Un automate d'état finis est composé d'une bande finie en entrée, d'un organe de commande et d'une tête de lecture (figure 1). L'automate d'états finis n'a pas de mémoire auxiliaire.

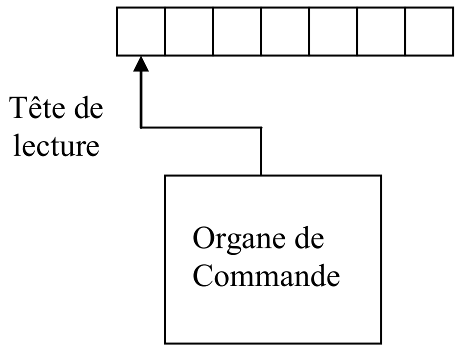


Figure 1: **Automate d'états finis**

1. Automate d'états finis simple déterministe

1.1 Définition: Un automate d'états finis simple déterministe, noté A<X, S, S_0, F, II>, est caractérisé par 5 paramètres que l'on définit comme suit :

- X est l'alphabet d'entrée,
- S est un ensemble fini d'états
- S_0 est l'état initial, $S_0 \in S$,
- F est l'ensemble des états finaux, $F \subseteq S$,
- II est l'ensemble des instructions (fonction de transition), II : S x X $\rightarrow$ S.

Les éléments de II (ensemble d'instructions) sont des triplets (S_i, x_i, S_j) qu'on lit de la manière suivante : l'automate étant à l'état $S_i \in S$ et lisant une lettre $x_i \in X$ en entrée, passe à l'état $S_j \in S$.

1.2 Représentation d'un automate d'états finis simple déterministe

Pour décrire un automate d'états finis déterministe, il faut spécifier tous ses paramètres, en particulier l'ensemble des instructions que l'on peut représenter par une table de transition ou par un multigraphe.

1.2.1 Table de transition (représentation matricielle)

Pour décrire l'ensemble des instructions II, on peut utiliser une matrice (figure 2).

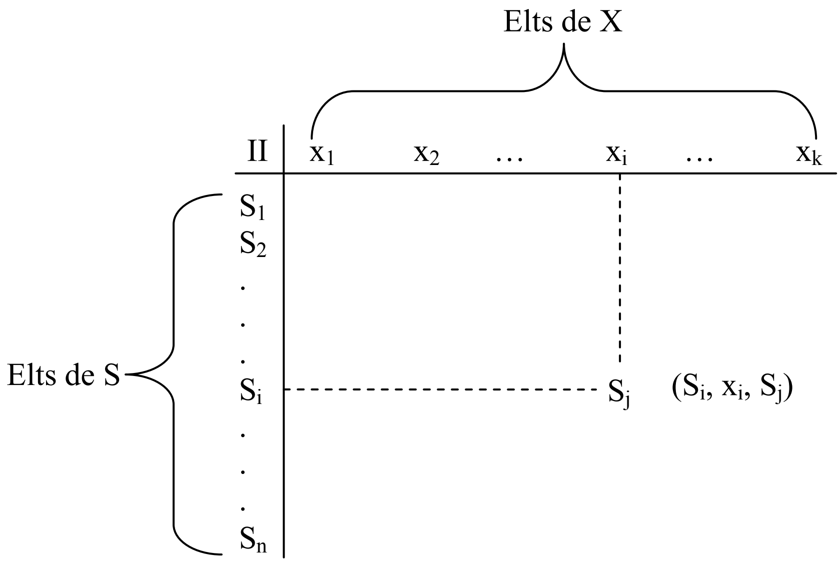


Figure 2: Représentation matricielle de l'ensemble des instructions II.

Une entrée est un état Sj auquel l'automate arrive après avoir été à l'état S_i et avoir lu, en entrée, la lettre x_i.

1.2.2. Représentation graphique

Un automate d'états finis déterministe peut-être représenté par un multigraphe orienté où chaque sommet est un état de S et les arcs sont étiquetés par des lettres de X. Une instruction (S_i, x_i, S_j) est représentée comme suit (figure 3) :

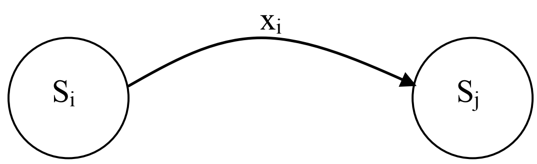


Figure 3 : **Représentation graphique de l'instruction (S_i, x_i, S_j).**

L'état initial et les états finaux sont représentés respectivement comme dans la figure 4.a et 4.b.

Figure 4: **a. Etat Initial** **b. Etat Final.**

Exemple 10 : Automate d’états finis simple déterministe

Soit A<X, S, S_0, F, II> un automate d’état finis simple déterministe où X = {a, b}, S={S_0,S_1,S_2,S_3,S_4,S_5}, F = {S_0, S_2, S_4 } et II = {(S_0, a, S_0), (S_0, b, S_1), (S_1, b, S_0), (S_1, a, S_2), (S_2, a, S_2), (S_3, b, S_0), (S_4, a, S_4), (S_4, b, S_5)}

Représentation graphique de l’automate A :

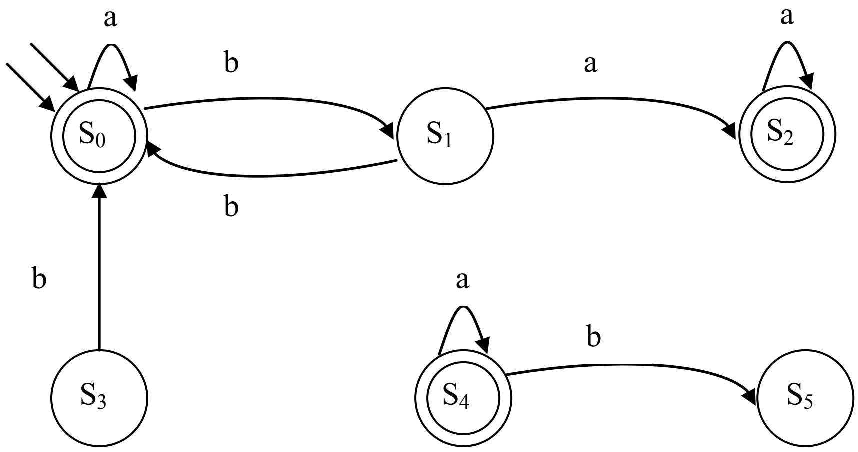


Figure 5 : Automate d’états finis simple déterministe.

Définition: Soit A<X, S, S_0, F, II> un automate d’états finis. A est simple si le passage d’un état à un autre se fait à la lecture d’une lettre de X.

L’automate d’états finis de la figure 5 est simple.

Définition : Soit A<X, S, S_0, F, II> un automate d’états finis simple. A est déterministe si et seulement si :

$$\forall\, S_i, S_j, S_k \in S, \text{ et } \forall\, w_i \in X \text{ Si } (S_i, w_i, S_j) \in S \text{ et } (S_i, w_i, S_k) \in S \text{ alors } S_j = S_k.$$

L’automate d’états finis de la figure 5 est déterministe.

2 Langage reconnu par un automate d’états finis

Définition: Soit A<X, S, S_0, F, II> un automate d’états finis simple. La lecture d’une lettre w_i $\in$ X en entrée, fait passer l’automate A de l’état S_i à l’état S_j si et seulement si (S_i, w_i, S_j) $\in$ II. Cette lecture est notée $S_i \;\Big|\!\frac{w_i}{A}\; S_j$.

Définition: Soit A<X, S, S_0, F, II> un automate d'états finis simple et déterministe. La lecture d'un mot $w \in X^*$ ($w = w_1\ w_2 \ldots\ w_n$) fait passer l'automate A de l'état $S_i \in S$ à l'état $S_j \in S$, que l'on note $S_i \underset{A}{\overset{w}{\vdash}} S_j$, s'il existe n-1 états $S_1, S_2, \ldots, S_{n-1}$ tels que :

$S_i \underset{A}{\overset{w_1}{\vdash}} S_1 \underset{A}{\overset{w_2}{\vdash}} S_2 \ldots S_{n-1} \underset{A}{\overset{w_n}{\vdash}} S_j$. w est appelé un chemin.

Exemple 11 : le mot w= bab est un chemin. Il fait passer l'automate de l'état S_1 à l'état S_1

$$S_1 \underset{A}{\overset{b}{\vdash}} S_0 \underset{A}{\overset{a}{\vdash}} S_0 \underset{A}{\overset{b}{\vdash}} S_1$$

On note $S_i \underset{A}{\overset{\varepsilon}{\vdash}} S_i$ le chemin vide.

Définition : Soit A<X, S, S_0, F, II> un automate d'états finis simple. Un mot $w \in X^*$ ($w = w_1\ w_2 \ldots\ w_n$) est reconnu par l'automate A s'il fait passer l'automate de l'état S_0 à un état $S_k \in F$: $S_0 \underset{A}{\overset{w}{\vdash}} S_k$.

w est appelé un chemin réussi.

Exemple 12 : le mot w= baa est un chemin réussi. Il fait passer l'automate de l'état S_1 à l'état S_1.

$$S_0 \underset{A}{\overset{b}{\vdash}} S_1 \underset{A}{\overset{a}{\vdash}} S_2 \underset{A}{\overset{a}{\vdash}} S_2$$

Définition : Soit A<X, S, S_0, F, II> un automate d'états finis simple et déterministe. Le langage reconnu par A, que l'on note L(A), est l'ensemble des mots reconnus par A :

$$L(A) = \{w \in X^* \ /\ \exists\ S_k \in F \text{ tq } S_0 \underset{A}{\overset{w}{\vdash}} S_k \}.$$

Définition : Soient A<X, S, S_0, F, II> et A'<X', S', S'_0, F', II'> deux automates d'états finis. On dit que A et A' sont équivalents si et seulement si L(A) = L(A').

3 Réduction d'un automate simple

Définition : Soit A<X, S, S_0, F, II> un automate d'états finis simple. On dit qu'un état $S_i \in S$ est accessible dans l'automate A s'il existe un mot $w \in X^*$ tel que $S_0 \underset{A}{\overset{w}{\vdash}} S_i$.

Exemple 13 : L’état S_2 est accessible dans l’automate de la figure 5. Si on prend w = ba, on a bien $S_0 \underset{A}{\overset{ba}{\vdash}} S_i$.

Définition : Soit *A*<X, S, S_0, F, II> un automate d’états finis simple. On dit qu’un état $S_j \in S$ est co-accessible s’il existe un mot $w \in X^*$ et $S_k \in F$ tels que $S_j \underset{A}{\overset{w}{\vdash}} S_k$.

Exemple 1 : L’état S_1, de l’automate de la figure 5, est co-accessible. Si on prend w = ba, on a bien $S_1 \underset{A}{\overset{aa}{\vdash}} S_2$.

Proposition : si *A*<X, S, S_0, F, II> est un automate d’états finis simple, et si *A’*<X,S’,S_0,F’,II’> est l’automate obtenu à partir de *A* en éliminant les états non accessibles et non co-accessibles alors L(*A*) = L(*A’*). On dit que *A* et *A*’ sont équivalents.

__Exercice 1.1 :__ **Démontrer la proposition 1.** Cette démonstration se décompose en deux parties. La première consiste à donner l’algorithme de construction de *A’* à partier de A. Puis de montrer que L(*A*) = L(*A’*).

Exemple 14 : Automate réduit

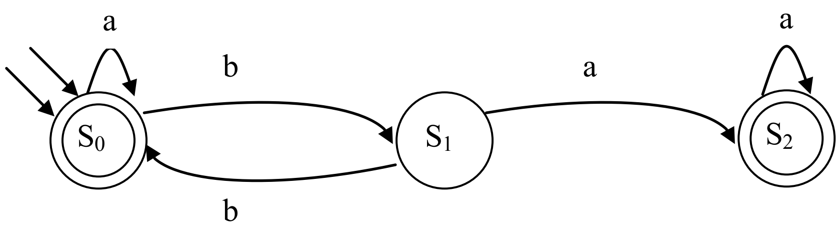


Figure 6 : Automate d’états finis simple déterministe réduit.

La figure 6 représente l’automate réduit équivalent à l’automate de la figure 5.

Automate d’états finis simple déterministe complet

On dit qu’un automate d’états finis simple déterministe *A*<X, S, S_0, F, II> est complet si et seulement si :

$$\forall S_i \in S \text{ et } \forall w_i \in X, \exists S_j \in S \text{ tels que : } S_i \underset{A}{\overset{w_i}{\vdash}} S_j.$$

Proposition : Pour tout automate d’états finis simple déterministe A_{sd}<X, S, S_0, F, II>, il existe un automate d’EF simple déterministe complet A_{sdc}<X, S’, S_0, F, II’> équivalent à A_{sd}.

__Exercice :__ Donner la procédure qui permet de construire un automate simple déterministe complet *A’* à partir d’un automate simple déterministe *A*.

4. Automate d'états finis simple non déterministe

Définition 1 : Soit A_s<X, S, S_0, F, II> un automate d'états finis simple. On dit que A_s est non déterministe s'il existe au moins un w $\in$ X* qui admet 2 lectures différentes dans A_s.

Définition 2 : Un automate d'états finis simple non déterministe est un quintuplet A_{snd}<X, S, S_{in}, F, II> où :

- X est l'alphabet d'entrée,
- S est un ensemble fini d'états de l'automate,
- S_{in} est l'ensemble des états initiaux ($S_{in} \subseteq S$),
- F est l'ensemble des états finaux ($F \subseteq S$),
- II est l'ensemble des instructions (fonction de transition).

I $\quad$ $I : X \times S \rightarrow P(S)$

$(S_i, w_i) \rightarrow \{ S_j \in S_i \quad \underset{A_{snd}}{\overset{w_i}{\vdash\!\!\!\longrightarrow}} \quad S_j \}$

Remarque : Dans ce qui suit, nous supposerons que l'automate non déterministe n'a qu'un seul état initial.

Exemple 15: Soit A_{snd}<{a, b}, {S_0, S_1, S_2}, S_0, { S_2}, II> l'automate suivant:

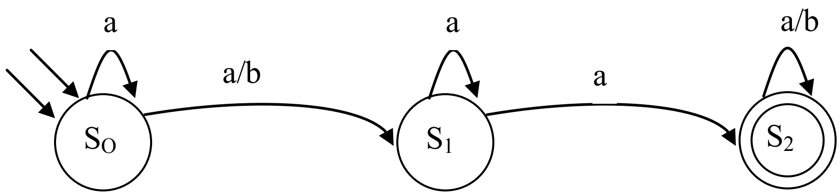


Figure 7:**Automate non déterministe.**

Cet automate n'est pas déterministe :

Selon la définition 1, On peut trouver un mot w = aaa pour lequel il existe au moins 2 lectures différentes (2 chemins différents). En fait, il existe plusieurs lectures possibles pour aaa. Quelques lectures sont données ci-dessous:

$$S_0 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_0 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_0 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_0$$

$$S_0 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_0 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_1 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_1$$

$$S_0 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_1 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_1 \underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}} S_2$$

Selon la définition 2, il existe au moins un état à partir duquel la lecture d'une lettre de X peut faire passer l'automate à deux ou plusieurs états différents. L'état S_0 répond à cette définition :

$$S_0 \;\underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}}\; S_0 \quad \text{et} \quad S_0 \;\underset{A_{snd}}{\overset{a}{\vdash\!\!\!\longrightarrow}}\; S_1$$

A partir de S_0, l'automate peut lire la lettre a est rester à l'état S_0 ou passer à l'état S_1.

Exemple 16: Soit A_{snd}<{0, 1}, {S_0, S_1, S_2}, S_0, { S_2}, II> l'automate suivant:

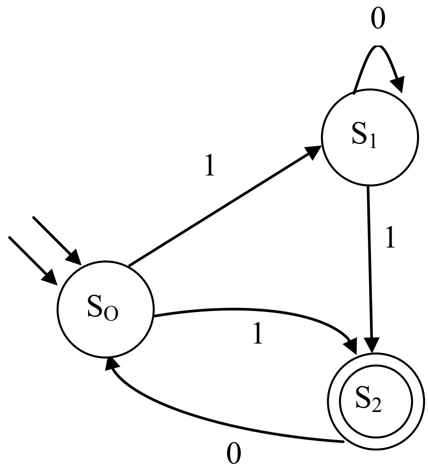


Figure 8. : Automate non déterministe

Voyons maintenant le comportement de cet automate à travers la lecture du mot w=1011.

Il y a trois chemins possibles pour la lecture du mot 101, deux chemins mènent à S_2 et un à S_1 (Figure 9).

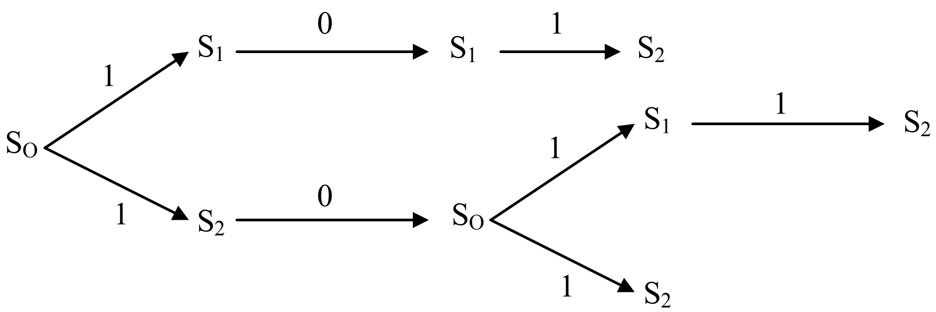


Figure 9 : **Lecture du mot 1011 par l'automate *A_{snd}*.**

Lorsque l'automate arrive à l'état S_2, il lui reste à lire la dernière lettre 1. A_{snd} se bloque car il n'existe aucune instruction de type (S_2, 1, S_i) dans II. Cependant, si A_{snd} se trouve à l'état S_1 et lit 1, il se déplace vers l'état final S_2. Le mot est alors accepté par l'automate. Il existe donc un chemin réussi pour ce mot.

4.1 Equivalence des automates d'états finis déterministe et non déterministes

Pour déterminer si un mot w est accepté par une automate simple non déterministe A_{snd}, il faut trouver au moins un chemin réussi pour ce mot dans A_{snd}. Cela revient à faire faire plusieurs lectures de ce mot par A_{snd} jusqu'à trouver un chemin réussi s'il existe (figure 9).

Pour construire un automate d'états finis simple déterministe A_{sd}, on peut simuler le comportement de l'automate d'états finis simple non déterministe. On peut ainsi supposer que l'automate déterministe équivalent passe par des combinaisons d'états (ensemble d'états), à chaque instant.

Exemple 17 : l'exemple de la figure 9 peut être représenté comme suit :

$$\{S_0\} \xrightarrow{1} \{S_1, S_2\} \xrightarrow{0} \{S_0, S_1\} \xrightarrow{1} \{S_1, S_2\} \xrightarrow{1} \{S_2\}$$

Le sous-ensemble d'états vers lequel un mot w mène l'automate à partir de son état initial est appelé ensemble d'états accessibles pour ce mot. Les états de $\{S_1, S_2\}$ sont accessibles pour w = 101. En appliquant le concept de sous ensemble d'états accessibles, on peut décrire le comportement de l'automate sur des mots de X^* :

Initialement, A_{snd} se trouve à l'état S_0 puis à la lecture de 1, il peut se déplacer soit à S_1 ou S_2. Si A_{snd} se trouve à l'état S_0 et lit 0, l'automate se bloque :

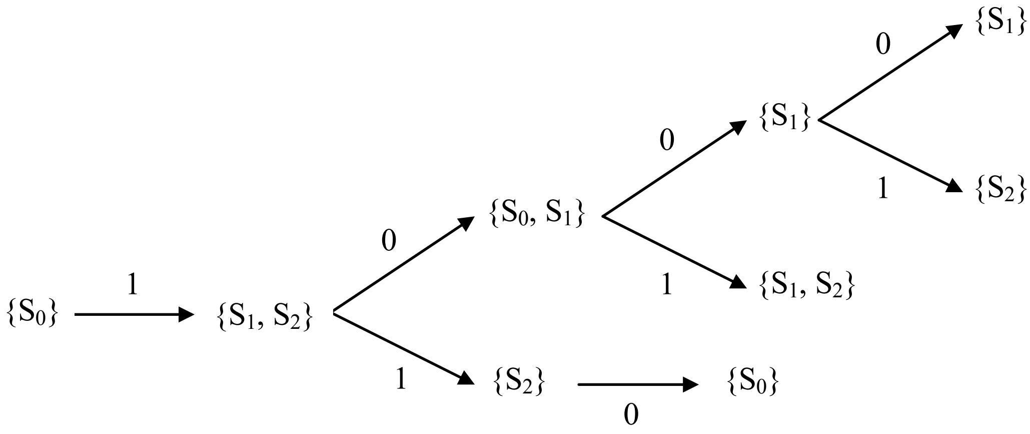
$$\{S_0\} \xrightarrow{1} \{S_1, S_2\}$$

Nous pouvons considérer que l'automate déterministe A_{sd} à la lecture de 1 vas se déplacer vers un état $\{S_1, S_2\}$.

Si A_{snd} se trouve à l'état S_1 ou S_2 et lit 0, l'automate se déplace soit à l'état S_0 ou S_1 (il se déplace vers S_0 si l'état courant est S_2 et vers S_1 si l'état courant est S_1. S'il lit 1, l'automate se déplace à l'état S_2, seule possibilité :

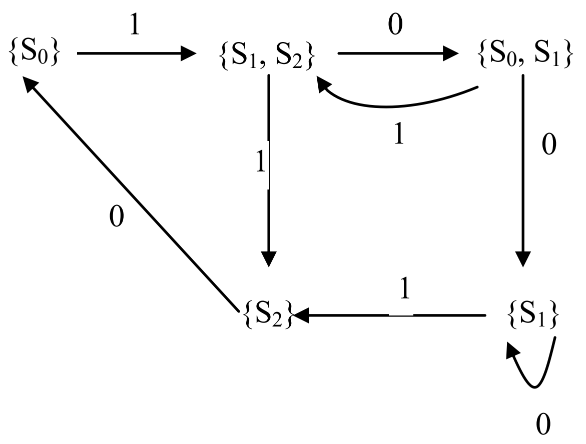
$$\begin{tikzcd} & & \{S_0, S_1\} \\ \{S_0\} \arrow[r, "1"] & \{S_1, S_2\} \arrow[ur, "0"] \arrow[dr, "1"'] & \\ & & \{S_2\} \end{tikzcd}$$

En appliquant cette procédure, on obtient le diagramme suivant :



Nous remarquons que certains sous ensembles se répètent. Nous allons les fusionner, ce qui donne le diagramme de la figure 10.a. Ce diagramme défini l’ensemble des instructions de l’automate déterministe (figure 10.b).

(a)



(b)

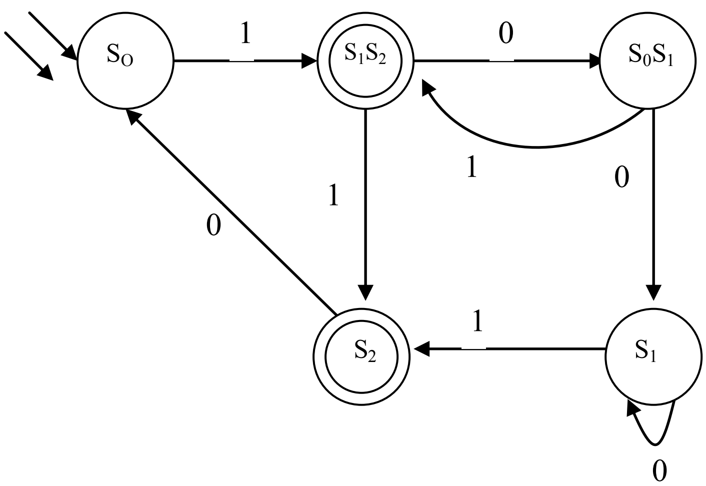


Figure 10 : Construction d’un automate d’états finis déterministe.

L'exemple 17 donne les étapes de la construction d'un automate simple déterministe à partir d'un automate simple non déterministe.

Soit A_{snd}<X, S, S_0, F, II> un automate d'états finis simple non déterministe et soit A_{sd}<X,S',S_0,F',II'> l'automate simple déterministe que l'on veut construire tel que L(A_{snd})=L(A_{sd}). Si A_{snd} lit le mot w $\in$ X^*, l'automate peut se déplacer à l'un des états possibles. Soit $S_{[w]}$ l'ensemble de ces états:

$$S_{[w]} = \{S_i \in S / \quad S_0 \; \underset{A_{sn}}{\overset{w}{\vdash\!\!-\!\!-}} \; S_i \}$$

$S_{[w]}$ contient tous les états accessibles pour le mot w dans A_{snd}. L'état initial de A_{sd} est défini comme suit : $S_{[\varepsilon]} = \{S_0\}$.

Pour chaque mot w=y.z, on peut exprimer $S_{[y.z]}$ en fonction de $S_{[y]}$. Un état S_i est accessible pour y.z si et seulement si : $S_k \; \underset{A_{sd}}{\overset{z}{\vdash\!\!-\!\!-}} \; S_i$ avec $S_k \in S_{[y]}$.

$S_{[y.z]} = \{S_i \in S' / \; S_k \; \underset{A_{sd}}{\overset{z}{\vdash\!\!-\!\!-}} \; S_i$ pour des $S_k \in S_{[y]}\}$.

Le nombre d'ensembles accessibles distinct est fini puisque chaque ensemble est un sous ensemble S, S étant fini.

Si on fait correspondre à l'ensemble des états de l'automate A_{sd}, S', chaque ensemble accessible avec les transitions telles que définies dans l'exemple 2.3, nous obtenons un automate d'états finis simple déterministe. Les états finaux seront tous les sous-ensembles d'états de S' contenant un état final de A_{snd}.

Proposition : Pour tout automate d'états finis simple non déterministe A_{snd}<X, S, S_0, F, II>, il existe un automate simple déterministe A'_{sd}<X, S', S_0, F', II'> équivalent.

Démonstration : La preuve se divise en deux parties :

1. Construire l'automate simple déterministe A_{sd}<X, S', S_0, F', II'> à partir de l'automate d'états finis simple non déterministe A_{snd}<X, S, S_0, F, II>. Cette construction revient à définir chaque paramètre de A_{sd} à partir de ceux de A_{snd}.
2. Montrer que A_{snd}<X, S, S_0, F, II> et A'_{sd}<X, S', S_0, F', II'> sont équivalents (L(A_{snd})= L(A'_{sd})).

1. Construction de A'_{sd}<X, S', S_0, F', II'>

Nous pouvons résumer les détails de la construction comme suit :

a) $S' = \{S_{[w]} / w \in X^*\}$
b) $S_{[\varepsilon]}$ est l'état initial de l'automate A_{sd}
c) l'ensemble des états finaux de A_{sd}, F' est défini comme suit

$F' = \{ S_{[w]} \in S' / S_{[w]} \cap F \neq \emptyset \}$

d) Une transition de II' est le triplet $(S_{[w]}, x_i, S_{[w.xi]})$ où $S_{[w.xi]} = \{S_k / (S_i, x_i, S_k) \in II$ pour des $S_i \in S_{[w]}\}$

2. Démonstration de ($L(A_{snd}) = L(A_{sd})$)

Un état $S_{[w]}$ de l'automate A_{sd} est un état final si et seulement si $S_{[w]}$ contient un état final de l'automate A_{snd}. Pour montrer que $L(A_{snd}) = L(A_{sd})$, Il suffit de montrer que chaque mot w, lu par A_{snd}, mène l'automate A_{sd} de l'état initial $S_{[\varepsilon]}$ à l'état $S_{[w]}$. Cette démonstration se fera par récurrence sur la longueur de w.

Cas particulier : $|w| = 0 \Leftrightarrow w = \varepsilon$

$$S_{[\varepsilon]} \left|\frac{\varepsilon}{A_{sd}}\right. S_{[\varepsilon]}$$

Hypothèse de récurrence : Supposons que

$$S_{[\varepsilon]} \left|\frac{y}{A_{sd}}\right. S_{[y]}$$

Montrons que pour $w = y.x_i$ on a $S_{[\varepsilon]} \left|\frac{y}{A_{sd}}\right. S_{[y]}$

Un état S_k est accessible pour w si

$$S_0 \left|\frac{y}{A}\right. S_i \left|\frac{x_i}{A}\right. S_k \quad \text{pour des états } S_i \in S_{[y]} ,$$

$S_k \in S_{[w]}$ ssi $S_i \left|\frac{x_i}{A_{snd}}\right. S_k$ pour des $S_i \in S_{[y]}$ (Etape 4.de la méthode de construction de A_{sd})

En conclusion on a :

$$S_{[\varepsilon]} \left|\frac{w}{A_{sd}}\right. S_{[w]}$$

5. Automate généralisé

Définition : Un automate d'états finis généralisé est un 5-Uplets A_G<X^*, S_G, S_{0G}, F_G, II_G> où

- X est l'alphabet d'entrée,
- S_G est un ensemble fini d'états de l'automate,

- S_{0G} est l'état initial, $S_{0G} \in S_G$[1],
- F_G est l'ensemble des états finaux, $F_G \subseteq S_G$,
- II_G l'ensemble des instructions (fonction de transition), $II_G : S_G \times X^* \rightarrow P(S_G)$.

Définition : Un automate partiellement généralisé est un 5-uplets A_{PG}<$X \cup \{\varepsilon\}$, S_{PG}, S_{oPG}, F_{PG}, II_{PG}> où

- S_{PG} est un ensemble fini d'états de l'automate,
- S_{0PG} est l'état initial, $S_{0G} \in S_G$,
- F_{PG} est l'ensemble des états finaux, $F_{PG} \subseteq S_{PG}$,
- II l'ensemble des instructions (fonction de transition), II : $S_{PG} \times X \cup \{\varepsilon\} \rightarrow P(S_{PG})$.

Les transitions dans ces automates sont causées par des lettres de X ou par le mot vide.

Exemple 18:

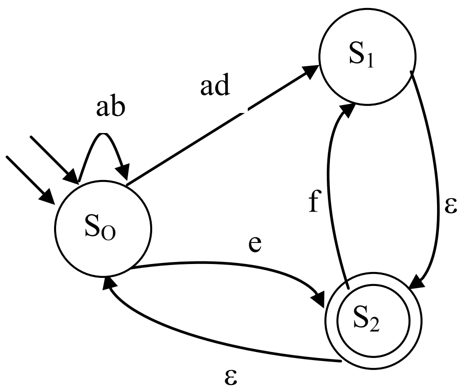


Figure 11 : Automate généralisé.

Le passage de l'état S_0 à l'état S_1 se fait à la lecture du mot ab, de longueur 2.
Le passage de l'état S_0 à l'état S_2 se fait à la lecture de la lettre e
Le passage de l'état S_1 à l'état S_2 se fait par transition spontanée, aucune lettre n'est lue pour ce passage.

Dans un automate généralisé, il existe trois types de transitions :

- Transitions causées par une lettre de X,
- Transitions causées par des mots de longueur supérieure ou égale à 2,

Transition causées par des mots vides que l'on appelle transitions spontanées.

Proposition : Pour tout automate généralisé A_G<X^*, S_G, S_{oG}, F_G, II_G>, il existe un automate simple équivalent A<X, S, S_o, F, II>.

Démonstration :
Pour construire l'automate simple A, nous allons éliminer les transitions causées par des mots de longueur supérieure ou égale à 2, puis celles causées par les transitions spontanées. Pour

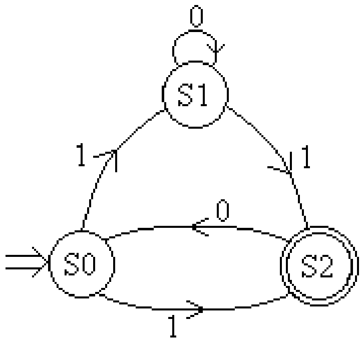
[1] Un automate généralisé peut avoir plusieurs états initiaux, cependant dans ce qui suit nous n'en utiliserons qu'un seul.

cela nous allons faire appel aux automates partiellement généralisés dont la définition est donnée dans ce qui suit

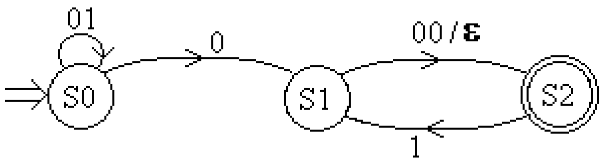
La démonstration de cette proposition se fera en cours.

Série d'exercices

3.1. Trouver l'automate simple déterministe B tel que L(B) = L(A). A étant l'automate suivant:



3.2. Soit *A* l'automate d'états finis suivant:



- Donner l'automate d'états finis simple déterministe *B* tel que L(*B*) = L(*A*)
- Donner l'automate d'états finis complet C tel que L(*C*) = L(*B*)
- Donner l'automate d'états finis D tel que L(*D*) = $\overline{L(C)}$

3.3. Soit A<X, (S0,S1,S2}, S0, (S2}, II > l'automate d'états finis dont la table de transition est donnée ci-dessous:

	ε	a	b	c
S0	-	S0,S1	-	-
S1	S0	-	S2	-
S2	S1	-	-	S0

Construire l'automate *B* tel que L(*B*) = $L^R(A)$

3.4. Construire les automates non-déterministes qui reconnaissent les langages suivants sur l'alphabet {a,b}, rendre ces automates déterministes et donner les automates compléments.

- Tous les mots qui contiennent abab
- Tous les mots qui se terminent par abb ou par aaa.
- Tous les mots qui commencent par abba ou ont la forme abababab....
- $L_1=\{w \in \{a, b\}^*, |w|_a = 2p, |w|_b = 2q+1, p, q \in N\}$
- $L_2=\{w \in \{a, b, c\}^*, |w|_a \equiv (|w|_b + |w|_c) \bmod 2 \}$

3.5. Construire l'automate reconnaissant le langage des entiers binaires qui sont multiples de 3 puis généraliser au cas des multiples de p, p ≥4.

3.6. On représente un texte par un mot sur l'alphabet _ = {$,#, ⊔,″ , c} en adoptant les conventions suivantes :
– $ représente une fin de texte ;
– # représente une fin de ligne ;
– ⊔ reprèsente le caractère espace ;
– ″ est le caractère quote ;
– c représente un caractère alphabétique quelconque.

Dans un tel texte, une suite de caractères est dite quotée si elle est précédée d'une quote ouvrante et suivie d'une quote fermante. Un texte est dit correctement écrit s'il vérifie les conditions suivantes :
1. les lignes ne commencent pas et ne finissent pas par un espace ;
2. il n'y a jamais 2 espaces consécutifs, sauf éventuellement à l'intérieur d'un texte quoté ;
3. chaque ligne contient un nombre pair de quotes (en particulier, un texte quoté ne peut contenir de
fin de ligne, ni de fin de texte) ;
4. une quote ouvrante est soit le premier caractère du texte ou est toujours précédée d'un espace ou d'une fin de ligne et une quote fermante est toujours suivie d'un espace ou d'une fin de ligne ou d'une fin de texte ;
5. une ligne peut être vide. Le texte se termine obligatoirement par $.

Construire un automate d'états fini déterministe pour reconnaître un texte correctement écrit.

3.7. Problème du loup, la chèvre et le chou.

Mr. M. amène le loup L., la chèvre C. le chou H. au bord de la rivière qu'il veut traverser dans un bateau. Le bateau est tellement petit, que l'homme rentre dedans seul ou avec un seul compagnon. Sans surveillance humaine L. mange immédiatement C., et C. mange H. Comment toute l'équipe peut traverser la rivière?

1. Construire un automate ***A*** qui modélise la situation. Il faut retrouver à partir de l'énoncé

- Les états de l'automate (S)
- Les actions possibles
- L'état initial (S_0)
- Les états finaux (F)
- Et les instructions (II)

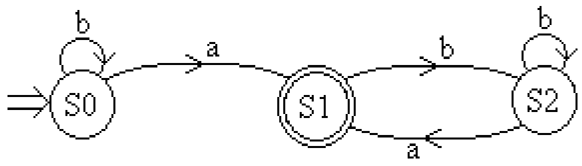
2. Aider Mr. M. à trouver la solution.

3.8. Construire les automates d'états finis simple déterministe acceptant les langages suivants:
- a^*b^*
- X* abX* X = {a, b}
- ((ab)*(baa)*)*aa
- (ab*a(a∪b)*∪aa(a∪b)*)

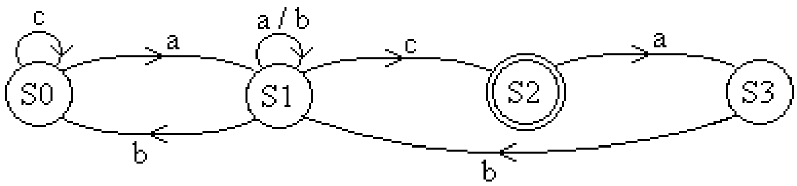
- ((a*bc*)*acb*)*
- 10 ∪ (0 ∪ 11) 0* 1.
- (a ∪ b)aab(a ∪ b)*
- $(b \cup \varepsilon)((a \cup ab)^* \cup (bb)^*)^*$
- $b((aab \cup b)^* a(aa)^*)^* b^*$

3.9. Pour les 4 automates suivants, déterminer les expressions régulières dénotant les langages reconnus par ces automates :

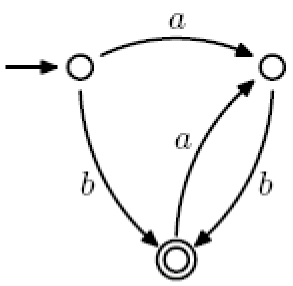
a/



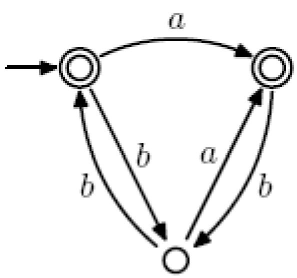
b/



c/



d/



3. 10. Pour chacun des langages suivants, construire l'automate d'états finis qui le reconnaît:

$L_1 = \{w \in X^* \ / \ w$ commence par u $\}$

$L_2 = \{w \in X^* \ / \ w$ se termine par v $\}$

$L_3 = \{w \in X^* \ / \ w$ admet z comme facteur propre $\}$

Montrer que le langage L est un langage régulier :

L = { w $\in$ X* /w commence par u se termine par v et n'admet pas z comme facteur propre }

3. 11. Soit L un langage régulier, on note:

$FG(L) = \{w \in X^* / \exists u \in X^* \text{ et } wu \in L \}$

$FGP(L) = \{w \in X^* / \exists u \in X^+ \text{ et } wu \in L \}$

$FD(L) = \{w \in X^* / \exists u \in X^* \text{ et } uw \in L \}$

$FDP(L) = \{w \in X^* / \exists u \in X^+ \text{ et } uw \in L \}$

$L//u = \{w \in X^* / uw \in L\}$

Etudier la nature ces langages.

3.12. Soit L le langage défini par l'expression régulière suivante

$E = 0\ 1\ [((1\ 0)^* \cup 111)^* \cup 0\ ]^*\ 1$

Donner l'automate déterministe complet reconnaissant le langage L // 010111101

3.13. Soit une grammaire G <X, P >

$S \rightarrow a\ S\ /\ b\ S\ /\ a\ A$

$A \rightarrow a\ D\ /\ b\ D$

$D \rightarrow A\ /\ \varepsilon$

Donner la grammaire régulière droite du complément de L(G)

3.14. Les langages suivants sont-ils réguliers? Justifier vos réponses

$L_1 = \{w \in 0, 1\} / w\, w^R \}$

$L_2 = \{w \in a, b, c\} / |w|_a = |w|_b \}$

$L_3 = \{a^i\, b^j\, a^j \text{ avec } i, j \geq 0 \}$

$L_4 = \{a^n\, b^p / n > p\}$

$L_5 = \{(01)^j\ aa\ 0^{i+j}, i, j \geq 0\}$

$L_6 = \{(ab)^n\ c^m\ (be)^{n+m}, n, m \geq 0\}$

$L_7 = \{a^i\, b^j\ c\ b^{i+j}, i > j\}$

3.15. Exercice du contrôle final 2012

Soient $L_1 = \{w \in \{a, b, c\}^* \text{ tq } |w|_a \equiv 0\ [2] \}$ et $L_2 = \{ w \in \{a, b\}^* \text{ tq } |w|_b \equiv 1\ [2]\}$ deux langages réguliers.

1. Donner les automates A_1 et A_2 reconnaissant respectivement les langages L_1 et L_2,

2. Construire l'automate A reconnaissant $L_1 \cap L_2$

3. Soient L_1 et L_2 deux langages réguliers, montrer que $L_1 \cap L_2$ est régulier. (Justifier vos réponses).

3.16 Exercice du contrôle final 2018

Soit l'expression régulière

$E= b^*a(a^*b \text{ U } a)^*a^+$

1. Donner l'automate simple déterministe complet reconnaissant L(E) (Utiliser les dérivées).
2. Donner la grammaire régulière droite du complément de L(E)//b*aa

3.17 Exercice du contrôle final 2018

Donnez les automates les plus adéquats reconnaissant les langages suivants

$L_1 = \{a^i b^{2i} a^i \ i \geq 0\}$
$L_2 = \{0^i 1^j 2^k \mid i \neq j \text{ or } j \neq k\}$