

Chapitre 4:

Automate fini déterministe et non déterministe

Automates à Etats finis

- ▶ Les langages sont reconnus par des machines formelles appelées: **automates** ou systèmes reconnaisseurs qui étant donnée un mot sont capable de dire si c'est mot appartient ou pas à un langage.
- ▶ Un automate à états finis (AF) est un modèle d'un système et de son évolution, c'est-à-dire une description formelle du système et de la manière dont il se comporte.

Automates à Etats finis

- ▶ Un AF est composé d'un ***ensemble fini d'états*** (représentés graphiquement par des cercles), ***d'une fonction de transition*** décrivant l'action qui permet de passer d'un état à l'autre (ce sont les flèches), ***d'un ou plusieurs états initiaux*** (désignés par des flèche entrantes), ***d'un ou plusieurs états finaux*** (désignées par le symbole $\vdash$).
- ▶ Un AF est donc un graph orienté où les nœuds correspondent aux états et les arcs aux flèches.

Automate fini déterministe

Définition 1 (Automate Fini Déterministe AFD (complet)).

Il est défini par un quintuplet $A = (Q, X, \delta, q_0, F)$, où :

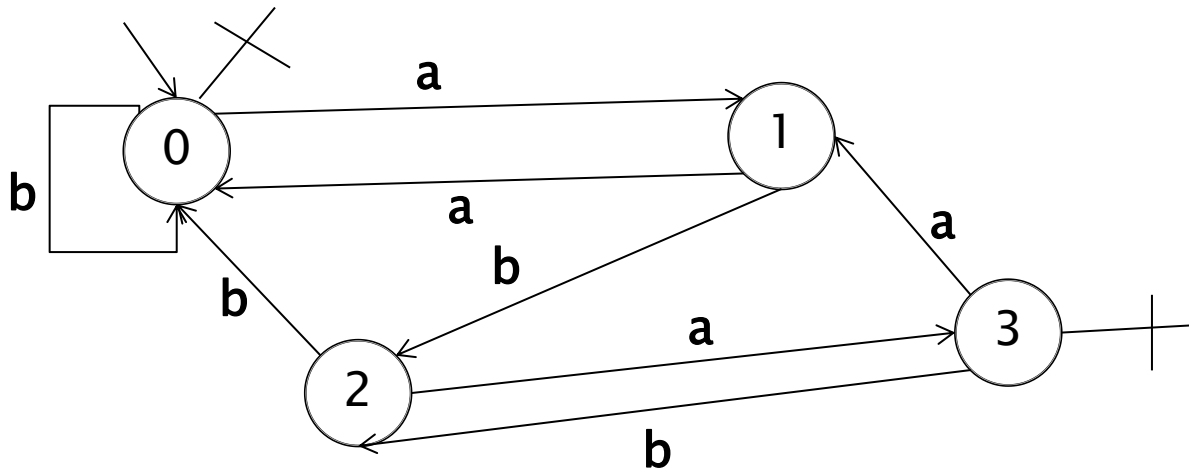
- Q : est un ensemble fini d'états
- X : est un ensemble fini de symboles (l'alphabet)
- $q_0 \in Q$: est l'état initial de l'automate
- $F \subset Q$: est l'ensemble des états finaux (ou terminaux) de l'automate.
- δ est une application de $Q * X$ dans Q , appelée fonction de transition de l'automate

Automate fini déterministe

- ▶ Un AF est donc un graphe orienté qui peut être décrit de la façon suivante:
- ▶ – Chaque état est représenté par un sommet du graphe.
- ▶ – on crée un arc (q, q') étiqueté a si $q' = \delta(q, a)$
- ▶ – l'état initial est différencié par une flèche entrante et les états finaux sont désignés par le symbole $+$.

Automate fini déterministe

- ▶ **Exemple:**
- ▶ Soit l'automate fini B dont la représentation graphique est donnée dans la figure suivante.

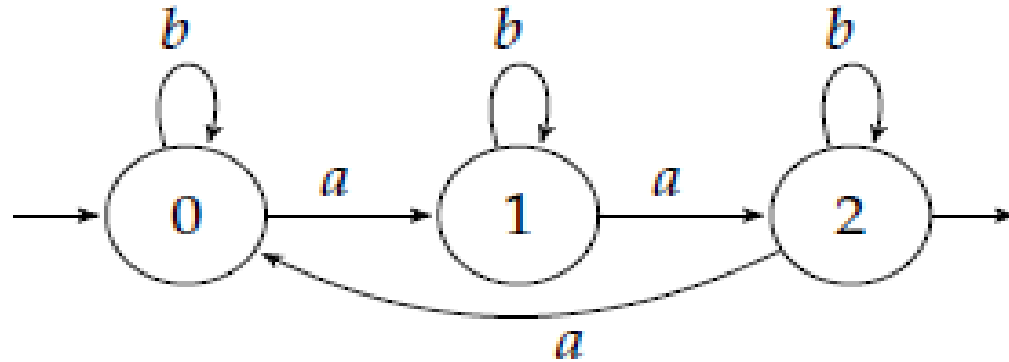


Automate fini déterministe

- ▶ La définition de B en terme de (X, Q, q_0, F, δ) est la suivante:
- ▶ 1. $X = \{a, b\}$
- ▶ 2. $Q = \{0, 1, 2, 3\}$
- ▶ 3. $q_0 = 0$: l'état initial
- ▶ 4. $F = \{0, 3\}$: les états finaux
- ▶ 5. δ : la fonction de transition est donnée par la table suivante:

δ	a	b
0	1	0
1	0	2
2	3	0
3	1	1

Automate fini déterministe



- ▶ Dans cette représentation, l'état initial 0 est marqué par un arc entrant sans origine et les états finaux (ici l'unique état final est 2) par un arc sortant sans destination.

Automate fini déterministe

- ▶ La fonction de transition correspondant à ce graphe s'exprime matriciellement par :

δ	a	b
0	1	0
1	2	1
2	0	2

La Fonction δ^*

- ▶ La fonction $\delta: Q \times X \rightarrow Q$ exprime le symbole (ou l'action) qui permet le passage d'un état à l'état suivant.
- ▶ Son extension est la fonction $\delta^*: Q \times X^* \rightarrow Q$: qui fournit l'état suivant pour une séquence de symboles.
- ▶ Rappelons que X^* correspond à l'ensemble de tous les mots que l'on peut former sur X .

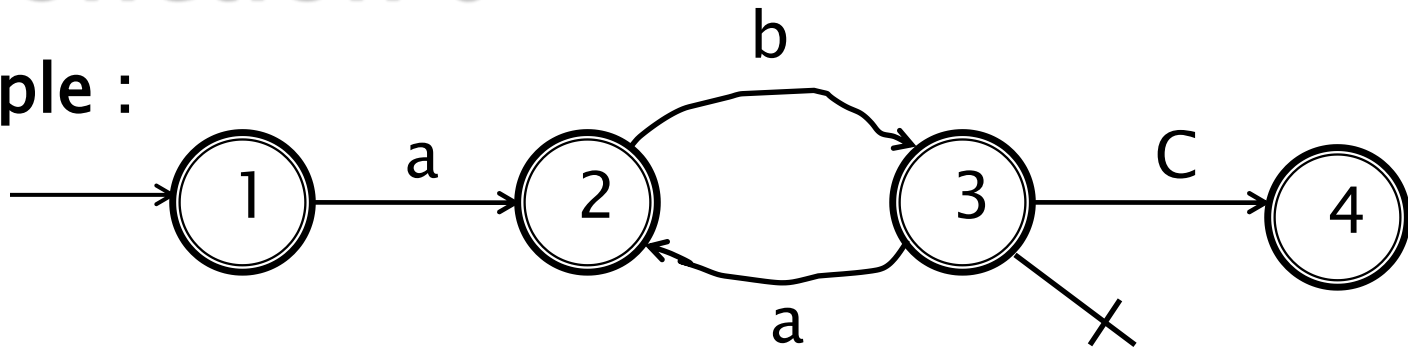
La Fonction δ^*

Définition:

- ▶ 1. $\delta^*(q, \varepsilon) = q$
- ▶ 2. $\delta^*(q, au) = \delta(\delta^*(q, u), a)$ avec $u \in X^*$ et $a \in X$.
 $= \delta^*(\delta(q, a), u)$
- ▶ Un mot $w \in X^*$ est reconnu (ou accepté) par l'automate fini $A = (Q, X, \delta, q_0, F)$ si $\delta^*(q_0, w) \in F$.
- ▶ Le langage reconnu par A est alors l'ensemble $L(A) = \{w \in X^* / \delta^*(q_0, w) \in F\}$

La Fonction δ^*

Exemple :



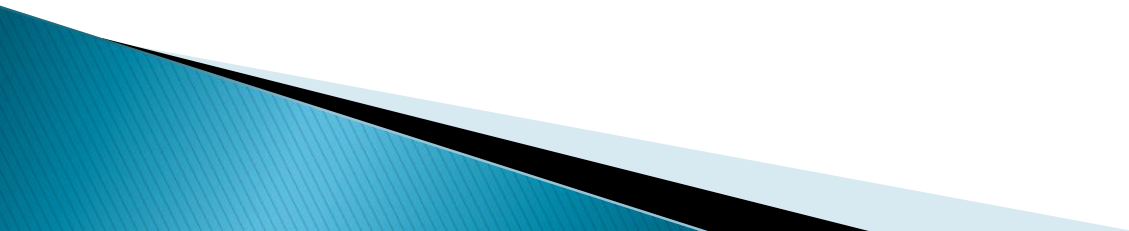
Automate B

- ▶ Le mot **abc** appartient-il à $L(B)$?

Exercice :

1. Donner un automate A qui lit le langage
$$L = \{ w \in \{a, b\}^* \mid w = a^*b \}$$
1. Définir A en terme de (Q, X, δ, q_0, F)
2. Montrer que $aaab \in L(A)$

Solution



Configuration et dérivations

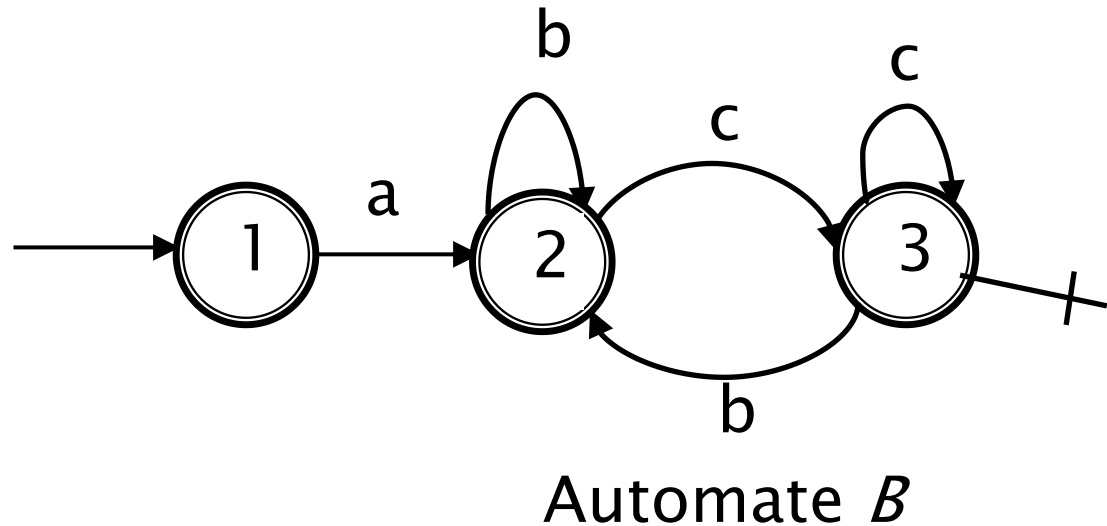
- ▶ Résumé: on a défini ce que représente un mot accepté en terme de fonction de transition: **un mot w est accepté par un automate M ssi $\delta^*(q_0, w) \in F$.**
- ▶ Afin de préciser le chemin parcouru par un automate, on introduit la notion de *configuration*.
- ▶ Une configuration d'un automate est un couple $(q, w) \in Q \times X^*$ où q représente l'état courant dans le quel se trouve l'automate et w est le mot en entrée qui reste à traiter.

Configuration et dérivations

- ▶ L'automate passe d'une configuration à une autre. Ce passage s'appelle *dérivation* et représente une (ou plusieurs) étapes d'exécution de l'automate.
- ▶ **Définition (Dérivation en une étape)**
- ▶ Pour un automate M , la dérivation en une étape de la configuration (q, w) à la configuration (q', w') est notée par
$$(q, w) \vdash (q', w') \text{ si } w = a.w' \text{ et } q' = \delta(q, a)$$

Configuration et dérivation

- ▶ Soit B l'automate suivant :



abbc appartient-il à $L(B)$?

Configuration et dérivation

- ▶ Définition (Dérivation en plusieurs étapes)
- ▶ Pour un automate M , la dérivation en plusieurs étapes de la configuration (q, w) à la configuration (q', w') est notée par $(q, w) \vdash\!\!\!\vdash (q', w')$ si $w = w'.w''$ et $q' = \delta^* (q, w'')$

Configuration et dérivation

- ▶ Cette notation permet d'exprimer formellement l'exécution d'un automate sur un mot w à partir d'un état q :

$$(q, w) \vdash (q_1, w_1) \vdash (q_2, w_2) \vdash \dots (q_n, w_n)$$

- ▶ Un mot w est accepté par un automate M si

$$(q_0, w) \vdash^* (q, \varepsilon) \text{ avec } q \in F$$

- ▶ Le langage accepté par M peut être exprimé par:

$$L(A) = \{ w \in X^* / \exists q \in F; (q_0, w) \vdash^* (q, \varepsilon) \}$$

Configuration et dérivation

- ▶ Un AFD accepte une chaîne d'entrée w si et seulement s'il existe un seul chemin dans le graphe de transition entre l'état initial et un état d'acceptation tel que les étiquettes d'arcs le long de ce chemin épellent le mot w .
- ▶ **Exemple1 (configuration en plusieurs étapes):**

$(1, abbc) \vdash (2, bbc) \vdash (2, bc) \vdash (2, c) \vdash (3, \varepsilon)$

$(1, abbc) \vdash (3, \varepsilon)$
 $\quad \quad \quad \underbrace{\hspace{1.5cm}}_w$
 $\quad \quad \quad q$

$(3, \varepsilon)$
 $\quad \quad \quad q' \quad w'$

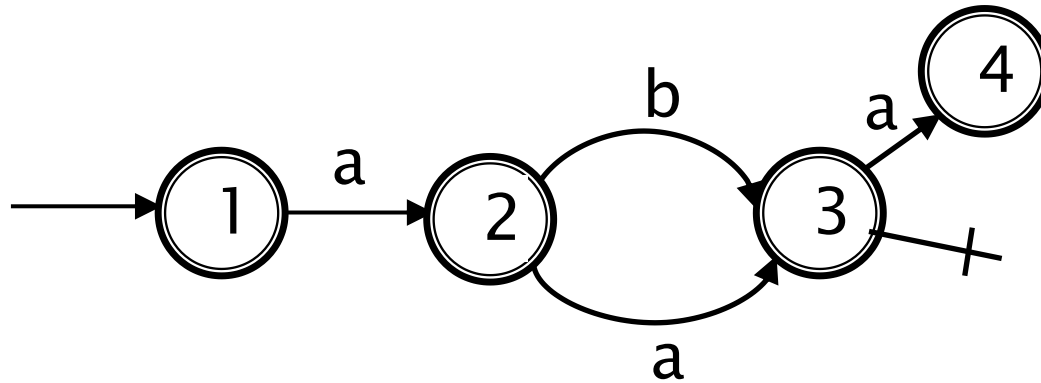
$$w'' = w.w'$$

$$\delta^*(1, abbc) = 3$$

Configuration et dérivation

Exemple :

- ▶ $L(A) = \{w \in X^* \mid q \in F; (q_0, w) \vdash^* (q, \varepsilon)\}$
- ▶ Soit l'automate A suivant :



Langage régulier

- ▶ Soit A un AFD qui a pour alphabet X . Nous dénotons par $L(A)$ l'ensemble de toutes les séquences de symboles reconnus par l'automate A .
- ▶ Nous dirons que $L(A)$ est le langage reconnu (ou accepté) par A .
- ▶ **Définition:** un langage L est régulier s'il existe un automate fini A tel que $L = L(A)$
 Reg: désignera l'ensemble des langages réguliers.
- ▶ Une façon pour montrer qu'un langage est régulier c'est le fait de construire un automate qui reconnaît ce langage.

Langage régulier

- ▶ **Exemple:**
- ▶ Soit $L = \{w \in \{a, b\}^* \mid w = a^*\}$
- ▶ L est-il régulier?

- ▶ **Solution:**

Déterminisme et non déterminisme

- ▶ Les automates défini précédemment possèdent une fonction de transition de $Q \times X \rightarrow Q$, c'est-à-dire qu'à partir d'un état et d'un symbole, un unique état peut être «atteint».
- ▶ On appelle ce type d'automates des *automates déterministes* (AFD).
- ▶ Cependant, il existe des modèles d'automates dits automates à états fini *non déterministes* (AFN) qui possèdent en outre la particularité suivante: à partir d'un état et pour un symbole donné, plusieurs états peuvent être atteints; le choix entre les différents «chemins» est équiprobable.

Déterminisme et non déterminisme

- ▶ **Définition (AFND: Automate Fini Non Déterministe)**
- ▶ C'est un modèle mathématique qui consiste en un quintuplet $A = (X, Q, Q_0, F, \delta)$ où:
 - X est l'ensemble des alphabets
 - Q est un ensemble fini d'états
 - $Q_0 \subset Q$ est l'ensemble des états initiaux de l'automate.
 - $F \in Q$ est l'ensemble des états finaux (ou terminaux) de l'automate.

Déterminisme et non déterminisme

- δ est une application de $Q \times \{X \cup \{\varepsilon\}\}$ dans $P(Q)$ appelé fonction de transition de l'automate. Cette fonction de transition fait correspondre des couples (état, symbole) à des ensemble d'états: $q \in Q, a \in X, \delta(q, a) = q'$ avec $q' \in P(Q)$.

Déterminisme et non déterminisme

- ▶ Intuitivement cet automate présente trois types de non-déterminisme:
 - Il peut exister plusieurs états initiaux
 - Peut lire le mot vide ε entre deux lettres du mot d'entrée.
 - A un couple (q, a) ; $a \in X \cup \{\varepsilon\}$, on peut associer plusieurs états, l'automate choisissant la transition qu'il veut.

- Avec la définition précédentes, le langage reconnu par un AFND est alors:

$$L(A) = \{w \in X^* / \exists q_0 \in Q_0, \exists q \in F: (q_0, w) \xrightarrow{*} (q, \varepsilon)\}$$

Autrement dit, il existe au moins un moyen de reconnaître le mot w .

Déterminisme et non déterminisme

- ▶ **Remarque:** Un AFD est un cas particulier d'un AFND dans lequel:
 - Aucun état n'a de ε -transition, c'est-à-dire de transition sur l'entrée ε .
 - Pour chaque état q et chaque symbole d'entrée a , il y a un seul arc étiqueté a qui quitte q :
$$\forall q \in Q, \forall a \in X, \exists q' \in Q \mid \delta(q, a) = q'$$
- Pour tout automate fini non déterministe A , il existe un automate fini déterministe B tel que $L(A) = L(B)$

Déterministe et non déterministe

► Automate fini déterministe

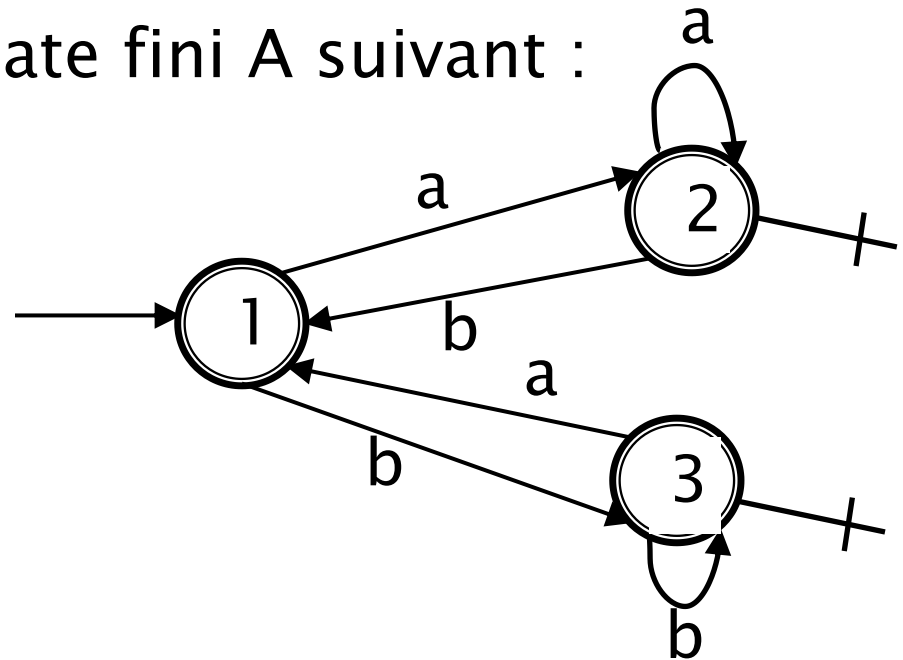
► Exemple : soit l'automate fini A suivant :

$X = \{a, b\}$; $Q = \{1, 2, 3\}$

$\delta(1, a) = 2$; $\delta(1, b) = 3$

$\delta(2, a) = 2$; $\delta(2, b) = 1$

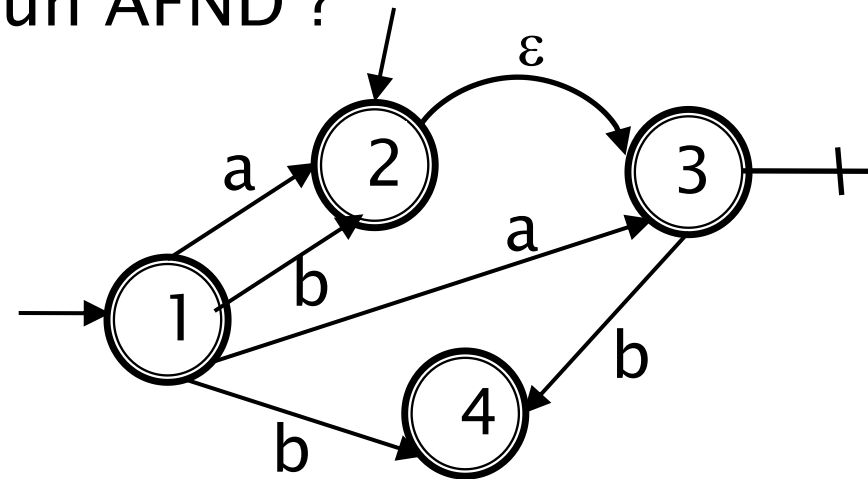
$\delta(3, a) = 1$; $\delta(3, b) = 3$



► A est-il AFD?

Déterministe et non déterministe

- ▶ **Automate fini non-déterministe**
- ▶ Soit l'AF A suivant : A est-il un AFND ?



Déterministe et non déterministe

AFD

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

- ▶ Un seul état initial
- ▶ Pour chaque lettre et chaque état on peut avoir une seule transitions sortante
- ▶ Pas de transition par la lettre ε

AFND

$$A = (X, Q, Q_0, F, \delta)$$

- Un ou plusieurs états initiaux
- Pour un état et une lettre on peut avoir plusieurs transitions sortantes
- On peut avoir des transitions par ε

Pour chaque automate fini non déterministe A il existe un automate fini déterministe B tel que $L(A) = L(B)$

Déterministe et non déterministe

- ▶ **Exercice** : donner un AFND qui reconnaît le langage dont les mots commencent par **a** suivi de 0 ou plusieurs occurrences de **b** et se terminent par **a**.
- ▶ **Solution** :

Grammaire et automates

- ▶ Deux aspects de représentation du langage:
 - Génération: grammaire ou définition récursive.
 - Reconnaissance: automate ou définition par propriété.
- ▶ Soit un langage reconnu par un automate, comment trouver la grammaire qui l'engendre et inversement. Pour répondre à cette question, il faut d'abord identifier le type de la grammaire:
 - Type 0: machine de Turing
 - Type 1: automate à mémoire
 - Type 2: automate à pile
 - Type 3: automate fini

Grammaire et automates

- ▶ Nous allons montrer que les grammaires régulières génèrent exactement les langages reconnus par les automates finis (déterministes ou non déterministes).
- ▶ Malgré les fortes restrictions imposés à la forme des règles de réécriture des grammaires régulières, elles génèrent l'ensemble des langages reconnaissables par les automates finis.

Grammaire et automates

- ▶ **Théorème:** Pour chaque alphabet X , l'ensemble des langages générés par toutes les grammaires régulières est égal à l'ensemble des langages reconnus par tous les automates finis.

$$\{L(G): G \text{ est une grammaire régulières sur } X\} = \{L(M): M \text{ est un automate fini sur } X\}$$

Conversion Grammaire/Automate

- ▶ Soit $G = (X, N, S, P)$ une grammaire régulière sur X . On peut transformer la grammaire G en une grammaire $G' = (X, N', S, P')$ qui génère le même langage que G mais en transformant toutes règles de la forme $A \rightarrow a$ avec $a \in X$ en $A \rightarrow aC$ et en ajoutant la règle $C \rightarrow \varepsilon$.
- ▶ On définit ensuite un automate fini $M = (X, Q, q_0, F, \delta)$ comme suit:
 - Le même alphabet
 - $Q = N'$
 - $q_0 = S$
 - $F = \{ A / A \rightarrow \varepsilon \text{ est une règle de } P' \}$
 - $\delta(A, a) = B$ si et seulement s'il existe une règle:
$$A \rightarrow aB \in P'$$

Conversion grammaire / automate

Soit G la grammaire suivante :

$$G : \left\{ \begin{array}{l} A \rightarrow aB \quad (1) \\ B \rightarrow b \\ A \rightarrow a \end{array} \right.$$

Conversion grammaire / automate

► Exercice

- Trouver un automate fini A tel que $L(A) = L(G)$
- G est défini comme suit :

$G :$

- $S \rightarrow aA$
- $S \rightarrow bS$
- $A \rightarrow aA$
- $A \rightarrow a$
- $S \rightarrow a$
- $S \rightarrow b$

Conversion Automate / Grammaire

- ▶ Soit $M = (X, Q, q_0, F, \delta)$ un automate fini. La grammaire $G = (X, N, S, P)$ est définie comme suit:
 - Le même alphabet
 - $N = Q$
 - $S = q_0$
 - $P = \{A \rightarrow aB \mid \delta(A, a) = B\} \cup \{B \rightarrow \varepsilon \mid B \in F\}$

Il s'agit simplement de la construction inverse de celle qui précède

Conversion automate / Grammaire

$R = (X, Q, Q_0, F, \delta)$

X

Q

Q_0

δ

F

$G = (X, N, S, P)$

X

N

S

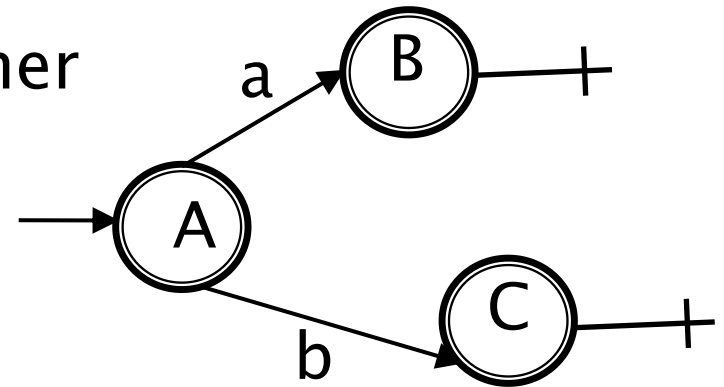
$P \Rightarrow A \rightarrow aB$

$C \rightarrow \varepsilon$

► **Exemple :**

soit l'automate suivant chercher la grammaire correspondante

► On cherche $G(X, N, S, P)$



- **Exercice** : soit l'automate suivant, trouver la grammaire correspondante.

