Automates à états finis

1-**Les automates :** Un automate est une machine abstraite de reconnaissance des mots d'un langage. L'objectif principal des automates est de présenter dans un format lisible un générateur de mots d'un langage, y compris la vérification si un mot appartient à ce langage ou non. Un automate reconnait un langage engendré par une grammaire, alors que chaque type de grammaire a un type d'automate adéquat.

2-Classification des automates Comme les grammaires, les automates peuvent être classés en 4 classes selon la hiérarchie de Chomsky. Il ne faut surtout pas oublier que la classification de Chomsky est une classification de complexité. Par conséquent, les automates reconnaissant les langages simples (par exemple ceux de la classe 3) sont plus simples que les automates reconnaissant les langages complexes. 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 ;

–Type 1 ou automate à bornes linéaires (ABL) : 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'ABL 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 :

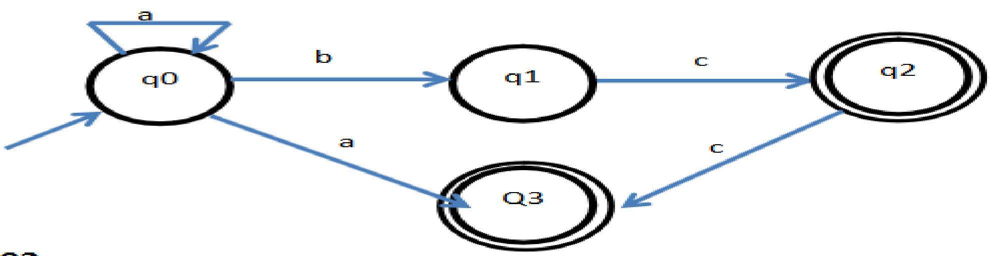
Grammaire	**Langage**	**Automate**
Type 0	Récursivement énumérable	Machine de Turing
Type 1 ou contextuelle	Contextuel	Machine de Turing à borne linéaire
Type 2 ou hors-contexte	Algébrique	Automate à pile
Type 3 ou régulière	Régulier ou rationnel	Automate à états fini

3-Un automate d'états finis : un automate d'états finis(AEF) est le plus simple des automates. Un AEF reconnait les langages de type 3 sans nécessiter d'avoir une mémoire. Un AEF est défini par un quintuplé A=(X, Q, q0, F, δ) tel que :

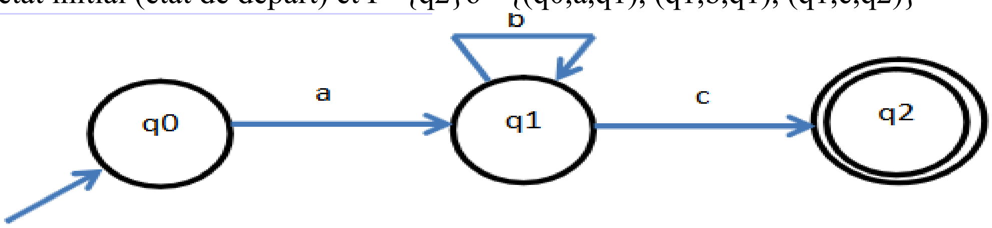
- X : est l'ensemble des symboles (lettre de l'alphabet du langage reconnue par cet automate.)

- Q : est l'ensemble des états de cet automate (on peut les considérer comme les non terminaux d'une grammaire)
- q0-et F sont deux états appartiennent a Q. tel que q0est l'état initial (on peut le considérer comme l'axiome d'une grammaire) et F est un ensemble des états finaux nommés par fois états d'acceptation.
- δ est la fonction de transition qui passe d'un état à un autre en lisant un symbole de X.
- Remarque : Deux automates sont équivalents s'ils reconnaissent le même langage.
- Exemple : Soit A un automate =(X, Q, q0, F, δ):X={a,b,c}Q, ={q0,q1,q2} / q0 est l'état initial (état de départ) F={q2} δ= {(q0,a,q1), (q1,b,q1), (q1,c,q2)}
- **4. Représentation des automates d'états finis** Les automates d'états finis sont un autre formalisme de langage type3, il ya trois types de représentation d'AEF, la représentation à travers la définition, la représentation graphique et la représentation par table.

  4.1. Représentation à travers la définition c'est -à -dire en expliquant les cinq paramètres formels avec en particulier la définition précise de δ sur chaque élément du domaine lorsque δ existe pour un couple donné.
- **Exemple A :** Soit A un automate =(X, Q, q0, F, δ):X={a,b,c,d}Q, ={q0,q1,q2,q3} /q0 est l'état initial (état de départ)F={q3}δ= {(q0,a,q1),(q1,b,q2), (q1,c,q1),(q2,d,q3)}
- **4.2-Représentation graphique(ou diagramme de transition)** Les états d'un AEF dans une représentation graphique sont représentés par un graph orienté, les nœuds du graph sont les états de l'automate et les arcs sont les transitons étiquetés par les symboles de l'AEF. Dans le cas où on a plusieurs symboles pour le même arc on peux utiliser virgule pour enter les symboles. L'état initial est represanté par une flèche et les états finaux par deux cercles.
- **Exemple 01:** Soit A un automate = (X, Q, q0, F, δ) : X={a,b,c}, Q, ={q0,q1q2,q3} / q0 est l'état initial (état de départ) et F={q2,q3}δ= {(q0,a,q0),(q0,b,q1), (q0,a,q3), (q1,c,q2), (q2,c,q3)}



- **Exemple 02 :** Soit A un automate =(X, Q, q0, F, δ):X={a, b, c}, Q, ={q0,q1q2} / q0 est l'état initial (état de départ) et F={q2}δ= {(q0,a,q1), (q1,b,q1), (q1,c,q2)}



4.3-Représentation par table : Les AEF peuvent être représentés par des tables ou les colonnes de la table contiennent les symboles de l'automate, les lignes contient les états de l'automate, les cases de la table représentent les transitions. On remplit la table par la transition adéquate de la ligne et la colonne d'une case donné.

Exemple : Soit A un automate =(X, Q, q0, F, δ):X={a, b, c}, Q, ={q0,q1q2} / q0 est l'état initial (état de départ) et F={q2}δ= {(q0,a,q1), (q1,b,q1), (q1,c,q2)}.

Etats / lettre	a	b	c
q0	q1	/	/
q1	/	q1	q2
q2	/	/	/

5. langage reconnu par un automate d'états finis :

Cas 1 : Un mot est acceptable par un automate autrement dit appartient au langage reconnu par cet automate, noté L(A),si et seulement si on peut lire toutes les lettres du mot et à la fin on arrive à un état final.

Cas 2 : Un mot est rejeté par un automate (n'appartient pas au langage) si on a fini de lire toutes les lettres de ce mot sans arriver à un état final.

Cas 3 : Un mot est rejeté par un automate (n'appartient pas au langage) s'il n y a pas une transition qui peux lire la lettre courante dans l'automate .On dit que l'automate est bloqué.

en utilisant l'automate de l'exemple A, faisons le fonctionner sur quelques mots :

abd,ac^2,acba, Ɛ

Mot abd:

δ*(q0,abd)= δ*[δ(q0,a),bd]=δ*[q1,bd]=δ*[δ(q1,b),d]=δ*[q2,d]= δ(q,d)=q3∈F donc abd ∈ L(A)

Mot ac^2:

δ*(q0,acc)= δ*[δ(q0,a),cc]=δ*[q1,cc]=δ*[δ(q1,c),c]=δ*[q1,c]= δ(q1,c)=q1∉F donc acc ∉ L(A)

Mot acba:

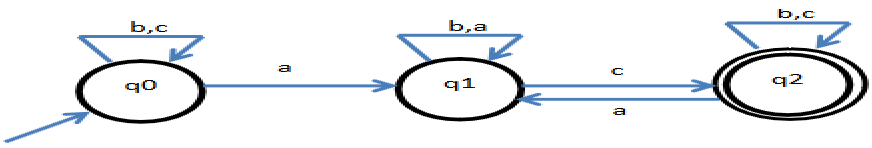
δ*(q0,acba)= δ*[δ(q0,a),cba]=δ*[q1,cba]=δ*[δ(q1,c),ba]=δ*[q1,ba]= δ*[δ(q1,b),a]=δ*[q2,a]= δ(q2,a)=? on a donc δ(q2,a) n'existe pas , il y a donc un blocage et acba ∉ L(A)

Mot Ɛ:

δ*(q0, Ɛ)=? blocage donc Ɛ∉ L(A)

6. Automate complet : soit A=(X, Q, q0, F, δ) un automate d'états fini, l'automate A est complet si et seulement si pour chaque état de Q il y'a une transition appartient à δ pour chaque symbole de X.

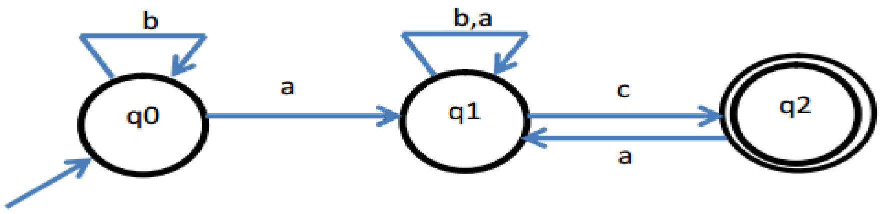
Exemple : Soit A un automate =(X, Q, q0, F, δ):

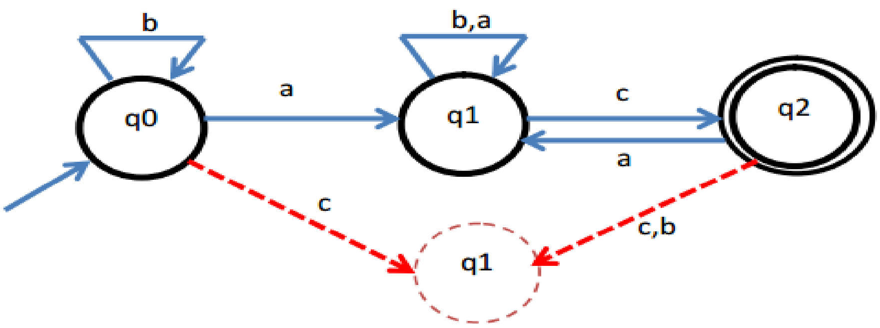


L'algorithme pour compléter un automate est d'ajouter un état dit état puis (n'est pas final) et ajouter les arcs manquantes de chaque état de Q pour avoir un automate complet.

Exemple:

Soit A un automate =(X, Q, q0, F, δ) :



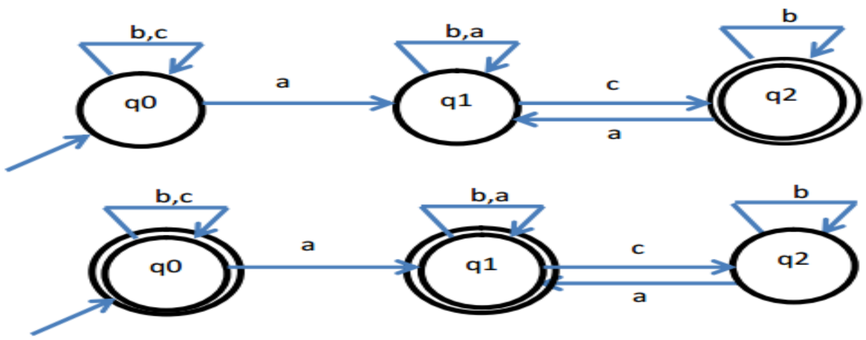


7. Union d'automates : L'union de deux automates A1=(X, Q1, q0, F1, δ1) et A2=(X, Q2, q1, F2, δ2) sert à construire l'automate de l'union de deux langages reconnu par A1 et A2. Le nouveau automate sera défini comme suit A1UA2 : (X, Q1UQ2, q0Uq1, F1UF2, δ1Uδ2)

8.Le complément :soit A un automate d'états finis, le complément de l'automate A est un nouvel automate d'états finis ou tous les états finaux devient non-finaux et vice versa. Dans le cas où l'automate n'est pas complet il faut le compléter en ajoutant des états puis.

Exemple: (Complément)

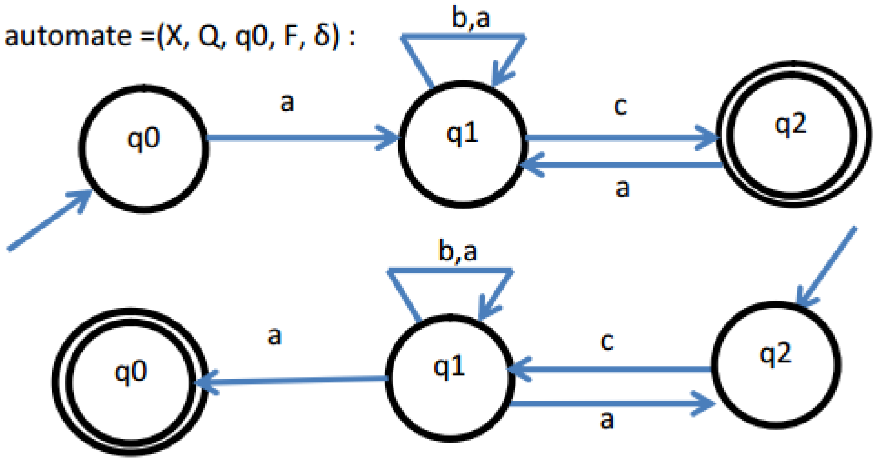
Soit A un automate =(X, Q, q0, F, δ) :



9. L’automate miroir : l’automate miroir est le l’automate qui reconnait le langage miroir de l’automate original. La procédure de trouver l’automate miroir est simple, il suffit juste d’inverser les transitions (les arcs) et le statut des états initiaux et finaux.

Exemple:

Soit A un automate =(X, Q, q0, F, δ) :



10. Le déterminisme d’un automate d’états finis :

Un automate d’états finis déterministe est un automate où l’on ne peut pas trouver deux transitions avec le même symbole à partir d’un seul état, pour enlever l’ambiguïté de lire un mot dans un automate d’états finis. Tout automate fini non déterministe a un automate fini déterministe équivalent ou chaque état possède au plus une transition avec le même symbole et il ne contient qu’un seul état initial.

Algorithme de détermination d’un automate d’états finis

Rabin et Scott ont prouvé que tout langage reconnu par un automate d’états finis non déterministe est également reconnu par un AEF déterministe équivalent. On suppose de construire un nouveaux automate d’états finis déterministe et qui reconnait tous les mots du langage initial.

Algorithme de déterminisation d’un AEF:

Soit A=(X, Q, q0, F, δ) un automate fini non-déterministe, on construit l’automate fini déterministe A’ = (X, Q’, q0’, F’, δ’) ou Q’ c’est un sous-ensemble de Q.

δ’ ← ϕ

q0’ ← q0 ∪ {qi / (q0, Ɛ, δ) Є δ }

Pour chaque nouveau état q’ Є Q’ faire

Pour chaque symbole c de X faire

q’’ ← { y / il existe x ⍰ q’ , y ⍰ Q tels que (x,σ,y) ⍰ δ }

si q’’ ≠ ϕ alors

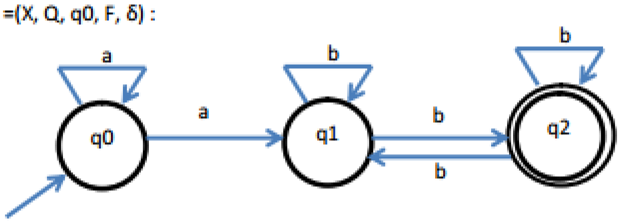
q’’ ← q’’∪ {y /il existe y ∊ q’’ et z ∊ Q tels que (y,Ɛ,z) ⍰ δ }

δ’ ← δ’ ∪ { (q’, c, q’’) Q’ ← Q’ ∪ {q’’}

F’ ← {q’ tels que q’ ∩ F ≠ ⍰ } }

Exemple:

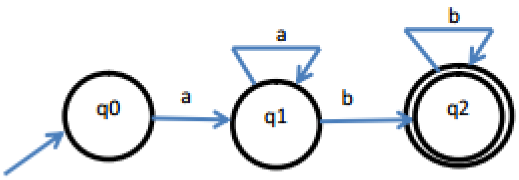
Soit A un automate =(X, Q, q0, F, δ) :



état	a	b
q0	q0,q1	

état	a	b
q0	q0,q1	
q0,q1	q0,q1	q1,q2

état	a	b
q0	q0,q1	
q0,q1	q0,q1	q1,q2
q1,q2		q1,q2



11. La minimisation d’un automate d’états finis :

Soit A=(X,Q,q0,F, δ) un automate d'état fini déterministe qui reconnait un langage L. d'après Nerode-Myhill tout langage de type 3 peut être représenté par un automate qui comporte le nombre minimum d'états et équivalent à tous les automates qui représente L.

Algorithme de minimisation d'un AEF :

La minimisation d'un AEF déterministe sert à enlever tous les états non utiles ou bien redondantes, on peut le faire par deux étapes différentes, premièrement, on enlève tous les états inaccessibles et les états puis, les états inaccessibles sont des états ou on ne peut pas attendre cet état à partir d'un état initial, et les états puis sont les états ou il n'a pas un chemin vers un état final à partir de cet état. Deuxièmement, grouper des états dit équivalents, deux états sont équivalents si l'on lit le même symbole et on arrive à un état final.

Algorithme de minimisation d'un AEF :

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

- Créer deux Classes C1 et C2 // C1 contient les états finaux et C2 contient les états non-finaux

Répéter

- Pour chaque deux états différentes q1 et q2, s'il existe deux transitions différentes qui commencent de q1 et q2 respectivement en lisant le même symbole et point vers deux classes différentes alors :

Créer une nouvelle classe Ci et séparer q1 de q2.

Jusqu'à ce que il n y a pas de deux classes à séparer.

Exemple:

Soit A un automate =(X, Q, q0, F, δ) :

état	a	b
q1	q2	q4
q2	q3	q5
q3	q6	q1
q4	q3	q7
q5	q8	q6
q6	q3	q6
q7	q8	q6
q8	q8	q7
q9	q7	q5

1- Enlever l'état q9 parce que c'est un état inaccessible
2- chercher les états équivalents

C1= [q1,q2,q3,q4,q5,q6,q7] ; C2= [q8]

C1= [q1,q2,q3,q4,q6] ;C3 = [q5,q7] ; C2=[q8] (séparés par a).

C1= [q1,q3,q6] ; C4= [q2,q4] ; C3= [q5,q7] ; C2= [q8] (séparés par b).

C1= [q1] ; C5= [q3,q6] ; C4= [q2,q4] ; C3= [q5,q7] ; C2= [q8] (séparés par a).

C1= [q1] ; C5= [q3] ; C6 =[q6] ; C4= [q2,q4] ; C3= [q5,q7] ; C2= [q8] (séparés par b).

On ne peut plus séparer des états

état	a	b
q1	q24	q24
q3	q6	q1
q24	q3	q57
q6	q3	q6
q57	q8	q6
q8	q8	q57

Exercice (2.1) :

Donnez si possible les automates à états finis qui reconnaissent les langages suivants :

1- Tous les mots qui commencent par un a dans l'alphabet A= {a, b}

2- Tous les mots qui terminent par un a dans l'alphabet A= {a, b, c}

3- Tous les mots qui contiennent au moins un a dans l'alphabet A= {a, b, c}

4- Tous les mots qui contiennent le facteur bc dans l'alphabet A= {a, b}

5- Tous les mots qui commencent par un a et se terminent par un b dans l'alphabet

A= {a, b}

6- Tous les mots qui contiennent le suffixe abc dans l'alphabet A= {a, b, c}

7- Tous les mots qui contiennent le préfixe abc dans l'alphabet A= {a, b, c}

8- Tous les mots qui contiennent un nombre de a plus que le nombre de b dans l'alphabet A={a, b}

9- Tous les mots qui contiennent un nombre de a égal le e nombre de b dans l'alphabet A= {a, b} ;

10- Tous les mots (nombre binaire) qui acceptent la division sur 2 par un a dans l'alphabet A = {0,1} ;

Exercice (2. 2) :

Soit A un AEF qui se définit comme suit :

A={a,b,c}, Q, ={q0,q1q2} , q0, F={q2}, δ= {(q0,a,q1), (q1,b,q1), (q1,c,q2)}

1- Dans quel état se trouve l'automate A après lecture des mots a, ab, abac, abb ?

2- Montrer que le mot abbbbc est reconnu par A.

3- Montrer que le mot aac n'est pas reconnu par A.

4- Montrer qu'un mot se terminant par b ne peut pas être reconnu par A.

5- Décrire les mots reconnus par l'automate A.