Informatique théorique III
(Automates, langages & calculabilité)

Formulaire du Cours 2004-2005
Partie 6
EPFL – I&C

Uwe Nestmann
Sébastien Briais
Daniel C. Bünzli

14 février 2005

Bibliographie

[HMU03] John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. *Introduction to Automata Theory, Languages and Computation*. Pearson Education International, 2003. ISBN 0321210298.

[Koz97] Dexter Kozen. *Automata and Computability*. Springer Verlag New York Inc., 1997.

[Sch95] Uwe Schöning. *Theoretische Informatik — kurzgefaßt*. Spektrum Lehrbuch. Spektrum Akademischer Verlag, 1995. 2. Auflage.

Table des matières

$Id: notes-6.tex,v 1.14 2004/11/25 10:57:31 uwe Exp $

semaine 6

Voir [HMU03] §4.4, [Koz97] §15–16, [Sch95] §1.2.5.

2.7 Régularité par équivalences

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 $x\ R\ y$, alors $\forall a \in \Sigma\,.\, xa\ R\ ya$.
2. R raffine $L \times L$, c'est à dire : si $x\ R\ y$, 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 : $\#(\Sigma^*/_R) < \infty$.

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

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

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

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

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

2.7.5 Lemme Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD. Alors $\#(\Sigma^*/_{\equiv_M}) \subseteq \#(Q)$.

On a l'égalité $=$ à la place de l'inclusion $\subseteq$ si tous tes états de M sont accessibles (avec Définition 2.8.1 : si $M = \mathrm{acc}(M)$).

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

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

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

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.

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.

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

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

1. Un état $q \in Q$ est dit *accessible*
   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 :

$$
\begin{aligned}
Q' &\triangleq \{\, q \in Q \mid q \ \text{ accessible} \,\} \\
\delta' &\triangleq \delta \restriction_{Q' \times \Sigma} \\
F' &\triangleq F \cap Q'
\end{aligned}
$$

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 $f(q) \in F_2$.

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 :

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

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

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

2.8.5 Notation Lorsqu'il n'y a pas d'ambiguïté sur l'automate M dont on parle, on écrit $\approx$ à la place de $\approx_M$.

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

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

Il est alors possible de construire l'automate $M/_{\approx_M}$.

2.8.7 Lemme Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD et $M/_{\approx} = (Q/_{\approx}, \Sigma, \delta/_{\approx}, s/_{\approx}, F/_{\approx})$.

1. $p \in F$ ssi $[p]_{\approx} \in F/_{\approx}$.
2. Si $[p]_{\approx} = [q]_{\approx}$, alors $\forall a \in \Sigma \,.\, [\delta(p, a)]_{\approx} = [\delta(q, a)]_{\approx}$.
3. $\forall z \in \Sigma^* \,.\, \big(\widehat{\delta/_{\approx}}([p], z) = [\widehat{\delta}(p, z)]\big)$.
4. $\mathrm{L}(M/_{\approx}) = \mathrm{L}(M)$. □

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

2.8.9 Lemme Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD. Soit $p, q \in Q$ et $a \in \Sigma$.

1. Si $p \in F$ et $q \notin F$, alors $p \not\approx_M q$.
2. Si $\delta(p, a) \not\approx_M \delta(q, a)$, alors $p \not\approx_M q$.

2.8.10 Notation Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD. Soit $p, q \in Q$ et $a \in \Sigma$.
Nous écrivons $\{p, q\} \xrightarrow{a} \{p', q'\}$ si $\delta(p, a) = p'$ et $\delta(q, a) = q'$.
Nous définissons $\mathcal{P}_2(Q) \triangleq \{\, P \subseteq Q \mid \#(P) = 2 \,\}$.

En d'autres mots : $\mathcal{P}_2(Q)$ contient comme éléments tous les *couples non-ordonnés* $\{p, q\}$ avec $p, q \in Q$ et $p \neq q$. (Noter que $\{p, p\}$ dénote un ensemble de taille 1, donc $\{p, p\} \notin \mathcal{P}_2(Q)$.)

2.8.11 Définition (Algorithme de minimisation) Soit M un AFD.

1. Transformer M en $\mathrm{acc}(M) = (Q, \Sigma, \delta, s, F)$.
2. Calculer l'ensemble $R_{\surd} \subseteq \mathcal{P}_2(Q)$ par la procédure suivante :
   (a) Construire un tableau avec une entrée par élément de $\mathcal{P}_2(Q)$.
   (b) Marquer tous les couples $\{p, q\}$ tels que $p \in F$ et $q \notin F$:
   INIT Soit $p, q \in Q$.
   Si $p \in F$ et $q \notin F$, alors $\{p, q\} \in R_{\surd}$.
   (c) Répéter l'application de la règle suivante tant qu'elle permet de rajouter des nouveaux éléments à $R_{\surd}$:
   STEP Soit $a \in \Sigma$ et $p, p'q, q' \in Q$.
   Si $\{p, q\} \xrightarrow{a} \{p', q'\}$ et $\{p', q'\} \in R_{\surd}$, alors $\{p, q\} \in R_{\surd}$.

3. Posons $R \triangleq \{ (p, q) \mid \{ p, q \} \notin R_{\surd} \}$.
   Noter que $R \subseteq Q \times Q$ et que $\mathrm{Id}_Q \subseteq R$.
4. L'automate minimal est l'AFD $\mathrm{acc}(M)/_R$.

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

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