Chapitre III : Les Langages Reguliers

III.1 Grammaire Regulière

Une grammaire $G = \langle T, N, S, P \rangle$ est regulière droite si toutes ses productions sont de la forme :

$A \to wB$ ou $A \to w$ avec $A, B \in N$ et $w \in T^*$

———————————————— gauche —

———————————————— :

$A \to Bw$ ————————————————

proposition :

Le langage généré par une grammaire regulière gauche ou droite est un langage regulier (Type 3).

Rappel :

Type 3 :

- RD : $A \to wB$ ou $A \to w$ $\quad w \in T^*$
- RG : $A \to Bw$ ou $A \to w$ $\quad w \in T^*$

Type 2 :

$A \to w \quad w \in (T \cup N)^*$

Type 1 :

- contexte lié : $\alpha A \beta \to \alpha w \beta$
- monotone : $\alpha \to \beta$ avec $|\alpha| \leqslant |\beta|$

III.2 Les Automates

Un automate (ou système de reconnaissance) est une machine qui permet de lire un mot et reconnaître si il $\in$ ou non à un langage donné.

<u>Remarque</u> : A tout type de langage on associe un automate.

Les langages de type 3 : on leur associe les automates à états finis.

Les langages de type 2 : on leur associe les automates à pile.

Les langages de type 0 : on leur associe les machines de TURING.

Definition Formelle d'un automate :

un automate à etats finis est un Cinq-uplet $A(X, Q, q_0, \delta, F)$ où :

- Q : est un ensemble fini d'états.
- X : est un alphabet d'entrée (fini et $\neg$ vide).
- q_0 : etat initial ($q_0 \in Q$).
- F : ensemble des états finaux (terminaux) $F \subseteq Q$
- δ : est une relation de transition qui associe à chaque couple (etat x symbole) un ensemble d'états d'arrivée.

$$\delta \subseteq \underbrace{Q \times X \times Q}_{\text{produit Cartesien}}$$

on peut aussi representer δ comme suit :

$$\delta : Q \times X \longrightarrow 2^Q \quad \text{avec} \quad \delta(Q, X) \subseteq 2^Q$$

Mapping et non une application

Exemple 1 : Automate à états finis

$X = \{a, b\}$

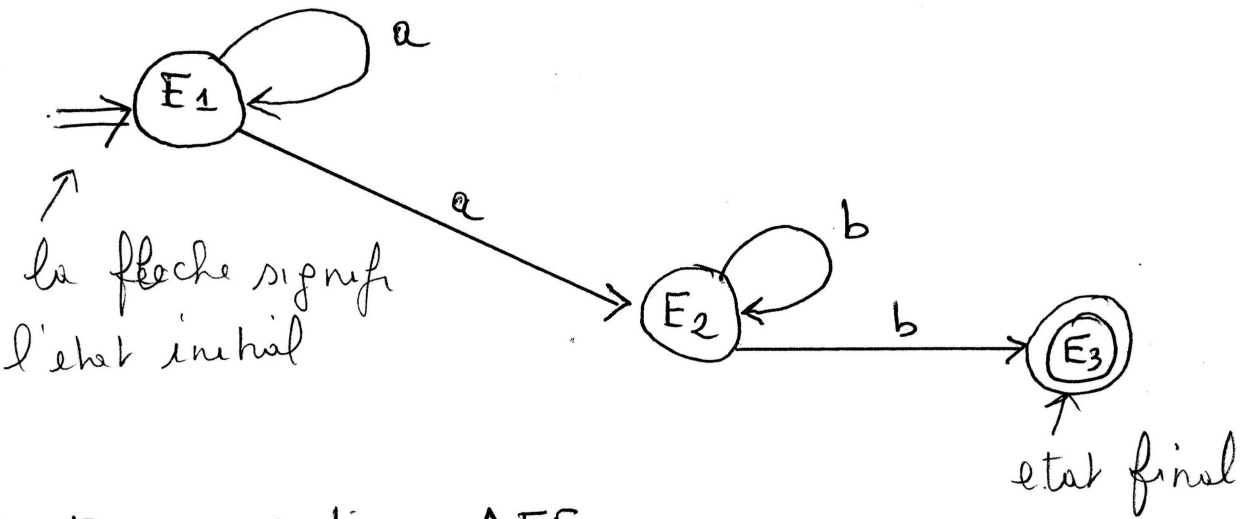
$Q = \{E_1, E_2, E_3\}$

$q_0 = \{E_1\}$

$F = \{E_3\}$

$\delta = \{\{E_1, a, E_1\}, \{E_1, a, E_2\}, \{E_2, b, E_2\}, \{E_2, b, E_3\}\}$

on represente graphiquement cet automate comme suit :



Fonctionnement d'un AEF

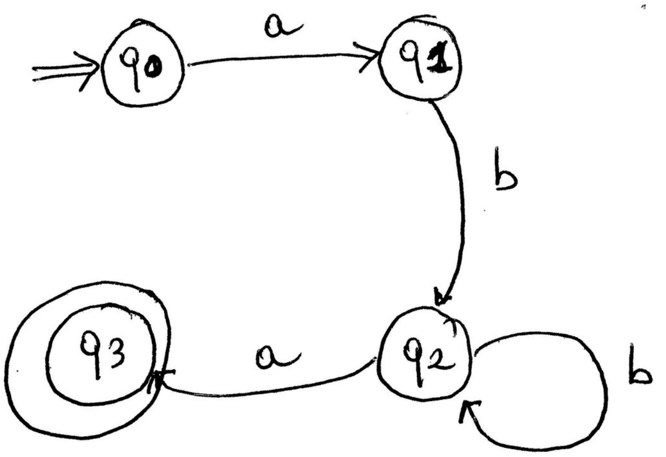
Les AEFs sont utilisés pour la reconnaissance des mots d'un langage. Initialement, l'AEF se trouve dans l'etat initial. A la lecture des symboles du mot à reconnaître l'AEF réalise des deplacements entre les états. Si l'AEF atteind un état final après la lecture du dernier symbole du mot alors le mot est accepté par l'automate.

Convention de représentation graphique :

Un AEF peut être représenté par un graphe orienté dont les sommets correspondent aux états et les arcs aux transitions.
L'arc ayant pour origine le sommet $q_i^{\in Q}$, pour extrémité le sommet $q_j^{\in Q}$ et pour étiquette le symbole $a \in X$, représente la transition $\{q_i, a, q_j\}$ (ou $\delta(q_i, a) = q_k$).
L'état initial est représenté avec une flèche
" final " " par deux cercles concentriques.

Exemple 2 : Langage reconnu par un automate

Soit $A < X, Q, q_0, \delta, F >$ où
- $X = \{a, b\}$
- $Q = \{q_0, q_1, q_2, q_3\}$
- $F = \{q_3\}$
- $\delta(q_0, a) = q_1$
  $\delta(q_1, b) = q_2$
  $\delta(q_2, b) = q_2$
  $\delta(q_2, a) = q_3$



langage reconnu :

aba
abba
abbba
⋮

$$L = \{a\, b^n a \mid n \geq 1\}$$

Representation Matricielle

Les AEFs peuvent être aussi representés par une matrice dont les indices des lignes correspondent aux états et les indices de colonne correspondent aux elements de X.
Un élément de la matrice de ligne q et de colonne x correspond au resultat de la transition $\delta(q, x)$

Representation matricielle de l'exemple 2 :

	a	b
q_0	q_1	—
q_1	—	q_2
q_2	q_3	q_2
q_3	—	—

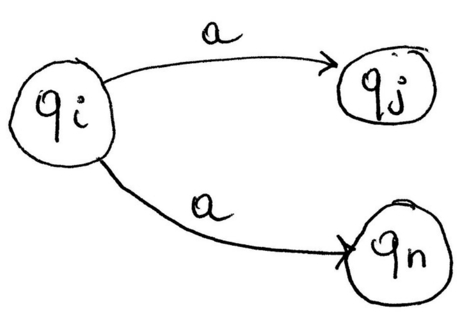
III.3 Les Differents types d'automates

III.3.1 Les automates deterministes

un automate fini est deterministe si un etat courant et un symbole ne permet qu'une seule transition possible. Dans ce cas, la relation δ est une application c'à d

$$\delta : Q \times X \rightarrow Q$$

Exemple :



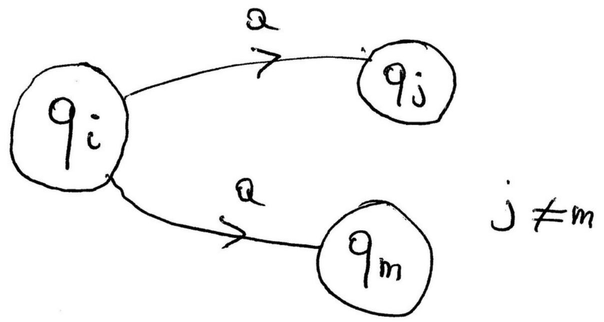
Cet automate n'est pas deterministe

Definition
Un automate deterministe est dit complet si a toute paire $(q, x) \in Q - F \times X$, la fonction δ associe exactement un état

III.3.2 Les automates non deterministes

un automate à etats finis non deterministe est un automate où on permet plusieurs transitions correspondant à la même etiquette. Dans ce cas, δ n'est plus une application mais une relation mathematique :

$$\delta : Q \times X \longrightarrow 2^{Q}$$



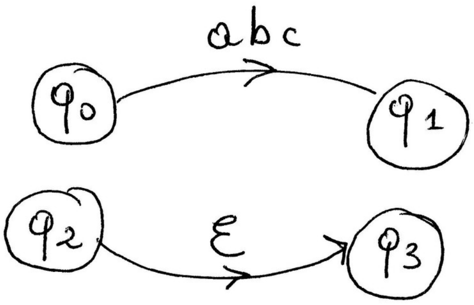
III.3.3 Les automates generalisés

Dans le cas d'un automate deterministe ~~ou~~ non deterministe, toute transition est definie par un seul elt de l'alphabet. Dans un automate generalisé, les transitions directes peuvent être declenchées par des mots eventuellement le mot vide.

Les transitions declenchées par le mot vide sont appelées "transition spontanée". Une transition spontanée (ε-transition) correspond à la situation où l'automate peut changer d'état sans lire de symbole en entrée.

la fonction δ est alors definie :

$$Q \times X^{*} \longrightarrow 2^{Q}$$

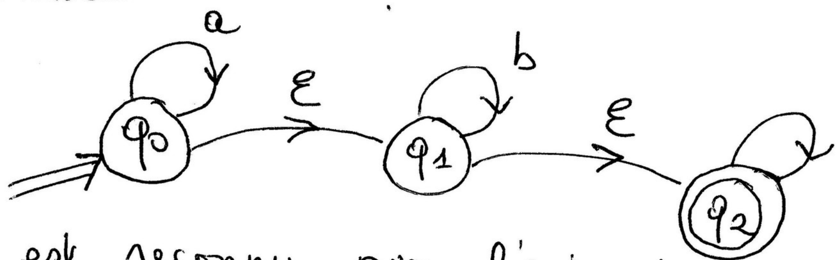


Notation

Par opposition aux automates generalisés, les automates qui passent d'un état à un autre par un seul caractère sont appelés "Automates Simples"

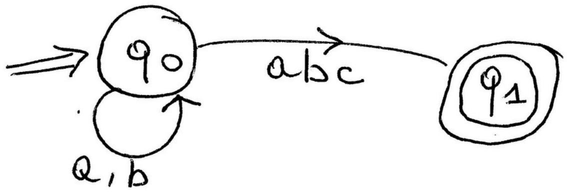
Exemple 1 : de transition spontanée

Soit l'automate suivant :



Le mot "aaabbc" est reconnu par l'automate

Exemple 2 : Automate generalisé



$X = \{a, b, c\}$

Le langage reconnu par cet automate est :

$(a \cup b)^*.abc$ ou $X^* abc$

III.4 Langage reconnu par un automate

Soit $A = \langle Q, X, q_0, \delta, F \rangle$ un AEF non deterministe. La configuration de l'automate à un moment donné est composée de l'etat courant et du mot qu'il reste à lire (à réconnaitre). On note cet etat par $\langle q, w' \rangle$ où q est l'etat courant et w' le mot qui reste à lire.

La configuration initiale est $\langle q_0, w \rangle$ où w est le mot à reconnaitre. Si w est accepté par l'automate alors la configuration finale est $\langle q_f, \varepsilon \rangle$ tq $q_f \in F$.

Conclusion : Pour qu'un mot (ω) soit accepté par un automate, il faut trouver un ensemble de derivations qui fait passer l'automate de l'etat $\langle q_0, w \rangle$ à l'etat $\langle q_f, \varepsilon \rangle$ où $q_f \in F$. Dans ce cas, on note $\langle q_0, w \rangle \vdash^* \langle q_f, \varepsilon \rangle$

Rmq : L'ensemble de derivations peut être vide

__Definition__ : Soit $A = \langle X, Q, q_0, \delta, F \rangle$ un AEF deterministe ou non. Le langage reconnu par l'automate A est noté $L(A)$ tel que :

$$L(A) = \{ w \in X^* \,/\, \exists\, q_f \in F \text{ et } \langle q_0, w \rangle \vdash^* \langle q_f, \varepsilon \rangle \}$$

__Proposition__ : Un langage L sur un alphabet X est regulier (reconnaissable) si il existe au moins un automate fini A ayant X comme alphabet tel que $L = L(A)$.

on note $Rec(X^*)$, la famille des langages reconnaissables sur l'alphabet X.

__Automate equivalent :__

Deux automates A_1 et A_2 sont equivalents si et seulement si ils acceptent le même langage :

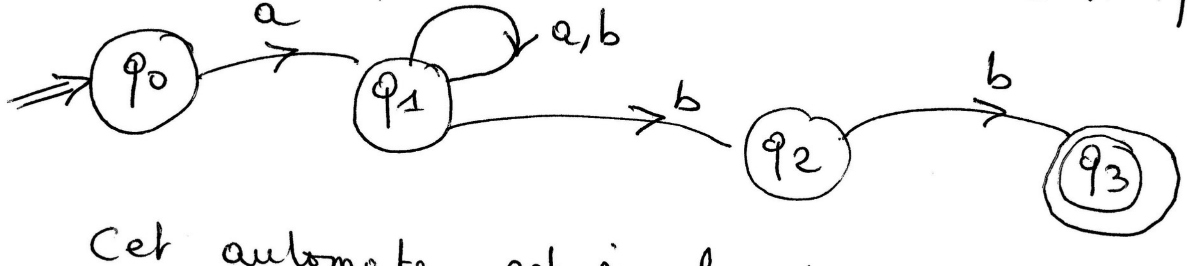
$$A_1 \equiv A_2 \iff L(A_1) = L(A_2)$$

__Exemple__ : Langage regulier

. Soit $L = \{ w \,/\, w \in \{a,b\}^*$ et w commence par a et se termine par $bb \}$

montrer que ce langage est regulier ?

Il suffit pour cela de trouver un automate A tq $L = L(A)$



Cet automate est simple et non deterministe donc L est regulier.

III.5 Transformation des automates

Il est possible de transformer un Automate Fini Généralisé (AFG) vers un Automate Fini Deterministe (AFD).

Le passage d'un AFG vers un AFD se fait en deux etapes :

a) Passage de l'AFG vers un automate simple fini non deterministe (AFND).

b) Passage d'un AFND vers un AFD

$$AFG \xrightarrow{\text{etape a)}} AFND \xrightarrow{\text{b) etape}} AFD$$

III.5.1 Passage d'un AFND vers un AFD (etape b)

<u>Proposition</u> : Pour tout automate fini non deterministe $A = \langle X, Q, q_0, \delta, F)$ il existe un automate fini deterministe equivalent $A' = \langle X', Q', q_0', \delta', F' \rangle$ tel que $L(A) = L(A')$.

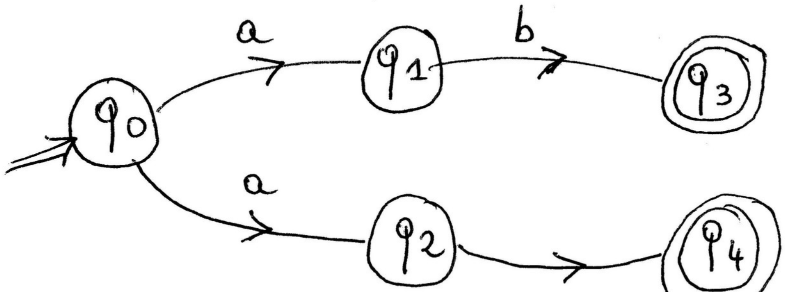
A' est construit comme suit :

$X' = X$; $q_0' = q_0$; $Q' \subseteq 2^Q$

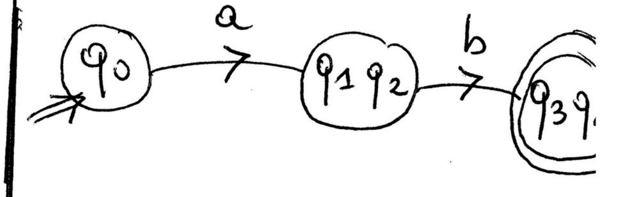
$\delta'(q, a) = \bigcup \delta(p, a)$ tel que $p \in q$, $q \in Q'$

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

Exemple 1 : manière intuitive



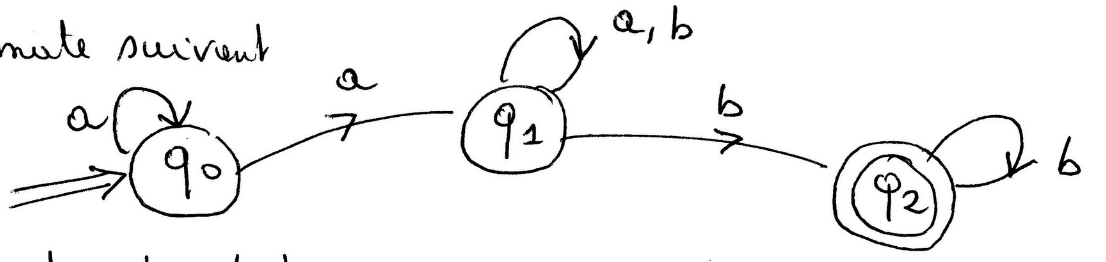
AFND



AFD equivalent

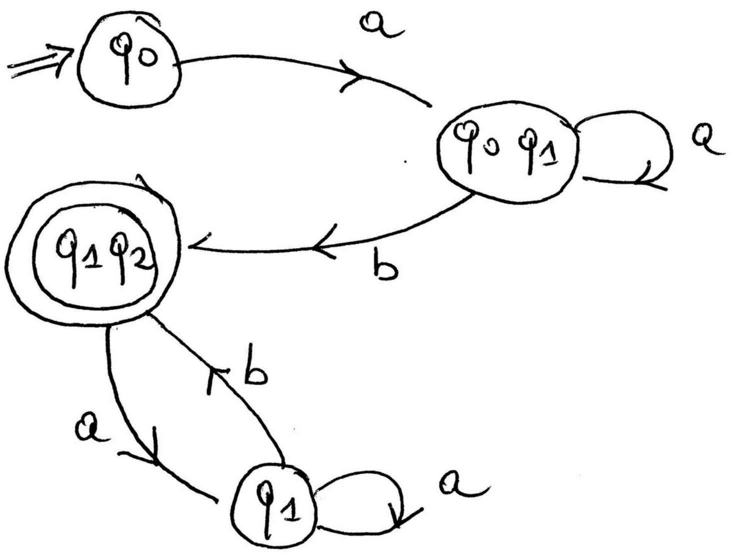
Exemple 2 :

Soit l'automate suivent



Rendre cet automate determeniste ?

	a	b
q_0	q_0, q_1	—
q_1	q_1	q_1, q_2
q_2	—	q_2
→ q_0	$q_0 q_1$	—
$q_0 q_1$	$q_0 q_1$	$q_1 q_2$
◎ $q_1 q_2$	q_1	$q_1 q_2$
q_1	q_1	$q_1 q_2$



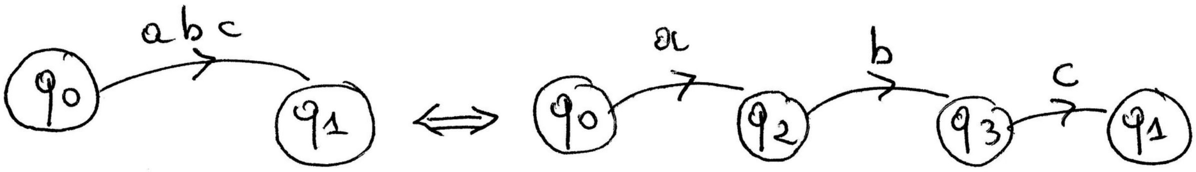
III.5.2 Passage d'un AFG vers un AFND

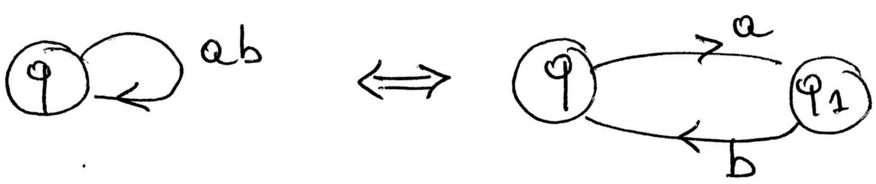
proposition : pour tout AFG il existe un automate simple (deterministe ou pas)

cet automate simple est obtenu par les regles suiventes :

Regle 1 : Eliminer les transitions declenchées par des mots de X^* de longueur ≥ 2

Exple





Regle 2 : Eliminer les ε-transitions

III.6 Grammaire et Automates

Théorème : Pour toute grammaire régulière droite $G = T, N, S, P$ il existe un automate à états finis généralisé équivalent.

Construction d'un AFG à partir d'une grammaire régulière droite : il s'agit de construire un automate $A < X, Q, q_0, \delta, F >$ tel que $L(G) = L(A)$. Les cinq éléments de l'automate sont définis comme suit :

$X = T$; $Q = N \cup \{q_f\}$; $q_0 = S$; $F = \{q_f\}$;

δ : Si $(A \to wB) \in P$ alors $B \in \delta(A, w)$

Si $(A \to w) \in P$ alors $q_f \in \delta(A, w)$

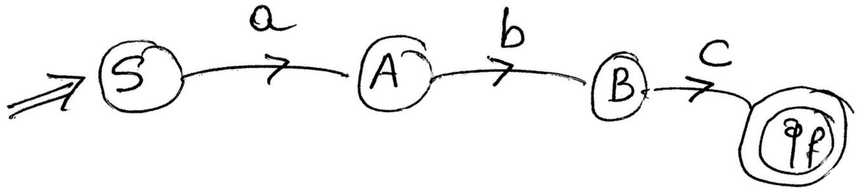
Exemple 1 :

Soit G tel que

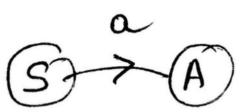
$S \to aA$ $\Longrightarrow$

$A \to bB$

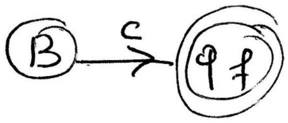
$B \to c$



$(S \to aA) \Rightarrow$



$B \to c \Rightarrow$



Exemple 2 construire un automate équivalent à G :

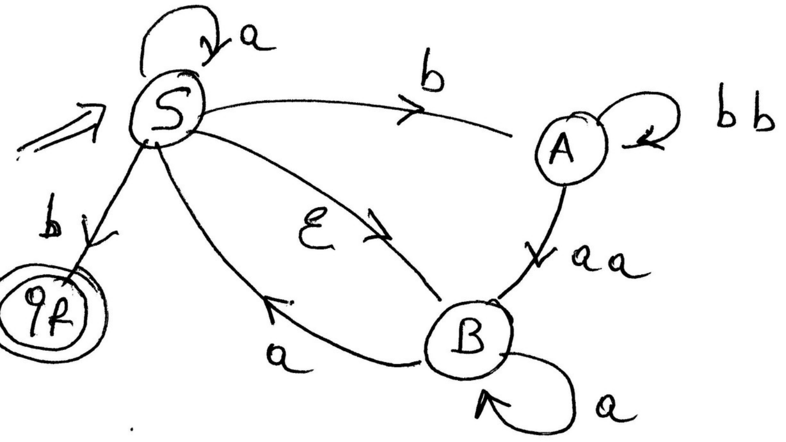
$S \to aS \,/\, bA \,/\, B \,/\, b$

$A \to bbA \,/\, aaB$

$B \to aB \,/\, aS$

$A \langle X, Q, q_0, \delta, F \rangle$

$X = \{a, b\}$; $Q = \{S, A, B, q_f\}$; $q_0 = \{S\}$; $F = \{q_f\}$



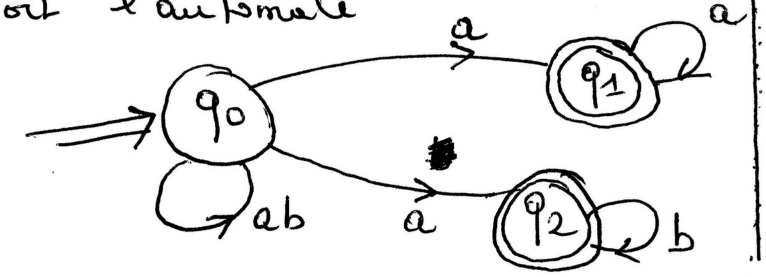
Théorème : pour tout automate fini généralisé, il existe une grammaire régulière droite équivalente c'àd générant le langage reconnu par l'automate

Construction d'1 grammaire régulière droite à partir d'1 AFG

Soit un automate généralisé $A \langle X, Q, q_0, \delta, F \rangle$, la grammaire régulière droite $G \langle T, N, S, P \rangle$ équivalente à l'automate A est définie comme suit :

Exemple 2 :

Soit l'automate



grammaire équivalente :

$T = \{a, b\}$

$N = \{q_0, q_1, q_2\}$

$S = \{q_0\}$

$P :$ $q_0 \rightarrow ab\,q_0 / a\,q_1 / a\,q_2$

$q_1 \rightarrow a\,q_1 / \varepsilon$

$q_2 \rightarrow b\,q_2 / \varepsilon$