Informatique théorique III
(Automates, langages & calculabilité)

Formulaire du Cours 2004-2005
Partie 11
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.

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

Table des matières

Chapitre 4

Langages récursifs et enumérables

$Id: notes-11.tex,v 1.17 2005/01/13 09:28:28 uwe Exp $

semaine 11

4.1 Les machines de Turing

Voir [HMU03] §8 (en particulier §8.2).

4.1.1 Définition (MT) Une *machine de Turing* (MT) est un septuplet

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$$

où

- Q est un ensemble fini d'*états*,
- Σ est un ensemble fini de symboles, *l'alphabet d'entrée*,
- $\Gamma \supset \Sigma$ est un ensemble fini de symboles, *l'alphabet du ruban*,
- $\delta : (Q \times \Gamma) \to (Q \times \Gamma \times \{-1, 0, 1\})$ est la *fonction de transition*,
- $q_0 \in Q$ est l'*état initial* (ou de *départ*),
- $\mathrm{B} \in \Gamma \setminus \Sigma$ est le *symbole blanc*,
- $F \subseteq Q$ est l'ensemble des *états accepteurs* (ou *finaux*).

En général, δ est une fonction partielle.
Il existe aussi des versions des MT où δ est de type

$$\begin{aligned} \textit{multi/}k\textit{-rubans}: &\quad (Q \times \Gamma^k) \to (Q \times \Gamma^k \times \{-1, 0, 1\}^k) \\ \textit{non-déterministe}: &\quad (Q \times \Gamma) \to \mathcal{P}(Q \times \Gamma \times \{-1, 0, 1\}) \end{aligned}$$

4.1.2 Notation On adopte les conventions d'écriture suivantes :

- $p, q, \ldots \in Q$
- $a, b, \ldots \in \Sigma$ et $u, v \ldots \in \Sigma^*$
- $X, Y, Z \in \Gamma$ et $\alpha, \beta, \gamma \in \Gamma^*$

4.1.3 Notation (Graphe d'une MT) On utilise la représentation des AAP mais les flèches sont dessinées comme suit :

Si $\delta(q, X) = (q', Y, d)$,
on dessine une flèche étiquetée par $X/Y, d$ de q à q'.

4.1.4 Définition (Configurations et calculs)
Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une MT.

1. Une *configuration* de M est un triplet (q, i, rub) où
   - $q \in Q$ est l'état dans lequel se trouve la machine M,
   - $i \in \mathbb{Z}$ est l'emplacement courant de la tête de lecture/écriture, et
   - $\mathrm{rub} : \mathbb{Z} \to \Gamma$ est une application représentant l'état du ruban telle que $\mathrm{rub}(k) \neq \mathrm{B}$ pour au plus un nombre fini d'entiers k.

   Si $q \in F$, on dit que (q, i, rub) est une *configuration acceptante* de M.
2. Soit $c = (q, i, \mathrm{rub})$ une configuration de M et $X = \mathrm{rub}(i)$.
   Si $\delta(q, X) = (q', Y, d)$,
   alors la configuration c *se réduit en une étape de calcul*
   en la configuration $c' = (q', i + d, \mathrm{rub}\{i \mapsto Y\})$, noté $c \vdash_M c'$, avec

   $$\begin{aligned} \mathrm{rub}\{i \mapsto Y\} : \mathbb{Z} &\to \Gamma \\ k &\mapsto Y && \text{si } k = i \\ k &\mapsto \mathrm{rub}(k) && \text{sinon} \end{aligned}$$

   Comme d'habitude, on note $\vdash_M^*$ la fermeture réflexive et transitive de la relation $\vdash_M$ ainsi définie.
3. On identifie tout mot $w \in \Gamma^*$ avec le ruban $w : \mathbb{Z} \to \Gamma$ défini par :

   $$\begin{aligned} w : \mathbb{Z} &\to \Gamma \\ k &\mapsto (w)_k && \text{si } 1 \leq k \leq |w| \\ k &\mapsto \mathrm{B} && \text{sinon} \end{aligned}$$

   Autrement dit, le ruban w est un ruban rempli de caractères blancs si ce n'est que le mot w est écrit sur celui-ci à partir de la position 1.
4. Soit $w \in \Sigma^*$. Un *calcul* de M pour w est une séquence

   $$c_0 \vdash_M c_1 \vdash_M \ldots \vdash_M c_i \vdash_M \ldots$$

   où pour tout k, c_k est une configuration de M et $c_0 = (q_0, 1, w)$ est la configuration initiale du calcul.
5. Soit c un configuration de M. S'il n'existe aucune configuration c' telle que $c \vdash_M c'$, on note $c \not\vdash_M$.

Lorsque la machine M dont on parle est claire dans le contexte, on écrit simplement $\vdash$ à la place de $\vdash_M$.

4.1.5 Définition (Langage accepté par une MT)
Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une MT.

1. M accepte le mot $w \in \Sigma^*$
   s'il existe une configurations c acceptante telle que $(q_0, 1, w) \vdash_M^* c$.

2. Le *langage accepté par M* est défini par :

$$\mathrm{L}(M) \triangleq \{\, w \in \Sigma^* \mid w \text{ accepté par } M \,\}$$

4.1.6 Théorème Soit Σ un alphabet et $L \subseteq \Sigma^*$.
Alors les deux propositions suivantes sont équivalentes :

1. Il existe une MT M telle que $L = \mathrm{L}(M)$.
2. Il existe une grammaire G de type 0 telle que $L = \mathrm{L}(G)$.

4.2 Calculabilité

Voir [HMU03] §8.2.6 et [Sch95] §2.2.

4.2.1 Définition (Arrêt)
Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une MT.
On dit que M s'arrête pour $w \in \Sigma^*$
s'il existe une configuration c telle que $(q_0, 1, w) \vdash_M^* c \not\vdash_M$.
Si M s'arrête pour tout $w \in \Sigma^*$, alors M est dite *totale*.

4.2.2 Notation Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une MT.

1. Si l'on ne s'intéresse pas au langage accepté par une MT M, mais seulement à la question de l'arrêt, on peut poser $F = \emptyset$ ou bien omettre F.
2. Si l'on veut imposer qu'une MT s'arrête au moment où elle accepte (c-à-d, où elle passe dans un état accepteur), on peut restreindre le type de δ à

$$((Q \setminus F) \times \Gamma) \to (Q \times \Gamma \times \{-1, 0, 1\})$$

4.2.3 Définition (Turing-calculabilité)

1. Une fonction $f : \Sigma^* \to \Sigma^*$ est dite *Turing-calculable*,
s'il existe une MT $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, \emptyset)$
telle que pour tout $x, y \in \Sigma^*$

$$\begin{gathered} f(x) = y \\ \text{ssi} \\ (q_0, 1, x) \vdash^* (q, 1, y) \not\vdash \end{gathered}$$

2. Soit $\mathrm{rep}(n)$ la suite $1 \cdots 1$ de longueur $n \in \mathbb{N}$.
Une fonction $f : \mathbb{N}^k \to \mathbb{N}$ est dite *Turing-calculable*,
s'il existe une MT $M = (Q, \{1\}, \{1, \#\} \cup \Gamma, \delta, q_0, \mathrm{B}, \emptyset)$
telle que pour tout $n_1 \ldots, n_k, m \in \mathbb{N}$

$$\begin{gathered} f(n_1 \ldots, n_k) = m \\ \text{ssi} \\ (q_0, 1, \mathrm{rep}(n_1)\# \cdots \# \,\mathrm{rep}(n_k)) \vdash^* (q, 1, \mathrm{rep}(m)) \not\vdash \end{gathered}$$

4.2.4 Thèse (Church-Turing) Tout *algorithme*, c-à-d toute *procédure effective*, peut être représenté par une MT.

4.3 Décidabilité des langages

Voir [HMU03] §8.2.6 et §9.2.1–2.

4.3.1 Définition Soit L un langage sur l'alphabet Σ.

1. L est dit *décidable* (ou : *récursif*)
   s'il existe une MT M telle que $\mathrm{L}(M) = L$ et M est totale.
2. L est dit *semi-décidable* (ou : *récursivement énumerable*, ou : *r.e.*)
   s'il existe une MT M telle que $\mathrm{L}(M) = L$.
3. L est dit *co-semi-décidable* (ou : *co-r.e.*)
   si son complément $\overline{L}$ est semi-décidable.
4. L est dit *indécidable*
   si L n'est pas décidable.
5. L est dit *non-semi-décidable* (ou : *non-r.e.*)
   si L n'est pas semi-décidable.

4.3.2 Proposition Soit L un langage sur l'alphabet Σ.

1. Si L est décidable, alors son complément $\overline{L}$ est aussi décidable.
2. Si L et $\overline{L}$ sont semi-décidables, alors L est décidable.