Cours: Théorie des langages & Compilation

Chapitre 3: Langages réguliers et Automates finis (partie A)

Réalisé par:
Dr. Sakka Rouis Taoufik

A.I. Introduction

Les grammaires représentent un moyen qui permet de *décrire un langage d'une manière* inductive. Elles montrent comment les mots du langage sont construits.
Considérons à présent un langage L, on se propose de répondre à la question w ∈ L ou non?
On peut répondre à cette question de plusieurs façons.

- On peut vérifier l'existence de w dans la liste des mots de L (impossible à réaliser si le langage est infini).
- Si L est défini par compréhension, on peut alors vérifier si w respecte la propriété du langage.
- Si L est défini par une grammaire, on vérifie l'existence d'une chaîne de dérivation pour w, le cas échéant on conclut que w ∈ L.
- Il existe en réalité un autre moyen permettant de répondre à cette question : **les automates**.

A.II. Les Automates

Définition : Un automate est une machine abstraite qui permet de lire un mot et de répondre à la question : "un mot w appartient-il à un langage L ?" par oui ou non.

Formellement, un automate contient au minimum :
– Un alphabet pour les mots en entrée noté X ;
– Un ensemble non vide d'états noté Q;
– Un état initial noté $q_0 \in Q$;
– Un ensemble non vide d'états finaux $F \subseteq Q$;
– Une fonction de transition (permettant de changer d'état) notée δ.

A.III. Représentation des automates

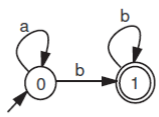
Exemple : L'automate qui reconnaît les mots de la forme $a^n b^m$ ($n >= 0$, $m > 0$) est le suivant :

<{a, b}, {0, 1}, 0, δ, {1}> tel que δ est donnée:

➢**par table:**

État	a	b
0	0	1
1	-	1

➢**Graphiquement:**



A.IV. Classification des automates

Comme les grammaires, les automates peuvent être classés en 4 classes selon la hiérarchie de Chomsky.
La classification de Chomsky pour les automates consiste à définir, pour chaque classe de langage, **l'automate minimal** permettant de répondre à la question "un mot w appartient-il à un langage ?".
Nous avons quatre classes d'automates :

– **Type 3** ou **automate à états fini (AEF)** : il reconnaît les langages de type 3. Sa structure est la suivante :
– bande en entrée finie ;
– sens de lecture de gauche à droite ;
– pas d'écriture sur la bande et pas de mémoire auxiliaire.

– **Type 2** ou **automate à pile** : il reconnaît les langages de type 2. Sa structure est similaire à l'AEF mais dispose en plus d'une mémoire organisée sous forme d'une pile infinie ;

Ch3: Langages réguliers et Automates finis

A.IV. Classification des automates

– **Type 1** ou **machine de Turing à bornes linéaires (MTBL) :** il reconnaît les langages de type 1. Sa structure est la suivante :
– Bande en entrée **finie accessible en lecture/écriture ;**
– Lecture dans les deux sens ;
– Pas de mémoire auxiliaire.

– **Type 0** ou **machine de Turing :** il reconnaît les langages de type 0. Sa structure est la même que l'MTBL mais la bande en entrée est infinie.

Le tableau suivant résume les différentes classes de grammaires, les langages générés et les types d'automates qui les reconnaissent :

Grammaire	Langage	Automate
Type 0	Récursivement énumérable	Machine de Turing
Type 1 ou contextuelle	Contextuel	Machine de Turing à borne linéaire
Type 2 ou hors-contexte	Algébrique	Automate à pile
Type 3 ou régulière	Régulier ou rationnel	Automate à états fini

A.V. Automate à états fini

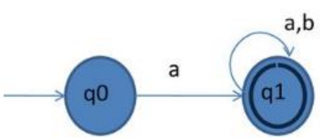
Définition: Un automate fini est dit **complet** sur un vocabulaire X **ssi** pour chaque état q et chaque symbole s, il existe au moins une transition qui quitte q avec le symbole s.

Définition : Un automate fini est dit **non ambigu** sur un vocabulaire X **ssi** pour chaque état q et chaque symbole s, il existe **au plus** une transition qui quitte q avec le symbole s.

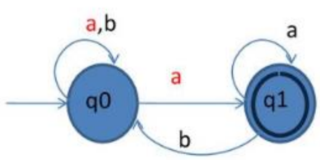
Définition : Un automate fini est dit **déterministe** sur un vocabulaire X ssi il possède un unique état initial, ne comporte aucune transition vide et admet au plus une transition pour chaque couple (état, symbole).

Rq. On peut conclure qu'un automate fini **complet** et **non ambigu** est toujours **déterministe (l'inverse n'est pas toujours vrai).**
(pour chaque état q et chaque symbole s, il existe **une et une seule** transition qui quitte q avec le symbole s ➔ Déterministe).

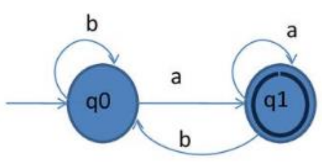
A.V. Automate à états fini



Non complet sur {a, b} car
$\delta(q0, b) = \phi$
L(A) = a(a+b)*



Ambigu sur {a, b} car
$\delta(q0, a) = \{q0, q1\}$
$L(A) = ((a+b)^*a^+)^+$



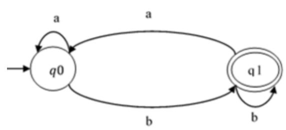
$L(A) = (b^*a^+)^+$

Automate fini déterministe sur {a, b} car il est complet et non ambigu:
$\delta(q0, a) = \{q1\}$ $\delta(q0, b) = \{q0\}$ $\delta(q1, a) = \{q1\}$ $\delta(q1, b) = \{q0\}$

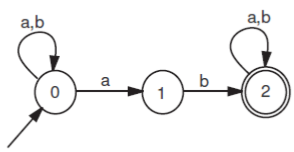
A.V. Automate à états fini

Définition formelle : Un AEF est dit déterministe si les deux conditions suivantes sont vérifiées:

- $\forall q_i \in Q, \forall a \in X$, il existe au plus un état q_j tel que $\delta(q_i, a) = q_j$;
- L'automate ne comporte pas de ε -transitions.



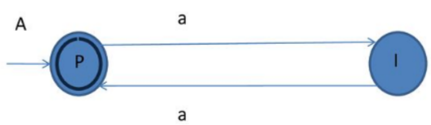
Sinon l'automate est dit fini non déterministe (AFN)



Ch3: Langages réguliers et Automates finis

A.V. Automate à états fini

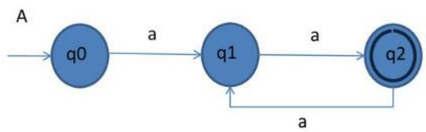
Exemple d'un automate fini qui accepte le langage L1
$L1= \{\omega \in \{a\}^* \ / \ | \omega | = 2k, k >= 0\}$. (aa)*



L'état initial devient un état final puisque le mot ε est accepté (pour k = 0). En étant à l'état initial de l'automate, on peut ne rien lire (c-a-d lire 0 symboles donc lire ε) et on est déjà à un état final.

A.V. Automate à états fini

Exemple d'un automate fini qui accepte le langage L2
L2= {ω ∈ {a}* / | ω | = 2k, k > 0}. aa(aa)*



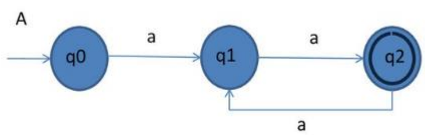
Le plus petit mot accepté est aa (On atteint l'état final après la lecture de aa).
Après il faudra continuer à avoir un nombre paire de a. Donc quand on se trouve à l'état d'acceptation ou l'état final, il faudra continuer à lire des séquences de 2a pour revenir à l'état final.

Ch3: Langages réguliers et Automates finis

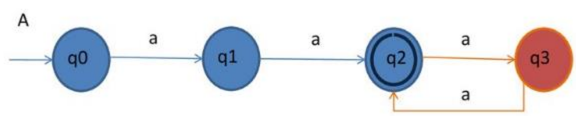
A.V. Automate à états fini

Exemple d'un automate fini qui accepte le langage L2
L2= {ω ∈ {a}* / | ω | = 2k, k > 0}. aa(aa)*

Sol1:



Sol2:



Le plus petit mot accepté est aa (On atteint l'état final après la lecture de aa).

A.V. Automate à états fini

Exercice1 : Pour chaque langage, construire un automate fini qui l'accepte.

L3= $\{\omega \in \{a, b\}^* \ / \ |\omega|_a = 3k+2, k >= 0\}$.

L4= $\{\omega \in \{a\}^* \ / \ |\omega| = 3k+1, k >= 0\}$.

L5= $\{\omega \in \{a, b,c\}^* \ / \ |\omega|_a = 3k+2, k >= 0\}$.

Ch3: Langages réguliers et Automates finis

A.V. Automate à états fini

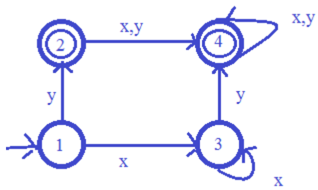
Exercice 2:
a) Donner le graphe de transition représentant l'automate suivant:
A=< {0,1,2,3},{a,b,c}, 0, δ , {2}>
Avec δ :

δ(0,a)=3	δ(0,b)=1	δ(0,c)=1
δ(1,a)=1	δ(1,b)=2	δ(1,c)=3
δ(2,a)=0	δ(2,b)=3	δ(2,c)=3
δ(3,a)=3	δ(3,b)=3	δ(3,c)=3

b) Dites si cet automate est déterministe ou non

A.V. Automate à états fini

Exercice 3:
Soit l'automate suivant:



a) Dites si les mots ci-dessous sont acceptés:

xxy x yyx xxyxy xyx

b)Dite si cet automate est déterministe ou non

Ch3: Langages réguliers et Automates finis

A.V. Automate à états fini

Exercice 4: Construire des automates reconnaissant les langages suivants:

$L6= \{\omega \in \{x, y\}^* \ / \ \omega = x^i , i >= 0\}$.

$L7= \{\omega \in \{x, y\}^* \ / \ \omega = x^i , i > 0\}$.

$L8= \{\varepsilon\}$.

$L9= \{\omega \in \{x, y\}^* \ / \ \omega = x^i y^j , i >= 0 \text{ et } j>0\}$.

$L10= \{\omega \in \{x, y\}^* \ / \ |\omega| \text{ est paire}\}$.