(27)

3.5 les types des AEF (Automate d'Etats Finis)

il existe deux types déterministe et indéterministe

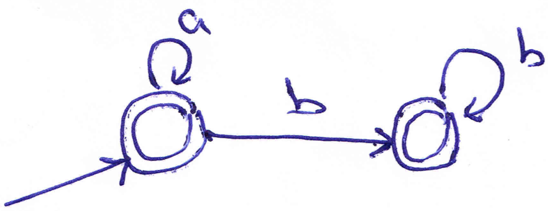
un AEF (X, Q, I, δ, F) est dit déterministe si les deux conditions sont vérifiées :

- $\forall q_i \in Q, \forall x_i \in X$, il existe au plus un état q_j tel que $\delta(q_i, x_i) = q_j$
- l'AEF ne comporte qu'un seul état initial

sinon

cet AEF est dit indéterministe

exemple



est un AEF déterministe

3.6. Configuration d'AEF

une configuration d'AEF pour un mot donné, c'est l'image instantané de l'automate donnée par

- l'état dans lequel est l'automate
- ce qui lui reste à lire de ce mot.

Formellement : une configuration est un couple de $Q \times X^*$

3.7 Mouvement d'AEF

le mouvement d'AEF est décrit par le passage d'une configuration à une autre, cette dernière est obtenue en appliquant la relation de transition δ sur le premier symbole du mot reste à lire

3.7.1 Notations :

- la relation binaire entre les configurations qu'on note $\vdash$ est définie par

$$\vdash \; : \; QX^* \longrightarrow QX^*$$

exemple : $(q_2, cabaa) \vdash (q_3, abaa)$ si $\delta(q_2, c) = q_3$

on dit alors que on passe de $(q_2, cabaa)$ à $(q_3, abaa)$ en une étape

- la fermeture réflexive et transitive de $\vdash$ est notée par $\overset{*}{\vdash}$

c'est à dire :

$(q, w) \overset{*}{\vdash} (q', w')$ signifie qu'on passe de (q, w) à (q', w') en zéro, une ou plusieurs étapes.

Remarque :

- étape signifie qu'il n'existe pas une transition et dans ce cas la configuration reste elle même (c'à-d : $q = q'$ et $w = w'$)

et on écrit $(q, w) \overset{0}{\vdash} (q, w)$

3.8 langage reconnu par AEF déterministe

Définitions : soit $A(X, Q, q_0, \delta, F)$ un AEF déterministe

- on appelle configuration initiale d'un AEF déterministe le couple (q_0, w) avec w le mot à lire.

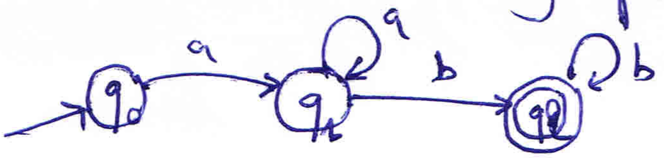
(29) • On appelle configuration finale d'un AEF déterministe le couple (q_F, ε) avec $q_F \in F$

donc : le langage reconnu par un AEF déterministe est l'ensemble des mots entièrement lus à partir de la configuration initiale jusqu'à la configuration finale.

Formellement :

$$L(A) = \{ w \in X^* / (q_0, w) \vdash^* (q_F, \varepsilon) \text{ avec } q_F \in F \}$$

exemple : soit A l'AEF défini par le graphe suivant :



est ce que les mots suivants appartiennent au langage reconnu par cet AEF.

a^2b , a^2 , a^2ba , ε.

$w \in L(A) \Leftrightarrow (q_0, w) \vdash^* (q_F, \varepsilon)$ tel que $q_F \in F$

1/ $a^2b \overset{?}{\in} L(A)$

$(q_0, aab) \vdash (q_1, ab) \vdash (q_1, b) \vdash (q_2, \varepsilon)$ et $q_2 \in F$ donc $a^2b \in L(A)$

2/ $a^2 \overset{?}{\in} L(A)$

$(q_0, aa) \vdash (q_1, a) \vdash (q_1, \varepsilon)$ mais $q_1 \notin F$ donc $a^2 \notin L(A)$

3/ $a^2ba \overset{?}{\in} L(A)$

$(q_0, aaba) \vdash (q_1, aba) \vdash (q_1, ba) \vdash (q_2, a)$ il y a un blocage puisque $\delta(q_2, a)$ n'existe pas et le mot n'est pas entièrement lu donc $a^2ba \notin L(A)$

(30)

4/ $\varepsilon \overset{?}{\in} L(A)$

(q_0, ε) il n'existe pas une transition ; puisque $\delta(q_0, \varepsilon)$ n'existe pas et $q_0 \notin F$ donc $\varepsilon \notin L(A)$

Exemple : soit A l'automate d'états finis défini par le graphe suivant

même questions avec les mots a^2b, a^1, ε

- $(q_0, aab) \vdash (q_0, ab) \vdash (q_0, b)$ il y a un blocage et le mot n'est pas entièrement lu donc $a^2b \notin L(A)$
- $(q_0, a) \vdash (q_0, a) \vdash (q_0, \varepsilon)$ et $q_0 \in F$ donc $a^1 \in L(A)$
- (q_0, ε) il n'existe pas une transition puisque $\delta(q_0, \varepsilon)$ n'est pas définie mais q_0 est aussi $\in F$ donc $\varepsilon \in L(A)$.

3.8- langage reconnu par AEF indéterministe

Définition : soit $A(X, Q, I, \delta, F)$ un AEF indéterministe

- on appelle configuration initiale d'un AEF indéterministe le couple (q_i, w) avec $q_i \in I$
- on appelle configuration finale d'un AEF indéterministe le couple (q_f, ε) avec $q_f \in F$

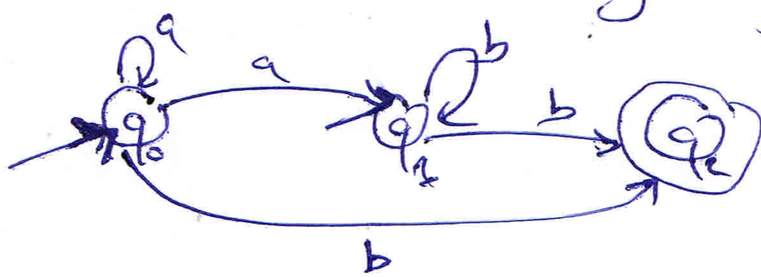
donc le langage reconnu par AEF indéterministe est l'ensemble des mots lus à partir de la configuration initiale jusqu'à la config finale.

Formellement $L(A) = \{ w \in X^* / (q_i, w) \overset{*}{\vdash} (q_F, \varepsilon) \}$

avec $q_i \in I$ et $q_F \in F$

(3 A)

exemple Soit A l'automate d'états fini défini par le graphe suivant



même question avec les mots a^2b, a^2, aba

1/ $a^2b \overset{?}{\in} L(A)$ $\quad I = \{q_0\}$, $F = \{q_2\}$

$(q_0, aab) \vdash (q_0, ab) \vdash (q_0, b) \vdash (q_2, \varepsilon)$ et $q_2 \in F$ donc $a^2b \in L(A)$

$\qquad\qquad\qquad\qquad\quad \vdash (q_1, b) \vdash (q_1, \varepsilon)$ mais $q_1 \notin F \Rightarrow \notin L(A)$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad \vdash (q_2, \varepsilon) \Rightarrow a^2b \in L(A).$ et $q_2 \in F$

$\qquad\quad \vdash (q_1, ab)$

∃ un blocage.

$\boxed{a^2b \in L(A)}$

2/ $(q_0, aa) \vdash (q_0, a) \vdash (q_0, \varepsilon)$ mais $q_0 \notin F \Rightarrow a^2 \notin L(A)$

$\qquad\qquad\qquad\qquad \vdash (q_1, \varepsilon)$ mais $q_1 \notin F \Rightarrow a^2 \notin L(A)$

$\qquad \vdash (q_1, a)$ ∃ un blocage. $\Rightarrow a^2 \notin L(A)$

$\boxed{\text{donc } a^2 \notin L(A)}$

3/ $(q_0, aba) \vdash (q_0, ba) \vdash (q_2, a)$ ∃ un blocage $\Rightarrow aba \notin L(A)$

$\qquad \vdash (q_1, ba) \vdash (q_2, a)$ ∃ un blg $\Rightarrow aba \notin L(A)$

$\qquad\qquad\qquad\quad \vdash (q_1, a)$

$\boxed{\text{donc } aba \notin L(A)}$

$L(A) = \{a^+ b^+, a^* b\}$

$= \{\{a\}\{a\}^*\{b\}\{b\}^*\} \cup \{\{a\}\cdot\{a\})^* \{b\}\}$

(32)

Remarque:

D'une manière générale, plus un automate → a le choix de transition pour lire un mot plus l'automate prendra du temps à reconnaître ce mot, d'où la nécessité de déterminiser des AEF indéterministes.

Théorème : pour tout AEF indéterministe A, il existe un AEF déterministe A' équivalent càd $\boxed{L(A) = L(A')}$

3.9 Algorithme de construire d'AEF déterministe à partir d'AEF indéterministe

① regrouper les états initiaux → (cet ensemble représente le nouveau état initial)

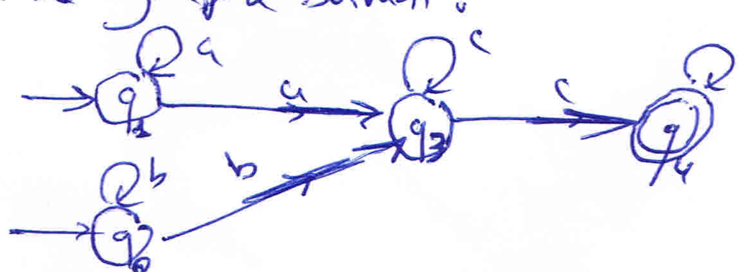
② rajouter dans la table de transition, tous les nouveaux états produits avec leurs transitions

③ recommencer $\boxed{2}$ jusqu'à ce qu'il n'y ait plus de nouvel état

ensuite ④ tous les états contenant au moins un état → final deviennent finaux.

⑤ Renuméroter alors les états.

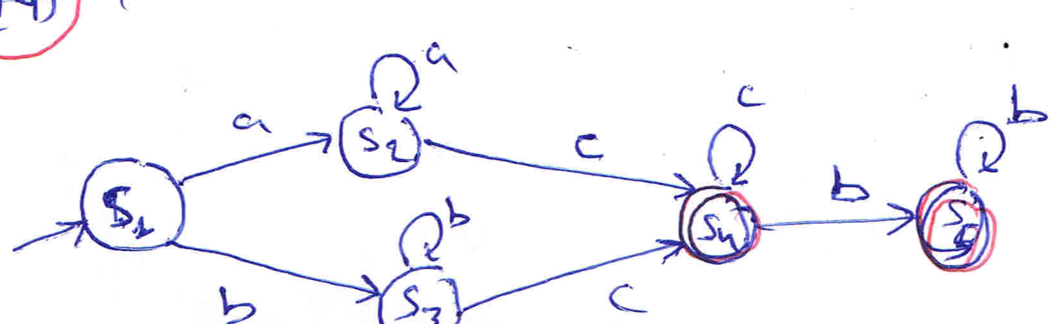
exemple : on reprend l'exemple d'AEF indéterministe défini par le graphe suivant :



Déterminiser cet automate ?

33

		a	b	c
S_1	$\{q_1, q_2\}$	$\{q_1, q_3\}$	$\{q_2, q_3\}$	—
S_2	$\{q_1, q_3\}$	$\{q_1, q_3\}$	—	$\{q_3, q_4\}$
S_3	$\{q_2, q_3\}$	—	$\{q_2, q_3\}$	$\{q_3, q_4\}$
S_4	$\{q_3, q_4\}$	—	q_4	$\{q_3, q_4\}$
S_5	$\{q_4\}$	—	$\{q_4\}$	—



Remarque : d'une manière générale, moins un automate contient d'états moins il prendra d'espace en mémoire de sauvegarde d'où la nécessité de minimiser les automates AEF déterministe puisque l'AEFD obtenu contient en général plus d'états que son AEFI.

3.10 Définitions

- un état est accessible, s'il y a un chemin d'un état initial vers lui.
- un état est co-accessible, s'il y a un chemin de cet état vers un état final.

Remarque : on peut retirer les états inaccessibles et non-coaccessibles puisque ils ne changent pas la représentation d'un langage

3.11 AEFD complet : un AEFD est dit complet si la relation de transition est complètement définie pour chaque couple $(q_i, x_i) \in Q \times X$. c.à.d. avec sa relation chaque chaque couple a exactement un état.

(34) • on complète un AEFD en lui ajoutant un état puits non final (état mort, état poubelle) cet état qui boucle sur lui même avec chaque symbole de X et vers lequel → arrivent toutes les transitions non définies (manquantes) (qui manquent)

c'à-d : au niveau de la table de transition, cela consiste à boucher les trous par un état puits non final qui boucle sur lui même avec chaque symbole de X.

Remarque : avec un automate AEFDC, chaque mot est entièrement lu

Dans ce cas, c'est l'état de sortie qui fait la reconnaissance

si cet état est un → état final alors le mot ∈

si cet état est un état → poubelle " " ∉.

3.12 algorithme de minimisation d'AEF déterministe

pour minimiser un AEF déterministe

1. éliminer tous les états inaccessibles et non co-accessibles
2. partitionner l'ensemble des états en deux classes :
   - classe des états finaux
   - classe des autres états.
3. • dans chaque classe obtenue d'une étape précédente, on regroupe les états équivalents (c'à-d : les états ayant un même comportement sur chaque symbole) en éclatant ainsi cette classe en d'autres classes de telle sorte que chacune contient des états équivalent.

• on répète → l'étape 3 jusqu'à ce qu'il n'y ait plus d'éclatement possible