Automates et Calculabilité

– Automates finis non déterministes –

Leçon 4 : 10 nov 2003

Uwe Nestmann

EPFL-I&C

Exemple (I)

Soit $\Sigma = \{ 0, 1 \}$.
$A = \{ x \in \Sigma^* \mid$ l'avant-dernier symbole dans x est un $1 \}$

Observation :
$\forall x \in \Sigma^*$:
$(|x| \geq 2 \Rightarrow (\exists x_1, x_2 \in \Sigma^* : x = x_1 \cdot x_2 \wedge x_2 \in \{ 00, 01, 10, 11 \}$

Idée : définissons un automate M avec les états

$$\begin{array}{rcl} \mathcal{Q} & \stackrel{\text{def}}{=} & \{ [00], [01], [10], [11] \} \\ s & \stackrel{\text{def}}{=} & [00] \\ F & \stackrel{\text{def}}{=} & \{ [10], [11] \} \end{array}$$

où les états stockent la valeur des deux derniers chiffres lus.

Lorsque $|x| < 2$, $x \in \{ \epsilon, 0, 1 \}$ et nous représenterons ϵ et 0 par $[00]$, et 1 par $[01]$.

Non déterminisme

- Le ***déterminisme*** dans les automates signifie que pour toute paire constituée d'un état et d'un symbole de l'alphabet, il existe ***exactement un état suivant***.

- Parfois, cela est trop exigeant pour exprimer l'essence d'un langage que l'on veut accepter avec un automate.

- Nous allons alléger cette contrainte sur les automates en admettant
  - pour tout symbole :
    zéro, un, ou plusieurs états suivants à un état, et
  - ***plusieurs états initiaux***.

Astuces générales

- Exprimer des *relations* $X \times Y$ par des *fonctions* $X \to 2^Y$:
  $\left\{ \begin{array}{c} x \mapsto y_1 \\ x \mapsto y_2 \end{array} \right\}$ peut être représenté par $\{x \mapsto \{\, y_1, y_2 \,\}\}$

- Élever une fonction $f : X \to Y$ à $f' : 2^X \to 2^Y$.

  Soit $V \in 2^X$. Alors $f'(V) \stackrel{\text{def}}{=} \{\, f(x) \mid x \in V \,\}$.

  NB: Pour tout $x \in V$, $f(x) \in Y$ est un élément (de Y).

- Élever une fonction $g : X \to 2^Y$ à $g' : 2^X \to 2^Y$.

  Soit $V \in 2^X$. Alors $g'(V) \stackrel{\text{def}}{=} \bigcup_{x \in V} g(x)$.

  Pourqoui utiliser l'union $\bigcup$?
  Parce que, pour tout $x \in V$, $g(x) \in 2^Y$ est un ensemble.

Formellement . . .

Un automate fini non déterministe (AFN) est une structure

$$M = (\mathcal{Q}, \Sigma, \Delta, S, F)$$

où

- $\mathcal{Q}$ est un ensemble fini d'*états*,
- Σ est un ensemble fini de symboles, *l'alphabet (d'entrée)*,
- $\Delta : \mathcal{Q} \times \Sigma \to 2^{\mathcal{Q}}$ est la *fonction de transition*,
- $S \subseteq \mathcal{Q}$ est un sous-ensemble de $\mathcal{Q}$, les *états initiaux*,
- $F \subseteq \mathcal{Q}$ est un sous-ensemble de $\mathcal{Q}$, les *états accepteurs*.

NB: Il est possible que $\Delta(p, a) = \emptyset$!

Exemple (II)

Reécrire l'automate précédent en utilisant le non déterminisme !

Comparer la taille
de la version déterministe et non déterministe
dans le cas où l'on s'intéresse au $n^{\text{ième}}$ chiffre de droite . . .

Combien d'états faut-il prévoir ?

Exemple (III)

$$A = \{ x \in \{ a \}^* \mid \exists n \in \mathbb{N} : |x| = 3 * n \quad \vee \quad |x| = 5 * n \}$$

1. Construire un AFN qui accepte A.

2. Construire un AFD qui accepte A.

Langage accepté par un AFN

Soit $M = (Q, \Sigma, \Delta, S, F)$ un AFN avec $\Delta : Q \times \Sigma \to 2^Q$,
on définit $\widehat{\Delta} : 2^Q \times \Sigma^* \to 2^Q$ par :

$$\widehat{\Delta}(P, \epsilon) \stackrel{\text{def}}{=} P \qquad\qquad \widehat{\Delta}(P, xa) \stackrel{\text{def}}{=} \bigcup_{q \in \widehat{\Delta}(P,x)} \Delta(q, a)$$

Un automate N accepte un mot $x \in \Sigma^*$ si

$$\widehat{\Delta}(S, x) \cap F \neq \emptyset$$

Le *langage accepté par* N, dénoté par $L(N)$,
est l'ensemble des mots qui sont acceptés par N.

$$L(N) \stackrel{\text{def}}{=} \{\, x \in \Sigma^* \mid \widehat{\Delta}(S, x) \cap F \neq \emptyset \,\}$$

Deux propriétés

Lemme:
Pour tout $x, y \in \Sigma^*$ et $P \subseteq Q$,

$$\widehat{\Delta}(P, xy) = \widehat{\Delta}(\widehat{\Delta}(P, x), y)$$

Lemme:
Pour tout $P_i \subseteq Q$ (avec $1 \leq i \leq n$ et $n \in \mathbb{N}$) et $x \in \Sigma^*$,

$$\widehat{\Delta}(\bigcup_i P_i, x) = \bigcup_i \widehat{\Delta}(P_i, x)$$

Conversion de AFN en AFD

Un AFN N est transformé en AFD M avec,

- pour $\mathcal{Q}_M$ des ensembles d'états de N, $\mathcal{Q}_M \subseteq 2^{\mathcal{Q}_N}$;
- pour $\delta_M(P, a)$, $q \in \delta_M(P, a)$
  s'il existe $p \in P$ tel que $q \in \Delta_N(p, a)$.

« Subset-construction », formellement

Soit $N = (\mathcal{Q}_N, \Sigma, \Delta_N, S_N, F_N)$

Alors, on construit $M = (\mathcal{Q}_M, \Sigma, \delta_M, s_M, F_M)$ avec :

$$\begin{array}{rcl}
\mathcal{Q}_M & \stackrel{\text{def}}{=} & 2^{\mathcal{Q}_N} \\
\delta_M(P, a) & \stackrel{\text{def}}{=} & \widehat{\Delta}_N(P, a) \\
s_M & \stackrel{\text{def}}{=} & S_N \\
F_M & \stackrel{\text{def}}{=} & \{\, P \subseteq \mathcal{Q}_N \mid P \cap F_N \neq \emptyset \,\}
\end{array}$$

Lemme: Pour tout $P \subseteq \mathcal{Q}_N$ et $x \in \Sigma^*$, $\quad \widehat{\delta}_M(P, x) = \widehat{\Delta}_N(P, x)$.

Théorème: N et M acceptent le même langage.