

# **Les Langages Réguliers**

**S. Mazouz & B. Laichi**

**FEI-USTHB**

**Département Informatique**

**2019-2020**

**[campusvirtuel.usthb.dz](http://campusvirtuel.usthb.dz)**

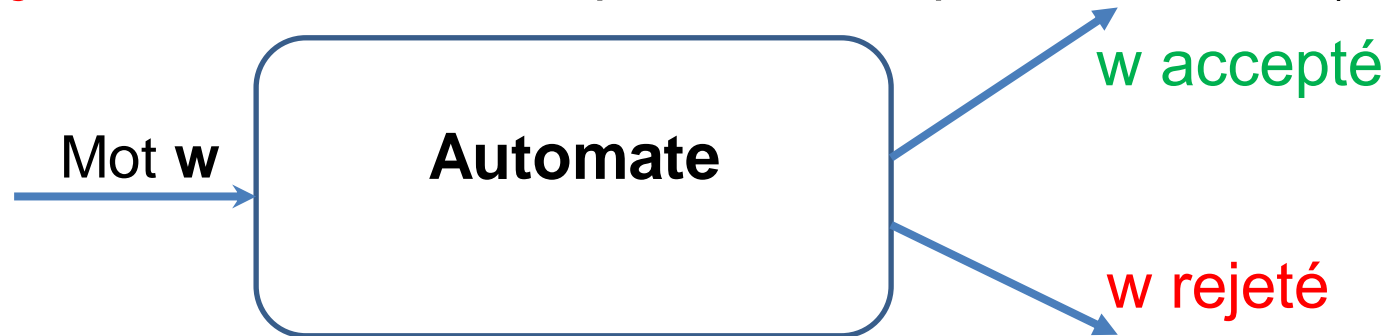
# Plan

- 1) Automates d'états finis
- 2) Grammaires régulières et automates d'états finis
- 3) Expressions régulières
- 4) Expressions régulières et automates d'états finis

# AUTOMATES D'ETATS FINIS

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



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'ETATS FINIS (AEF)

Un AEF lit les symboles d'un mot à reconnaître 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 1<sup>er</sup> 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 (état d'acceptation)**.

Ruban d'entrée de taille finie



**Unité de commande  
(Ensemble des transitions)**

# AUTOMATES D'ÉTATS FINIS

## Définition (Automate d'états finis)

Un automate d'états finis déterministe est un cinquiuplet

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

- $X$  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  $Q \times X$  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  $a$ .

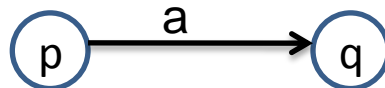
# AUTOMATES D'ÉTATS FINIS

## 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  $a \in X$

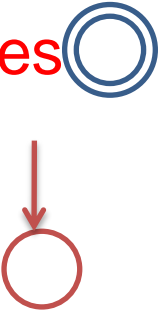
représente la **transition**  $\delta(p, a) = q$



# AUTOMATES D'ÉTATS FINIS

## 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 :

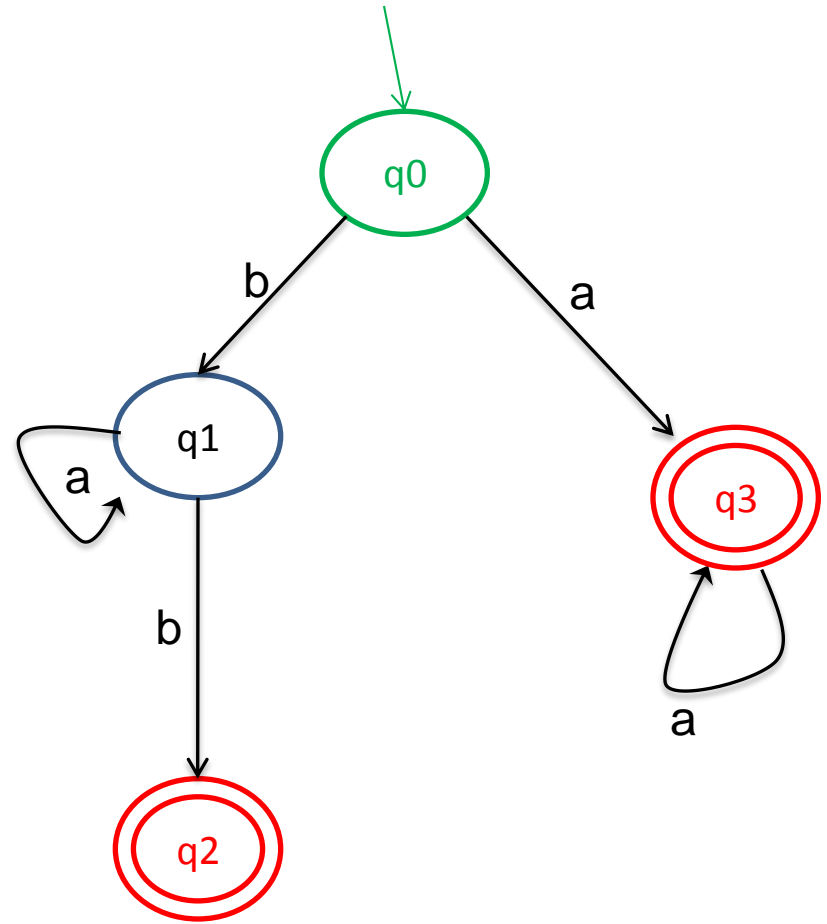


# AUTOMATES D'ÉTATS FINIS

**Exemple :** Soit l'AEF suivant

$A = (X, Q, q_0, \delta, F)$  tels que :

- $X = \{a, b\}$
- $Q = \{q_0, q_1, q_2, q_3\}$
- $q_0$  est l'état initial
- $\delta(q_0, a) = q_3$   
 $\delta(q_0, b) = q_1$
- $\delta(q_1, a) = q_1$   
 $\delta(q_1, b) = q_2$
- $\delta(q_3, a) = q_3$
- $F = \{q_2, q_3\}$



Cet Automate lit les mots =  $\{a, aa, aaa, \dots, bb, bab, baab, \dots\}$  mais pas  $\{ab, b\}$



# AUTOMATES D'ÉTATS FINIS

## Représentation matricielle :

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

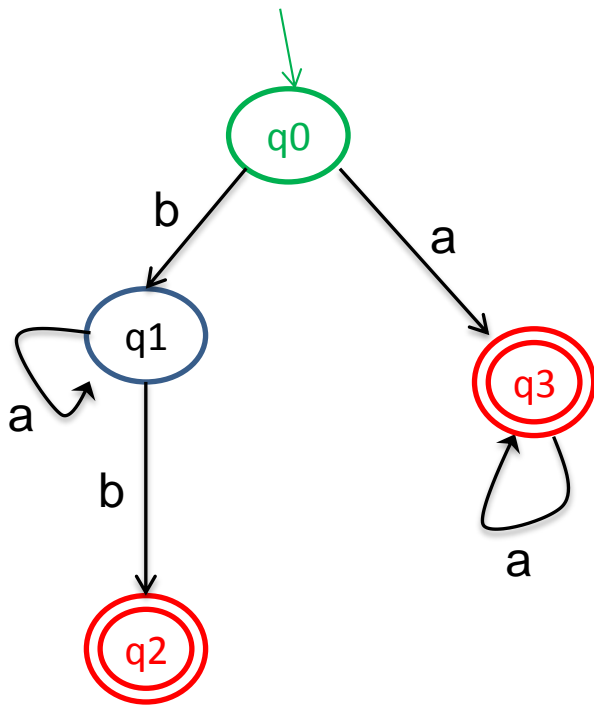
- les indices **des lignes** correspondent aux **états**
- les indices **des colonnes** correspondent aux **éléments de X**.
- un élément de la matrice de ligne **q** et colonne **a** correspond à la transition  **$\delta(q, a)$** .

état\lettre		a	
q		$\delta(q, a)$	

# AUTOMATES D'ETATS FINIS

Exemple :

Représentation  
Graphique



Représentation  
Matricielle

etat\lettre	a	b
<b>q0</b>	q3	q1
<b>q1</b>	q1	q2
<b>q2</b>	—	—
<b>q3</b>	q3	—

$\delta(q0, a)$

Transition non définie

# AUTOMATES D'ÉTATS FINIS

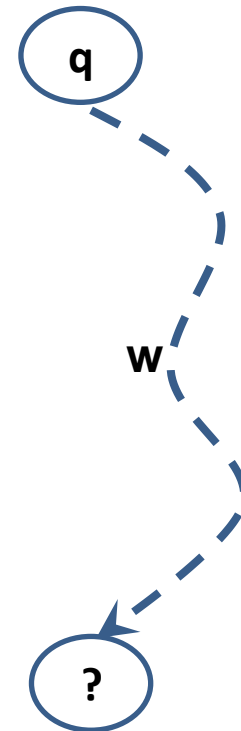
Langage reconnu par un automate d'états finis :

## 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.



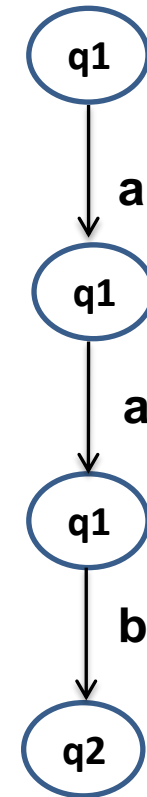
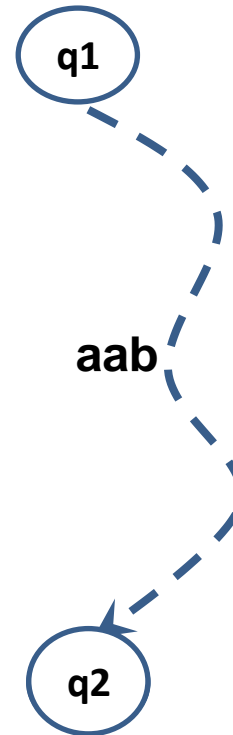
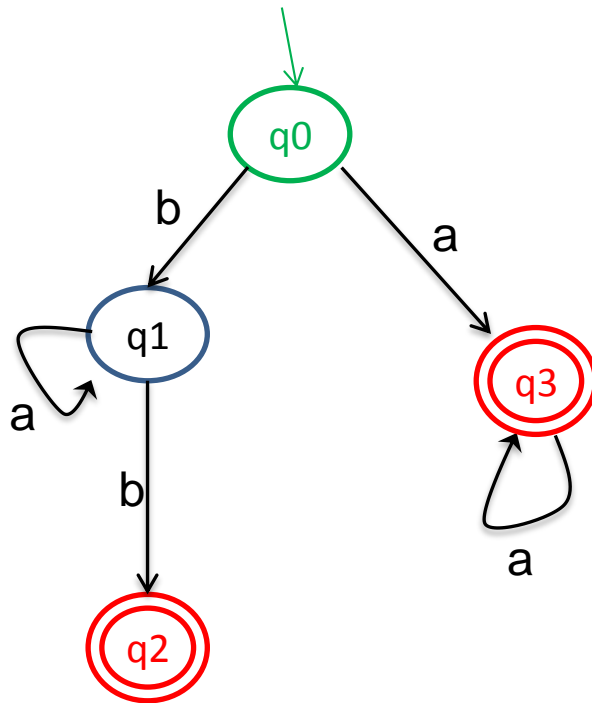
Définir la fonction de  
**succession de transitions.**



# AUTOMATES D'ETATS FINIS

## Fonction de succession de transitions :

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

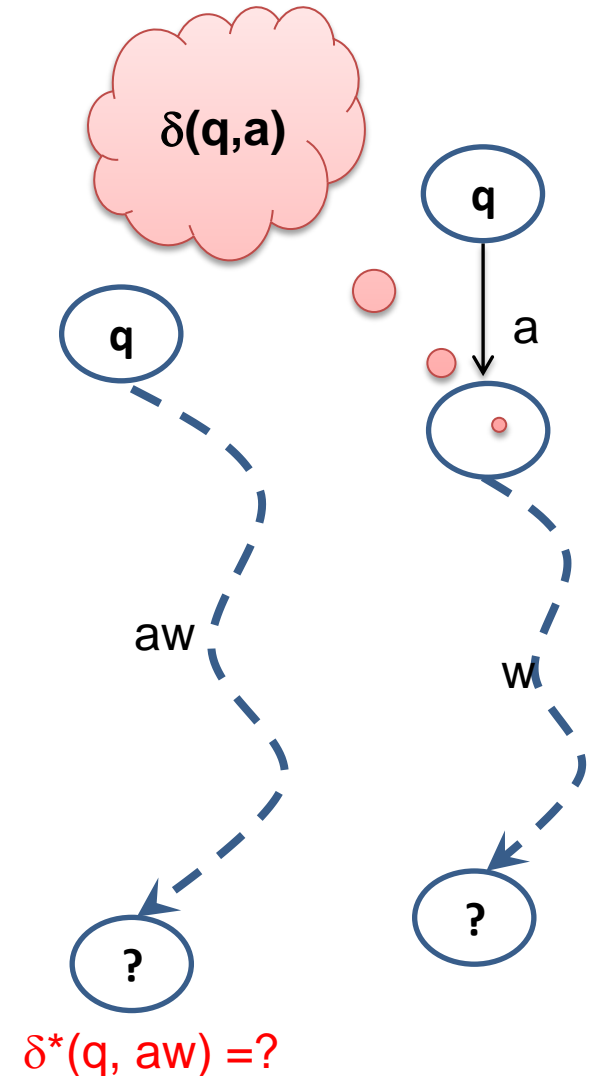


# AUTOMATES D'ÉTATS FINIS

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

On étend naturellement, la fonction de transition  $\delta$  à la fonction de succession de transitions  $\delta^*$  définie de  $Q \times X^*$  dans  $Q$  comme suit :

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



# AUTOMATES D'ÉTATS FINIS

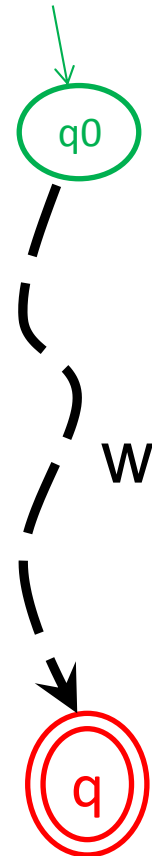
## Condition de reconnaissance :

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  $A$ ).



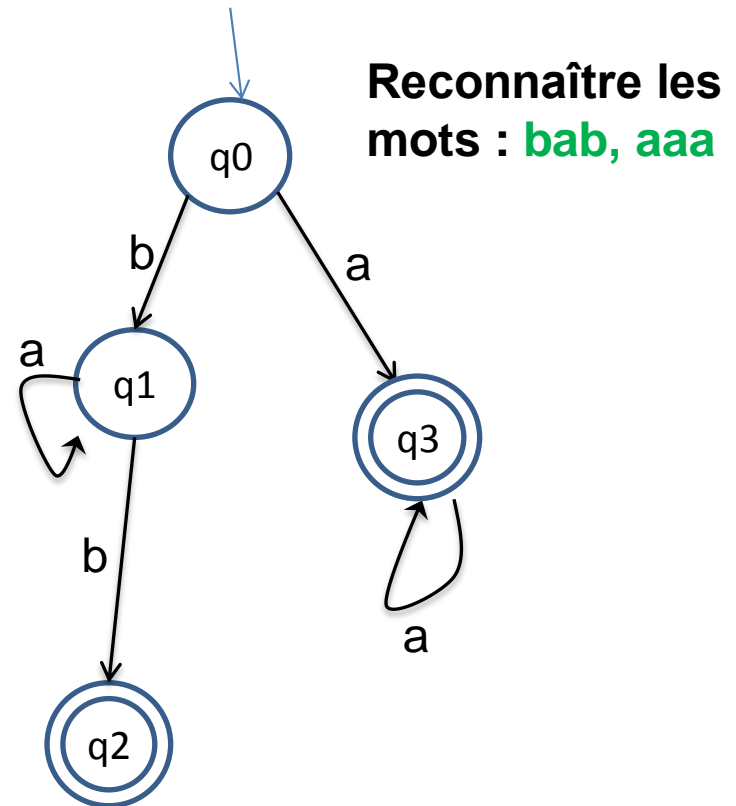
# AUTOMATES D'ÉTATS FINIS

$$\begin{aligned}\delta^*(q_0, bab) &= \delta^*(\delta(q_0, b), ab) \\ &= \delta^*(q_1, ab) \\ &= \delta^*(\delta(q_1, a), b) \\ &= \delta^*(q_1, b) \\ &= \delta(q_1, b) \\ &= q_2 \text{ et } q_2 \text{ est un état final}\end{aligned}$$

$$\begin{aligned}\delta^*(q_0, aaa) &= \delta^*(\delta(q_0, a), aa) \\ &= \delta^*(q_3, aa) \\ &= \delta^*(\delta(q_3, a), a) \\ &= \delta^*(q_3, a) \\ &= \delta(q_3, a) \\ &= q_3 \text{ et } q_3 \text{ est un état final}\end{aligned}$$

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

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



# AUTOMATES D'ÉTATS FINIS

$$\begin{aligned}\delta^*(q_0, ab) &= \delta^*(\delta(q_0, a), b) \\ &= \delta^*(q_3, b) \\ &= \delta(q_3, b)\end{aligned}$$

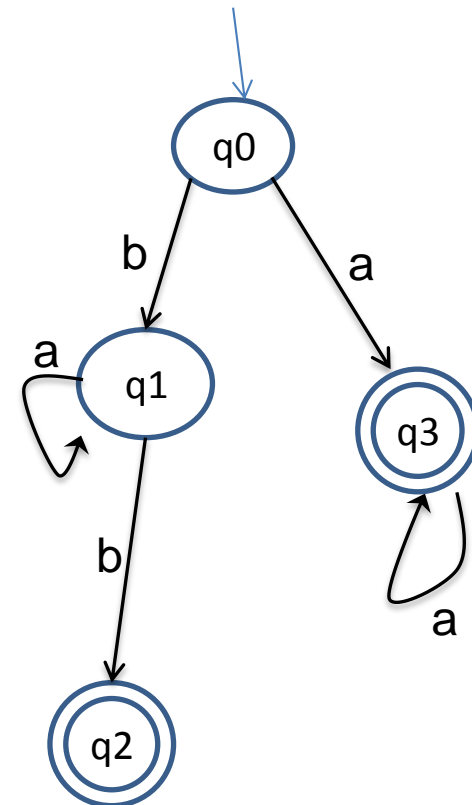
mais à l'état  $q_3$ , l'automate ne peut pas lire  $b$

$$\begin{aligned}\delta^*(q_0, ba) &= \delta^*(\delta(q_0, b), a) \\ &= \delta^*(q_1, a) \\ &= \delta(q_1, a) \\ &= q_1\end{aligned}$$

mais  $q_1$  n'est pas final

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

Reconnaître les  
mots :  $ab, ba$





# AUTOMATES D'ÉTATS FINIS

## 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

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

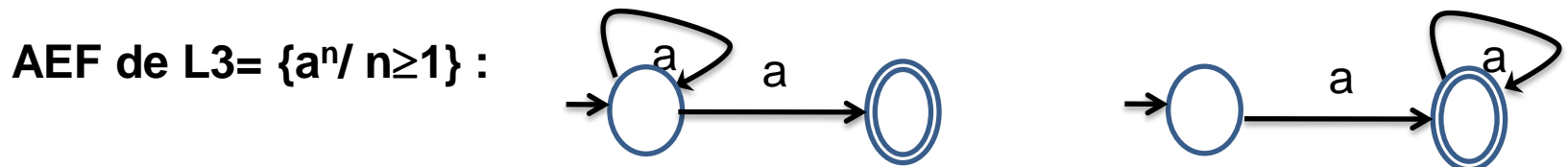
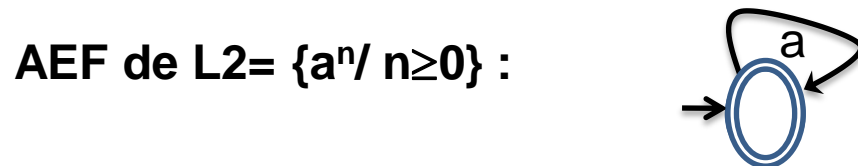
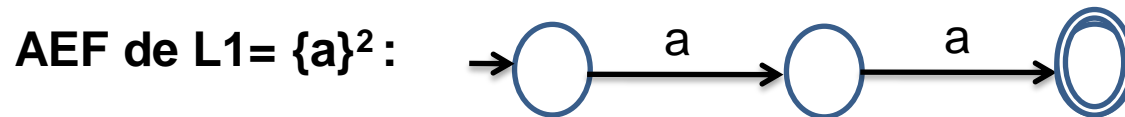
Un langage  $L$  sur  $X$  est **régulier** (ou **reconnaissable**) s'il existe au moins un automate d'états finis  $A$  ayant  $X$  comme alphabet d'entrée tel que  $L=L(A)$ .

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

# AUTOMATES D'ÉTATS FINIS

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

$L1 = \{a\}^2$ ,  $L2 = \{a^n / n \geq 0\}$  et  $L3 = \{a^n / n \geq 1\}$ ,



Cet AEF n'est pas déterministe

**Rappel :** Pour un AEF déterministe

La **fonction de transitions**  $\delta$  définie de  $Q \times X$  dans  $Q$

# AUTOMATES D'ÉTATS FINIS

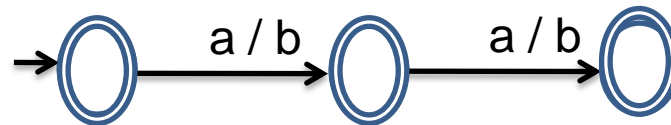
**Exemples :** Donner un AEF pour chacun des langages

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

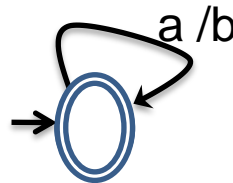


AEF de  $L2$  : on a  $\{a, b\}^2 \subseteq L2$  :

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

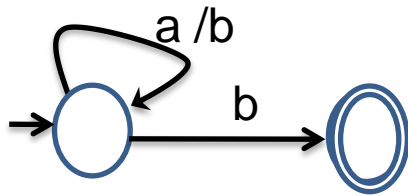


AEF de  $L3 = \{a, b\}^*$  :

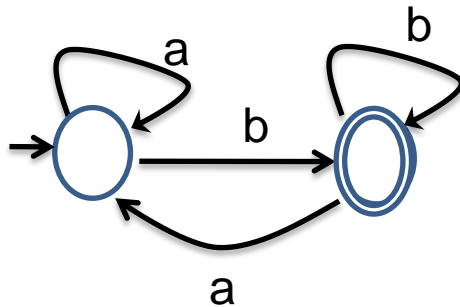


# AUTOMATES D'ETATS FINIS

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



**Cet AEF n'est pas déterministe**



**Cet AEF est déterministe**

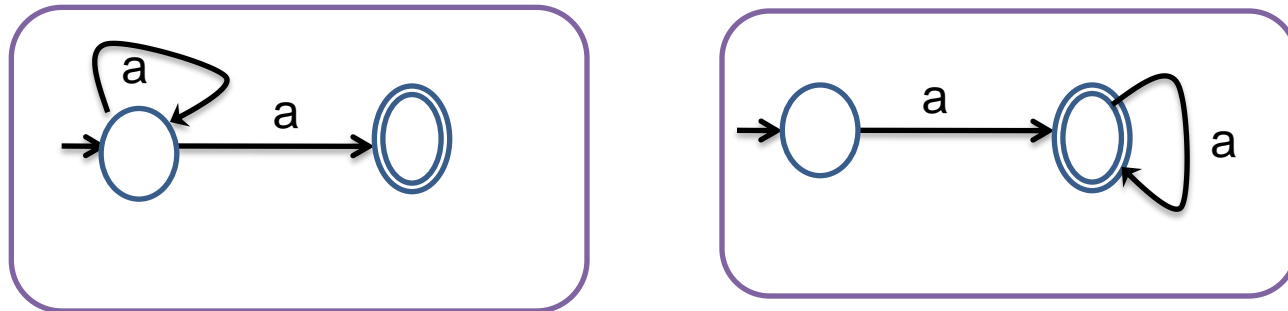
# AUTOMATES D'ÉTATS FINIS

## Définition (Equivalence de deux AEFs)

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

$$A1 \equiv A2 \Leftrightarrow L(A1) = L(A2)$$

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



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

# LES VARIANTES DES AUTOMATES D'ÉTATS FINIS

Il existe plusieurs variantes d'automates d'états finis:

- 1) Les automates déterministes.
- 2) Les automates non-déterministes.
- 3) Les automates généralisés.

# AUTOMATES DÉTERMINISTES

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

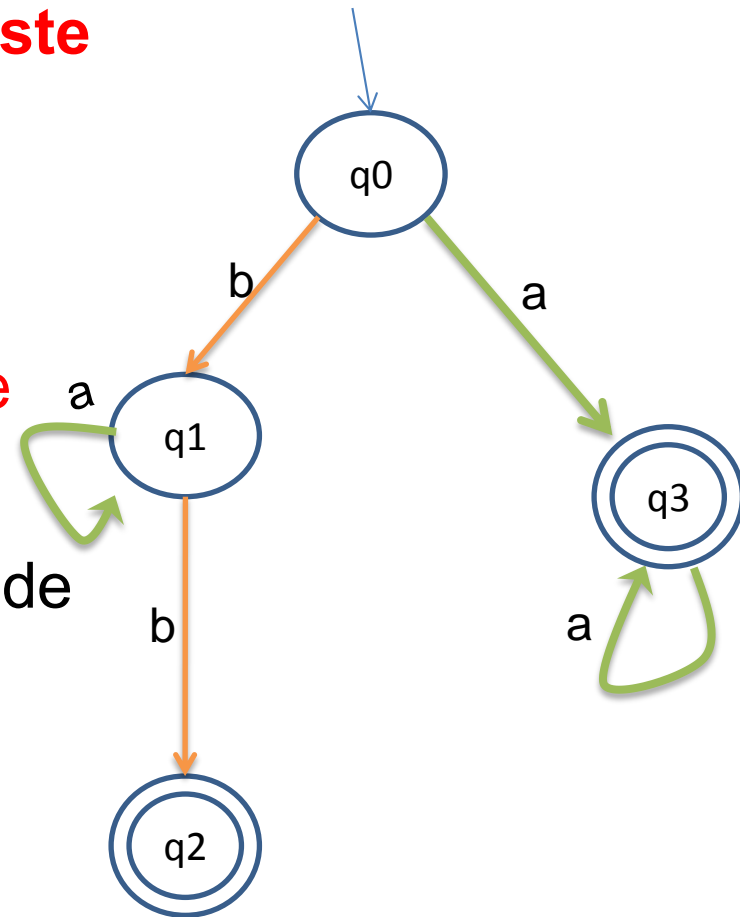
- à un état
- à un symbole d'entrée

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

Autrement dit, la fonction  $\delta$  est définie de  $Q \times X$  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  $(q,a) \in Q \times X$ , la fonction  $\delta$  associe  
**exactement un état**.

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

## Remarque :

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

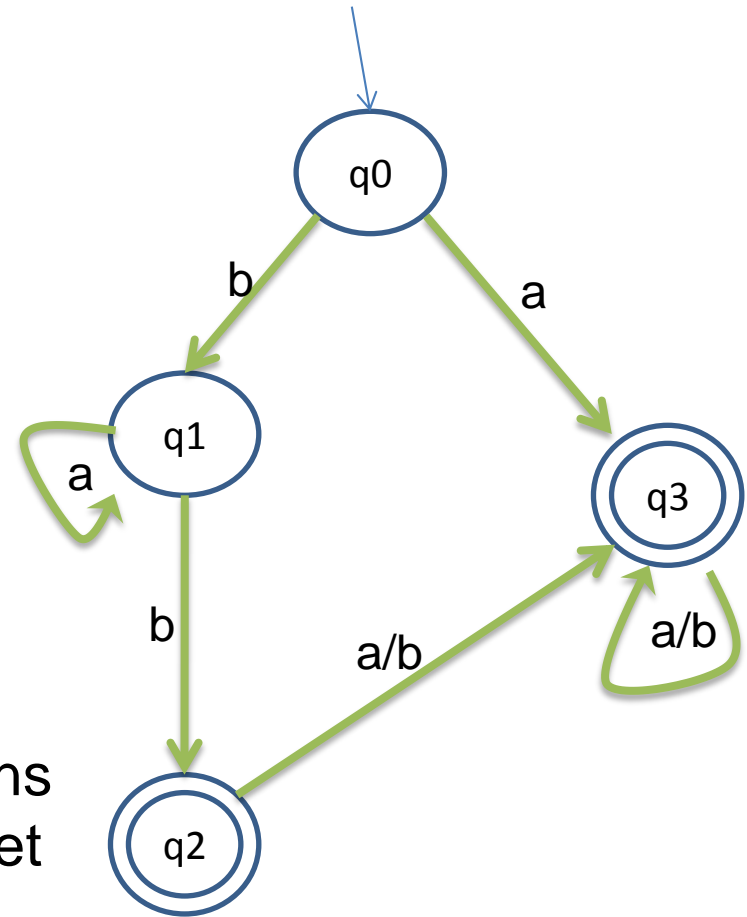


# AUTOMATES DÉTERMINISTES

**Exemple :**

état\lettre	a	b
q0	q3	q1
q1	q1	q2
q2	q3	q3
q3	q3	q3

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



**Automate déterministe et complet**

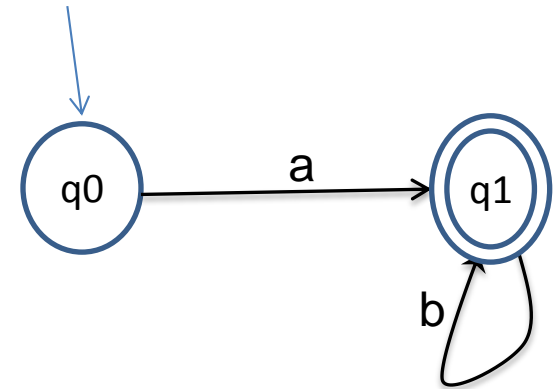
# AUTOMATES DÉTERMINISTES

**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 :  $\{ab^n / n \geq 0\}$ .

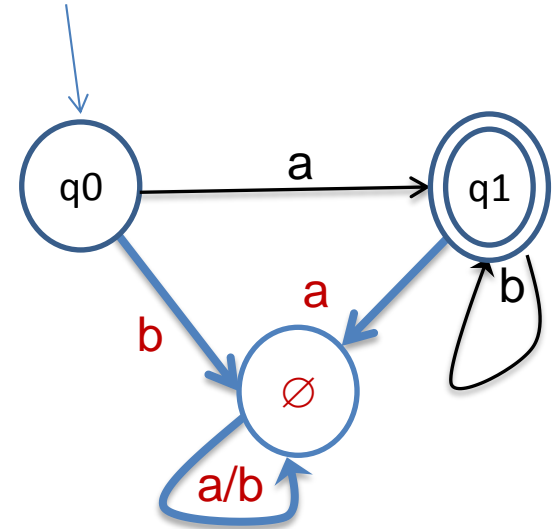


## Remarques :

- ✓ 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 «**états puits**»  
généralement noté  $\emptyset$ , et de rajouter  
toutes **les transitions manquantes vers cet état**.



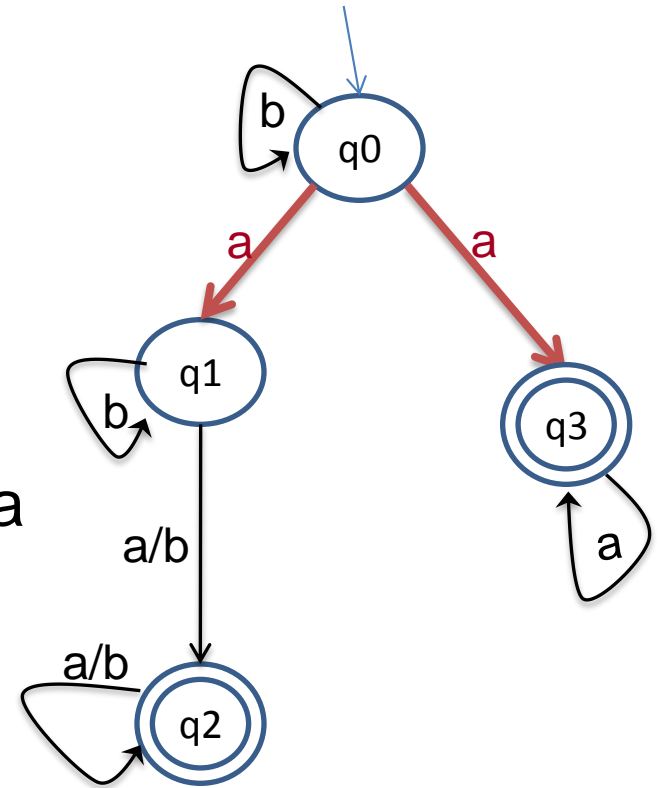
## Remarques :

- ✓ Un automate déterministe complet permet de lire tous les mots de l'alphabet (pas nécessairement les reconnaître)
- ✓ 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  $q_0$ , on a le choix entre **deux transitions par  $a$**  : l'une vers  $q_1$  et l'autre vers  $q_3$ .  
Donc,  $\delta(q_0, a) = \{q_1, q_3\}$

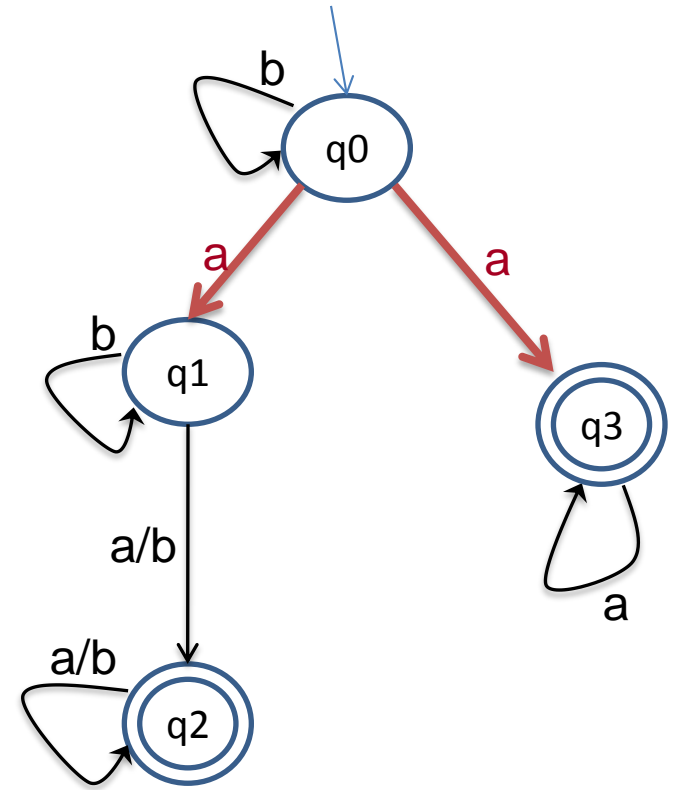


Dans ce cas,  $\delta$  est une fonction de transition définie de :  
 $Q \times X$  dans **l'ensemble des parties** de  $Q$ .  $\delta : Q \times X \rightarrow \mathcal{P}(Q)$

# AUTOMATES NON DÉTERMINISTES

Exemple :

etat\lettre	a	b
q0	<b>{q1, q3}</b>	{q0}
q1	{q2}	<b>{q1, q2}</b>
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'à maintenant, les transitions se faisaient en lisant un seul symbole à la fois).

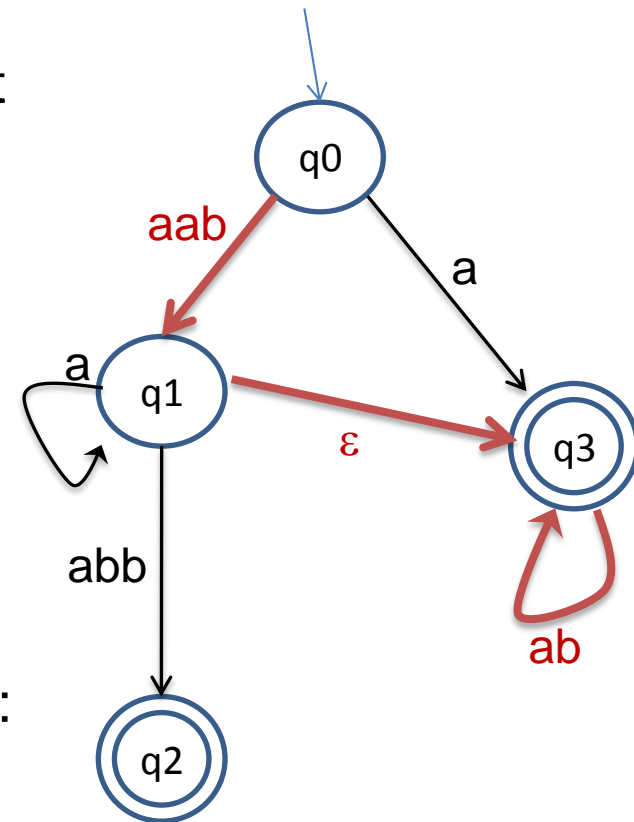
Les transitions directes **causées par le mot vide** sont appelées **transitions spontanées** ou  **$\varepsilon$ -transitions**.

Une  $\varepsilon$ -transition correspond à la situation où l'automate change d'état sans lire de symbole.

Dans l'exemple, à l'état  $q_1$ , on a 3 possibilités :

- lire **a** et rester dans  $q_1$
- lire **abb** et passer à  $q_2$
- passer directement à  **$q_3$**  sans lecture.

La fonction  $\delta$  est alors définie de:  **$Q \times X^* \rightarrow \mathcal{P}(Q)$**



$$\delta(q_0, \mathbf{aab}) = \{q_1\}$$

$$\delta(q_1, \boldsymbol{\varepsilon}) = \{q_3\}$$

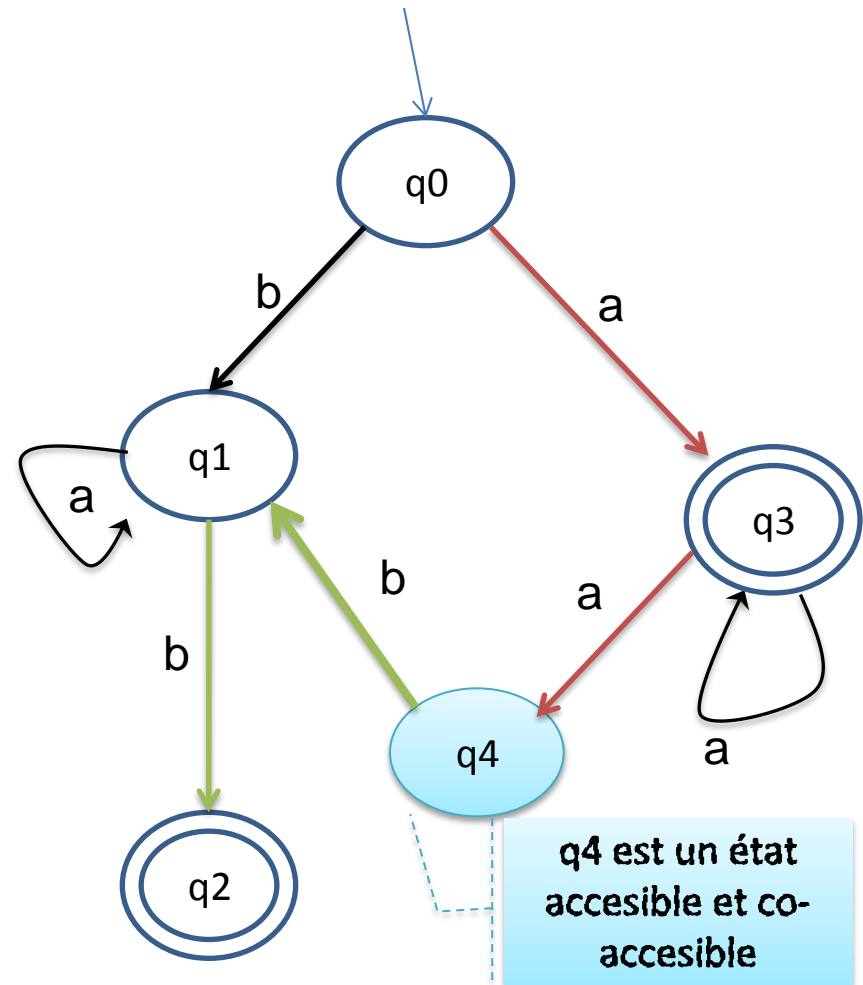
# AUTOMATES D'ÉTATS FINIS

## Définitions :

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

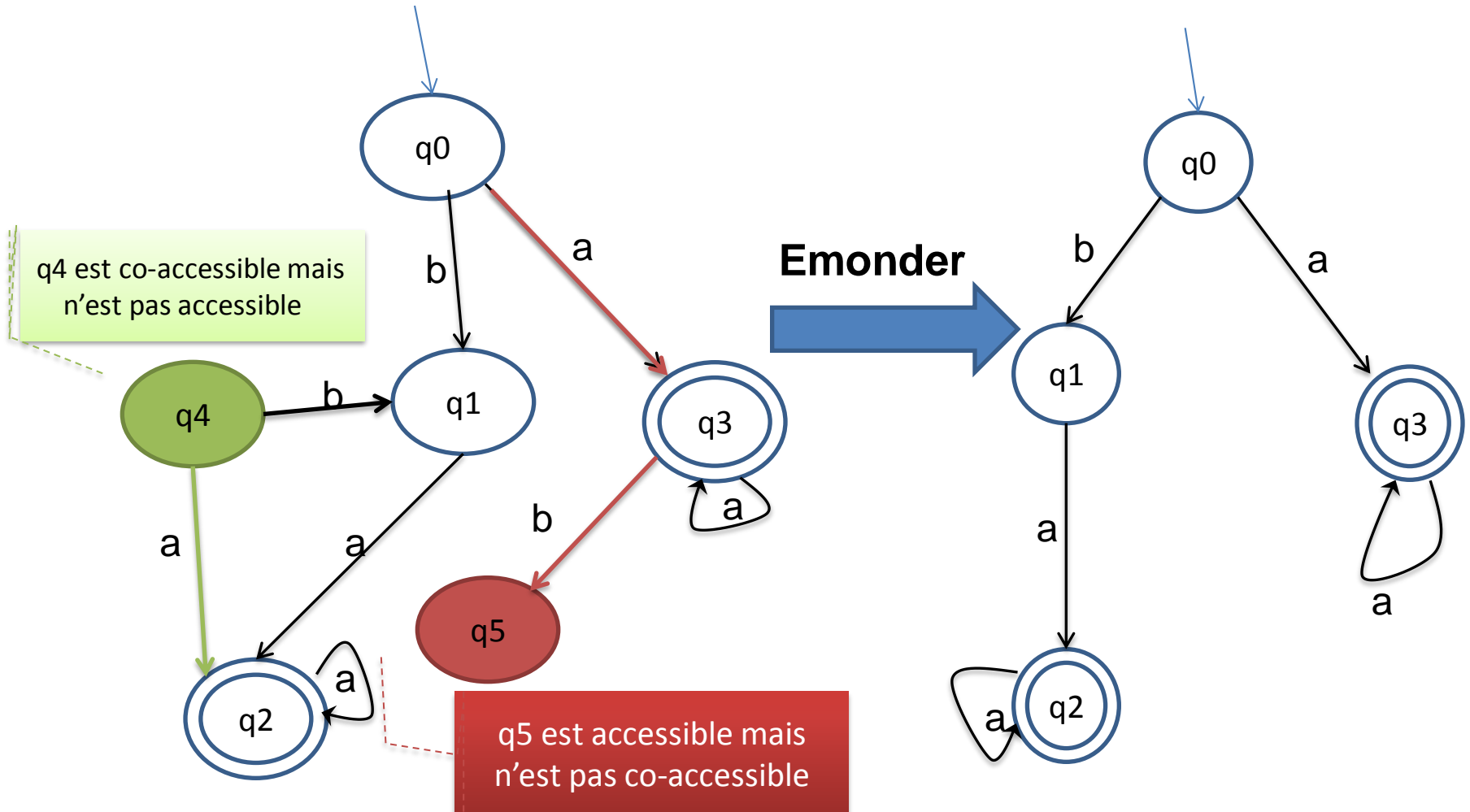
Un état  $q$  est **co-accessible** s'il existe un chemin de l'état  $q$  vers un état final.

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.





# AUTOMATE SIMPLE VS GÉNÉRALISÉ

## 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 à la fois)

# TRANSFORMATION DES AUTOMATES

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

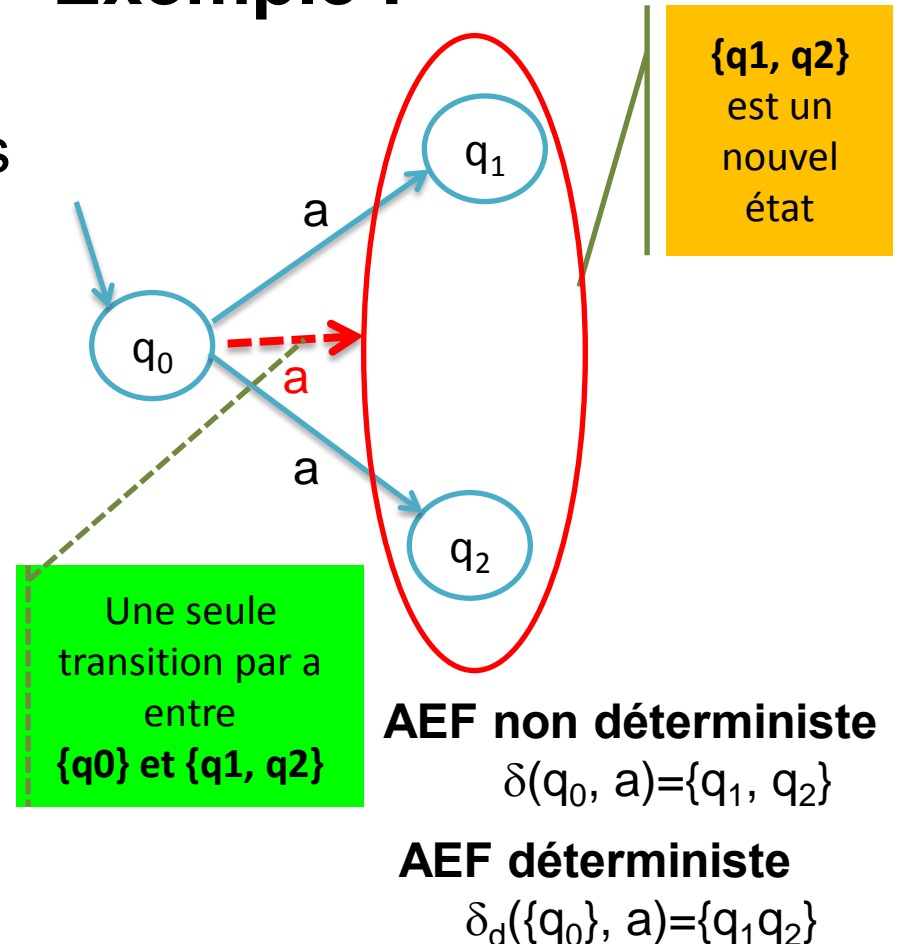
On commence d'abord par présenter la deuxième transformation

# 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).

## Exemple :



# 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 :

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


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  $\{q_0\}$** , on regroupe les transitions étiquetées par la même lettre issue de  $q_0$  :

etat\lettre	a1	.....	an
 $q_0$	$\delta(q_0, a_1)$		$\delta(q_0, a_n)$

AEF Non déterministe

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

etat\lettre	a1	.....	an
 $\{q_0\}$	$\delta(q_0, a_1)$		$\delta(q_0, a_n)$

AEF déterministe

# PASSAGE AEF NON DÉTERMINISTE VERS AEF DÉTERMINISTE

Pour chaque ensemble d'états  $q_d = \{q_1, \dots, q_n\}$  nouvellement obtenu et donc accessibles à partir de  $q_{0d}$ , on détermine les transitions issues de  $q_d = \{q_1, \dots, q_n\}$  i.e.  $\delta_d(q_d, a)$ ,  $\forall a \in X$ .

On **regroupe** les transitions étiquetées par la même lettre :

$$\delta_d(q_d, a) = \delta(q_1, a) \cup \dots \cup \delta(q_n, a)$$

Pour chaque lettre, on fait **l'union des cases** des différentes lignes associées aux états  $q_1, \dots, q_n$ .

etat\lettre	a	.....
q1	$\delta(q_1, a)$	
qn	$\delta(q_n, a)$	

AEF non déterministe

etat\lettre	a
qd	$\delta(q_1, a) \cup \dots \cup \delta(q_n, a)$

AEF déterministe

# PASSAGE AEF NON DÉTERMINISTE VERS AEF DÉTERMINISTE

On arrêtera la procédure dès 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

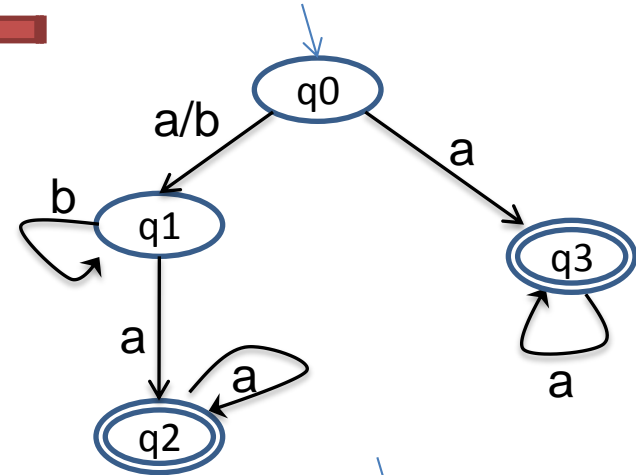
- Recopier la ligne de l'état initial.
- {q1,q3} et {q1} : nouveaux états accessibles
- Pour l'état {q1,q3}, on regroupe les transitions partant de q1 et celles partant de q3 par la même lettre et on obtient {q2,q3}.
- Pour l'état {q1}, il suffit de prendre les transitions de q1.
- On traite les nouveaux états accessibles de la même manière
- Les états {q1, q3}, {q2,q3} et {q2} sont finaux.



# PASSAGE AEF NON DÉTERMINISTE VERS AEF DÉTERMINISTE

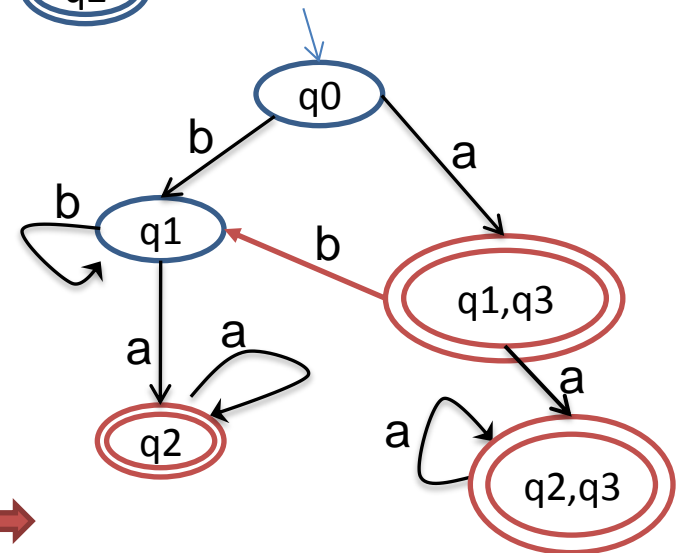
Représentation matricielle

etat\lettre	a	b
→ q0	{q1, q3}	{q1}
q1	{q2}	{q1}
⊙ q2	{q2}	—
⊙ q3	{q3}	—



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

# PASSAGE AEF GÉNÉRALISÉ VERS AEF SIMPLE

## Proposition :

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

## Principe de la construction :

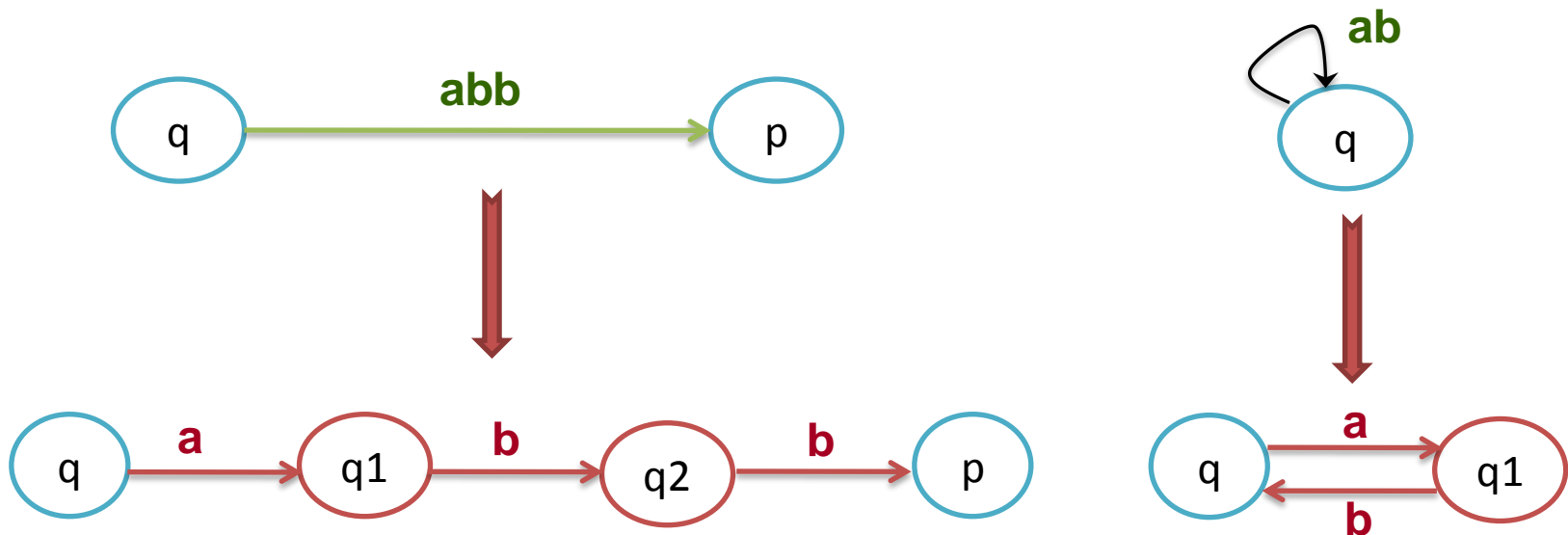
- **Éliminer les transitions de longueur strictement supérieure à 1.** On obtient un automate partiellement généralisé  $A_p$ .
- **Éliminer les transitions spontanées dans l'automate partiellement généralisé obtenu.** On obtient un automate simple.

# 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.



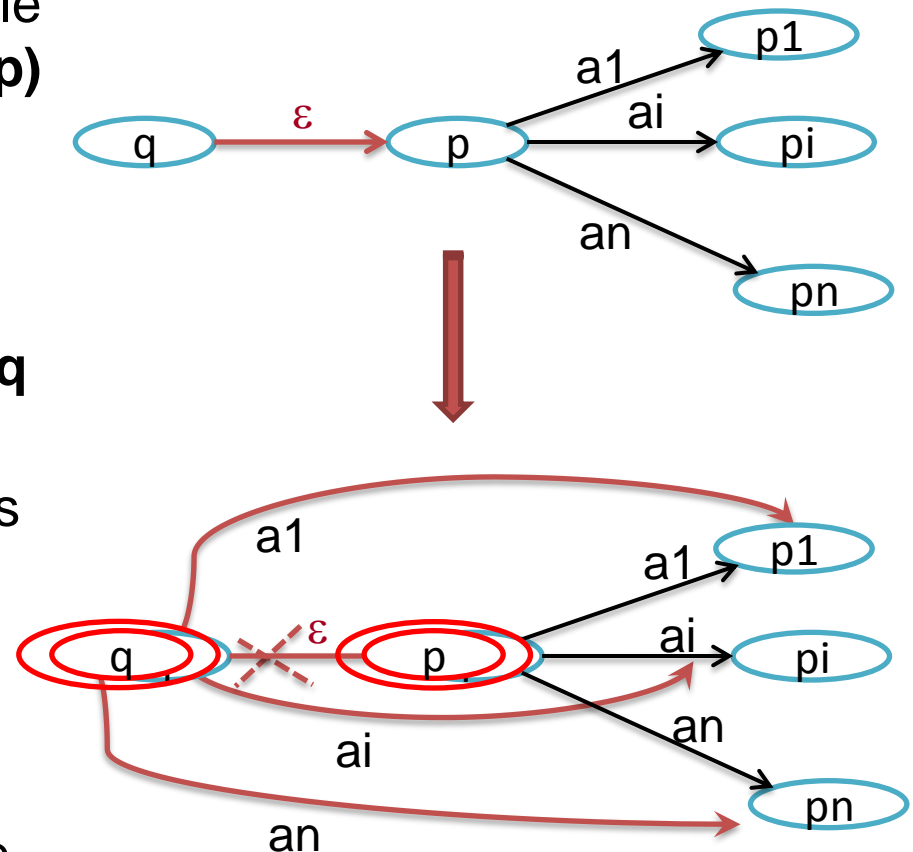
# PASSAGE AEF GÉNÉRALISÉ VERS AEF SIMPLE

## Obtention de l'Automate Simple :

On élimine les **transitions spontanées** comme illustré dans le schéma. Ainsi pour éliminer  $(q, \varepsilon, p)$  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).

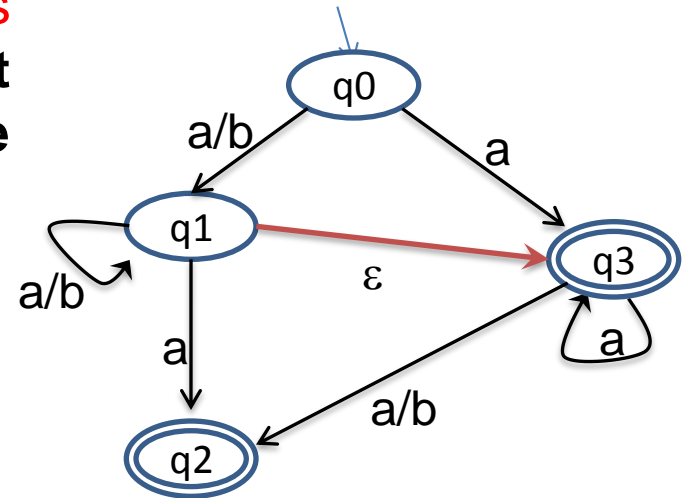
## Exemple :



# 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  $\varepsilon$ -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 l'état  $q_3$ .



Etat/let	a	b	$\varepsilon$
q0	{q1, q3}	{q1}	-
<del>q1</del>	<del>{q1, q2}</del>	<del>{q1}</del>	<del>{q3}</del>
q2	-	-	-
q3	{q2, q3}	{q2}	-
q1	{q1, q2, q3}	{q1, q2}	-

Comme l'état  $q_3$  est final, l'état  $q_1$  devient aussi final. L'automate obtenu est simple et dans ce cas il est non-déterministe.

# 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 :

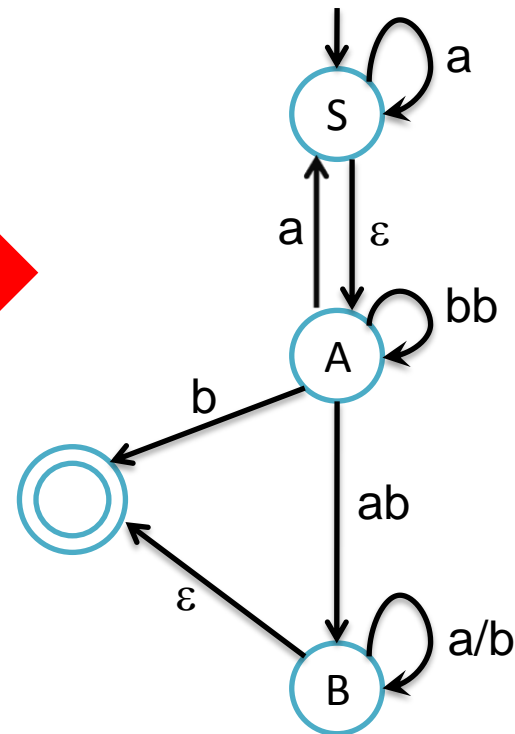
$S \rightarrow aS / A$

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

$B \rightarrow aB / bB / \varepsilon$



## 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**.

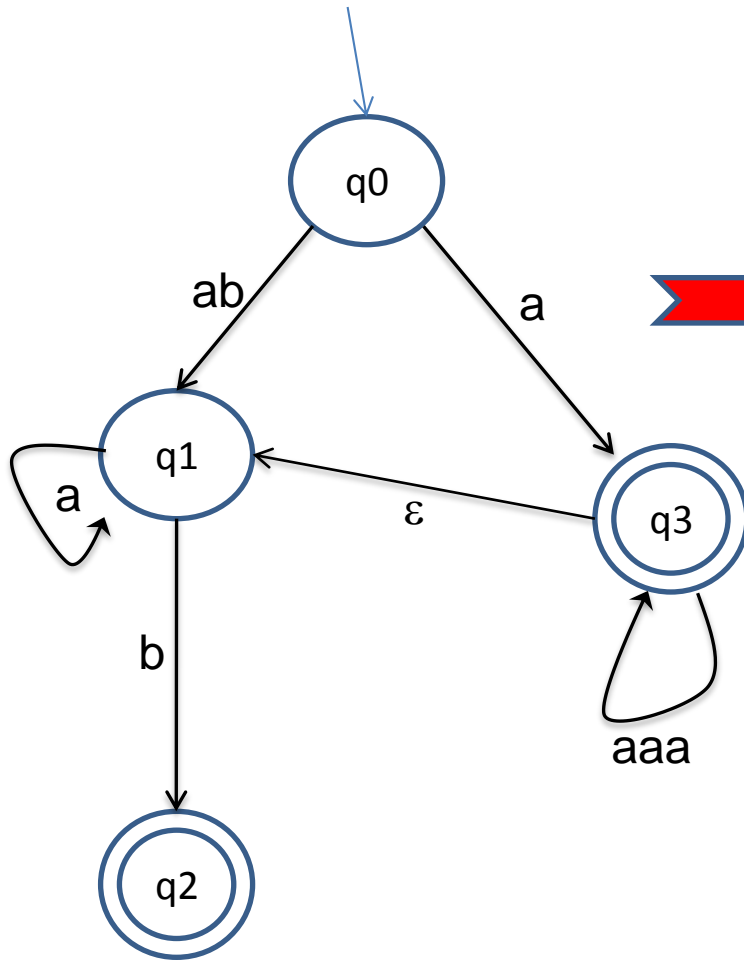
**Démonstration :** Il s'agit de déterminer une grammaire régulière droite  $G=(T, N, S, P)$  telle que  $L(A)=L(G)$ .

La grammaire  $G$  est construite comme suit :

- $T = X$  Même alphabet pour  $L(A)$  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  $(p \rightarrow wq) \in P$  Transitions deviennent des productions
- Si  $q \in F$  alors  $(q \rightarrow \varepsilon) \in P$  Etat final correspond à une production d'arrêt avec  $\varepsilon$



# GRAMMAIRES ET AUTOMATES



**Grammaire régulière droite :**

$G = (\{a, b\}, \{q_0, q_1, q_2, q_3\}, q_0, P)$

**P est défini par :**

**$q_0 \rightarrow ab\ q_1 / a\ q_3$**

**$q_1 \rightarrow a\ q_1 / b\ q_2$**

**$q_2 \rightarrow \varepsilon$**

**$q_3 \rightarrow aaa\ q_3 / q_1 / \varepsilon$**

# 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**.