Automates finis non déterministes (AFND). Déterminisation

Jean-Marie Rifflet - Octobre 2005

1. Fermeture de la famille $\mathcal{R}ec(X^*)$ pour les opérations ensemblistes

1.1. Fermeture pour le passage au complément

Étant donné un langage reconnaissable L sur l'alphabet X, le langage $X^* - L$, complément de L par rapport à X^* est lui-même reconnaissable. En effet :

- si L est reconnaissable, il existe un automate fini **complet** qui le reconnaît.
  Soit $\mathcal{A} =< Q, X, \delta, q_0, Q' >$ un tel automate.
- l'automate $\mathcal{B} =< Q, X, \delta, q_0, Q - Q' >$ reconnaît $X^* - L$:
  en effet $w \in X^* - L$ si et seulement si $\delta^*(q_0, w) \notin Q'$,
  c'est-à-dire si et seulement si $\delta^*(q_0, w) \in Q - Q'$.

L'hypothèse que l'automate $\mathcal{A}$ est complet (il ne faut pas que l'automate se bloque au cours d'un calcul) est nécessaire pour assurer que tous les mots de $X^* - L$ soient effectivement reconnus.

1.2. Fermeture pour l'union et l'intersection

1.2.1. Automate produit de deux automates

Avant de montrer la fermeture de la famille des langages reconnaissables pour les opérations d'intersection et d'union, nous allons associer à deux automates quelconques un ensemble d'automates qui ont tous la même structure (ils correspondent tous au même graphe dans leur représentation graphique) et simulent tous le fonctionnement en parallèle des deux automates de départ. C'est le choix des états terminaux dans un tel automate produit qui permettra de définir quel langage le nouvel automate reconnaît.

Un automate produit de deux AFD quelconques

- $\mathcal{A} =< Q_1, X, \delta_1, q_1, Q'_1 >$
- $\mathcal{B} =< Q_2, X, \delta_2, q_2, Q'_2 >$

est un automate $\mathcal{C}$ défini de la manière suivante :

$$\mathcal{C} =< Q_1 \times Q_2, X, \delta, [q_1, q_2], Q' \subset Q_1 \times Q_2 >$$

où si $s \in Q_1$, $t \in Q_2$ (c.a.d. $[s,t] \in Q_1 \times Q_2$) et $x \in X$, $\delta([s,t], x) = [\delta_1(s,x), \delta_2(t,x)]$.

1.2.2. Des AFD pour l'intersection et l'union de langages reconnaissables

Soient L_1 et L_2 deux langages reconnaissables et $\mathcal{A}_1$ et $\mathcal{A}_2$ des AFD supposés complets tels que $L_1 = L(\mathcal{A}_1)$ et $L_2 = L(\mathcal{A}_2)$.

• pour reconnaître $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$, on considère l'automate produit de $\mathcal{A}_1$ et $\mathcal{A}_2$ dans lequel l'ensemble des états terminaux est $Q' = (Q'_1 \times Q_2) \cup (Q_1 \times Q'_2)$ (**ici, l'hypothèse de complétude est nécessaire**) ;

Remarque : une union infinie de langages reconnaissables n'est pas nécessairement reconnaissable. Ainsi si pour un entier i donné, $L_i = \{a^i b^i\}$, l'union de tous les L_i est le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ dont nous avons montré qu'il n'est pas reconnaissable.

automates de départ n'est pas nécessaire).

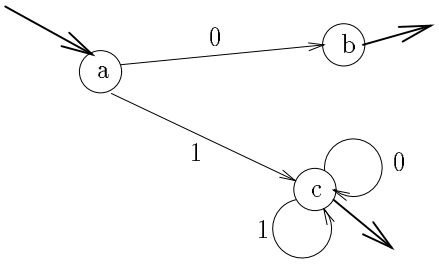
1.2.3. Exemple et application

• À titre d'exemple, reprenons l'exemple de l'automate $\mathcal{A}_6$ (page 4, notes de cours 2) reconnaissant les mutiples de 3 écrits en base 2. Le langage L_6 qu'il reconnaît contient le mot vide et des nombres comportant des 0 superflus en tête. Le langage L'_6 déduit de L_6 en y enlevant les mots de cette forme correspond plus à ce qu'on peut considérer comme un nombre sous une forme usuelle.

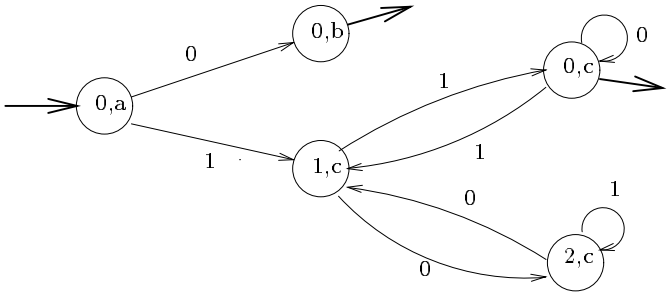
De fait on a : $L'_6 = L_6 \cap (\{0\} \cup (\{1\}.\{0,1\}^*))$.

Pour construire un automate reconnaissant le langage L'_6, on peut commencer par construire un automate $\mathcal{B}$ reconnaissant le langage $(\{0\} \cup \{1\}.\{0,1\}^*)$ et déduire de cet automate et de l'automate $\mathcal{A}_6$ un automate $\mathcal{A}'_6$ par le procédé que nous venons de décrire.

Un automate $\mathcal{B}$ pour le langage $\{0\} \cup (\{1\}.\{0,1\}^*)$ est par exemple :



L'automate $\mathcal{A}'_6$ obtenu est alors (on n'a conservé comme états de cet automate que les couples utiles, c'est-à-dire ceux qu'on peut atteindre depuis le couple constitué des deux états initiaux des deux automates de départ) :



• De la propriété de fermeture de la famille des langages reconnaissables pour l'intersection, on peut déduire, par exemple, que le langage $L = \{w \mid \mid w \mid_a = \mid w \mid_b\}$ n'est pas reconnaissable.

En effet, s'il l'était, le langage $\{a^n b^n \mid n \in \mathbb{N}\} = L \cap (\{a\}^*.\{b\}^*)$ le serait comme intersection de deux langages reconnaissables ($\{a\}^*.\{b\}^*$ étant de manière évidente reconnaissable).

2. L'opération miroir et les automates finins non déterministes [AFND]

2.1. Reconnaissance du miroir d'un langage

• Le langage $\widetilde{L}$ (miroir de L) est défini comme l'ensemble $\{\widetilde{u} \mid u \in L\}$

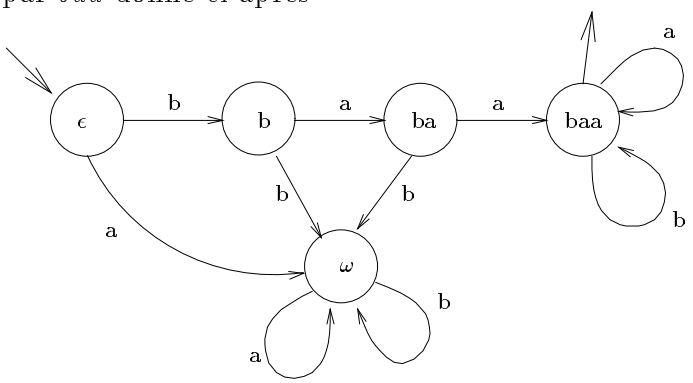
où, si $u = x_1 x_2 \ldots x_n$, alors $\widetilde{u} = x_n \ldots x_2 x_1$.

Si L est reconnaissable, il existe un automate fini déterministe $\mathcal{A} = < Q, X, \delta, q_0, Q' >$ qui le reconnaît. Si on considère la machine obtenue en inversant dans cet automate toutes

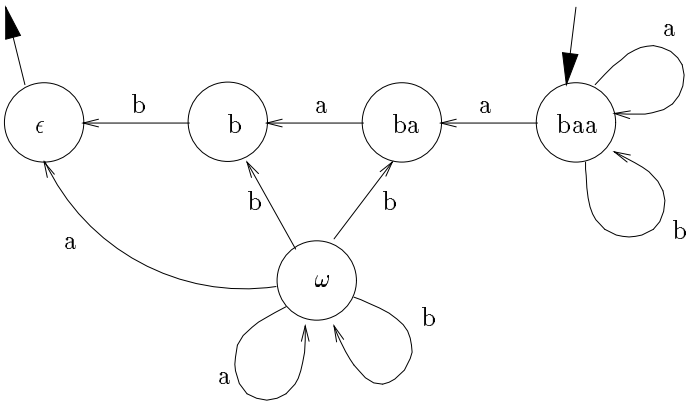
cela, on perd généralement la propriété de déterminisme :

- la machine peut avoir plusieurs états initiaux (si l'automate de départ a plusieurs états terminaux) ;
- la machine étant dans un état q donné et lisant une lettre x, plusieurs transitions sont possibles, conduisant à des états différents.

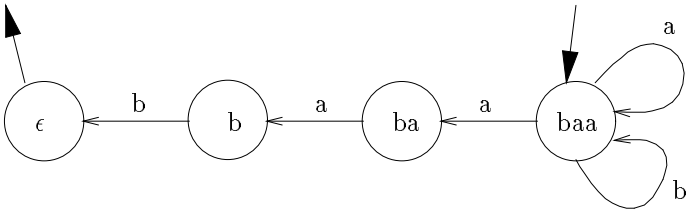
• Ainsi, partant de l'automate fini déterministe (complet) reconnaissant le langage L_1 des mots commençant par baa donné ci-après



le langage miroir $\widetilde{L_1}$ de L_1 (ensemble des mots finissant par aab) est reconnu, en appliquant le mécanisme que nous venons de décrire, par la machine suivante :



qu'on peut émonder en éliminant l'état ω inaccessible depuis l'état initial (et donc inutile) et les transitions le concernant pour obtenir la machine :



La machine ainsi obtenue n'est pas un automate fini déterministe (AFD) mais ce qu'on appelle un **automate fini non déterministe (AFND)**, type de machines que nous allons étudier maintenant et dont nous allons montrer que leur puissance de calcul est identique à celle des AFD : pour tout AFND, il est possible de construire effectivement un AFD équivalent, c'est-à-dire reconnaissant exactement le même langage.

• un automate fini non déterministe (AFND) $\mathcal{A}$ est un quintuplet

$\mathcal{A} = < Q, X, \delta, Q_0 \subset Q, Q' \subset Q >$ avec $\delta : Q \times X \longrightarrow \mathcal{P}(Q)$

($\mathcal{P}(Q)$ désigne l'ensemble des parties de Q) ;

• un calcul d'un automate fini non déterministe $\mathcal{A}$ correspond à la fonction

$\delta^* : Q \times X^* \longrightarrow \mathcal{P}(Q)$ définie par :

$$\forall q \in Q, \delta^*(q, \epsilon) = \{q\}$$

$$\forall q \in Q, \forall u \in X^*, \forall x \in X, \delta^*(q, ux) = \bigcup_{q' \in \delta^*(q,u)} \delta(q', x)$$

Le calcul d'un automate non déterministe revient donc à mener en parallèle tous les calculs possibles de cet automate tels que définis par la fonction de transitions δ ;

• le langage reconnu par un automate fini non déterministe $\mathcal{A}$ est

$L(\mathcal{A}) = \{w \mid w \in X^* \; et \; \exists q_0 \in Q_0 \; tel \; que \; \delta^*(q_0, w) \cap Q' \neq \emptyset\}$ (c'est-à-dire que parmi tous les calculs possibles ou tous les chemins du graphe il y en a un qui mène d'un état initial à un état termina).

2.2.2. Théorème

Pour tout automate fini non déterministe $\mathcal{A}$, il existe un automate fini déterministe $\mathcal{B}$ tel que $L(\mathcal{A}) = L(\mathcal{B})$

L'ensemble Q des états de l'automate non déterministe $\mathcal{A}$ étant fini, l'ensemble des parties $\mathcal{P}(\mathcal{Q})$ de Q est lui-même fini (il contient $2^{|Q|}$ éléments).

Un AFD $\mathcal{B}$ équivalent à un AFND $\mathcal{A}$ donné est défini par :

- $\mathcal{P}(Q)$ comme ensemble d'états
- Q_0 comme seul état initial
- $\{E \mid E \subset Q, E \cap Q' \neq \emptyset\}$ comme ensemble des états terminaux
- la fonction de transition $\delta_{\mathcal{B}} : \mathcal{P}(Q) \times X \longrightarrow \mathcal{P}(Q)$ telle que

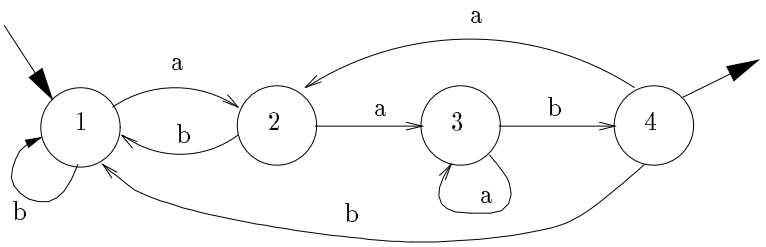
$$\forall S \subset Q, \forall x \in X, \delta_{\mathcal{B}}(S, x) = \bigcup_{s \in S} \delta(s, x)$$

D'un point de vue pratique, lors de la déterminisation d'un automate on ne conservera comme états que les éléments utiles de $\mathcal{P}(Q)$, c'est-à-dire ceux apparaissant dans le processus de déterminisation à partir de l'état initial du nouvel automate.

2.2.3. Application à l'exemple

La construction conduit à la table suivante où les états du nouvel automate sont des ensembles d'états de l'automate de départ, l'état initial étant l'ensemble $\{aab\}$ et les états terminaux tous les états atteignables contenant ϵ (ici il n'y en a qu'un, repéré par une étoile dans la table) :

état	a	b
$\{baa\}$ [1]	$\{baa, ba\}$ [2]	$\{baa\}$ [1]
$\{baa, ba\}$ [2]	$\{baa, ba, b\}$ [3]	$\{baa\}$ [1]
$\{baa, ba, b\}$ [3]	$\{baa, ba, b\}$ [2]	$\{baa, \epsilon\}$ [4]
$\{baa, \epsilon\}$ [4] *	$\{baa, ba\}$ [2]	$\{baa\}$ [1]



3. AFD standard

Avant de montrer la fermeture de la famille des langages reconnaissables par les opérations produit de langage et passage à l'étoile, nous introduisons la notion d'AFD standard.

3.1. Définition

Un **AFD** $\mathcal{A} =< Q, X, \delta, q_0, Q' >$ est dit **standard** s'il n'existe aucun couple $(q, x) \in Q \times X$ tel que $\delta(q, x) = q_0$.
Autrement dit, au cours d'un calcul, l'automate ne repasse jamais par son état initial.

3.2. Propriété : pour tout AFD, il existe un AFD standard équivalent

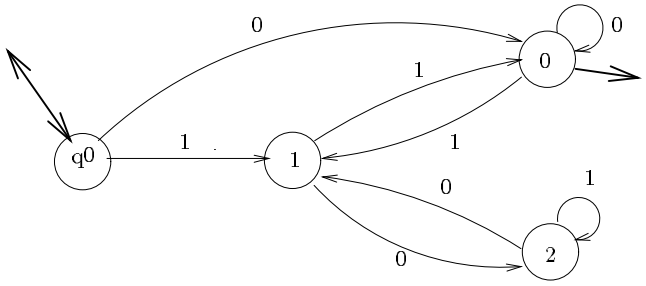
Soit un AFD $\mathcal{A} =< Q, X, \delta, q_0, Q' >$.
L'AFD $\mathcal{B} =< S, X, \delta', s_0, S' >$ tel que :

- $S = Q \cup \{s_0\}$ avec $s_0 \notin Q$
- la fonction δ' est définie par :
  - $\delta'(q, x) = \delta(q, x)$ pour tout $(q, x) \in Q \times X$
  - $\delta'(s_0, x) = \delta(q_0, x)$ pour tout $x \in X$
- l'ensemble des états terminaux S' est :
  - Q' si $_0 \notin Q'$
  - $Q' \cup \{s_0\}$ si $q_0 \in Q'$

est un AFD standard équivalent à $\mathcal{A}$.

3.3. Exemple

Reprenons l'exemple de l'automate $\mathcal{A}_6$ pour le langage L_6 des multiples de 3 en base 2.
Un automate standard $\mathcal{A}_6$" équivalent en est (l'état $q0$ a été ajouté et il est terminal puisque l'état initial de l'automate $\mathcal{A}_6$ l'est) :

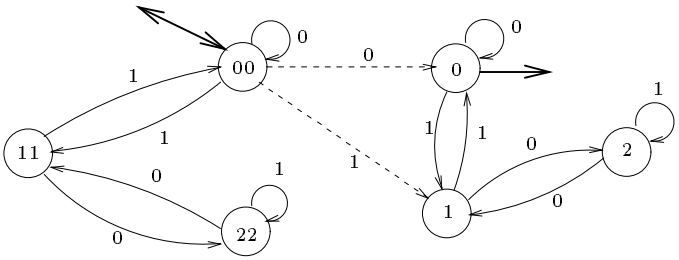


Pour montrer cette propriété nous allons construire, à partir de deux automates reconnaissant respectivement l'un des deux langages à composer, un automate non déterministe reconnaissant le produit de ces deux langages.
Soient $\mathcal{A} =< Q_1, X, \delta_1, q_1, Q'_1 >$ un AFD (non nécessairement standard) reconnaissant le langage L_1 et $\mathcal{B} =< Q_2, X, \delta_2, q_2, Q'_2 >$ un AFD **standard** reconnaissant le langage L_2.
On suppose que $Q_1 \cap Q_2 = \emptyset$, ce qui est toujours possible en renommant les états.
Le langage $L_1.L_2$ est reconnu par l'AFND $\mathcal{C} =< Q, X, \delta, Q_0, Q' >$ avec :

- ensemble des états : $Q = (Q_1 \cup Q_2) - \{q_2\}$
- état initial : $Q_0 = \{q_1\}$
- ensemble des états terminaux :
  - Q'_2 si $q_2 \notin Q'_2$ (c'est-à-dire si $\epsilon \notin L_2$)
  - $(Q'_1 \cup Q'_2) - \{q_2\}$ si $q_2 \in Q'_2$ (c.a.d. $\epsilon \in L_2$)
- fonction de transitions δ définie par
  - $\forall q \in (Q_1 - Q'_1), \forall x \in X,\ \delta(q, x) = \{\delta_1(q, x)\}$
  - $\forall q \in Q'_1, \forall x \in X,\ \delta(q, x) = \{\delta_1(q, x)\} \cup \{\delta(q_2, x)\}$
  - $\forall q \in Q_2 - \{q_2\}, \forall x \in X,\ \delta(q, x) = \{\delta_2(q, x)\}$

4.2. Exemple

Un automate reconnaissant le langage $L_6.L_6$ est obtenu par cette construction à partir par exemple de l'automate $\mathcal{A}_6$ (pour le membre gauche du produit dont l'automate utilisé n'est pas nécessairement standard) et de l'automate $\mathcal{A}_6$" (pour le membre droit du produit pour lequel l'automate utilisé doit être standard). L'état initial $q0$ de ce dernier automate et les transitions en sortant ont été supprimées et des transitions ajoutées à partir de l'état terminal de l'automate $\mathcal{A}_6$:



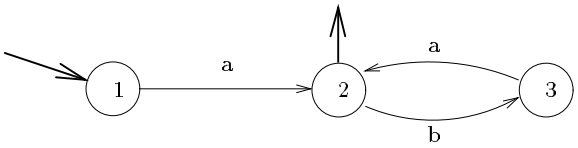
5. Fermeture de la famille $\mathcal{R}ec(\mathbf{X}^*)$ pour le passage à l'étoile

5.1. Automate reconnaissant l'étoile d'un langage

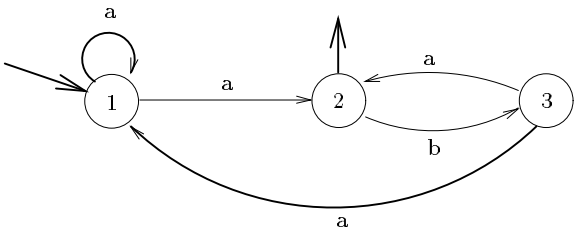
Soit L un langage reconnaissable et un AFD standard $\mathcal{A} =< Q, X, \delta, q_0, Q' >$ le reconnaissant.
Le langage L^* est reconnu par l'AFND $\mathcal{B} =< Q, X, \delta', \{q_0\}, Q' \cup \{q_0\} >$ où la fonction de transitions δ' est définie par :

$\forall q \in Q, \forall x \in X$

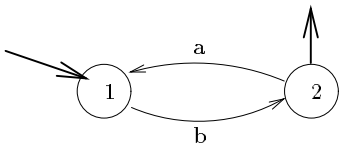
- $\delta'(q, x) = \{\delta(q, x)\}$ si $\delta(q, x) \notin Q'$;
- $\delta'(q, x) = \{\delta(q, x), q_0\}$ si $\delta(q, x) \in Q'$.



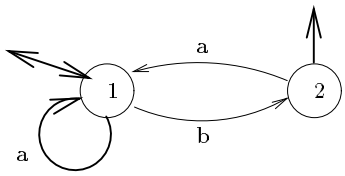
La construction standard décrite précédemment conduit à l'automate suivant (les arcs ajoutés à l'automate de départ sont en gras) pour le langage L^* :



Remarque : si on part de l'automate non standard suivant qui reconnaît également le langage L



le procédé décrit appliqué directement à cet automate conduit à l'automate suivant :



qui ne reconnaît pas le langage L^* : il reconnaît en effet tous les mots de la forme a^n dont aucun (hormis ϵ) n'appartient à L^*.