Informatique Théorique 3
2004/05

Semaine 6

Régularité

2.5.1 Théorème (Rappel) Soit Σ un alphabet, L un langage sur Σ. Les propositions suivantes sont équivalentes :

1. L est régulier.
2. Il existe un AFD M tel que $\mathrm{L}(M) = L$.
3. Il existe un AFN M tel que $\mathrm{L}(M) = L$.
4. Il existe un AFN$_\mathbf{e}$ M tel que $\mathrm{L}(M) = L$.
5. Il existe une expression régulière $\mathbf{x} \in \mathbf{RE}_\Sigma$ telle que $\mathrm{L}(\mathbf{x}) = L$.

6. Il existe une grammaire régulière G telle que L(G) = L
7. Il existe un AFNG M tel que L(M)=L
8. (... aujourd'hui ...)

2.7.10 Théorème (Myhill-Nerode) Soit Σ un alphabet et $L \subseteq \Sigma^*$. Alors L est régulier ssi $\#(\Sigma^*/_{\equiv_L}) < \infty$.

0.3.6 Définition (Relation d'équivalence) Une *relation d'équivalence* sur A est une relation $R \subseteq A \times A$ réflexive, symétrique et transitive.

La *classe d'équivalence* de $a \in A$ dans R est l'ensemble défini par

$$[R]a \triangleq \{\, b \in A \mid aRb \,\}$$

Lorsqu'il n'y a pas d'ambiguité sur la relation R dont on parle, on écrit $[a]$ pour $[a]_R$.

Le *quotient* $A/_R$ est défini par :

$$A/_R \triangleq \{\, [a]_R \mid a \in A \,\}$$

c'est l'ensemble des classes d'équivalences de A.

L'index de R est le nombre de classes d'équivalence de R. Il est égal à $\#(A/_R)$ (cf. 0.5.1). $\square$

0.3.7 Lemme Soit R une relation d'équivalence sur un ensemble A et $p, q \in A$, alors :

1. pRq ssi $[p] = [q]$.
2. Si $[p] \neq [q]$, alors $p \notin [q]$ et $q \notin [p]$.
3. $[p] = [q] \vee [p] \cap [q] = \emptyset$
4. $A = \bigcup_{r \in A}[r]$. $\square$

0.3.8 Lemme Soit R, S deux équivalences sur A avec index fini et $R \subseteq S$.

1. $\forall p \in A\,.\,[p]_R \subseteq [p]_S$.
2. $\#(A/_R) \geq \#(A/_S)$. $\square$

Deux équivalences

2.7.4 Définition Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.
On définit la relation $\boxed{\equiv_M \subseteq \Sigma^* \times \Sigma^*}$ par :

$$x \equiv_M y \quad \text{ssi} \quad \boxed{\widehat{\delta}(s,x) = \widehat{\delta}(s,y)}$$

2.7.7 Définition Soit Σ un alphabet et $L \subseteq \Sigma^*$.
On définit la relation $\boxed{\equiv_L \subseteq \Sigma^* \times \Sigma^*}$ par :

$$x \equiv_L y \quad \text{ssi} \quad \boxed{\forall z \in \Sigma^* \,.\, (xz \in L \iff yz \in L)}$$

Myhill-Nerode (M-N)

2.7.1 Définition Soit Σ un alphabet, $L \subseteq \Sigma^*$ et R une équivalence sur Σ^*.
R est une *L-congruence* si :

1. R est une congruence pour la concaténation de symboles à droite :
   si $x \mathrel{R} y$, alors $\forall a \in \Sigma \,.\, xa \mathrel{R} ya$.
2. R raffine $L \times L$, c'est à dire :
   si $x \mathrel{R} y$, alors $x \in L \iff y \in L$.

R est une *relation de Myhill-Nerode pour L* si, de plus,

3. L'index de R est fini : $\#(\Sigma^*/_R) < \infty$.

2.7.6 Proposition Soit M un AFD.
Alors $\equiv_M$ est une relation de Myhill-Nerode pour $\mathrm{L}(M)$.

2.7.8 Lemme Soit Σ un alphabet et $L \subseteq \Sigma^*$.

1. $\equiv_L$ est une L-congruence.
2. Pour toute L-congruence R, on a $R \subseteq \equiv_L$.

C'est-à-dire $\equiv_L$ est la plus grande L-congruence.

Un automate par équivalence M-N

2.7.2 Définition Soit R une relation de Myhill-Nerode pour $L \subseteq \Sigma^*$.
On définit l'AFD $M_R = (Q, \Sigma, \delta, s, F)$ par :

$$\begin{aligned}
Q &\triangleq \boxed{\Sigma^*/_R = \{\, [x]_R \mid x \in \Sigma^* \,\}} \\
s &\triangleq \boxed{[\epsilon]_R} \\
F &\triangleq \boxed{\{\, [x]_R \mid x \in L \,\}} \\
\delta &: \ \Sigma^*/_R \times \Sigma \to \Sigma^*/_R \\
&\quad \boxed{([x]_R, a) \mapsto [xa]_R}
\end{aligned}$$

2.7.3 Proposition Soit R une relation de Myhill-Nerode pour L.
Alors $\mathrm{L}(M_R) = L$.

2.7.8 Lemme Soit Σ un alphabet et $L \subseteq \Sigma^*$.

1. $\equiv_L$ est une L-congruence.
2. Pour toute L-congruence R, on a $R \subseteq \equiv_L$.

C'est-à-dire $\equiv_L$ est la plus grande L-congruence.

0.3.8 Lemme Soit R, S deux équivalences sur A avec index fini et $R \subseteq S$.

1. $\forall p \in A \,.\, [p]_R \subseteq [p]_S$.
2. $\#(A/_R) \geq \#(A/_S)$. □

2.7.9 Corollaire Soit M un AFD. Soit $L = \mathrm{L}(M)$.

1. $\equiv_M \subseteq \equiv_L$.
2. $\#(\Sigma^*/_{\equiv_L}) \leq \#(\Sigma^*/_{\equiv_M}) < \infty$
3. $\equiv_L$ est une relation de Myhill-Nerode pour L.

Théorème principal

2.7.10 Théorème (Myhill-Nerode) Soit Σ un alphabet et $L \subseteq \Sigma^*$. Alors L est régulier ssi $\#(\Sigma^*/_{\equiv_L}) < \infty$.

2.7.11 Définition Soit $L \subseteq \Sigma^*$ tel que $\equiv_L$ est d'index fini. Alors l'automate $M_{\equiv_L}$ est dit *automate des classes d'équivalences de L*.

2.7.12 Lemme Soit $L \subseteq \Sigma^*$ tel que $\equiv_L$ est d'index fini. Soit $Q_L = \Sigma^*/_{\equiv_L}$ l'ensemble d'états de l'automate des classes d'équivalences de L. Si $M = (Q, \Sigma, \delta, s, F)$ un AFD tel que $\mathrm{L}(M) = L$, alors $\#(Q) \geq \#(Q_L)$.

2.8 Minimisation des automates

Encore une équivalence, mais sur Q !

2.8.4 Définition Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD. On définit la relation $\approx_M \subseteq Q \times Q$ par :

$$p \approx_M q \quad \text{ssi} \quad \forall z \in \Sigma^* .\, (\widehat{\delta}(p,z) \in F \iff \widehat{\delta}(q,z) \in F)$$

2.8.6 Lemme Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.

1. $p \approx_M q$ ssi $L_p = L_q$. (voir Déf. 2.4.12)
2. $\approx_M$ est une équivalence (sur Q).

Etats non-accessibles

2.8.1 Définition Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.

1. Un état $q \in Q$ est dit *accessible* **(détecter)** s'il existe un mot $w \in \Sigma^*$ tel que $\widehat{\delta}(s, w) = q$.
2. L'AFD $\mathrm{acc}(M) = (Q', \Sigma, \delta', s, F')$ est défini par :

(éliminer)

$$Q' \triangleq \{ q \in Q \mid q \ \text{accessible} \}$$
$$\delta' \triangleq \delta \restriction_{Q' \times \Sigma}$$
$$F' \triangleq F \cap Q'$$

Algorithme de minimisation (I)

1. Transformer M en $\mathrm{acc}(M) = (Q, \Sigma, \delta, s, F)$.
2. Calculer l'ensemble $R_\surd$ par la procédure suivante :
   (a) Construire un tableau avec une entrée pour tous les couples *non-ordonnés* $\{p, q\}$ avec $p, q \in Q$.
   (b) Marquer tous les couples $\{p, q\}$ tels que $p \in F$ et $q \notin F$.

$$(\textsc{init})\ \frac{p \in F \qquad q \notin F}{\{p,q\} \in R_\surd}$$

   (c) Répéter jusqu'à plus soif (hic) :

4

$$(\textsc{step})\ \frac{a \in \Sigma \qquad \delta(p,a) = p' \quad 3 \quad \delta(q,a) = q' \qquad \{p',q'\} \in R_\surd}{2 \qquad \{p,q\} \in R_\surd}$$

1

Algorithme de minimisation (II)

3. Posons $R \triangleq \{ (p,q) \mid \{p,q\} \not\in R_\checkmark \}$.
L'automate minimal est l'AFD $\mathrm{acc}(M)/_R$.

2.8.10 Théorème Soit un AFD M et R la relation obtenue en appliquant l'algorithme de minimisation sur M. Alors $R = \approx_{\mathrm{acc}(M)}$.

2.8.11 Lemme Si M' est le résultat de la minimisation de M, et si M'' est l
résultat de la minimisation de M', alors $M' \cong M''$.

2.8.8 Théorème Soit M un AFD avec $\mathrm{acc}(M) = M$. Soit $L = \mathrm{L}(M)$
Alors $M/_{\approx_M} \cong M_{\equiv_L}$

Quotient d'un automate

2.8.3 Définition Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD et R une équivalence sur Q.
On définit l'automate $M/_R = (Q/_R, \Sigma, \delta/_R, s/_R, F/_R)$ par :

$$\begin{aligned}
Q/_R &\triangleq \{ [p]_R \mid p \in Q \} \\
s/_R &\triangleq [s]_R \\
F/_R &\triangleq \{ [p]_R \mid p \in F \} \\
\delta/_R &: Q/_R \times \Sigma \to Q/_R \\
&([p]_R, a) \mapsto [\delta(p,a)]_R
\end{aligned}$$

Le quotient est plus petit, mais est-il ***le plus petit*** des quotients qui ***acceptent le même langage*** que M ?

Comparaison

équivalence sur Q
$\subseteq Q \times Q$

R

Myhill-Nerode pour $L \subseteq \Sigma^*$
$\subseteq \Sigma^* \times \Sigma^*$

$$M/_R = (Q/_R, \Sigma, \delta/_R, s/_R, F/_R)$$

$$\begin{aligned}
Q/_R &\triangleq \{ [p]_R \mid p \in Q \} \\
s/_R &\triangleq [s]_R \\
F/_R &\triangleq \{ [p]_R \mid p \in F \} \\
\delta/_R &: Q/_R \times \Sigma \to Q/_R \\
&([p]_R, a) \mapsto [\delta(p,a)]_R
\end{aligned}$$

$$M_R = (Q, \Sigma, \delta, s, F)$$

$$\begin{aligned}
Q &\triangleq \Sigma^*/_R = \{ [x]_R \mid x \in \Sigma^* \} \\
s &\triangleq [\epsilon]_R \\
F &\triangleq \{ [x]_R \mid x \in L \} \\
\delta &: \Sigma^*/_R \times \Sigma \to \Sigma^*/_R \\
&([x]_R, a) \mapsto [xa]_R
\end{aligned}$$

Automates isomorphes

Renommer les états
de manière compatible
avec les composants d'un automate

2.8.2 Définition Deux AFD $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ et $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ sont dits *isomorphes*, noté $M_1 \cong M_2$, s'il existe une application bijective $f : Q_1 \to Q_2$ telle que

1. $f(s_1) = s_2$.
2. $\forall q \in Q_1 . \forall a \in \Sigma . f(\delta_1(q,a)) = \delta_2(f(q), a)$
3. $q \in F_1$ ssi $f(q) \in F_2$.