Licence ST Informatique – S4 – 2008/2009

Automates et langages

avril 2009

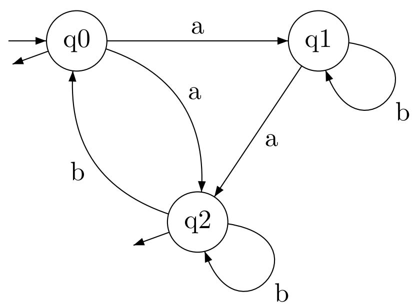
Résumé du cours 5 : Automates finis de mots

1 Automates et langages reconnaissables

Définition 1 : Un *automate fini de mots* est un quintuplet (X, Q, I, F, δ), avec

- X alphabet
- Q ensemble fini d'*états*
- $I \subseteq Q$ ensemble d'états *initiaux*
- $F \subseteq Q$ ensemble d'états *finaux*
- δ fonction de transition, de $Q \times X$ dans 2^Q.

Exemple 1



$Q = \{q_0, q_1, q_2\}$
$I = \{q_0\}$
$F = \{q_0, q_2\}$
$\delta(q_0, a) = \{q_1, q_2\}$
$\delta(q_0, b) = \{\}$
$\delta(q_1, a) = \{q_2\}$
$\delta(q_1, b) = \{q_1\}$
$\delta(q_2, a) = \{\}$
$\delta(q_2, b) = \{q_0, q_2\}$

On étend la fonction δ au domaine $2^Q \times X$:

$$\delta(S, x) = \bigcup_{s \in S} \delta(s, x)$$

Définition 2 : Si m est un mot sur X et S un ensemble d'états, on note $\delta^*(S, m)$ l'ensemble des états atteints en lisant m à partir des états de S.

- Si $m = \varepsilon$, $\delta^*(S, m) = S$
- Sinon, $m = a.m'$ pour a lettre de X, et $\delta^*(S, m) = \delta^*(\delta(S, a), m')$

Exemple 2 *Sur l'automate donné précédemment,* $\delta^*(\{q_0\}, aba) = \{q_1, q_2\}$

Définition 3 : Un mot m sur l'alphabet X est *reconnu* par un automate $M = (X, Q, I, F, \delta)$ si

$$\delta^*(I, m) \cap F \neq \emptyset$$

Exemple 3 *aba est reconnu par l'automate précédent car on peut atteindre l'état final* q_2 *à partir de l'ensemble* I *:* $\delta^*(\{q_0\}, aba) \cap F = \{q_2\}$.

Définition 4 : Le *langage* $L(M)$ *reconnu par un automate* M est l'ensemble des mots reconnus par M.

Définition 5 : Deux automates sont *équivalents* s'ils reconnaissent le même langage

Définition 6 : Un langage est *reconnaissable* s'il existe un automate qui le reconnaît.

2 Propriétés des automates

Définition 7 : Soit $M = (X, Q, I, F, \delta)$ un automate et soit q un état de Q.
q est dit *accessible* s'il existe un mot u sur l'alphabet X tel que $q \in \delta^*(I, u)$, et q est dit *coaccessible* s'il existe un mot u tel que $\delta^*(\{q\}, u) \cap F \neq \emptyset$.

Proposition 1 : Pour tout automate M, il existe un algorithme permettant de calculer l'ensemble des ses états accessibles, noté $\mathsf{acc}(M)$ et l'ensemble des ses états coaccessibles, noté $\mathsf{coacc}(M)$.

Définition 8 : Un automate est dit *émondé* si tous ses états sont accessibles et coaccessibles.

Proposition 2 : Pour tout automate M, il existe un automate émondé équivalent.

Définition 9 : Un automate $M = (X, Q, I, F, \delta)$ est *complet* si pour tout état q et pour toute lettre x, $||\delta(\{q\}, x)|| \geq 1$.

Proposition 3 : Pour tout automate M, il existe un automate complet équivalent.

CONSTRUCTION : On ajoute à M un nouvel état appelé *puits*.

- à chaque fois que $\delta(q, x) = \emptyset$ dans M, on ajoute la transition $\delta(q, x) = \{\text{puits}\}$.
- Pour toute lettre x de X, on ajoute $\delta(\text{puits}, x) = \{\text{puits}\}$

Définition 10 : Un automate $M = (X, Q, I, F, \delta)$ est *déterministe* si $||I|| = 1$ et si pour tout état q et pour toute lettre x, $||\delta(\{q\}, x)|| \leq 1$.

Proposition 4 : Pour tout automate M, il existe un automate déterministe équivalent.

En effet, on peut construire $M_d = (X, Q_d, I_d, F_d, \delta_d)$ automate déterministe équivalent à M, en définissant

- Q_d le plus petit ensemble des parties de Q tel que $\forall u \in X^*, \delta^*(I, u) \in Q_d$.
- $I_d = \{I\}$

- $F_d = \{s \in Q_d \mid s \cap F \neq \emptyset\}$
- $\forall x \in X,\ \forall s \in Q_d\ \delta_d(s,x) = \{\delta(s,x)\}$

ALGORITHME DE DÉTERMINISATION :

Soit $M = (X, Q, I, F, \delta)$ un automate non déterministe
$Q_1 \leftarrow \{I\}$; $Q_2 \leftarrow \{\}$
TantQue $Q_1 \neq Q_2$ Faire
    $Q_3 \leftarrow Q_1 - Q_2$
    $Q_2 \leftarrow Q_1$
    Pour $s \in Q_3$ Faire
        Pour $x \in X$ Faire
            $Q_1 \leftarrow Q_1 \cup \{\delta(s,x)\}$
        FinPour
    FinPour
FinTantQue

des exemples ont été traités en cours