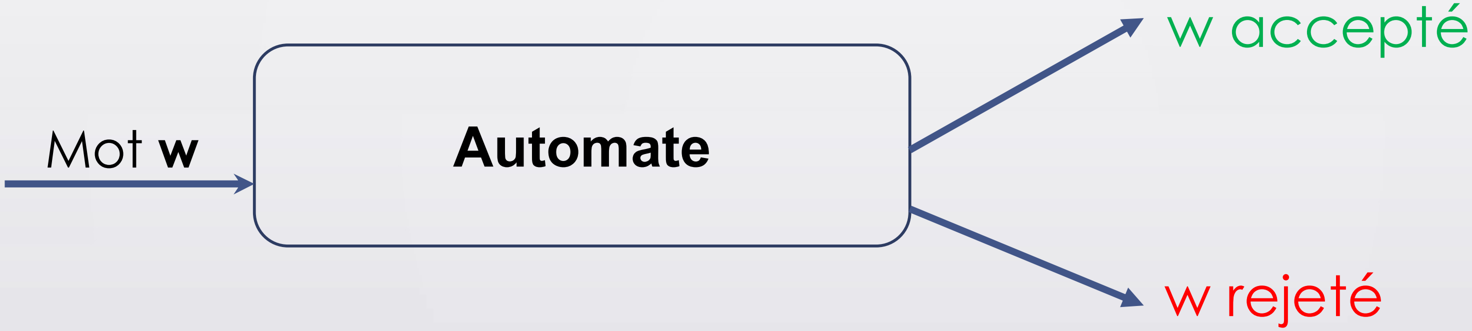
**2 ième année ING 2023/2024**

Dr H.BELHADI hib.belhadi@gmail.com

# Automates d'états finis (AEF)

**Définition** : Un automate **(ou système de reconnaissance)** est une **machine abstraite** qui permet de **reconnaître** les mots d'un langage. Il prend en entrée un mot **w**, et fournit comme résultat :

- **accepté :** si le mot est reconnu par l'automate ($\in$Langage)
- **rejeté :** si le mot n'est pas reconnu par l'automate ($\notin$Langage)




A chaque type de langage, on associe un type d'automate :

Aux langages de Type 3 $\longrightarrow$ Les Automates d'Etats Finis
Aux langages de Type 2 $\longrightarrow$ Les Automates à Piles
Aux langages de Type 1 $\longrightarrow$ Les Automates à Bornes Linéaires
Aux langages de Type 0 $\longrightarrow$ Les Machines de Turing

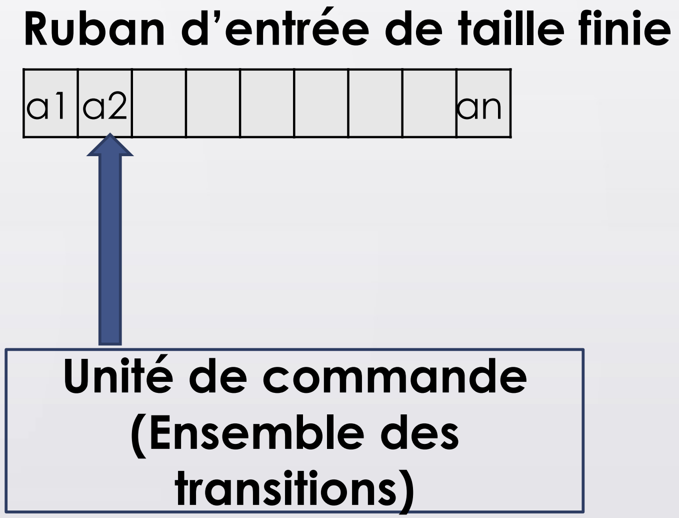
# Automates d'états finis (AEF)

Un AEF lit les symboles d'un mot à reconnaitre un par un et va d'état en état selon les transitions. Le mot lu est soit accepté par l'automate soit rejeté.

**Fonctionnement :**

Le ruban d'entrée contient le mot à reconnaître.

- L'automate **démarre à l'état initial.**
- La tête de lecture **est initialement sur le 1er symbole du mot d'entrée.**
- Les symboles du mot sont lus **de la gauche vers la droite.**
- L'automate avance d'une **case vers la droite** à chaque exécution d'une transition.
- L'automate évolue d'un état à un autre en fonction du **symbole lu** et de **l'état courant**.
- Le mot est **reconnu** si et seulement si :
  - l'automate **a terminé** la lecture du mot
  - **et** se trouve dans **un état final.**



# Automates d'états finis (AEF)

**Définition** (Automate d'états finis)

Un automate d'états finis déterministe est un cinquplet

$$\mathbf{A = (X, Q, q_0, \delta, F)} \text{ où :}$$

- X est un **alphabet d'entrée**, fini et non vide
- Q est un **ensemble d'états**, fini et non vide
- $q_0 \in Q$ est un **état initial**
- $F \subseteq Q$ est un **ensemble d'états finaux**
- $\delta$ est une **fonction de transitions** définie de Q x X dans Q et qui associe à un état donné **p** et un symbole donné **a** un état d'arrivée **q** i.e. $\delta(p, a)=q$

**Remarque :** $\delta(p, a)=q$ signifie que l'automate réalise un déplacement (une transition) de l'état **p** vers l'état **q** en lisant la lettre **a**.

# Automates d'états finis (AEF)

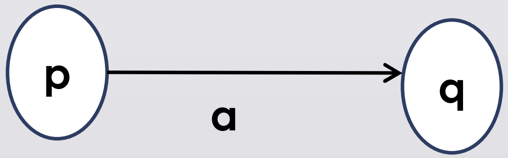
**Représentation graphique :**

Les automates d'états finis sont souvent représentés par des graphes orientés dont :

les sommets correspondent aux états.

- les arcs correspondent aux transitions
- l'arc ayant comme extrémité initiale $p \in Q$ et pour extrémité terminale $q \in Q$ et étiqueté par $a \in X$ représente la transition

$$\delta(p, a) = q$$



# Automates d’états finis (AEF)

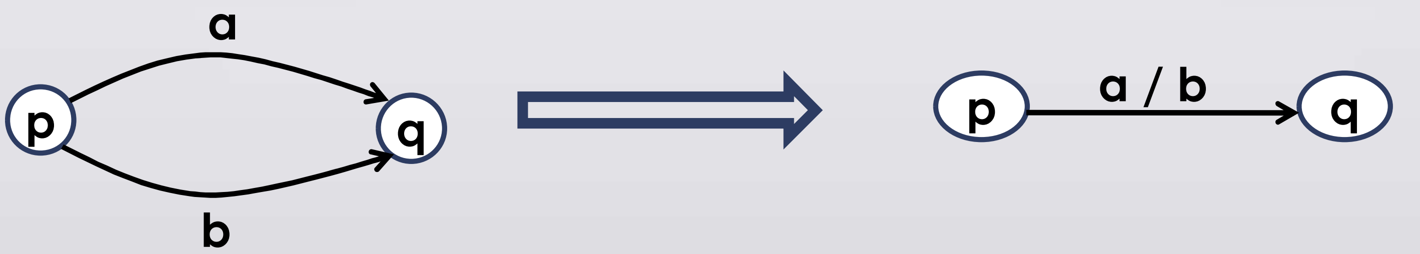
## Représentation graphique :

- un état final est représenté par deux cercles concentriques. 
- un état initial est représenté par une flèche incidente sur l’état initial. 

## Remarque :

Un AEF possède un seul état initial mais peut avoir plusieurs états finaux

**Notation :** Deux transitions ou plus entre deux états seront représentées par un seul arc entre ces deux états comme suit :

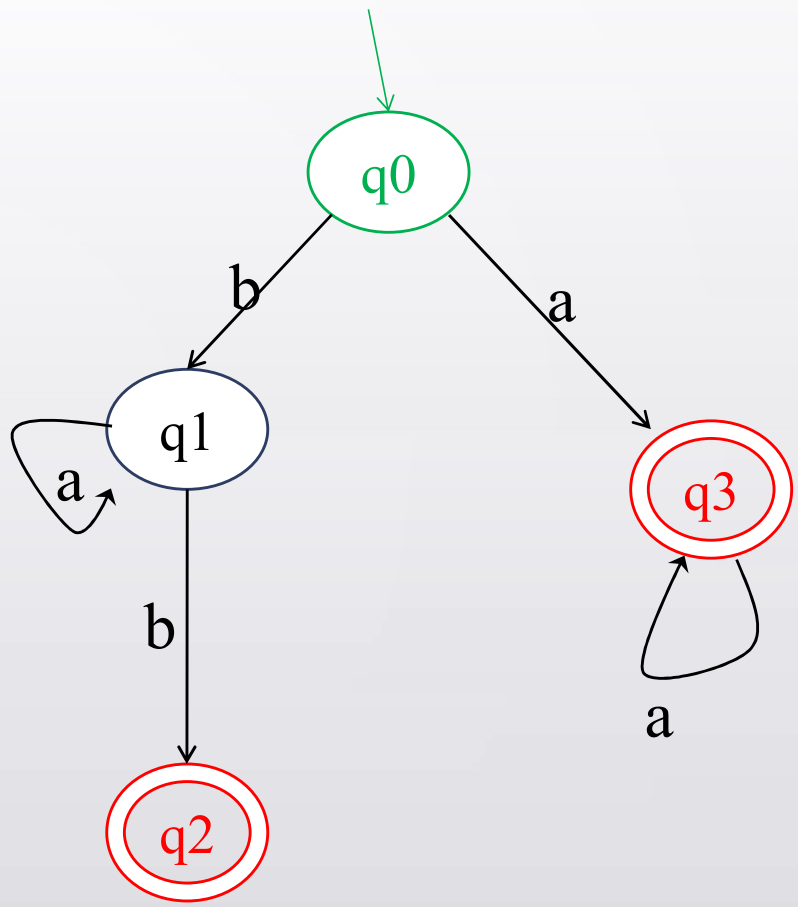


# Automates d'états finis (AEF)

**Exemple :** Soit l'AEF suivant

$A = (X, Q, q0, \delta, F)$ tels que :

- $X = \{a, b\}$
- $Q = \{q0, q1, q2, q3\}$
- q0 est l'état initial
- $\delta(q0, a) = q3$
  $\delta(q0, b) = q1$
  $\delta(q1, a) = q1$
  $\delta(q1, b) = q2$
  $\delta(q3, a) = q3$
- $F = \{q2, q3\}$



## Automates d'états finis (AEF)

**Représentation matricielle :**

Les automates d'états finis peuvent aussi être représentés par des matrices dont :

- les indices des lignes correspondent aux états
- les indices des colonnes correspondent aux éléments de X.
- un élément de la matrice de ligne **q** et colonne **a** correspond à la transition **$\delta$(q, a).**

Symbole

| état\lettre | | a | |
|---|---|---|---|
| | | | |
| **q** | | **$\delta$(q, a)** | |
| | | | |

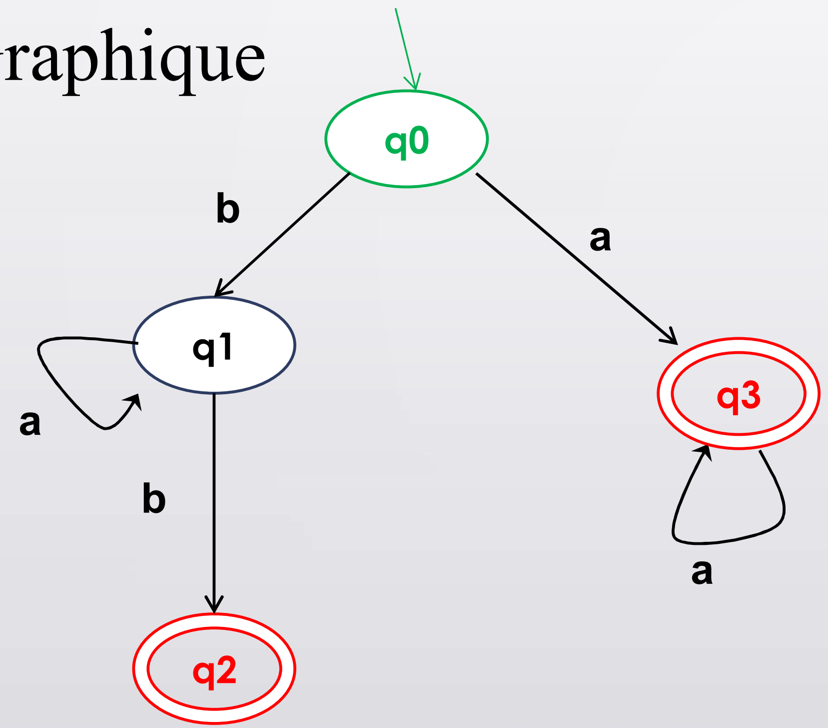
état

Transition

# Automates d'états finis (AEF)

## Exemple :

Représentation Graphique




Représentation Matricielle

| etat\lettre | a | b |
|---|---|---|
| q0 | q3 | q1 |
| q1 | q1 | q2 |
| q2 | _ | _ |
| q3 | q3 | _ |

$\delta(q0, a)$

**Transition non définie**

# Automates d'états finis (AEF)

**Question :**

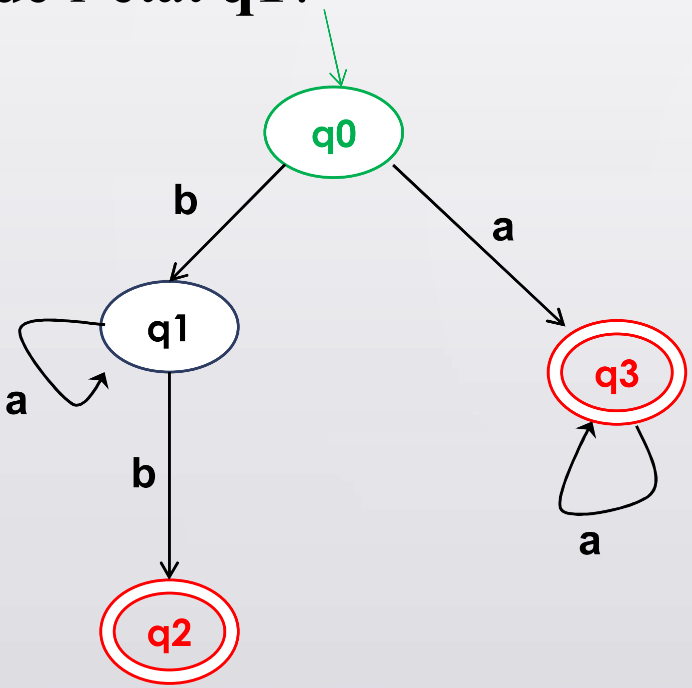
- Quelle est la **succession de transitions** permettant de lire un mot **w** à partir d'un état donné **q**, si elle existe ?
- Quel est l'état d'arrivée après cette succession de transitions.

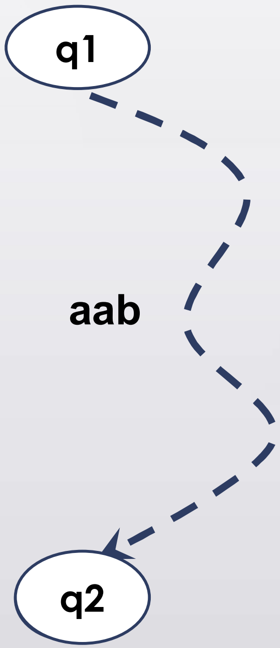

Définir la fonction de **succession de transitions**.

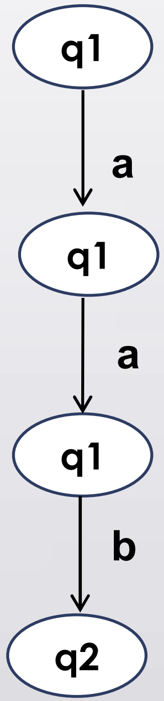
# Automates d'états finis (AEF)

**Fonction de succession de transitions :**

Illustration par un exemple **:** Quel est la succession de transitions pour lire **aab** à partir de l'état **q1**?



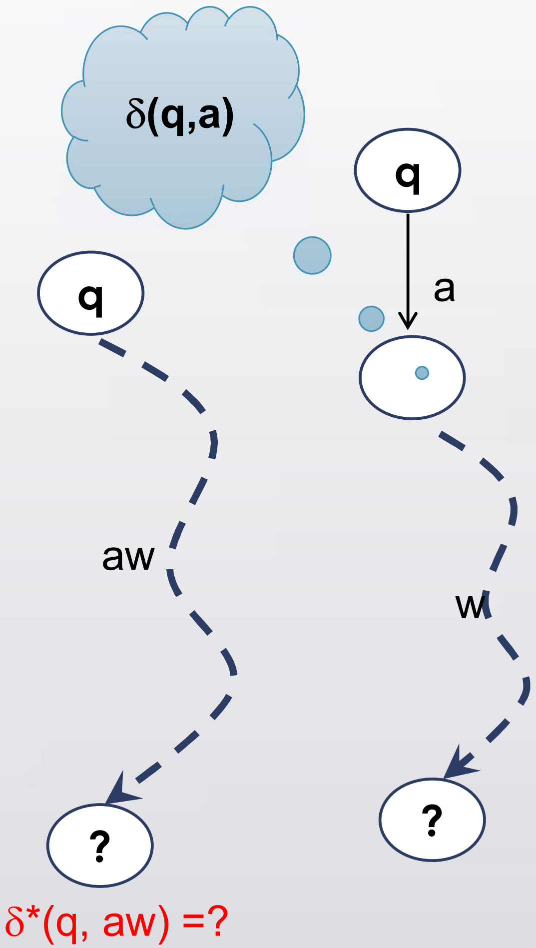



# Automates d'états finis (AEF)

Soit A=(X, Q, q0, δ, F) un automate d'états fini déterministe.

On étend naturellement, la fonction de transition δ à la fonction de succession de transitions δ* définie de **Q x X*** dans **Q** comme suit :

- $\delta^*(q, \varepsilon) = q$
- $\delta^*(q, a) = \delta(q, a) \quad a \in X$
- $\delta^*(q, aw) = \delta^*(\delta(q, a), w) \quad a \in X$ et $w \in X^*$



# Automates d'états finis (AEF)

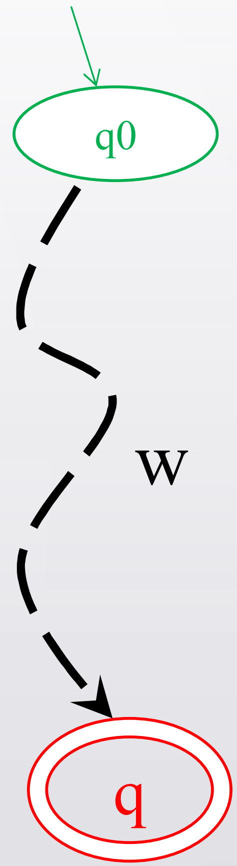
**Condition de reconnaissance :**

Un mot w est **reconnu (accepté)** par l'automate A ssi :

- l'automate A lit le mot w à partir de $q_0$
- et atteint un état final.

Autrement dit, **il existe un état final $q_F$**$\in$F tel que $\delta$ **$^*(q_0, w) = q_F$**.

On dit que **A accepte le mot w** (ou que w est accepté par A).



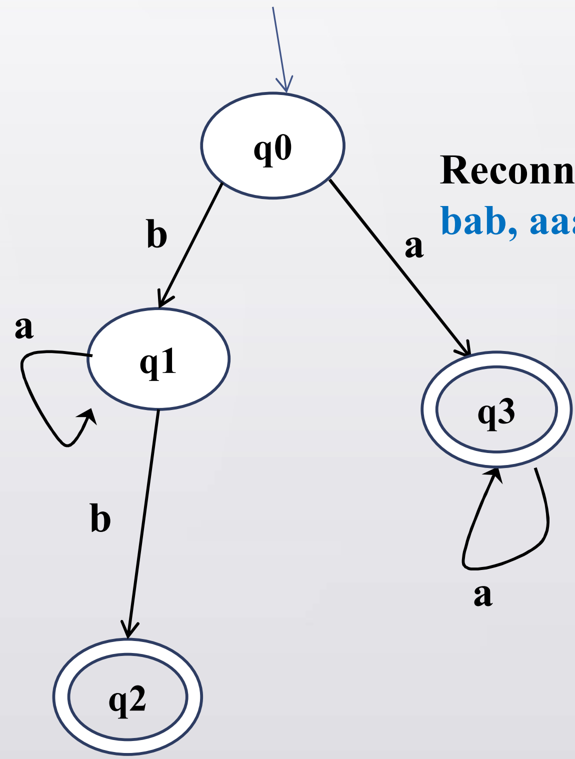
# Automates d'états finis (AEF)

$\delta^*(q0, bab) = \delta^*(\boldsymbol{\delta(q0, b)}, ab)$
$= \delta^*(\mathbf{q1}, ab)$
$= \delta^*(\boldsymbol{\delta(q1, a)}, b)$
$= \delta^*(q1,b)$
$= \delta(q1,b)$
$=$ **q2** et **q2 est un état final**

$\delta^*(q0, \mathbf{a}aa) = \delta^*(\boldsymbol{\delta(q0, a)}, aa)$
$= \delta^*(\mathbf{q3}, aa)$
$= \delta^*(\boldsymbol{\delta(q3, a)}, a)$
$= \delta^*(\mathbf{q3},a)$
$= \delta(q3,a)$
$= q3$ et **q3 est un état final**

**Les mots bab et aaa sont acceptés par cet automate**

$\delta^*(q, aw)=\delta^*(\delta\ \mathbf{(q, a)}, w)$, $a \in X$ et $w \in X^*$




**Reconnaître les mots :**
**bab, aaa**

# Automates d'états finis (AEF)

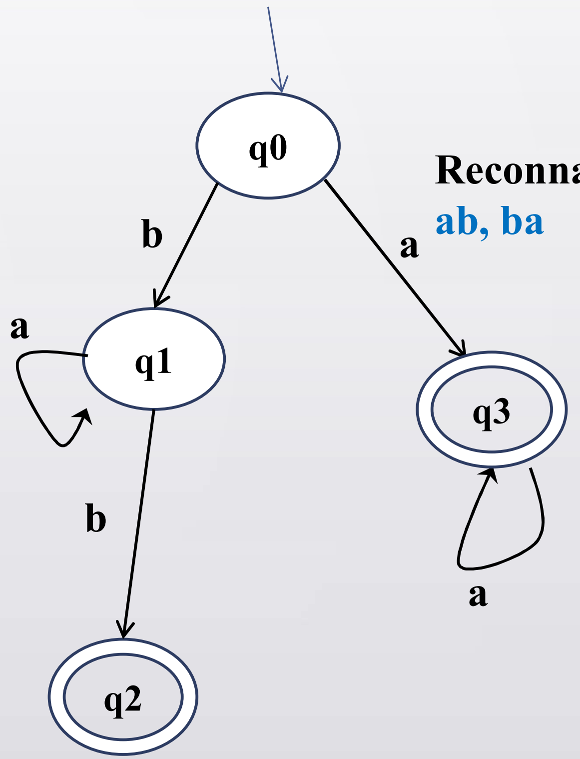
$\delta^*(q0, ab) = \delta^*(\delta(\mathbf{q0, a}), b)$

$= \delta^*(q3, b)$

$= \delta(q3, b)$

**mais à l'état q3, l'automate ne peut pas lire b**

$\delta^*(q, aw) = \delta^*(\delta\ (\mathbf{q, a}), w)$, $a \in X$ et $w \in X^*$




**Reconnaître les mots :**
**ab, ba**

$\delta^*(q0, ba) = \delta^*(\delta(\mathbf{q0, b}), a)$

$= \delta^*(\mathbf{q1}, a)$

$= \delta(\mathbf{q1}, a)$

$= q1$

**mais q1 n'est pas final**

**Les mots ab et ba ne sont pas acceptés**

# Automates d'états finis (AEF)

**Définition (Langage Reconnu par AEF)**

Soit $A=(X, Q, q_0, \delta, F)$ un automate déterministe.

Le langage reconnu par l'automate A est l'ensemble

$$\mathbf{L(A) = \{w \in X^* / \delta^*(q_0, w) \in F\}.}$$

Un langage L sur X est **régulier** (ou **reconnaissable**) s'il existe au moins un automate d'états finis A ayant X comme alphabet d'entrée tel que **L=L(A).**

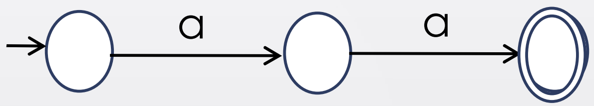
**Notation :** On note **Rec(X*)** la famille des langages reconnaissables sur l'alphabet X.

## Automates d'états finis (AEF)

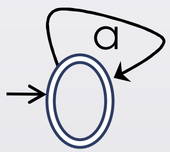
**Exemples :** Donner un AEF pour chacun des langages :
L1= $\{a\}^2$, L2= $\{a^n / n \geq 0\}$ et L3= $\{a^n / n \geq 1\}$,

$\{a\}\{a\}=\{aa\}$

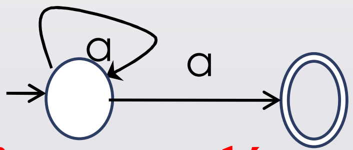
**AEF de L1= $\{a\}^2$ :**

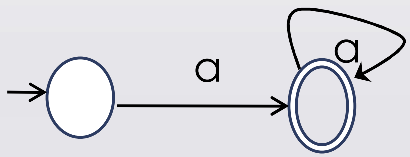



**AEF de L2= $\{a^n / n \geq 0\}$ :**




**AEF de L3= $\{a^n / n \geq 1\}$ :**




Cet AEF n'est pas déterministe

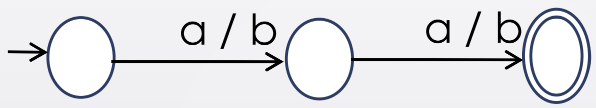
**Rappel : Pour un AEF déterministe**
La **fonction de transitions** δ définie de Q x X dans Q

# Automates d'états finis (AEF)

**Exemples :** Donner un AEF pour chacun des langages

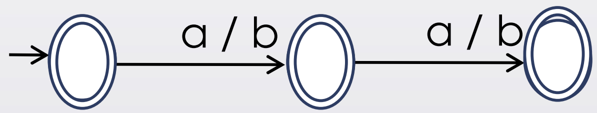
$L1= \{a, b\}^2$, $L2= \{w \ / \ w \in \{a, b\}^*, |w| \leq 2\}$ et $L3= \{a, b\}^*$

**AEF de $L1= \{a,b\}^2$ :**

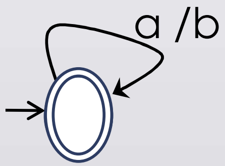



**AEF de L2 : on a $\{a,b\}^2 \subseteq L2$ :**

**$L2= \{a,b\}^2 \cup \{a, b, \varepsilon\}$ :**



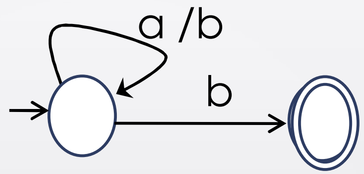

**AEF de $L3= \{a,b\}^*$ :**



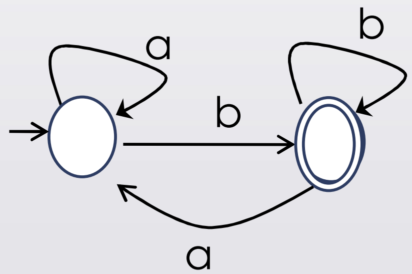
# Automates d'états finis (AEF)

**Exemple** Donner un AEF déterministe reconnaissant le langage $\{ wb / w \in \{a, b\}^* \}$

**Cet AEF n'est pas déterministe**




**Cet AEF est déterministe**



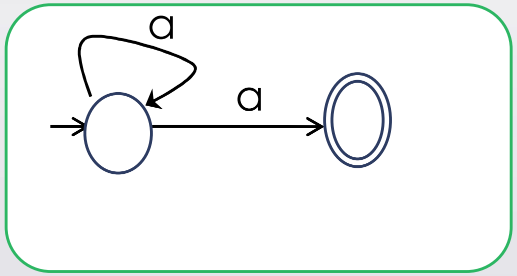
# Automates d'états finis (AEF)

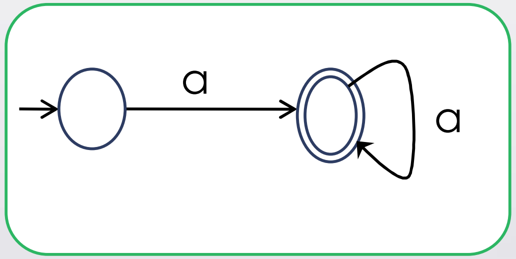
**Définition (Equivalence de deux AEFs)**

Deux automates d'états finis A1 et A2 sont équivalents, noté A1≡A2, si et seulement s'ils acceptent le même langage.

$$\mathbf{A1 \equiv A2 \iff L(A1)=L(A2)}$$

**Exemple :** Ces deux automates sont équivalents.

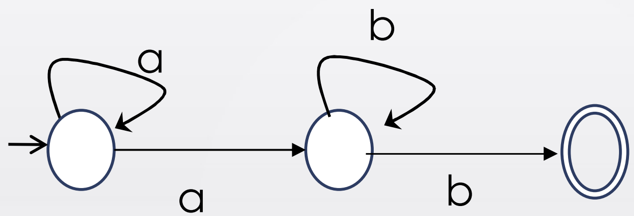



**Remarque** : Un langage peut être reconnu par plusieurs automates. Par contre un automate ne peut reconnaitre qu'un seul langage.

# Automates d’états finis (AEF)

$\{a^n b^m / \ n, m >= 1\}$

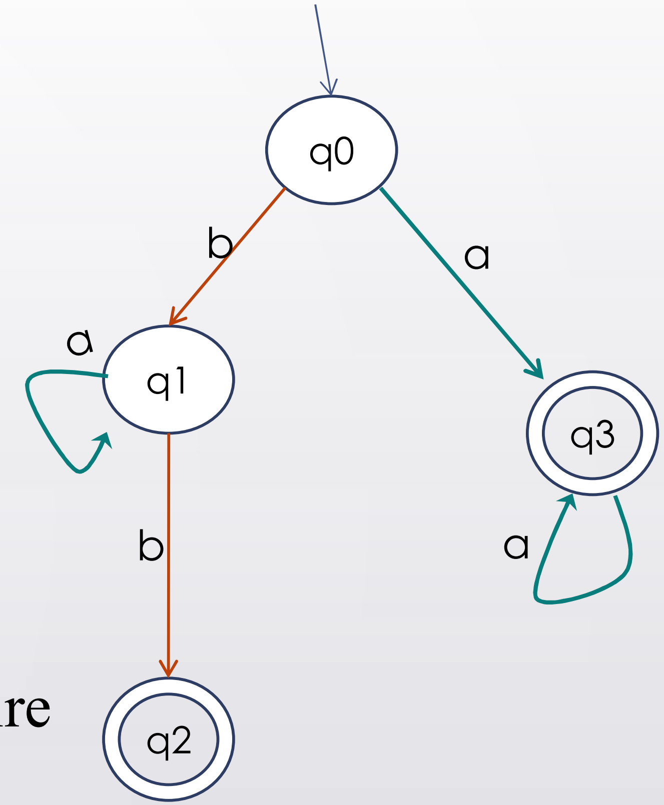


# Automates Déterministes

Un automate d'états fini est **déterministe** si et seulement si :

- à un état
- à un symbole d'entrée

la fonction $\delta$ associe au plus une seule transition.




Autrement dit, la fonction $\delta$ est définie de Q x X dans Q.

**Remarque :**
Dans les automates déterministes, il n y a pas de choix à faire pour l'état suivant après la lecture d'un certain symbole.

# Automates Déterministes

Un automate déterministe est dit complet ssi : à toute paire $(q,a) \in QxX$, la fonction $\delta$ associe exactement un état.

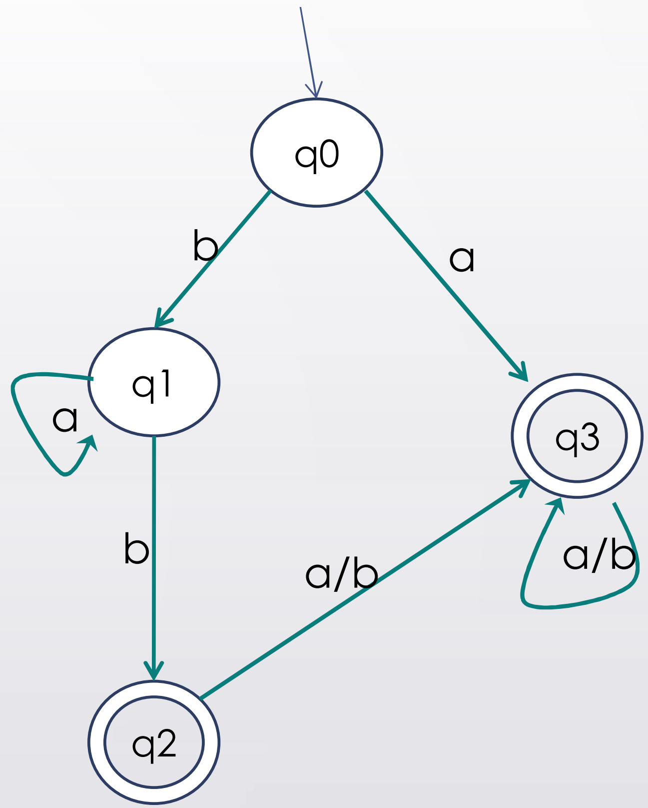
Ainsi, la fonction de transition $\delta$ est une **application fonctionnelle** : Q x X dans Q.

**Remarque :**
Dans un automate complet, il y a possibilité de lire n'importe quel symbole à partir de n'importe quel état.

# Automates Déterministes

**Exemple** : Automate déterministe et complet

| **état\lettre** | **a** | **b** |
|---|---|---|
| q0 | q3 | q1 |
| q1 | q1 | q2 |
| q2 | q3 | q3 |
| q3 | q3 | q3 |

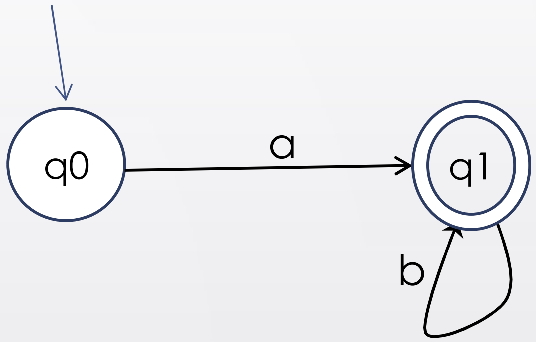



**Automate déterministe et complet**

**Remarque :** La matrice des transitions d'un automate déterministe et complet est pleine.

# Automates Déterministes

**Exemple** : Cet AEF est-il déterministe et complet ?




Cet automate est **déterministe mais non complet.**

En effet, dans l'état **q0**, on ne peut pas lire la lettre **b**.

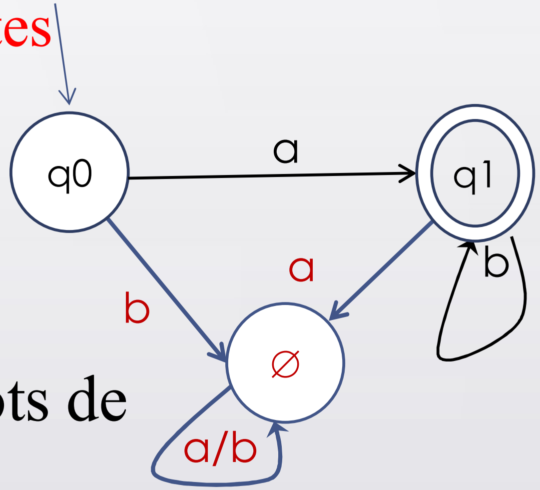
Le langage reconnu par cet automate est : $\{ab^n / n \geq 0\}$.

**Remarques :**

- ✓ Un automate déterministe non complet ne permet pas de lire certains mots de X*.
- ✓ Un automate peut lire des mots mais ne pas les reconnaître.

# Automates Déterministes

Pour rendre complet un «automate déterministe non complet» il suffit de :
rajouter un état, appelé «état puits»
généralement noté ∅, et de rajouter toutes les transitions manquantes vers cet état.



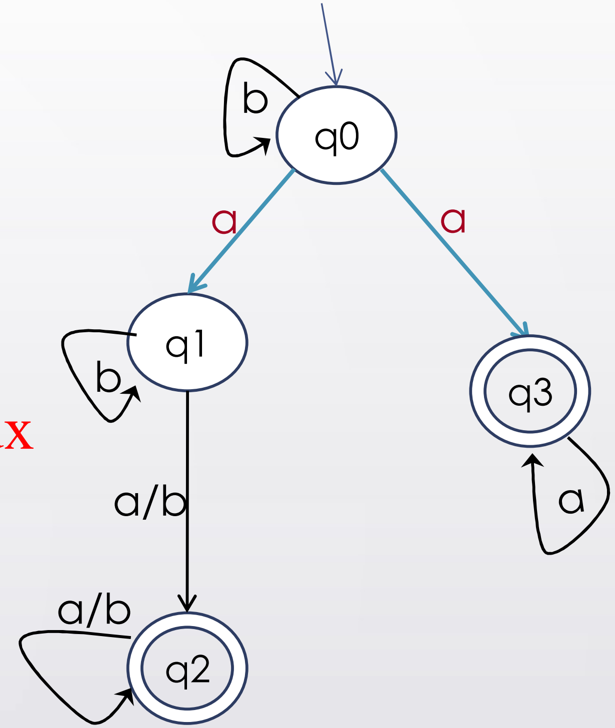

**Remarques :**

✓Un automate déterministe complet permet de lire tous les mots de l'alphabet (pas nécessairement les reconnaitre)

✓L'automate complet obtenu reconnaît le même langage que l'automate initial (l'état puits n'est pas un état final).

# Automates Non Déterministes

Les automates d’états finis non déterministes sont des automates où : l’on permet plusieurs transitions correspondant à la même lettre **à** partir des états de l’automate.



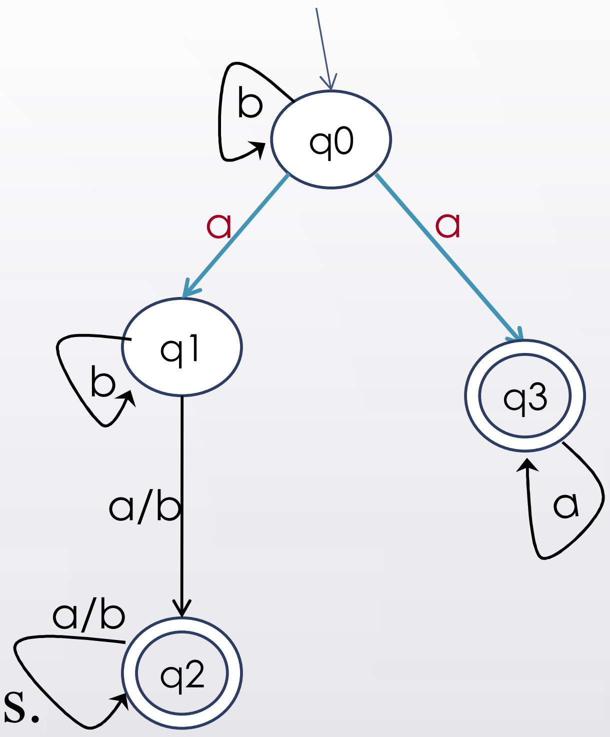

Dans l’exemple, à partir de q0, on a le choix entre deux transitions par a :
l’une vers q1 et l’autre vers q3.
Donc, δ (q0, a)=**{q1, q3}**

Dans ce cas, δ est une fonction de transition définie de :
QxX dans **l’ensemble des parties** de Q. **δ : QxX→ 𝒫 (Q)**

# Automates Non Déterministes

**Exemple:**

| etat\lettre | a | b |
| --- | --- | --- |
| q0 | {q1, q3} | {q0} |
| q1 | {q2} | {q1, q2} |
| q2 | {q2} | {q2} |
| q3 | {q3} | _ |

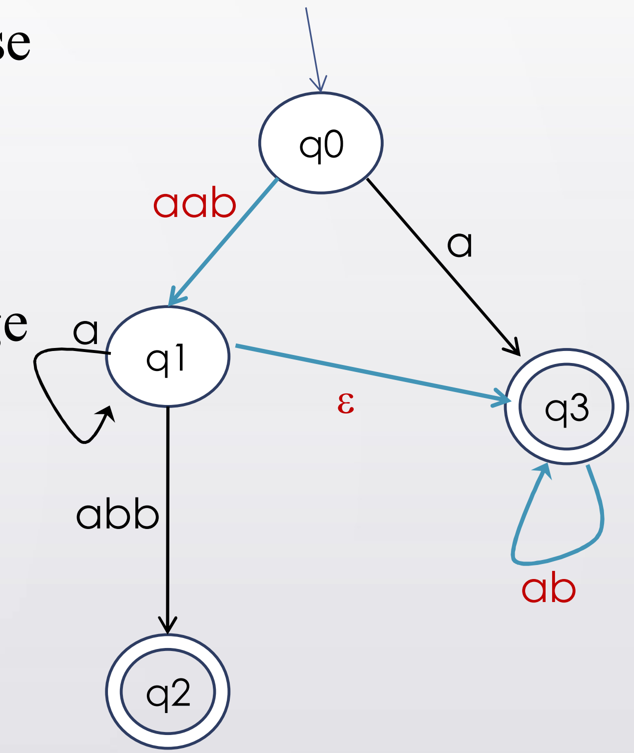



**Remarque :**

- ✓ Pour alléger les notations, on peut omettre les accolades.
- ✓ Dans les automates non déterministes, un choix est permis pour passer à l'état suivant.

# Automates Généralisés

- Dans un automate généralisé, les transitions directes peuvent être causées par des mots (jusqu'a maintenant, les transitions se faisaient en lisant un seul symbole à la fois).
- Les transitions directes causées par le mot vide sont appelées transitions spontanées ou ε-transitions.
- Une ε-transition correspond à la situation où l'automate change d'état sans lire de symbole.
- Dans l'exemple, à l'état q1, on a 3 possibilités :
  - lire **a** et rester dans q1
  - lire **abb** et passer à q2
  - passer directement à **q3** sans lecture.




La fonction δ est alors définie de: **QxX*→P(Q)**

$\delta(q0, \mathbf{aab})=\{q1\}$

$\delta(q1, \varepsilon)=\{q3\}$

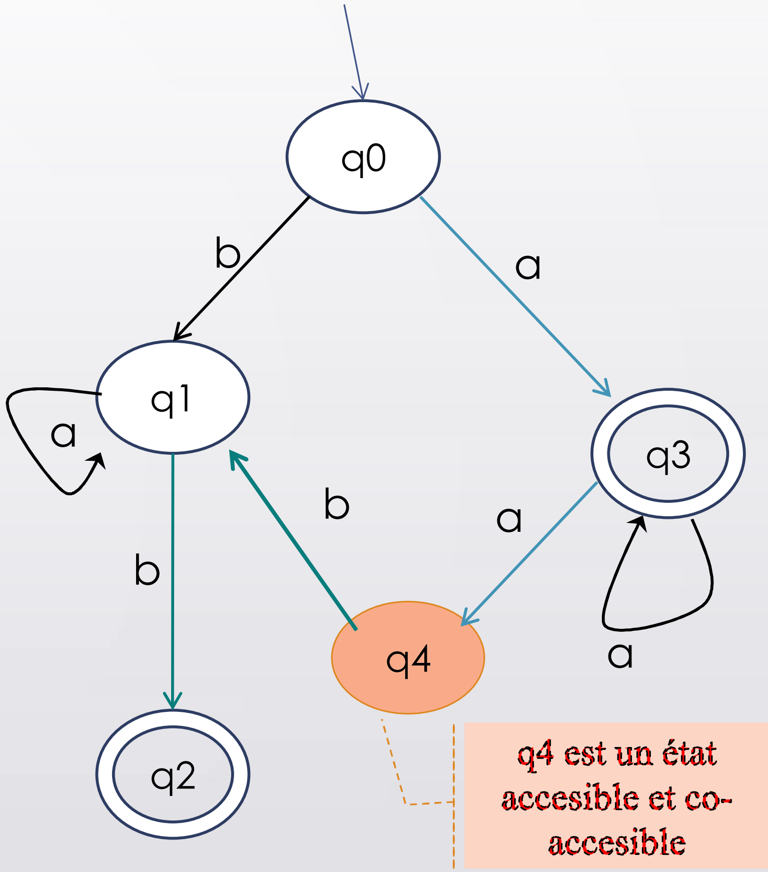
# Automates D’états Finis

**Définitions :**

Un état **q** est **accessible** s'il existe un chemin de l'état initial de l'automate vers **q**.

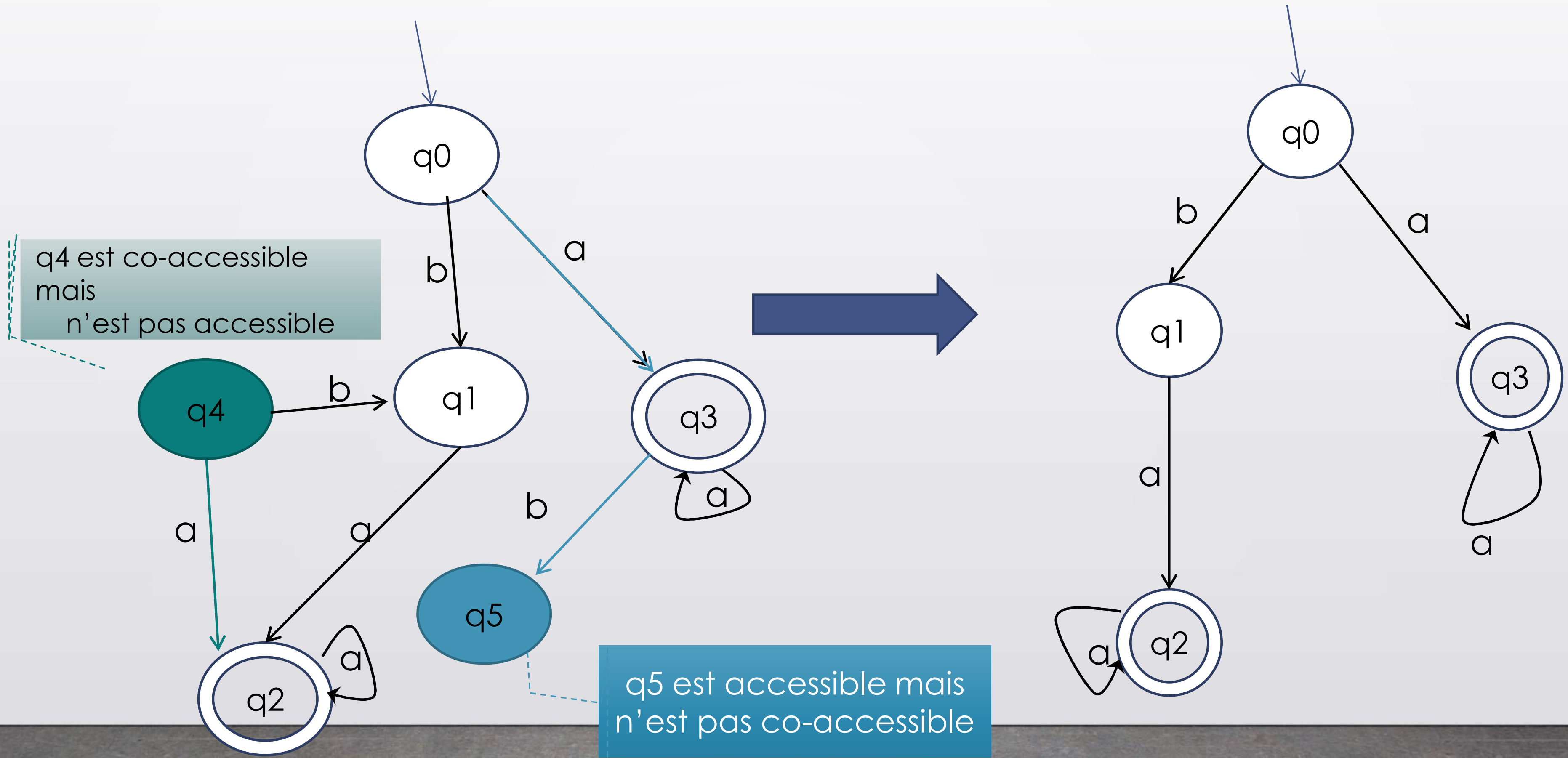
Un état **q** est **co-accessible** s'il existe un chemin de l'état **q** vers un état final.

Un automate est **émondé** si tous ses états sont accessibles et co-accessibles.



# Automates D’états Finis

Pour rendre un automate émondé, il suffit de supprimer tous les états non accessibles et non co-accessibles.



# Automates D'états Finis

**Remarques :**

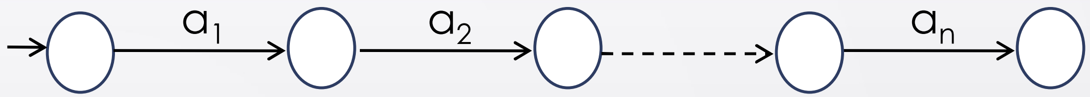
1. Dans les cas d'un automate déterministe et non déterministe, toute **transition directe** est causée par **une seule lettre de l'alphabet**.

2. Par opposition aux automates **généralisés**, les automates déterministes et non déterministes sont dits **simples** (toutes les transitions se font en lisant une seule lettre a la fois)

# Transformation des Automates

En pratique, les automates simples déterministes, sont très intéressants pour la **reconnaissance des mots**.

Exemple : Reconnaitre le mot : $a_1a_2\ldots a_n$




Donc, l'idéal serait de transformer n'importe quel type d'automate, vers un automate déterministe.

Cette transformation se fait en **deux étapes** :

1) Transformer un automate généralisé vers un automate simple (déterministe ou non déterministe)
2) Transformer l'automate simple vers un automate déterministe (s'il ne l'est pas déjà)

**Automate :** Généralisé→Simple Non-Déterministe→Simple Déterministe

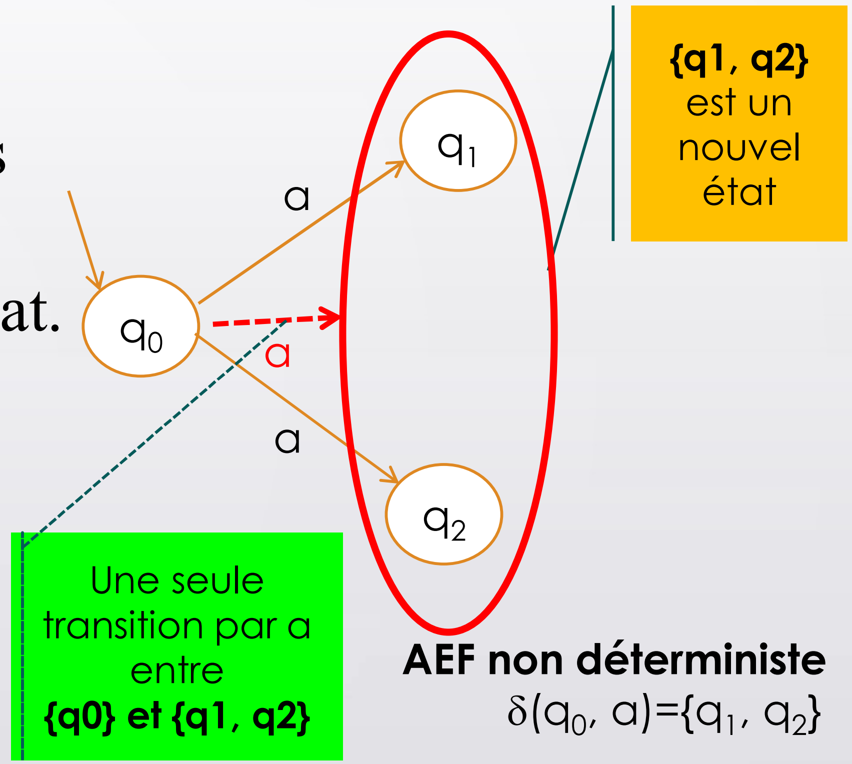
On commence d'abord par présenter la deuxième transformation

# Passage AEF NON Déterministe vers AEF Déterministe

## Principe :

1. Considérer des ensembles d'états plutôt que des états en regroupant toutes les transitions étiquetées par la même lettre issues du même état.

2. Un ensemble d'états (nouvel état obtenu par regroupement) est final ssi il contient un état final (de l'automate initial).




**AEF non déterministe**

$\delta(q_0, a)=\{q_1, q_2\}$

**AEF déterministe**

$\delta_d(\{q_0\}, a)=\{q_1 q_2\}$

# Passage AEF NON Déterministe vers AEF Déterministe

**Proposition :** Pour tout automate fini non déterministe $A = (X, Q, q_0, \delta, F)$ il existe un automate fini déterministe équivalent $A_d=(X_d, Q_d, q_{0d}, \delta_d, F_d)$ avec :

- $X_d=\mathbf{X}$
- $Q_d=\mathbf{2^{(Q)}}$ (ou $P(\mathbf{Q})$)
- $q_{0d}=\{\mathbf{q_0}\}$
- Pour tout état $q_d \in Q_d$ avec $q_d=\{q_1, q_2,\ldots,q_n\}$, on a
  $\delta_d(q_d, a) = \boldsymbol{\delta(q_1, a)} \cup \boldsymbol{\delta(q_2, a)} \cup \ldots\ldots \cup \boldsymbol{\delta(q_n, a)}, \forall a \in X$
- $F_d=\{\mathbf{q_d \in Q_d \,/\, q_d \cap F \neq \emptyset}\}$

Les états de $Q_d$ contenant au moins un état final de A sont **finaux.**

# Passage AEF NON Déterministe vers AEF Déterministe

Pour déterminiser un automate, il est plus pratique d'établir la table des transitions.

La construction de l'automate déterministe se fait sur des ensembles d'états obtenus à **partir de l'état initial** comme suit :

A partir de l'ensemble contenant **seulement l'état initial {q0}**, on regroupe les transitions étiquetées par la même lettre issue de q0 :

| etat\lettre | a1 | …… | an |
|---|---|---|---|
| → q0 | δ(q0, a1) | | δ(q0, an) |

Il suffit de reprendre (recopier) la ligne de l'état initial.

AEF Non déterministe

| etat\lettre | a1 | …… | an |
|---|---|---|---|
| → {q0} | δ(q0, a1) | | δ(q0, an) |

AEF déterministe

# Passage AEF NON Déterministe vers AEF Déterministe

Pour chaque ensemble d'états $q_d=\{q1, \ldots, qn\}$ nouvellement obtenu et donc accessibles à partir de $q_{0d}$, on détermine les transitions issues de $q_d=\{q1, \ldots, qn\}$ i.e. $\delta_d(q_d, a)$, $\forall a \in X$.

On regroupe les transitions étiquetées par la même lettre :

$$\boldsymbol{\delta_d(q_d, a) = \delta(q1, a) \cup \ldots\ldots \cup \delta(qn, a)}$$

Pour chaque lettre, on fait l'union des cases des différentes lignes associées aux états q1, …, qn.

| état\lettre | a | ……. . |
|---|---|---|
| | | |
| **q1** | **δ(q1, a)** | |
| | | |
| **qn** | **δ(qn, a)** | |

AEF non déterministe

| état\lettre | a |
|---|---|
| | |
| **qd** | **δ(q1, a ) ∪ …∪ δ(qn, a )** |
| | |
| | |

AEF déterministe

# Passage AEF NON Déterministe vers AEF Déterministe

On arrêtera la procédure dés que tous les ensembles d'états obtenus seront traités.

Un ensemble d'états (nouvel état obtenu par regroupement) $q_d$ est final s'il contient un état final (de l'automate initial).

$$F_d = \{q_d \in Q_d \,/\, q_d \cap F \neq \emptyset\}$$

# Passage AEF NON Déterministe vers AEF Déterministe

AEF non déterministe

| etat\lettre | a | b |
|---|---|---|
| → q0 | {q1, q3} | {q1} |
| q1 | {q2} | {q1} |
| q2 | {q2} | _ |
| q3 | {q3} | _ |

| Etat/Lettre | a | b |
|---|---|---|
| → {q0} | {q1, q3} | {q1} |
| {q1, q3} | {q2, q3} | {q1} |
| {q1} | {q2} | {q1} |
| {q2, q3} | {q2, q3} | - |
| {q2} | {q2} | - |

AEF déterministe

- Recopier la ligne de l'état initial.

- {q1,q3} et {q1} : nouveaux états accessibles

- Pour l'état {q1,q3}, on regroupe les transitions partant de q1 et celles partant de q3 par la même lettre et on obtient {q2,q3}.

- Pour l'état {q1}, il suffit de prendre les transitions de q1.

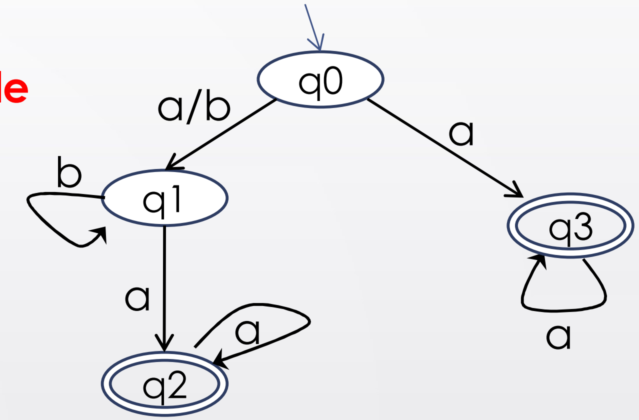
- On traite les nouveaux états accessibles de la même manière

- Les états {q1, q3}, {q2,q3} et {q2} sont finaux.

# Passage AEF NON Déterministe vers AEF Déterministe

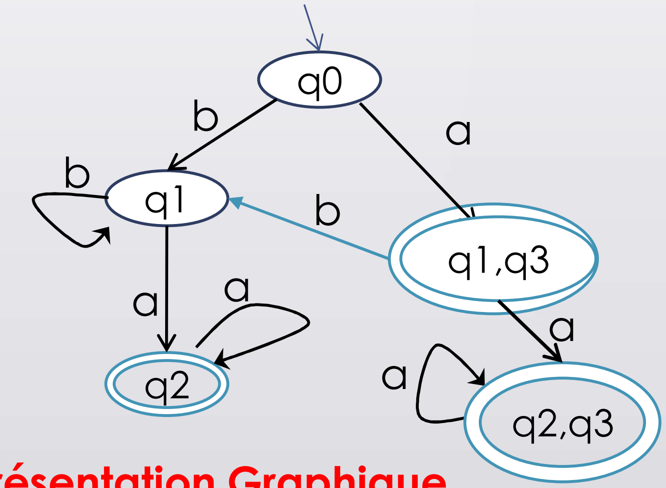
| etat\lettre | a | b |
|---|---|---|
| → q0 | {q1, q3} | {q1} |
| q1 | {q2} | {q1} |
| q2 | {q2} | _ |
| q3 | {q3} | _ |

Représentation matricielle




Déterminisation

| etat\lettre | a | b |
|---|---|---|
| → {q0} | {q1, q3} | {q1} |
| {q1, q3} | {q2, q3} | {q1} |
| {q1} | {q2} | {q1} |
| {q2, q3} | {q2,q3} | _ |
| {q2} | {q2} | _ |



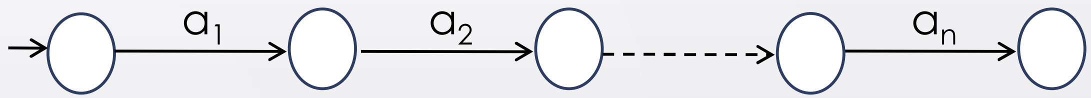

Représentation Graphique

# Transformation des Automates

## Transformation : AEF Généralisé→AEFSimple Déterministe

Pourquoi, on cherche à faire cette transformation?

Exemple : Reconnaitre le mot : $a_1a_2\ldots a_n$




Cette transformation se fait en deux étapes :

1) Transformer un automate généralisé vers un automate simple (déterministe ou non déterministe).
2) Transformer l'automate simple vers un automate déterministe (s'il ne l'est pas déjà).

# Passage AEF Généralisé vers AEF Simple

**Proposition :**

Pour tout automate fini généralisé $Ag=(X_g,Q_g,q0_g,\delta_g,F_g)$ il existe un automate simple équivalent $A_s=(X_s,Q_s,q_{0s},\delta_s,F_s)$.

**Principe de la construction :**

Éliminer les transitions de longueur strictement supérieure à 1. On obtient un automate partiellement généralisé Ap.

Éliminer les transitions spontanées dans l'automate partiellement généralisé obtenu. On obtient un automate simple.

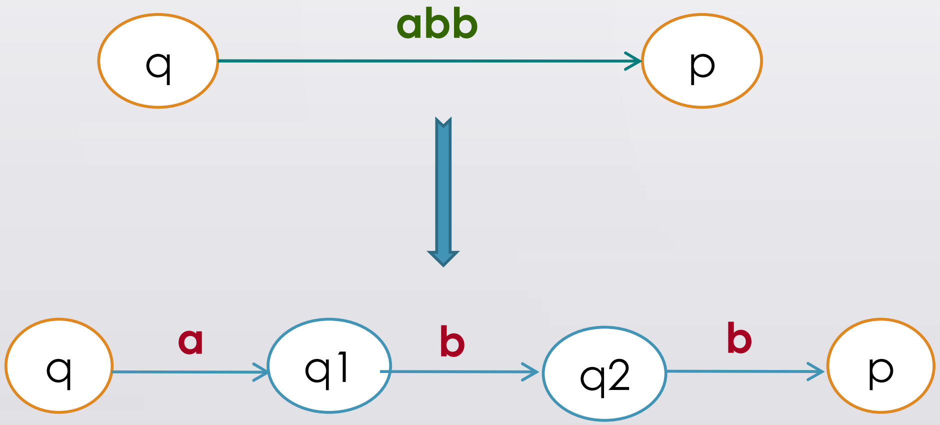
**AEF Généralisé → AEF partiellement Généralisé → AEF simple**

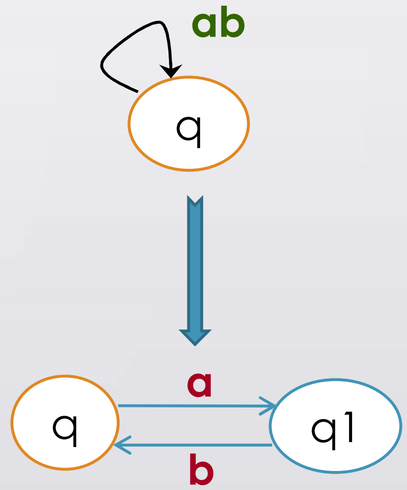
# Passage AEF Généralisé vers AEF Simple

**Obtention d'un Automate Partiellement Généralisé :**

Pour éliminer les transitions étiquetées par des mots de longueur supérieure ou égale à 2,

on ajoute des états intermédiaires.

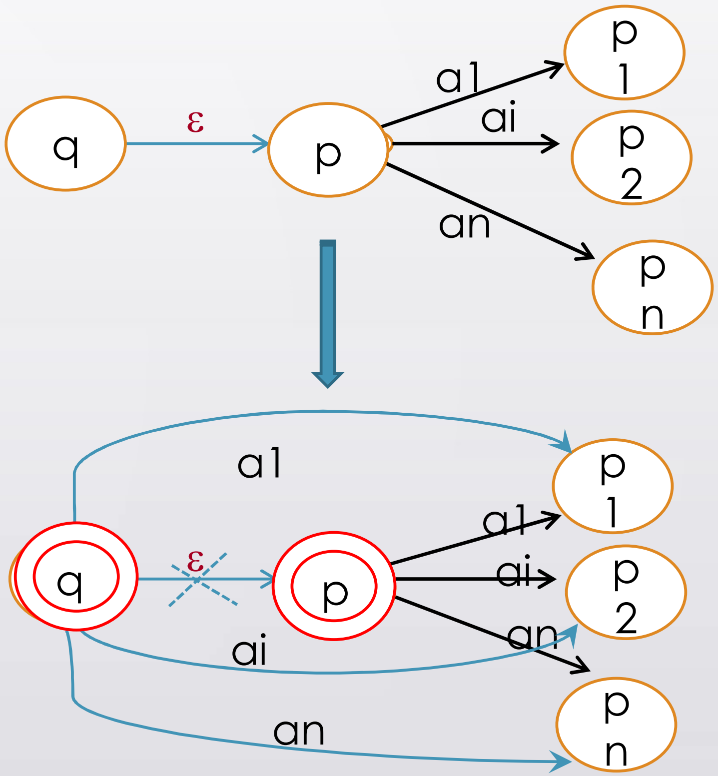


# Passage AEF Généralisé vers AEF Simple

**Obtention de l'Automate Simple :**

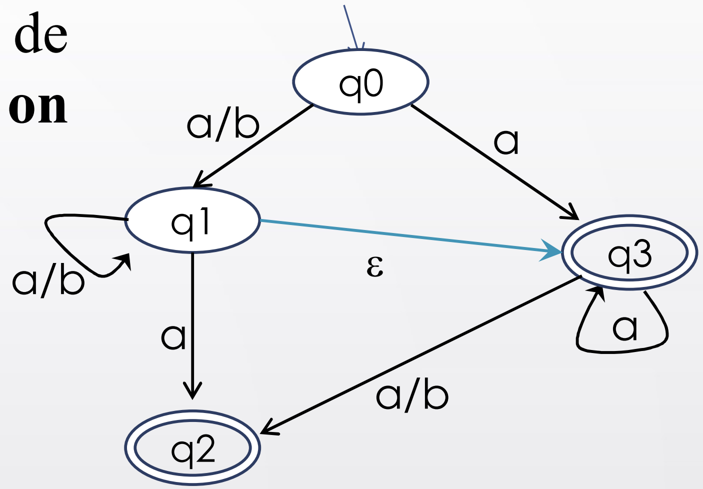
On élimine les transitions spontanées comme illustré dans le schéma. Ainsi pour éliminer **(q,ε,p)** on procède comme suit :




- pour toute transition reliant l'état p à un état $p_i$, **on ajoute une transition reliant directement q à $p_i$** et ayant la même étiquette.
- De plus, **si l'état p est final** alors **l'état q devient final.**
- On supprime la transition spontanée de q vers p .
- Cette opération est répétée jusqu'à l'obtention d'un automate simple (sans aucune transition spontanée).

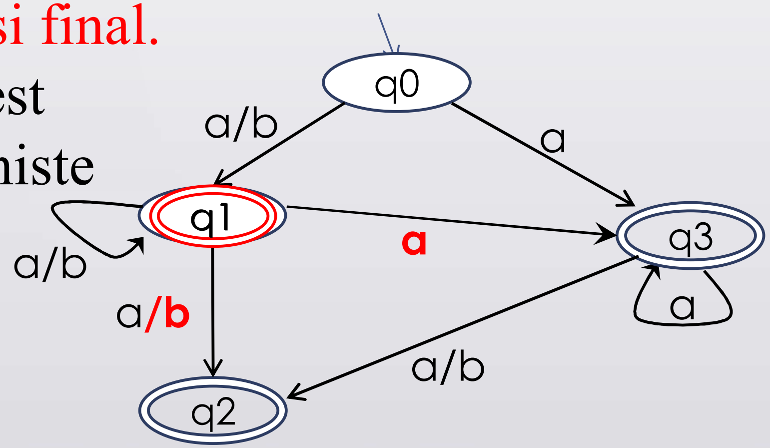
# Passage AEF généralisé vers AEF simple

En pratique, on utilise la table des transitions de l'automate **partiellement généralisé** dans laquelle **on ajoute une colonne pour la ε-transition.**




Pour éliminer la transition spontanée qui relie l'état $q_1$ à l'état $q_3$, on **reprend la ligne associée à l'état $q_1$** en lui ajoutant **case par case** (lettre par lettre) la ligne associée à $q_3$

Comme **l'état $q_3$ est final,**
l'état $q_1$ devient aussi final.
L'automate obtenu est simple non-déterministe




| Etat/let | a | b | ε |
|---|---|---|---|
| q0 | {q1, q3} | {q1} | - |
| ~~q1~~ | ~~{q1, q2}~~ | ~~{q1}~~ | ~~{q3}~~ |
| q2 | - | - | - |
| q3 | {q2, q3} | {q2} | - |
| q1 | {q1, q2, q3} | {q1, q2} | - |

# Grammaires et Automates

**Proposition :** Pour toute grammaire régulière droite $G=(T,N,S,P)$ il existe un **automate généralisé équivalent**.

**Démonstration :** Il s'agit de déterminer un automate fini $A=(X,Q,q_0,\delta,F)$ tel que $L(G)=L(A)$.

L'automate A est construit comme suit :

- $X = T$ Même alphabet pour L(G) et L(A)
- $Q = N\cup\{q_F\}$ et $q_F \notin N$ Les non-terminaux deviennent des états
- $q_0 = S$ L'axiome devient l'état initial
- $F = \{q_F\}$ Un seul état final = l'état spécial ajouté
- Si $A\rightarrow wB\in P$ alors $B\in\delta(A,w)$ Les productions deviennent des transitions
- Si $A\rightarrow w\in P$ alors $q_F\in\delta(A,w)$ Les productions d'arrêt deviennent des transitions vers l'état final

Les deux dernières règles sont schématisées comme suit :

# Grammaires et Automates

**Exemple :**

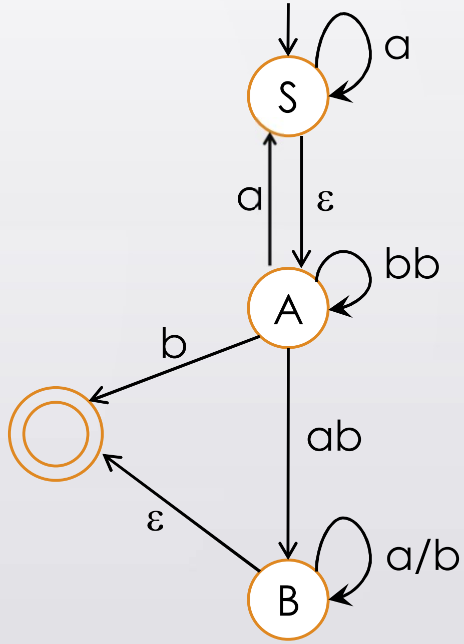
Soit G =({a, b}, {S, A, B}, S, P) une grammaire régulière droite telle que P est défini par :

$S \rightarrow aS \,/\, A$

$A \rightarrow bbA \,/\, aS \,/\, abB \,/\, b$

$B \rightarrow aB \,/\, bB \,/\, \varepsilon$

**AEF Equivalent**



# Grammaires et Automates

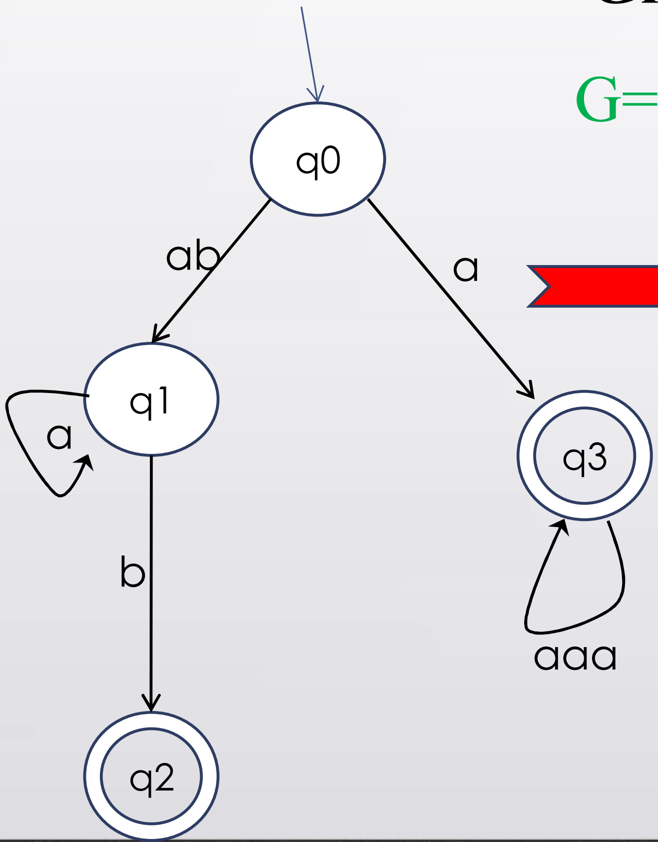
**Proposition :** Pour tout automate fini généralisé $A=(X,Q,q_0,\delta,F)$, il existe **une grammaire régulière droite équivalente.**

**Démonstration :** Il s'agit de déterminer une grammaire régulière droite $G=(T, N, S, P)$ telle que $L(A)=L(G)$.

La grammaire G est construite comme suit :

- $T = X$ Même alphabet pour (LA) et L(G)
- $N = Q$ Les états deviennent les non terminaux
- $S = q_0$ L'état initial devient l'axiome
- Si $q \in \delta(p, w)$ alors $(p \to wq) \in P$ Transitions deviennent des productions
- Si $q \in F$ alors $(q \to \varepsilon) \in P$ Etat final correspond à une production d'arrêt avec ε

# Grammaires et Automates




**Grammaire régulière droite :**

G=({a,b}, {q0,q1,q2,q3}, q0, P)

P est défini par :

**q0** → ab **q1** / a **q3**
**q1** → a **q1** / b **q2**
**q2** → ε
**q3** → aaa **q3** / **q1** / ε

# Fermetures des langages réguliers

**Théorème :**
La classe des langages réguliers est fermée par rapport aux opérations de l'union, la concaténation, l'étoile, le complément et le miroir.

Si L et M sont deux langages **réguliers** alors :

$L \cup M$, $L.M$, $L^*$, $\overline{L}$ et $L^R$ sont des langages **réguliers**.