Les langages Réguliers

S. Mazouz
Département Informatique
USTHB
2010-2011

Plan

I. Rappel sur les grammaires régulières

II. Automates d’états finis

III. Grammaire régulière et Automate d’états fini

IV. Expressions Régulières

V. Expression Régulière et Automate d’Etats Fini

Rappel Grammaire Régulière

Grammaire régulière droite

Une grammaire G=(T, N, S, P) est régulière (ou linéaire) droite si toutes ses productions sont de la forme $A \rightarrow w\mathbf{B}$ ou $A \rightarrow w$ avec A, $B \in N$ et $w \in T^*$.

Grammaire régulière gauche

Une grammaire G=(T, N, S, P) est régulière (ou linéaire) gauche si toutes ses productions sont de la forme $A \rightarrow \mathbf{B}w$ ou $A \rightarrow w$ avec A, $B \in N$ et $w \in T^*$.

Remarque

Une grammaire régulière droite génère les mots de la gauche vers la droite alors qu'une grammaire régulière gauche génère les mots de la droite vers la gauche.

Un langage est régulier s'il est généré par une grammaire régulière.

On notera Reg(X*) la famille des langages réguliers sur X*.

Rappel Grammaire Régulière

Exemple

La grammaire G_1= ({a, b}, {S, A}, S, P_1) avec P_1

S $\rightarrow$ **S**a/**A**a / **A**

A $\rightarrow$ **A**b/b

est **régulière gauche**.

Attention la grammaire G_2 = ({a, b}, {S, A}, S, P_2) avec P_2

S $\rightarrow$ a**S**/a**A**

A $\rightarrow$ **A**b/b

n'est ni régulière gauche ni régulière droite. En effet la règle **S**$\rightarrow$a**S** est de la forme **A** $\rightarrow$w**B** mais la règle **A**$\rightarrow$**A**b est de la forme **A** $\rightarrow$**B**w.

Automates d’Etats Finis (AEF)

Définition (Automate d’états finis)

Un automate d’états finis déterministe est un cinq-uplé *A* =(X, Q, q_0, δ, F) 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**
- δ est une **fonction de transitions** qui associe à chaque état et chaque symbole un état d’arrivée.

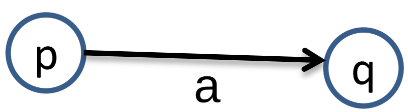
Automates d'Etats Finis

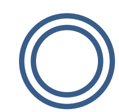
Représentation graphique

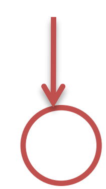
Les automates d'états finis sont souvent représentés par des graphes orientés dont :

- les **sommets** correspondent aux **états** et les **arcs** 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 $\delta(p, a)= q$



- un **état final** est représenté par **deux cercles concentriques :** 

- un **état initial** est représenté par une **flèche incidente sur l'état initial** 

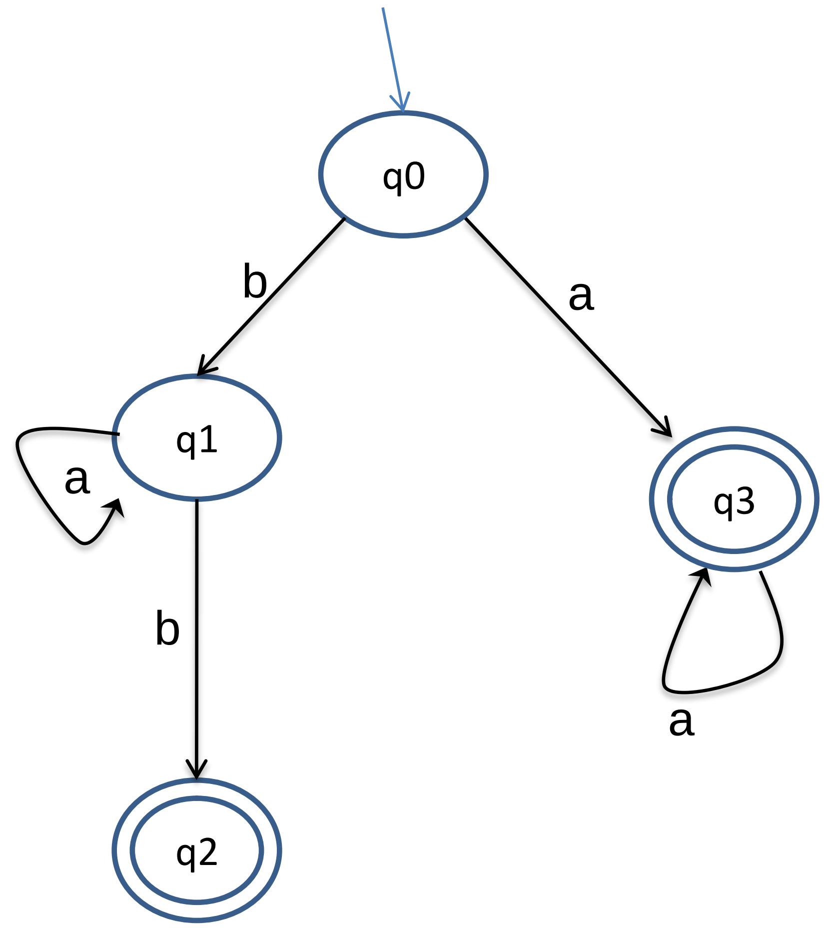
Automates d'Etats Finis

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

- $X=\{a, b\}$
- $Q=\{q_0, q_1, q_2, q_3\}$
- q0 est l'état initial
- $\delta(q_0, a)=q_3$, $\delta(q_0, b)=q_1$

  $\delta(q_1, a)=q_1$, $\delta(q_1, b)=q_2$

  $\delta(q_3, a)=q_3$
- $F=\{q2, q3\}$



Automates d'Etats Finis

Représentation matricielle

Les automates d'états finis peuvent aussi être représentés par une matrice dont :

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

Etat\lettre	a	b
q0	q3	q1
q1	q1	q2
q2	_	_
q3	q3	_

Transition non définie

Automates d'Etats Finis

Langage reconnu par un automate d'états finis

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

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

- $\delta^*(q, \varepsilon) = q$
- $\delta^*(q, aw) = \delta^*(\delta(q, a), w)$

Un **mot w est reconnu (accepté)** par l'automate si et seulement si il **existe un état final** $q_F \in F$ tel que $\delta^*(q_0, w) = q_F$.

Autrement dit, l'automate **A lit le mot w à partir de q0 et atteint un état final.**

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

Automates d'Etats Finis

Les mots bb et aaa sont acceptés par cet automate :

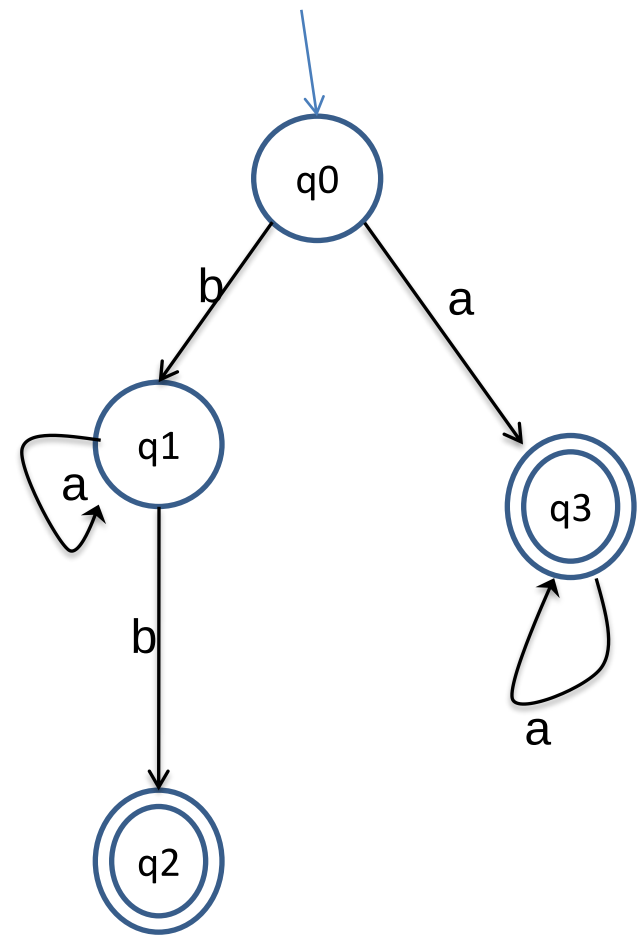
$\delta^*(q0, bb) = \delta^*(\boldsymbol{\delta(q0, b)}, b) = \delta^*(\mathbf{q1}, b)$

$= \delta(q1,b) =$ **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 ab et ba ne sont pas acceptés :

$\delta^*(q0, ab) = \delta^*(\boldsymbol{\delta(q0,a)}, b) = \delta^*(q3, b)$ mais à l'état q3, l'automate ne peut pas lire b

$\delta^*(q0, ba) = \delta^*(\boldsymbol{\delta(q0,b)}, a) = \delta^*(\mathbf{q1}, a) = q1$ mais q1 n'est pas final

Automates d'Etats Finis

Définition

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 reconnaissable** s'il existe au moins un automate fini A ayant X comme alphabet d'entrée et tel **que L= L(A).**

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

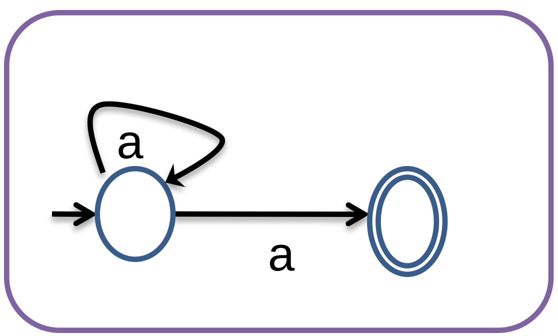
Automates d'Etats Finis

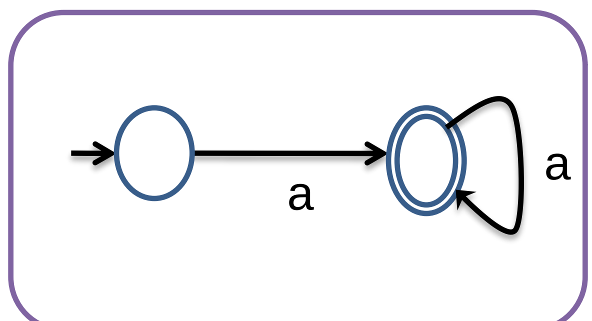
Définition

Deux automates finis A1 et A2 de sont équivalents si et seulement s'ils acceptent le même langage.

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

Exemple Ces deux automates sont équivalents . Ils génèrent le langage $\{a^n / n \geq 1\}$





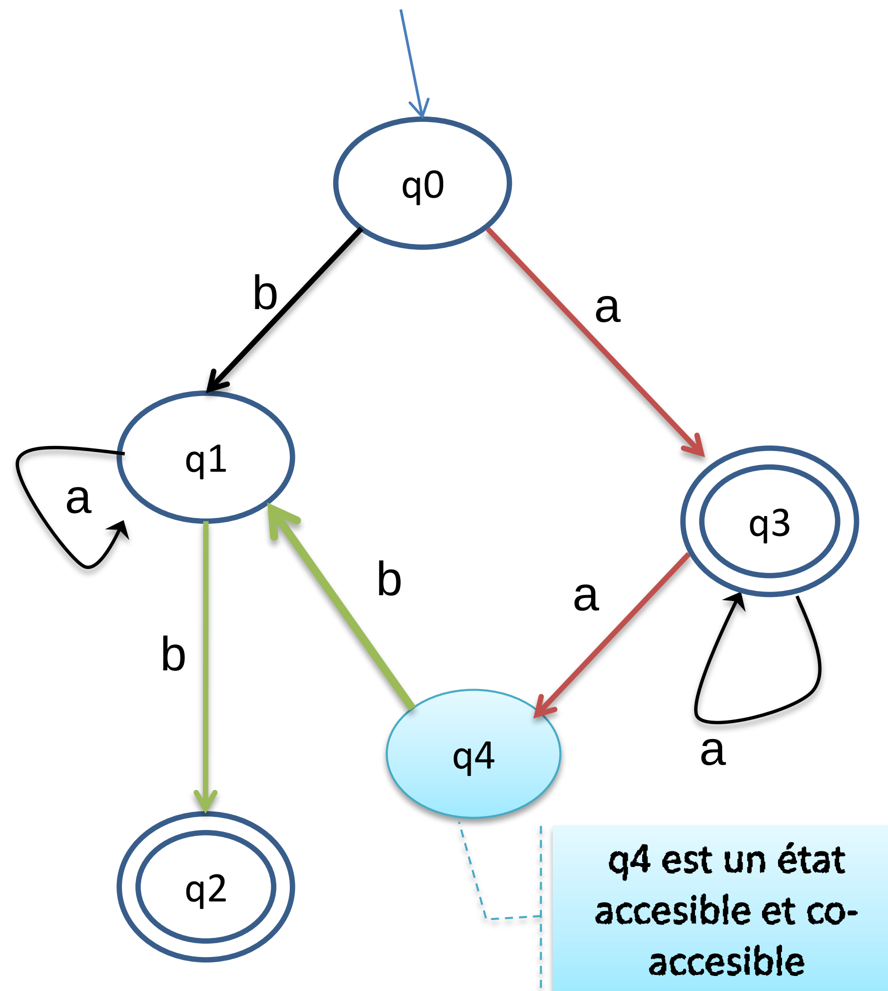
Automates d'Etats Finis

Définition

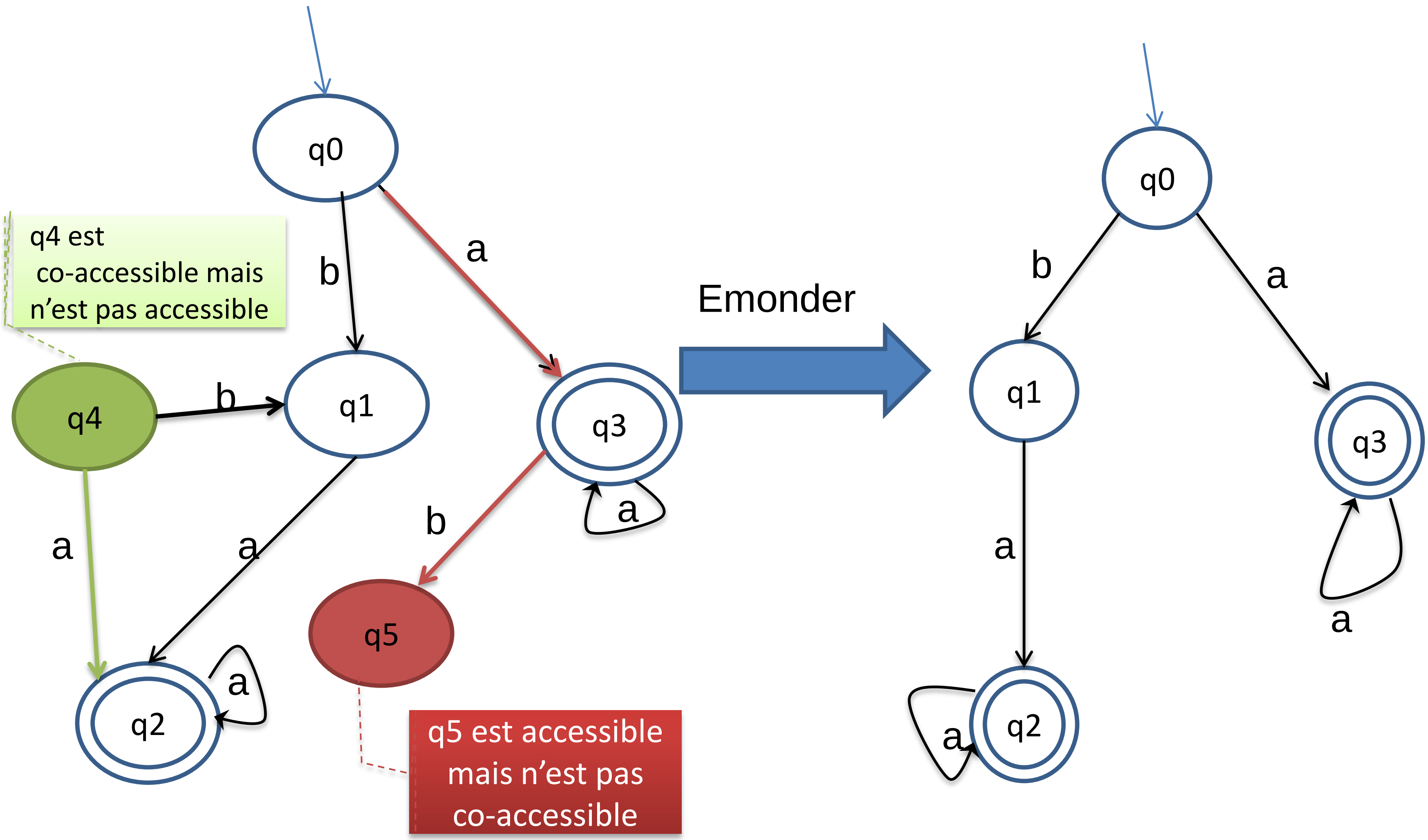
Un état q est **accessible** s'il existe un chemin de l'état initial de l'automate vers q.

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 coaccessibles



Automates d'Etats Finis



Les Variantes des Automates d'Etats Finis

1. Automates Déterministes

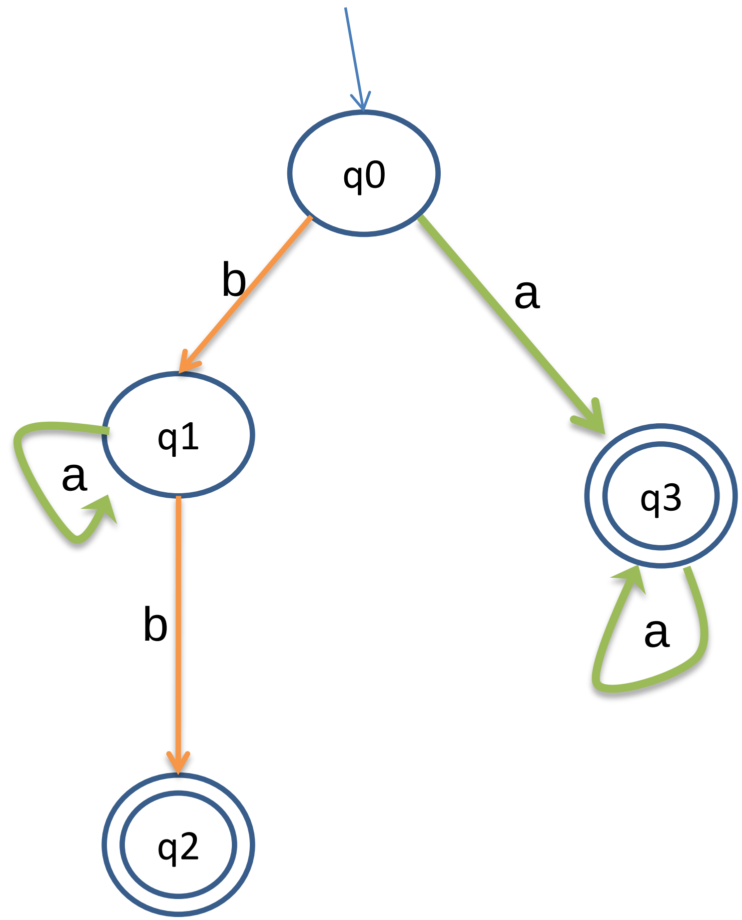
Un automate d'états fini est **déterministe** si et seulement si :

- à un état

- à un symbole d'entrée,

la fonction δ **associe <u>au plus</u> une transition**.

Autrement dit, la fonction δ est définie de QxX dans Q.

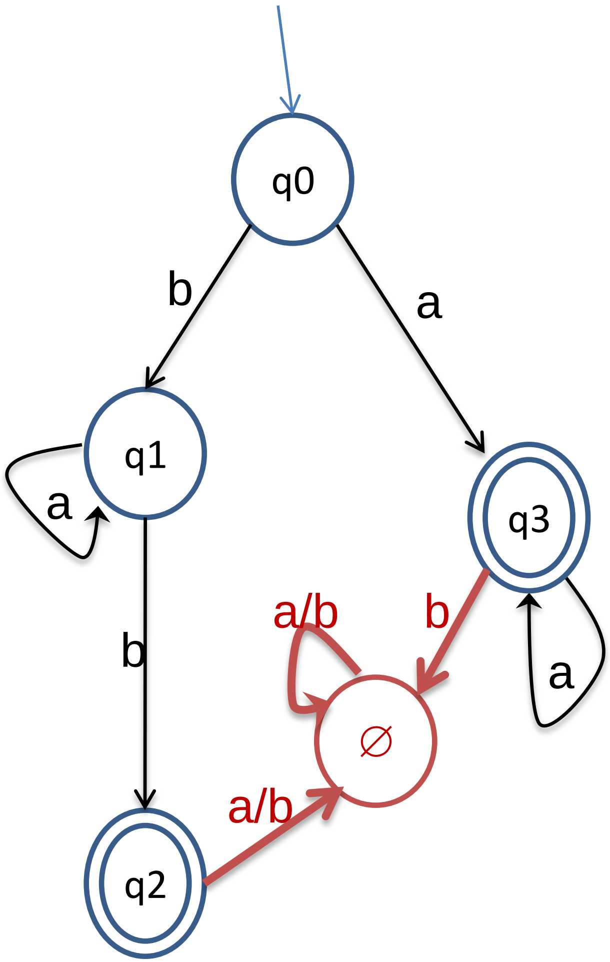


Les Variantes des Automates d'états Finis

Remarque

Un automate déterministe est **dit complet ou complètement spécifié** si à toute paire (q, a) ∈ QxX, **la fonction δ associe <u>exactement un</u> état**. Autrement dit la **fonction de transition δ** est une **application fonctionnelle** de QxX dans Q.

L'automate précédent est **déterministe mais non complet**. Pour le rendre complet, il suffit de rajouter un état, appelé « **état puit** », généralement noté ∅, et de rajouter toutes les transitions manquantes vers cet état



Les Variantes des Automates d'Etats Finis

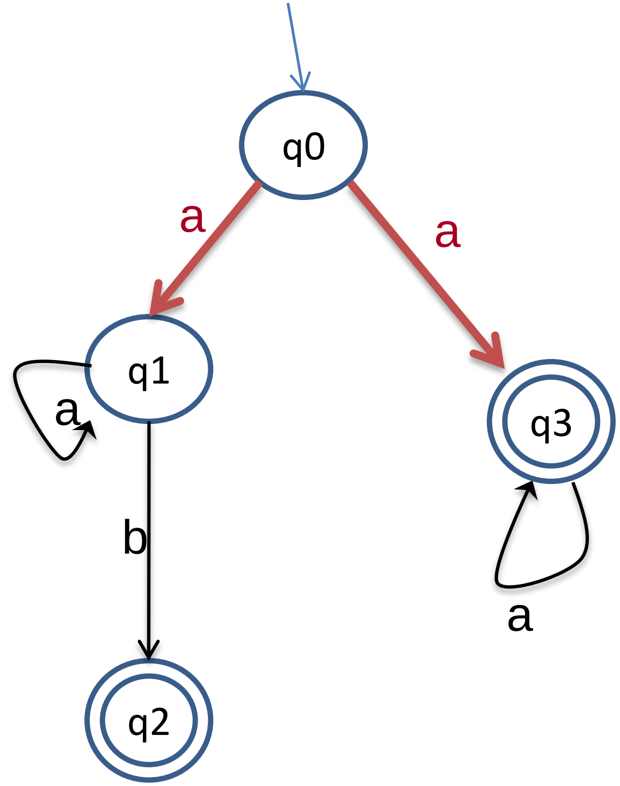
2. Automates Non Déterministes

Les automates d'états finis **non déterministes** sont des automates où l'on **permet plusieurs transitions** correspondant à **la même lettre dans chaque état**.

Donc δ est une fonction de transition définie de **QxX dans <u>l'ensemble des parties de Q</u>** (δ : QxX $\rightarrow \mathscr{P}$(Q))

Remarque

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



De q0, on a deux transitions par a, l'une vers q1 et l'autre vers q3.

Les Variantes des Automates d'Etats Finis

3. Automates Généralisés

Dans un automate **généralisé**, les transitions directes **peuvent être causées par des mots**.

Les transitions directes **causées par le mot vide** sont appelées **transitions spontanées ou ε-transition**.

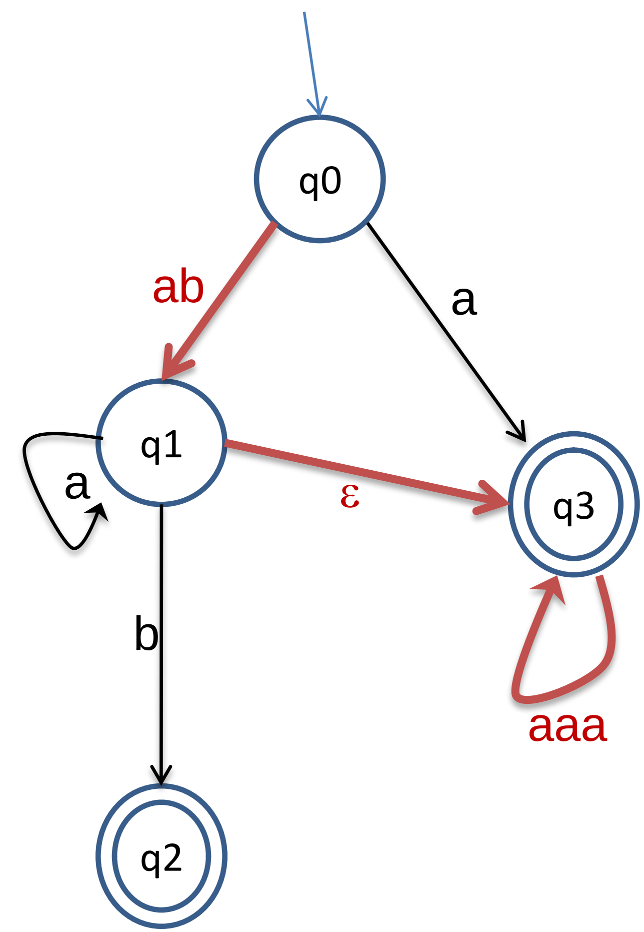
Une ε -transition correspond à la situation où l'automate **peut changer d'état sans lire de symbole**.

La fonction δ est alors définie de :

Qx X* dans 𝒫 (Q).

Remarque

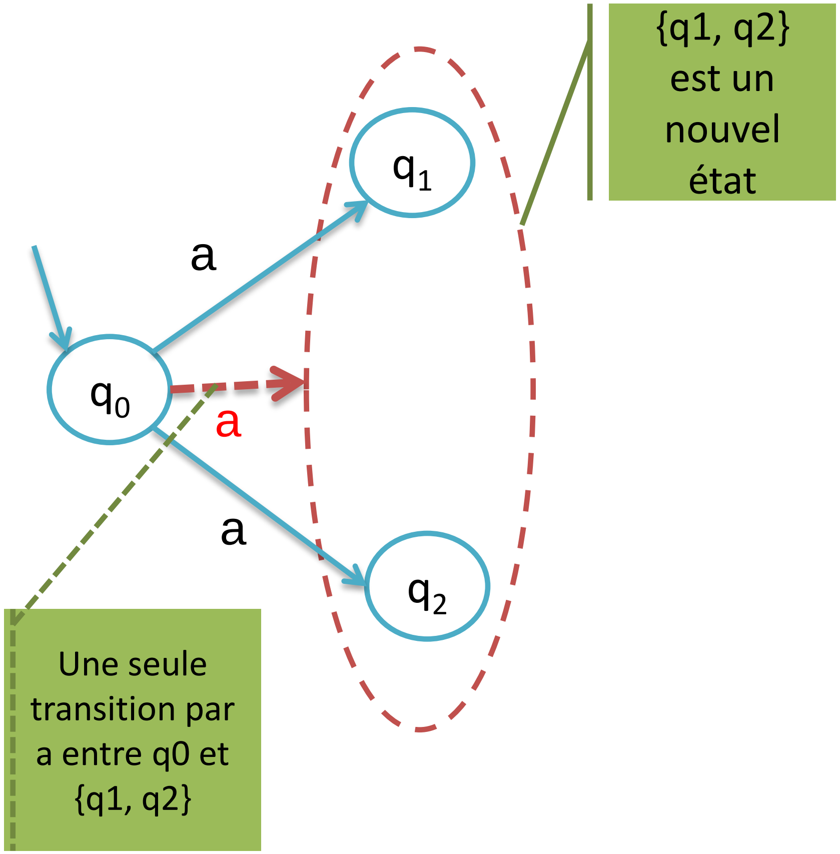
Par opposition aux automates généralisés, les automates déterministes et non déterministes sont dits simples.



Passage AEF non déterministe vers AEF déterministe

Principe :

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



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 déterministe équivalent $A_d=(X, Q_d, q_{0d}, \delta_d, F_d)$ avec

- $Q_d=2^{(Q)}$ (ou $\mathcal{P}(Q)$) au maximum car généralement ces états ne sont pas tous accessibles à partir de l'état initial et ainsi $Q_d \in \mathcal{P}(Q)$

- $q_{0d}=\{q_0\}$

- $\delta_d(q_d, a)= \bigcup_{q \in q_d} \delta(q, a)$

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

Passage AEF non déterministe vers AEF déterministe

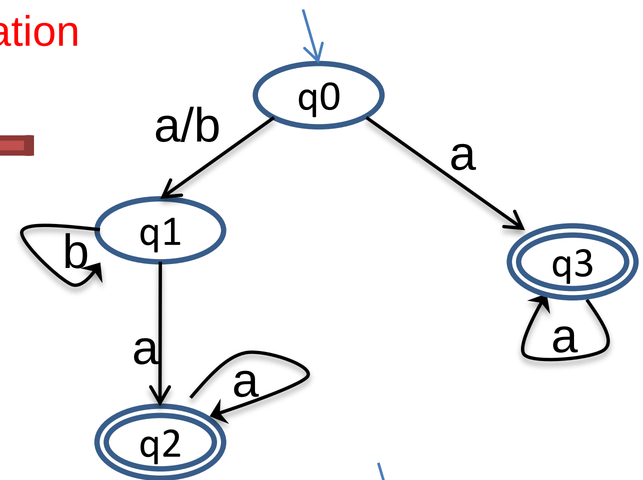
Pour déterminiser un automate, il est pratique d'établir la table des transitions. La construction de l'automate déterministe se fait sur des ensembles d'états obtenus à partir de l'état initial comme suit :

- A partir **de l'ensemble contenant seulement l'état initial**, on regroupe les transitions étiquetées par la même lettre issue de cet état. Il suffit de reprendre la ligne de l'état initial.
- Pour chaque ensemble d'états (nouvellement) obtenu, on regroupe les transitions étiquetées par la même lettre issue de cet ensemble. Donc, pour chaque lettre, on fait l'union des cases associées aux différents états de l'ensemble.
- On arrête la procédure dés que tous les ensembles d'états obtenus ont été traités.
- Un **ensemble d'états est final** s'il **contient un état final.**

Passage AEF non déterministe vers AEF déterministe

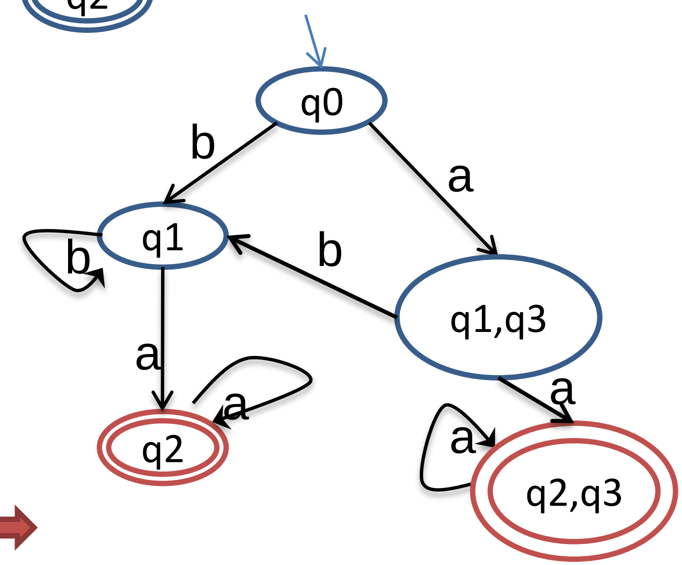
Etat\Lettre	a	b
q0	{q1, q3}	{q1}
q1	{q2}	{q1}
q2	{q2}	_
q3	{q3}	_

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

Passage AEF généralisé vers AEF simple

Proposition

Pour tout automate fini généralisé $Ag=(X^*, Qg, q_0, \delta g, Fg)$ il existe un automate simple équivalent $A_s=(X, Q_s, q_{0s}, \delta_s, F_s)$.

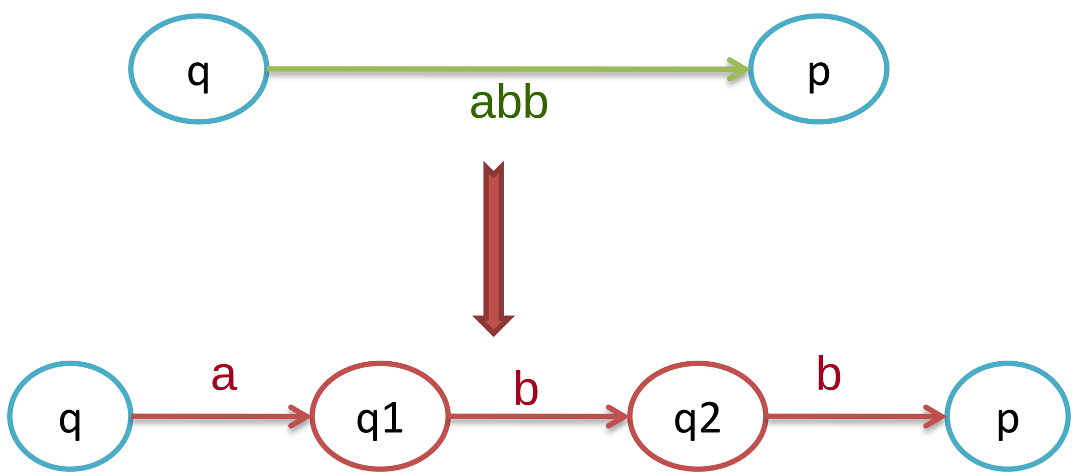
Principe de la construction

- 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 A.

Passage AEF généralisé vers AEF simple

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

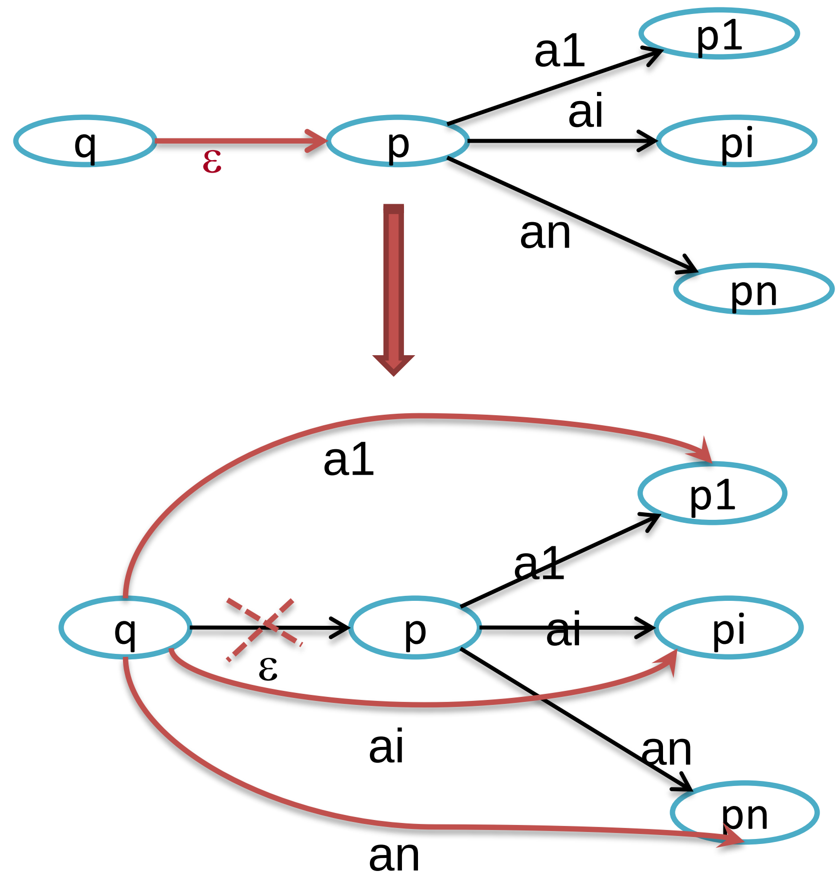
on ajoute des états intermédiaires.



Passage AEF généralisé vers AEF simple

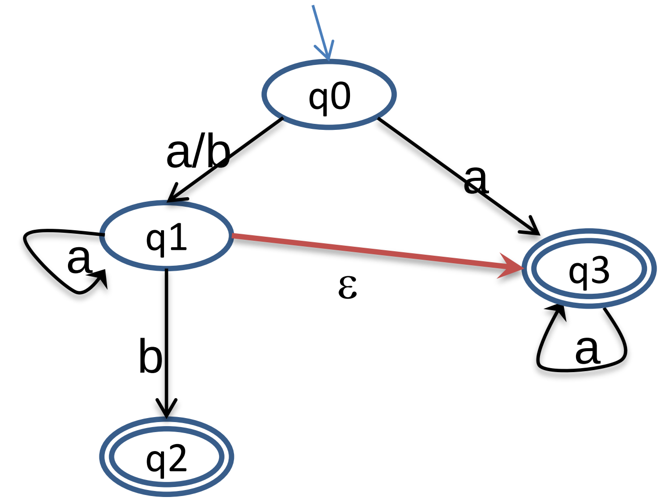
Ensuite, on élimine les transitions spontanées (comme illustré dans le schéma) :

- Ainsi, 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.**
- 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 l'état q_3.

Etat\Lettre	a	b	ε
q0	{q1, q3}	{q1}	_
~~q1~~	~~{q1}~~	~~{q2}~~	~~{q3}~~
q2	_	_	_
q3	{q3}	_	_
q1	{q1, q3}	{q2}	

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

Grammaire et Automate

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, δ, F) tel que L(G)=L(A):

- X= T ; Q =N$\cup$ {q_F} et $q_F \notin$N, q_0=S, F={q_F }
- Si A$\rightarrow$ wB $\in$ P alors B $\in$ δ(A, w)
- Si A $\rightarrow$ w $\in$ P alors q_F $\in$ δ(A, w)

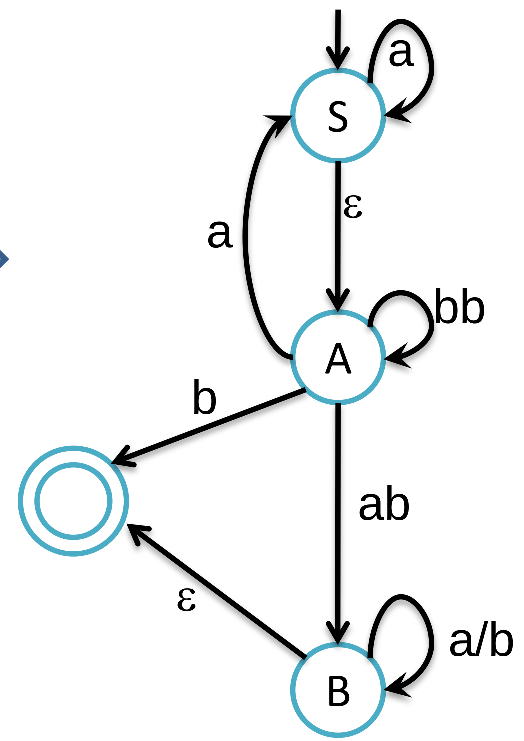
Grammaire et Automate

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

$S \rightarrow aS / A$

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

$B \rightarrow aB / bB / \varepsilon$



Grammaire et Automate

Proposition

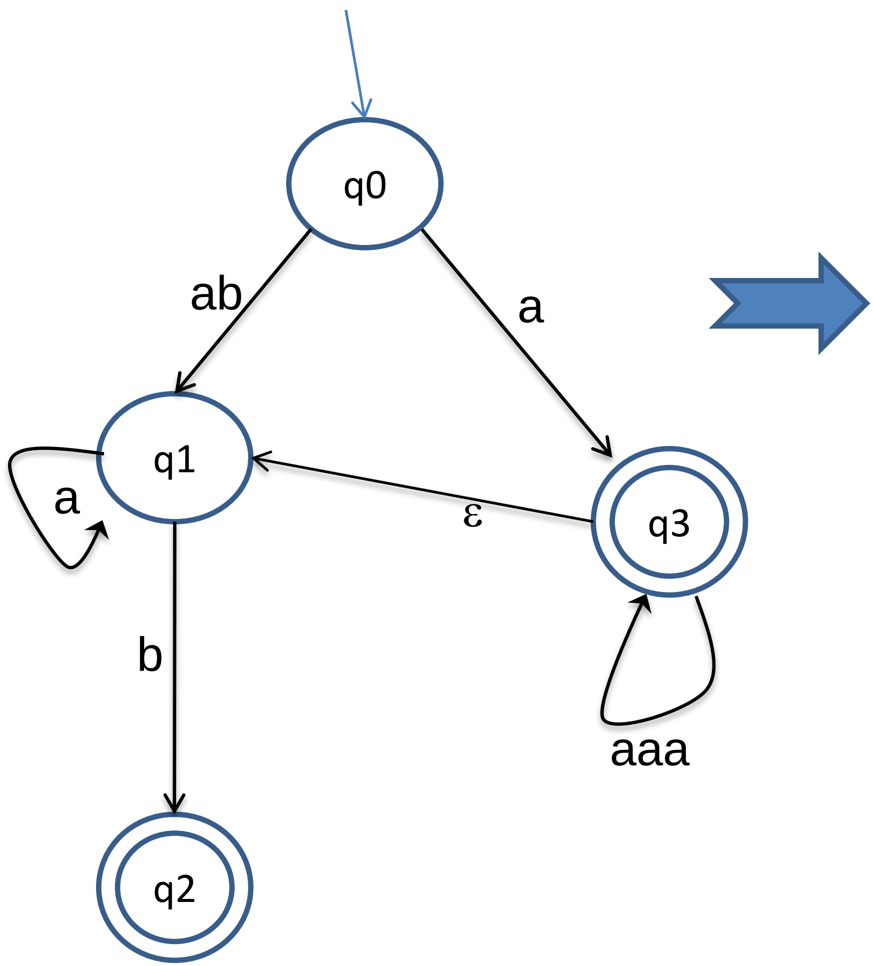
Pour tout automate fini généralisé il existe une grammaire régulière droite équivalente.

Démonstration

Soit un automate généralisé $A=(X, Q, q_0, \delta, F)$, il s'agit de déterminer une grammaire régulière droite $G=(T, N, S, P)$ telle que $L(A)=L(G)$:

- $T= X$; $N=Q$, $S= q_0$
- Si $q\in \delta(p, w)$ alors $(p \to wq) \in P$
- Si $q\in\delta(p, w)$ et $q\in F$ alors $(p \to w) \in P$
- Si $q_0 \in F$ alors $(q_0 \to \varepsilon) \in P$

Grammaire et Automate



Grammaire régulière

G=({a, b}, {q0, q1, q2, q3}, q0, P) tel que P est défini :

q0 $\rightarrow$ ab q1 / a q3 / a

q1 $\rightarrow$ a q1 / b q2 / b

q2 $\rightarrow$ ε

q3 $\rightarrow$ aaa q3 / q1 / aaa

Expressions Régulières

Les expressions régulières sur un alphabet X sont définies comme suit :

Cas de base

- $\varnothing$ est une expression régulière et **décrit le langage vide**
- **ε** est une expression régulière et **décrit le langage {ε}**
- **∀a∈X, a** est une expression régulière et **décrit le langage {a}**

Induction

- Si r et s sont deux expressions régulières sur décrivant respectivement les langages R et S alors :
- **r+s** est une expression régulière et **décrit le langage R∪S**
- **r.s** est une expression régulière et **décrit le langage R.S**
- **r*** est une expression régulière et **décrit le langage R***
- **(r)** est une expression régulière et **décrit le langage R.**

Expressions Régulières

Remarque

Pour simplifier les expressions, nous supposons que l'étoile '*' est plus prioritaire que la concaténation ' .'qui est plus prioritaire que l'addition '+'.

Exemples

- (a+b)* dénote le lange ({a}$\cup${b})* qui correspond à tous les mots sur {a, b}
- (a+b)*aab représente tous les mots de {a, b}* se terminant par aab
- (a+b)*aba(a+b)* représente tous les mots de {a, b}* ayant aba comme sous mot.

On note Rat(X*) l'ensemble des expressions régulières sur l'alphabet X.

Expressions Régulières

Définition

Deux expressions régulières sont équivalentes si elles dénotent le même langage.

Propriétés sur les expressions régulières

- Commutativité : $p + q = q + r$
- Associativité : $p+ (q + r) = (p + q) + r$ $\quad p(qr) = (pq)r$
- Distribution : $(p + q)r = pr + qr$ $\quad p(q+r) = pq+pr$
- Elément absorbant : $p. \varnothing = \varnothing.p= \varnothing$
- Elément neutre : $p.\varepsilon = \varepsilon.p = p$ $\quad p+ \varnothing = \varnothing +p = p$
- $p. p^* = p^*.p$ $\quad p^* = (p+ \varepsilon)^*$ $\quad p^*.p^* = p^*p^* = \varepsilon + p.p^*$

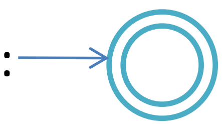
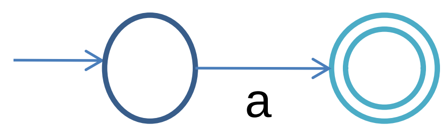
  $\varnothing^* = \varepsilon$ $\quad (p^*)^* = p^*$
- $(p^*+q^*)^* = (p+q)^* = (p^*.q^*)^*$

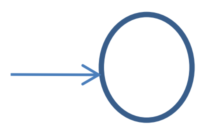
Expression Régulière et Automate d'Etats Fini

Proposition

A toute expression régulière E, il existe un automate d'états fini A(E) tel que reconnaissant le langage dénoté par E.

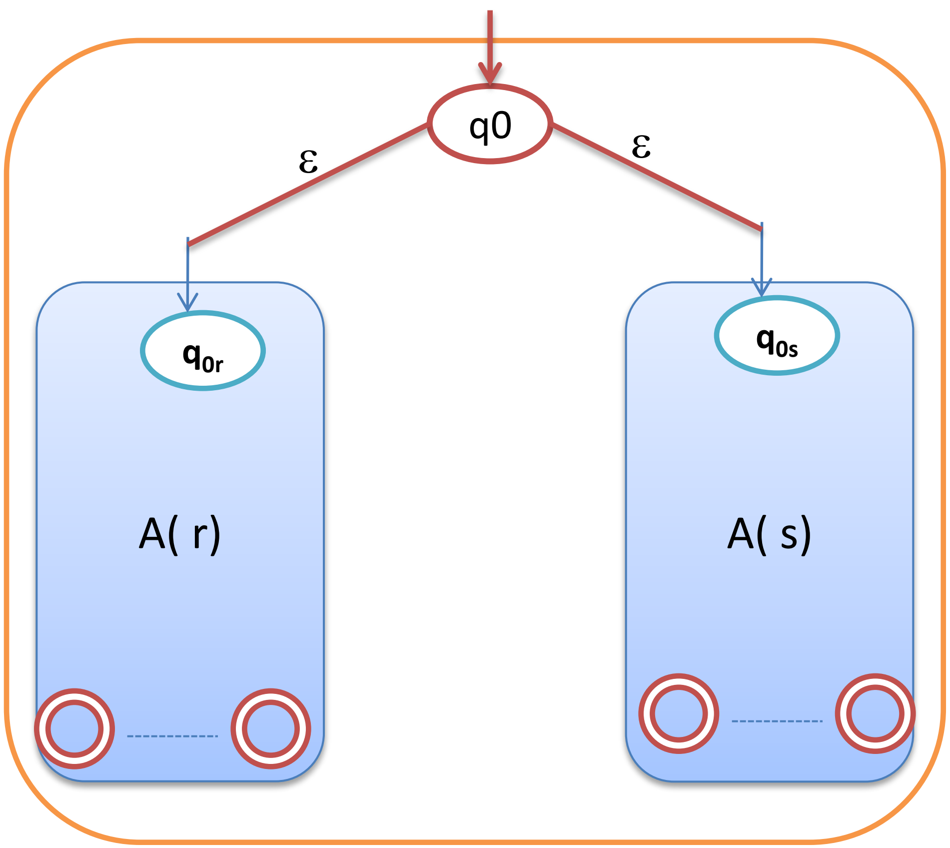
Démonstration

- Pour ε, on lui associe l'automate : 
- Pour a, on lui associe l'automate : 

- ∅, on associe l'automate : 

Expression Régulière et Automate d'Etats Fini

- Pour r+s, on associe l'automate A(r+s)



A(r)=(Xr, Qr, q0r, δr, Fr)

A(s)= (Xs,Qs,q0s, δs, Fs)

A(r+s)=(X, Q, q0, δ, F) tq

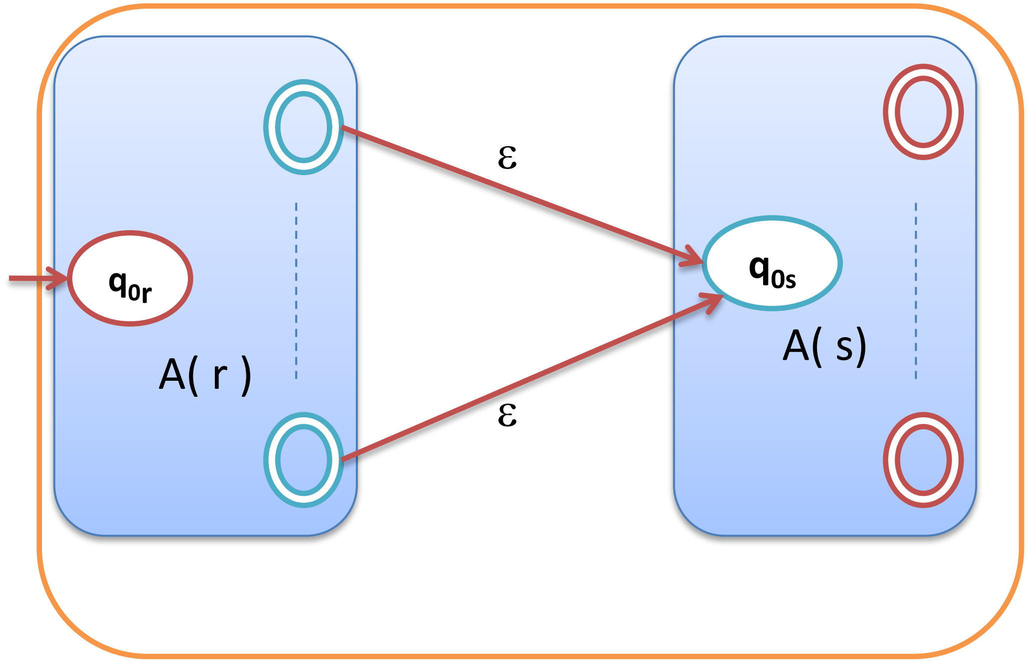
- X = Xr ∪ Xs
- Q = Qr ∪ Qs ∪ {q0}
- q0 / q0 ∉ Qr ∪ Qs
- δ = δr+ δs +
  δ(q0, ε)={q0r, q0s}
- F =Fr ∪ Fs

Expression Régulière et Automate d'Etats Fini

- Pour r.s, on associe l'automate A(r.s)

A(r)=(Xr, Qr, q0r, δr, Fr)

A(s)= (Xs,Qs,q0s, δs, Fs)



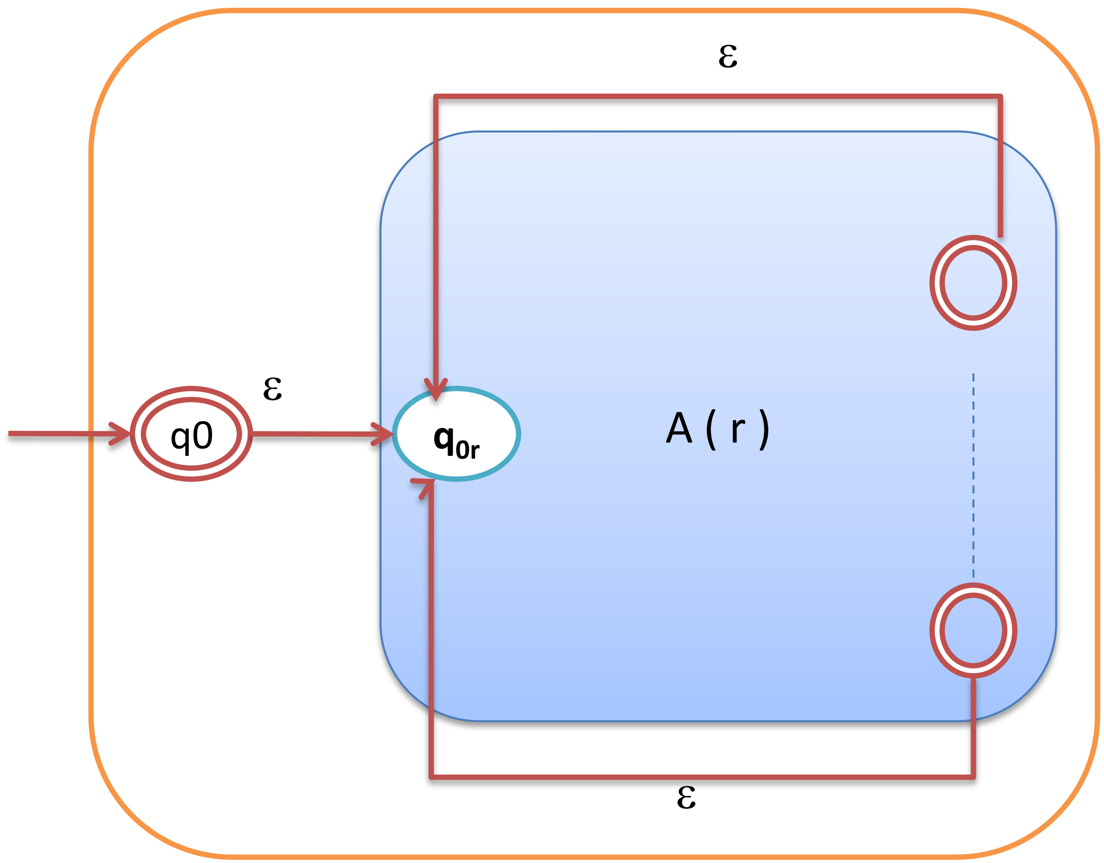
A(r.s)=(X, Q, q0, δ, F) tq

- X = Xr ∪ Xs
- Q = Qr ∪ Qs
- q0=q0r
- δ = δr+ δs +

  ∀q∈Fr, δ(q, ε)=q0s
- F =Fs

Expression Régulière et Automate d'Etats Fini

- Pour r*, on associe l'automate A(r*) :



A(r)=(Xr, Qr, q0r, δr, Fr)

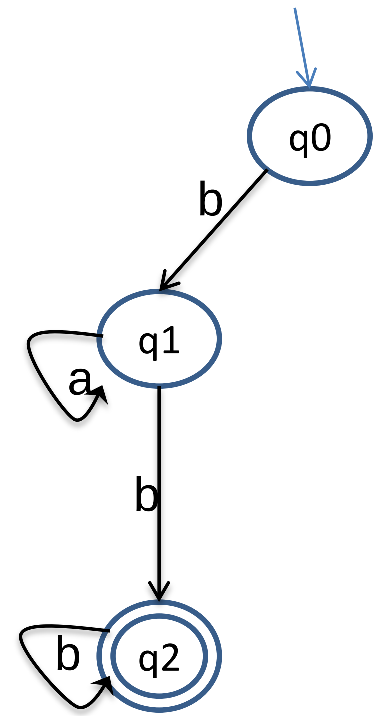
A(r*)=(X, Q, q0, δ, F) tq

- X = Xr
- Q = Qr ∪ {q0}
- q0 / q0 ∉ Qr
- δ = δr
  + δ(q0, ε)=q0r
  + ∀q∈Fr, q0r∈δ(q, ε)
- F =Fr ∪ {q0}

Expression Régulière et Automate d'Etats Fini

Proposition

A tout automate d'états fini A lui correspond une expression régulière qui le dénote.



Le langage reconnu par cet automate est dénoté par l'expression régulière ba*bb*

Remarque

Plusieurs méthodes permettant de trouver l'expression dénotant le langage reconnu par un automate d'états fini ont été proposées dans la littérature.