Chapitre 3

Automate à états finis

3.1 Introduction aux automates à états finis

Nous avons vu au premier chapitre quelques notions des langages réguliers. Les automates que nous allons introduire ici sont des machines permettant de reconnaˆıtre exactement ces langages. De façon générale, un automate est une machine qui a des entrées et des sorties discrètes et qui réagit à une modification de ses entrées en changeant ses sorties. Formellement un automate fini peut être défini de la façon suivante :

3.1.1 Définition formelle

Un automate fini déterministe (AFD) A est un quintuplet $A = (Q, \Sigma, \delta, q_0, Q_m)$ où :

-Q est l'ensemble fini des états.

-Σ est un ensemble fini de symboles d'entrée (ensemble des événements), appelé alphabet d'entrée.

-δ est la fonction de transition d'états de $Q \times \Sigma \longrightarrow Q$ qui associe un état d'arrivée q_k,à un état de départ q_i et à un symbole d'entrée σ_j : $\delta(q_i, \sigma_j) = q_k$.

-$q_0 \in Q$ est l'état initial.

-$Q_m \subseteq Q$ est l'ensemble des états marqués (états finaux).

3.1.2 Représentation graphique

Un automate à états finis peut être décrit par son graphe de transition d'états. Dans ce graphe :

-Les états sont symbolisés par des cercles : l'état initial est figuré par un cercle avec une flèche entrante et les états finaux sont indiqués par des doubles cercles.

-La fonction de transition d'états est représentée par des arcs orientés associés aux événements de Σ.

Exemple 1. *La figure (1) représente le graphe de transition d'états d'un automate fini déterministe A.*

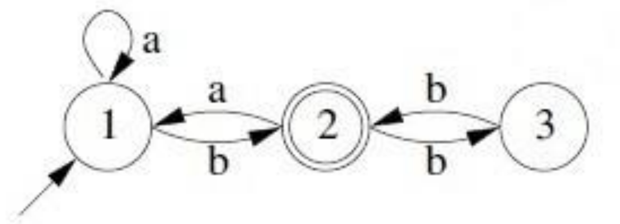


FIGURE 1 – Graphe de transition d'états d'un automate fini

Nous pouvons identifier $Q = \{1, 2, 3\}$, $\Sigma = \{a, b\}$ *et* $Q_m = \{1, 2\}$, $q_0 = 1$.
La fonction de transition d'état δ *est définie par :*

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

$\delta(1,b) = 2$ *est figurée par un arc orienté de l'état* 1 *à l'état* 2 *et étiqueté par le symbole* b.

On remarque aussi que dans le cas de ce modèle, la fonction de transition d'état est partielle, car elle n'est pas définie pour tout élément de produit cartésien $Q \times \Sigma$. *Par exemple* $\delta(3, a) = \emptyset$.
On peut représenter la fonction de transition δ *par le tableau suivant :*

	$\delta(q,a)$	$\delta(q,b)$
1	*1*	*2*
2	*1*	*3*
3	$\emptyset$	*2*

Aussi, on peut présenter un AFD par un tableau comme suit :

	$\delta'(q,a)$	$\delta'(q,b)$
$\rightarrow 1$	*1*	*2*
$\leftarrow 2$	*1*	*3*
3	$\emptyset$	*2*

La flèche entrante et les flèches sortantes désignent respectivement l'état initial et les états finaux.

Remarque : Il faut bien noter qu'on ne peut pas avoir dans un AFD deux ou plusieurs transitions avec la même étiquette en sortie d'un état.

2.1 Langage d'un automate à états finis déterministe

2.1.1 Langage généré

Soit $A = (Q, \Sigma, \delta, q_0, Q_m)$, un langage généré par l'automate à états finis A est constitué de l'ensemble de tous les mots s générés tels que la transition $\delta(q_0, s)$ est définie. Il est noté $L(A)$:

$$L(A) = \{s \in \Sigma^* \ |\delta(q_0, s) \ \text{ est définie}\}$$

En d'autres termes un mot $s \in L(A) \iff$ il existe une séquence de transition étiquetée par s et commençant en q_0.

2.1.2 Langage accepté (marqué)

Soit $A = (Q, \Sigma, \delta, q_0, Q_m)$, un langage accepté par l'automate à états finis A est constitué de l'ensemble des mots s générés, provoquant une évolution de l'état initial à un état marqué (final).

$$L_m(A) = \{s \in \Sigma^* \ |\delta(q_0, s) \in Q_m\}$$

Remarque : Le langage $L_m(A) \subseteq L(A)$ puisque $Q_m \subseteq Q$.

Exemple 2. *Si on poursuit l'exemple précédent, l'automate A de la figure (2.1) accepte le mot abbb car on a, partant de l'état initial, le parcours suivant :*

$$\delta(1, abbb) = 2 \in Q_m$$

Par contre, abb n'est pas accepté par A car :

$$\delta(1, abb) = 3 \notin Q_m$$

3.2 Propriétés

3.2.1 Accessibilité

Un état $q \in Q$ est dit accessible(ou réalisable) s'il existe une chaîne $s \in \Sigma^*$ telle que $q = \delta(q_0, s)$, c'est-a-dire que l'automate peut y accéder depuis l'état initial. Par extension, l'automate A est accessible si tout état $q \in Q$ est accessible.

3.2.2 Co-accessibilité

Un état $q \in Q$ est dit co-accessible (ou co-réalisable) s'il existe une chaîne $s \in \Sigma^*$ telle que $\delta(q, s) \in Q_m$, c'est-à-dire qu'à partir de cet état l'automate peut atteindre un état marqué. Par extension, l'automate A est co-accessible si tout état $q \in Q$ est co-accessible.

Exemple 3. *Soient les deux automates A, B :*

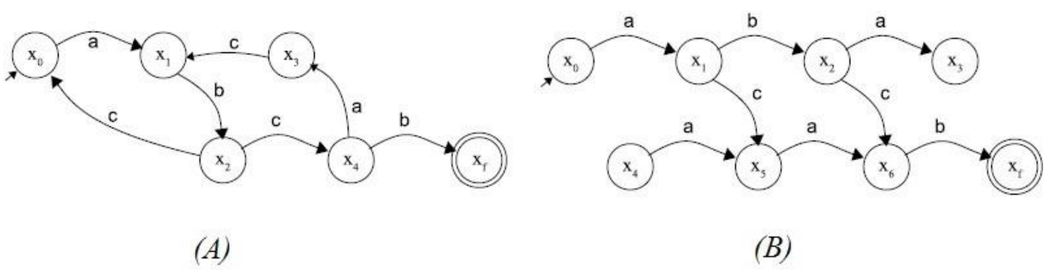


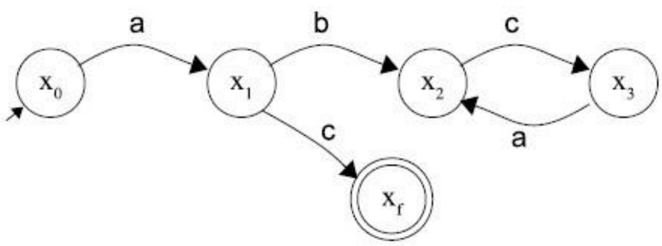
FIGURE 3 – A accessible et co-accessible, B non accessible et non co-accessible

Dans l'automate A tous les états sont accessibles et co-accessibles. C'est pourquoi cet automate et accessible et co-accessible.
Dans l'automate B, les états ($x_1, x_2, x_3, x_5, x_6, x_f$) sont des états accessibles mais x_4 n'est pas accessible. Aussi, tous les états de l'automate B sont co-accessibles sauf x_3 qui n'est pas co-accessible. Donc, l'automate B est ni accessible, ni co-accessible.

3.2.3 Automate bloquant / non bloquant

Un état $q \in Q$ est dit bloquant s'il est accessible mais pas co-accessible ; Un automate A est dit non bloquant si tous ses états sont non bloquants.

Exemple 4. *Soit l'automate suivant :*



L'automate présente un automate bloquant car les états x_2 et x_3 sont des états accessibles mais pas co-accessibles.

3.2.4 Automate complet

Un automate A est dit complet si la fonction de transition est définie pour tout élément de $Q \times \Sigma$ i.e. pour chaque état $q \in Q$, il part un arc étiqueté par chacune des éléments de Σ.

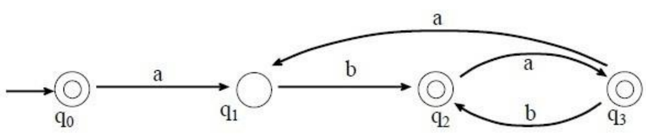
On notera que si A est un automate complet alors $L(A) = \Sigma^*$.

Par exemple, l'automate présenté dans l'exemple précédant, n'est pas complet car la fonc- tion de transition $\delta(x_0, c)$ n'est pas définie.

Il est toujours possible de compléter un automate si on complète la fonction de transition : i.e (rendre δ complète) . Cela se fait en ajoutant un nouvel état q_d à Q, et toutes les parties de la fonction de transition qui ne sont pas définies dans l'automate sont affectés à l'état q_d. Formellement, on définit la fonction de transition δ^* telle que :

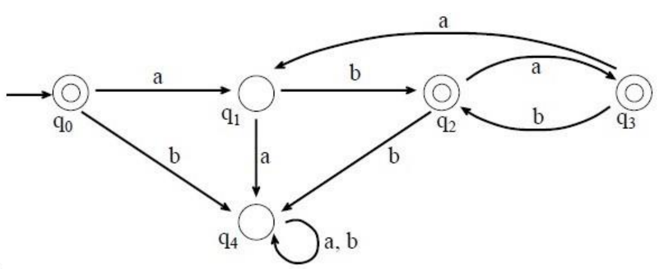
$$\delta^*(q,e) = \begin{cases} \delta(q,e) & si \quad \delta(q,e) \quad est\ définie \\ q_d & si \quad \delta(q,e) \quad n'est\ pas\ définie \end{cases}$$

Exemple 5. *L'automate dans la figure suivante n'est pas complet car par exemple $\delta(q_0, b)$ n'est pas définie.*



Automate non complet

En complétant l'automate non complet, on obtient l'automate suivant :



Automate complet

Remarques :

- Un automate complet A accepte le même langage de son automate non complet B mais il ne génère pas que le même langage ($L(A) = \Sigma^*$).
- Les propriétés des deux automates sont différents : en particulier, A est certainement bloquant, vu que le nouvel état n'est pas co-accessible.

3.3 Automate à états finis non déterministe (AFND)

3.3.1 Définition

Un automate fini déterministe A est un quintuplet $A = (Q, \Sigma, \Delta, Q_0, Q_m)$ où :

-Q est l'ensemble fini des états.
-Σ est un ensemble fini de symboles d'entrée (ensemble des événements), appelé alphabet d'entrée.
-Δ est la fonction de transition d'états de $Q \times \Sigma \cup \{\varepsilon\} \rightarrow 2^Q$ qui associe un état d'arrivée
q_k, à un état de départ q_i et à un symbole d'entrée σ_j : $\Delta(q_i, \sigma_j) = q_k$.
-$Q_0 \subseteq Q$ est l'ensemble des états initiaux.
-$Q_m \subseteq Q$ est l'ensemble des états marqués (états finaux).

Un automate fini non-déterministe (AFND) peut représenter par deux types :

- **Automate fini non-déterministe avec ε-transition (AFN-ε) :** est un automate tel que dans un état donné, il peut y avoir des événements étiquetés avec le mot vide ε.
- **Automate fini non-déterministe sans ε-transition (AFN) :** est un automate tel que dans un état donné, il peut y avoir plusieurs événements avec le même symbole mais aucun événement étiqueté avec le mot vide ε.

Exemple 5. *Les deux automates présentés dans la figure suivante sont non déterministe. L'automate* (*a*) *est un AFN et* (*b*) *est un AFN-ε*

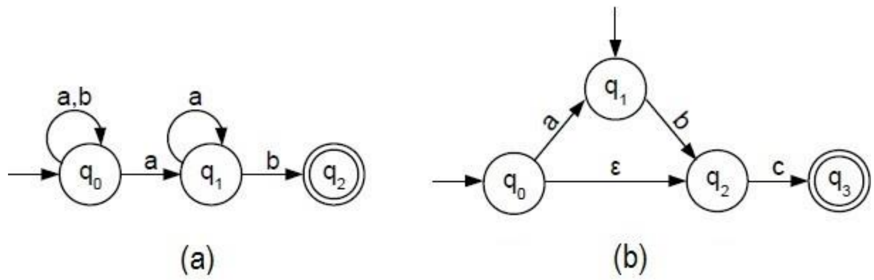


FIGURE 2.3 – Automates non déterministes

3.4 Différence entre AFD et AFND :

On résume la différence entre un AFD et un AFND dans le tableau suivant :

AFD	AFND
$A = (Q, \Sigma, \delta, q_0, Q_m)$	$A = (Q, \Sigma, \Delta, Q_0, Q_m)$
Un seul état initial	Un ou plusieurs états initiaux
Pour chaque lettre et chaque état on peut avoir une seule transitions sortante.	Pour un état et une lettre on peut avoir plusieurs transitions sortantes.
Pas de transition vide ε	On peut avoir des transitions vides ε

Remarques : Un AFD est un cas particulier d'un AFND. Donc tout langage reconnu par un AFD est reconnu par un AFND.

3.5 Langage d'un automate à états finis non déterministe

3.5.1 Langage généré

Soit $A = (Q, \Sigma, \Delta, Q_0, Q_m)$, un langage généré par AFND A est constitué de l'ensemble de tous les mots s générés tels que la transition $\Delta(q_0, s)$ est définie. Il est noté $L(A)$:

$$L(A) = \{s \in \Sigma^* \ |\Delta(q_0, s) \ \ \textit{est définie}\}$$

3.5.2 Langage accepté (marqué)

Soit $A = (Q, \Sigma, \Delta, Q_0, Q_m)$, un langage accepté par AFND A est constitué de l'ensemble des mots s générés, provoquant une évolution de l'état initial à un état marqué (final).

$$L_m(A) = \{s \in \Sigma^* \ |\delta(q_0, s) \in Q_m\} \subseteq L(A)$$

3.5.3 Propriétés

Toutes les propriétés d'un AFD sont valables pour un AFND. On résume les propriétés d'un AFND dans la définition suivante :

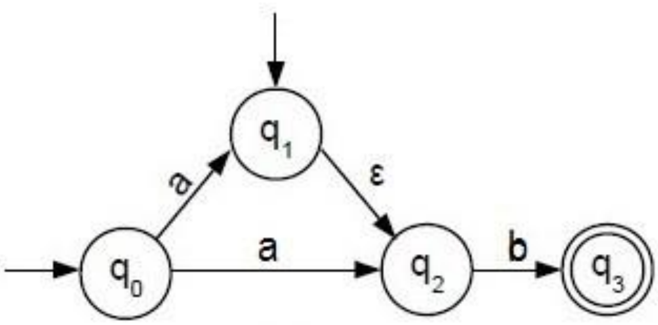
Un AFND $A = (Q, \Sigma, \Delta, Q_0, Q_m)$ est dit :

-Accessible si tout état $q \in Q$ est accessible.

-Co-accessible si tout état $q \in Q$ est co-accessible.

-Non bloquant si tous ses états sont non bloquants.

Exemple 6. *Dans l'automate suivant,*



– *Le mot $s = ab$ peut être généré par deux séquence :*

$$q_0 \longrightarrow q_2 \longrightarrow q_3$$

$$q_0 \longrightarrow q_1 \longrightarrow q_2 \longrightarrow q_3$$

le deux séquences conduisent à l'état final et donc le mot $s = ab$ est accepté.

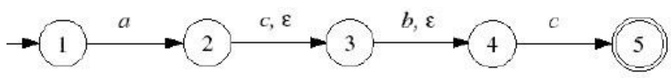
– *L'automate présente un AFND accessible, co-accessible. et non bloquant.*

3.6 Transformation d'un AFN-ε en AFN

Il existe un algorithme dit algorithme d'élimination des ε qui calcule pour tout automate AFN-ε A un automate AFN A^J tel que $L(A) = L(A^J)$.

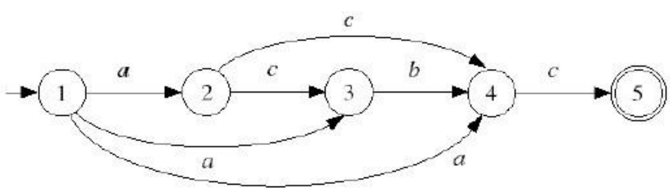
Le principe de l'algorithme consiste à remplacer chaque chemin commençant par une transition ε par une nouvelle transition qui décrit ce chemin.

Exemple 7. *Soit le AFN-ε suivant :*



Langage reconnu est $L = \{acbc, acc, abc, ac\}$.

On ajoute des nouvelles transitions et on supprime des transitions ε, on obtient l'automate AFN suivant :



Langage reconnu est $L = \{acbc, acc, abc, ac\}$.

3.7 Equivalence entre AFD et AFND

Un automate fini déterministe est aussi non-déterministe. Donc tout langage reconnu par un automate fini déterministe est reconnu par un automate fini non-déterministe. Plus surprenant, la réciproque est aussi vraie.
Avant de donner les méthodes de déterminisation d'un AFND, on doit connaitre la notion de la notion de ε-fermeture.

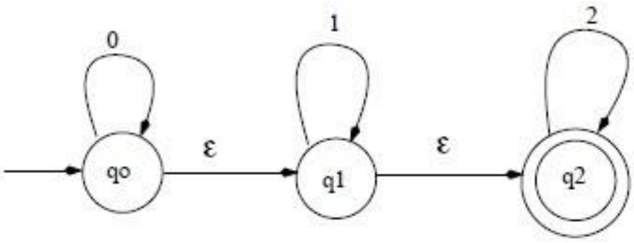
3.7.1 La notion de ε-fermeture

La définition de l'ε-fermeture d'un état q, que l'on note par $E(q)$, et de son extension sont utiles pour l'élimination des ε-transitions.
Etant donné un état $q \in Q$, on définit l'ε-fermeture de q par :

$$E(q) = \{q^J \mid \Delta(q, \varepsilon) = q^J\}$$

En d'autres termes $E(q)$ est l'ensemble de tous les états dans Q que l'on pourrait atteindre à partir de q en effectuant 0, 1 ou plusieurs ε-transitions.

Exemple 8. *L'ε-fermeture de chaque état de l'automate suivant :*



$E(q_0) = \{q_0, q_1, q_2\}$
$E(q_1) = \{q_1, q_2\}$
$E(q_2) = \{q_2\}$
$E(\{q_1, q_2\}) = E(q_1) \cup E(q_2) = \{q_1, q_2\} \cup \{q_2\} = \{q_1, q_2\}$

3.7.2 Déterminisation d'un AFN (sans ε-transition)

Considérons un AFN $A = (Q, \Sigma, \Delta, q_0, Q_m)$ et construisons un AFD $A' = (Q', \Sigma, \Delta', q'_0, Q'_m)$ qui reconnaît exactement le même langage ($L_m(A) = L_m(A')$ et $L(A) = L(A')$).

-l'ensemble d'événements Σ reste le même avec ε / Σ.

-Q' est constitué de tous les sous-ensembles de Q.

Par exemple si $Q = \{q_0, q_1\}$, alors $Q' = \{\{q_0, \{q_1\}, \{q_0, q_1\}\}$.

-Q'_m est l'ensemble des sous-ensembles de Q qui contiennent au moins un élément de Q_m.

- Étant donné un sous ensemble S de Q et $a \in \Sigma$, la fonction Δ' est définie comme suite :

$$\Delta'(S, a) = \cup_{q \in S} \Delta(q, a)$$

On va expliquer les étapes de déterminisation d'un AFN à travers l'exemple suivant.

Exemple 9. *Soit AFN suivant :*

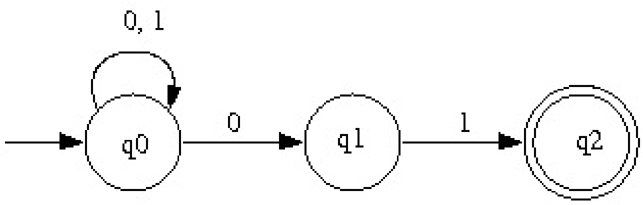


FIGURE 4 – Automate non déterministe simple

Cet automate a trois états $\{q_0, q_1, q_2\}$

– On calcule la fonction de transition pour chaque sous ensemble (on commence par l'état initial), on obtient le tableau suivant :

Q^J	$\Delta^J(q,0)$	$\Delta^J(q,1)$
$\{q_0\}$	$\{q_0.q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$	$\{q_0.q_1\}$	$\{q_0, q_2\}$
$\{q_0,q_2\}$	$\{q_0.q_1\}$	$\{q_0\}$

- *On sélectionne l'ensemble des états Q^J de AFD à partir de la première colonne du tableau. Donc $Q^J = \{q_0, \{q_0.q_1\}, \{q_0,q_2\}\}$.*
- *L'état initial est $\{q_0\}$ et l'état marqué est $\{q_0,q_2\}$.*

Finalement, AFD equivalent de AFN est donné comme suit :

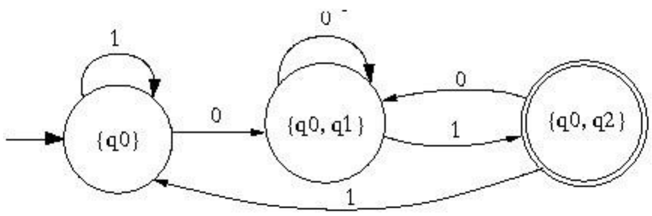


FIGURE 5 – Automate déterministe équivalent

Dans l'exemple précédent, le nombre des états de l'AFN égale à 3, et le nombre des états de l'AFD égale aussi 3. En général, on ne peut pas dire a priori lequel des deux automates a le plus grand nombres d'états. Le seul résultat général est le suivant :

Proposition : Soit A un AFN avec ensemble d'états Q. le nombre d'états dans Q^J d'un AFD A^J équivalent est donné par :

$$|Q^J| \leq 2^{|Q|} - 1$$

3.7.3 Déterminisation d'un AFN-ε

Considérons un AFN-ε $A = (Q, \Sigma, \Delta, q_0, Q_m)$ et construisons un AFD $A^J = (Q^J, \Sigma, \Delta^J, q_0^J, Q_m^J)$ qui reconnaît exactement le même langage ($L_m(A) = L_m(A^J)$ et $L(A) = L(A^J)$).

-l'ensemble d'événements Σ reste le même avec ε / Σ.

-Q^J est constitué de tous les sous-ensembles de Q.

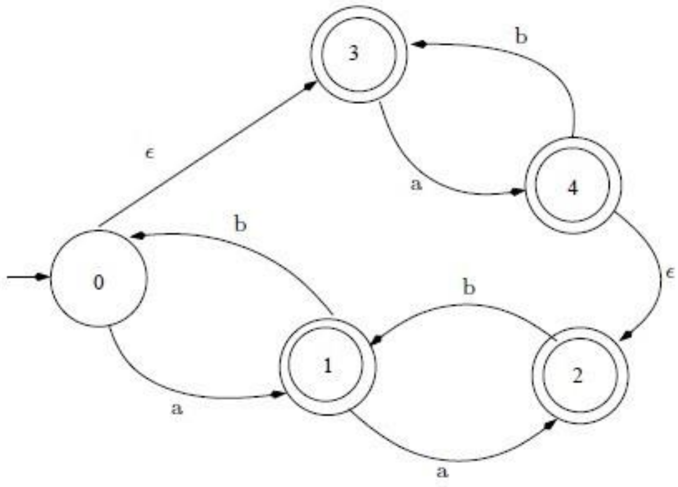
-Q_m^J est l'ensemble des sous-ensembles de Q qui contiennent au moins un élément deQ_m.

-L'état initial $q_0 = E(q_0)$.

Étant donné un sous ensemble S de Q et $a \in \Sigma$, la fonction Δ^J est définie comme suite :

$$\Delta^J(S,a) = \bigcup_{q \in S} \Delta(q), a)$$

Exemple 10. *Soit AFN-ε :*



L'état initial de l'automate est définie par ε-fermeture de l'état 0*, donc l'état initial de AFD est :* $E(0) = \{0, 3\}$.

La fonction de transition est définie par :

Q'	$\Delta'(q, a)$	$\Delta'(q, b)$
$\{0, 3\}$	$\{1, 2, 4\}$	$\emptyset$
$\{1, 2, 4\}$	$\{2\}$	$\{0, 1, 3\}$
$\{2\}$	$\emptyset$	$\{1\}$
$\{0, 1, 3\}$	$\{1, 2, 4\}$	$\{0, 3\}$
$\{1\}$	$\{2\}$	$\{0, 3\}$

Comme l'ensembles des états finaux de AFN-ε est $Q_m = \{1,2,3,4\}$*, alors l'ensemble des états finaux de AFD est* $Q'_m = \{\{1\}, \{2\}, \{0,3\}, \{0,1,3\}, \{1,2,4\}\}$.

AFD equivalent est donné comme suit :