Informatique théorique III
(Automates, langages & calculabilité)

Formulaire du Cours 2004-2005
Partie 2
EPFL – I&C

Uwe Nestmann
Sébastien Briais
Daniel C. Bünzli

14 février 2005

Table des matières

$Id: notes-2.tex,v 1.26 2005/01/28 10:25:22 uwe Exp $

semaine 2
lundi

1.4 La hiérarchie de Chomsky

1.4.1 Définition (Grammaire générative)
Une grammaire est un quadruplet $G \triangleq (V, \Sigma, P, S)$.

- V est un alphabet non vide de symboles *non-terminaux*.
- Σ est un alphabet non vide de symboles *terminaux* disjoint de V ($V \cap \Sigma = \emptyset$).
- $P \subseteq (\Gamma^+ \times \Gamma^*)$ (avec $\Gamma \triangleq V \cup \Sigma$) est un ensemble fini de *productions*.
- $S \in V$ est le *symbole de départ*. □

Nous utilisons des lettres grecques minuscules $\alpha, \beta, \ldots$ pour les éléments de $(V \cup \Sigma)^*$. Pour spécifier une production $(\alpha, \beta) \in P$, on note $\alpha \to \beta$ ou encore $\alpha \to_G \beta$.

1.4.2 Définition (Dérivation directe) Soit $G \triangleq (V, \Sigma, P, S)$ une grammaire. La relation de *dérivation directe* dans G entre deux mots $\alpha, \beta \in (V \cup \Sigma)^*$ est définie par,

$$(\alpha \Rightarrow_G \beta) \quad \Leftrightarrow \quad \exists \alpha', \beta', \gamma, \gamma' \in (V \cup \Sigma)^* : \begin{cases} \alpha' \to_G \beta' \\ \alpha = \gamma \alpha' \gamma' \\ \beta = \gamma \beta' \gamma' \end{cases}$$

On dit alors que β est *directement dérivable* de α dans G. □

1.4.3 Définition (Dérivation) Soit $G \triangleq (V, \Sigma, P, S)$ une grammaire. On définit la relation de dérivation $\Rightarrow_G^*$ comme étant la fermeture réflexive et transitive de $\Rightarrow_G$.

Une *dérivation* de α_0 en α_n dans G est une séquence finie $(\alpha_i)_{i \in [0,n]}$ d'éléments de $(V \cup \Sigma)^*$ telle que $\forall i \in [\![0, n-1]\!] : \alpha_i \Rightarrow_G \alpha_{i+1}$. On note généralement cette séquence $\alpha_o \Rightarrow_G \alpha_1 \Rightarrow_G \ldots \Rightarrow_G \alpha_n$. □

1.4.4 Définition (Langage d'une grammaire) Soit $G = (V, \Sigma, P, S)$ une grammaire.

1. Un mot $v \in \Sigma^*$ est généré par G ssi $S \Rightarrow_G^* v$. Dans ce cas on dit que v est un mot *terminal* de la grammaire.
2. Le langage généré par G est défini par :

$$\mathrm{L}(G) \triangleq \{ v \in \Sigma^* \mid S \Rightarrow_G^* v \}$$

1.4.5 Définition (Type d'une grammaire) Soit $G = (V, \Sigma, P, S)$ une grammaire. Un grammaire est dite de

Type 0 dans tous les cas.

Type 1 si pour toute production $\alpha \to_G \beta \in P$ on a,

1. $|\alpha| \leq |\beta|$
2. ou $\alpha = S \wedge \beta = \epsilon$.

La grammaire G est dite *contextuelle*.

Type 2 si pour toute production $\alpha \rightarrow_G \beta \in P$ on a $\alpha \in V$. La grammaire est dite *libre de contexte* ou *non-contextuelle*.

Type 3 si toutes les production $\alpha \rightarrow_G \beta \in P$ sont de la forme,

1. $A \rightarrow_G wB$
2. $A \rightarrow_G w$

avec $A, B \in V$ et $w \in \Sigma^*$. La grammaire est dite *régulière*.

Avec cette définition, on parle de grammaires *linéaires à droite*. Si l'on remplace ci-dessus wB par Bw, on parle de grammaires *linéaires à gauche*. Ces deux définitions sont équivalentes, un langage générable par une grammaire linéaire à droite l'est aussi par une grammaire linéaire à gauche, et vice-versa.

1.4.6 Définition L'ensemble des langages générés par les grammaires d'un type i est donné par :

$$\mathcal{L}_i = \{\mathrm{L}(G) \mid G \text{ est de type } i\}\,.$$

Un type de grammaire i est *inclus* dans un type j si $\mathcal{L}_i \subseteq \mathcal{L}_j$. □

1.4.7 Définition (Type d'un langage) Un langage L est de type i si $L \in \mathcal{L}_i$ et L est *strictement* de type i si $L \in \mathcal{L}_i \setminus \mathcal{L}_{i+1})$, c'est à dire si L est généré par une grammaire de type i et par aucune grammaire de type $i+1$.

1.4.8 Théorème (Hiérarchie de Chomsky)
La relation entre les types de langages est :

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$$

Les inclusions sont strictes.

Étant donné une grammaire $G = (V, \Sigma, P, S)$ de type 1, il existe un algorithme qui permet de décider en temps fini si un mot $x \in \Sigma^*$ appartient au langage $\mathrm{L}(G)$ ou non.

Chapitre 2

Langages réguliers

semaine 2
mercredi

2.1 Automates finis déterministes

2.1.1 Définition Un *automate fini déterministe (AFD)* est un quintuplet

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

où

- Q est un ensemble fini d'*états*,
- Σ est un ensemble fini de symboles, *l'alphabet (d'entrée)*,
- $\delta : Q \times \Sigma \to Q$ est la *fonction (totale) de transition*,
- $s \in Q$ est l'*état initial* (ou de *départ*),
- $F \subseteq Q$ est un sous-ensemble de Q, les *états accepteurs* (ou *finaux*).

2.1.2 Notation Un automate $(Q, \Sigma, \delta, s, F)$ peut aussi être défini en spécifiant son alphabet $\Sigma = \{ s_1, \ldots, s_n \}$, son ensemble d'état $Q = \{ q_1, \ldots, q_m \}$ *et* l'un des éléments suivants :

- un *graphe* :
  1. Pour tout état $q \in Q$ on dessine un noeud dans le graphe (q entouré d'un cercle).
  2. Pour toute transition $\delta(q, s) = q'$ on dessine une flèche étiquettée par s allant du noeud q à q'
  3. L'état initial s est distingué par une flèche dont la queue n'est attachée à aucun état et dont la tête pointe sur s.
  4. Pour tout état accepteur $q \in F$, on dessine un second cercle autour de son noeud correspondant.

- un *tableau* de la forme :

δ	s_1	$\cdots$	s_i	$\cdots$	s_n
q_1	$\delta(q_1, s_1)$	$\cdots$	$\delta(q_1, s_i)$	$\cdots$	$\delta(q_1, s_n)$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$F\, q_j$	$\delta(q_j, s_1)$	$\cdots$	$\delta(q_j, s_i)$	$\cdots$	$\delta(q_j, s_n)$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$SF\, q_m$	$\delta(q_m, s_1)$	$\cdots$	$\delta(q_m, s_i)$	$\cdots$	$\delta(q_m, s_n)$

où S et F sont utilisés pour marquer les états initiaux (ici : q_m) et finaux (ici : $\{ q_j, q_m \}$).

2.1.3 Définition (Fonction de transition sur les mots) Étant donné un AFD $(Q, \Sigma, \delta, s, F)$, on étend la fonction de transition δ en une fonction de transition $\widehat{\delta} : Q \times \Sigma^* \to Q$ sur les mots définie par :

$$\begin{array}{rcl} \widehat{\delta}(q, \epsilon) & \triangleq & q \\ \widehat{\delta}(q, wa) & \triangleq & \delta(\widehat{\delta}(q, w), a) \end{array}$$

2.1.4 Définition (Langage accepté par un automate) Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.

1. M *accepte* le mot $w \in \Sigma^*$ ssi $\widehat{\delta}(s, w) \in F$. Dans le cas contraire on dit que M *rejette* w.
2. Le *langage accepté par* M est défini par :

$$\mathrm{L}(M) \triangleq \{ w \in \Sigma^* \mid \widehat{\delta}(s, w) \in F \}$$

2.1.5 Définition (Langage régulier) Un langage $L \subseteq \Sigma^*$ est dit *régulier* s'il existe un automate fini M tel que $\mathrm{L}(M) = L$.

2.1.6 Théorème Soit L un langage sur Σ. Les deux propositions suivantes sont équivalentes :

1. Il existe un AFD $M = (Q, \Sigma, \delta, s, F)$ avec $\mathrm{L}(M) = L$.
2. Il existe une grammaire régulière $G = (V, \Sigma, P, S)$ avec $\mathrm{L}(G) = L$.