Rédigé avec LaTeX
Version du 6 novembre 2013

Langages et automates

L3 Informatique
Semestre 5

Cours donné par Christine Maurel – JP Arcangeli
Rédigé par Antoine de Roquemaurel

2013

Avant-Propos

- 3 séances CM
- 8 séances JPA
- 4 séances CM

MCC

Contrôle continue 30%

Contrôle terminal 70%

Objectifs Avoir les bases pour aborder la compilation lors du S7 en master.

Le but est de nous apprendre ce qu'est un **automate** et ce qu'est une **grammaire**.

Table des matières

Chapitre 1

Introduction

L'information doit être exprimée à l'aide d'une *grammaire* et reconnue, vérifiée grâce aux *automates*.

Les langages sont plus ou moins compliqué, il en existe plusieurs types :
- les langages les plus simples sont les langages *régulier*, soit une grammaire *linéaire à droite* et un automate fini.
- Les langages plus complexes sont les langages hors contexte lié à une grammaire hors contexte et un automate à pile.
- Langages sensible au contexte, avec une grammaire sensible au contexte et une machine de Turing

> R langage régulier $\subset$ langage hors contexte $\subset$ langage sensible au contexte

Chapitre 2

Langages et grammaires

2.1 Définitions

Langage Un langage est engendré par une grammaire et reconnu par un automate

Alphabet Ensemble fini non vide de symboles tous différents

> $X = \{0, 2, 4, 6, 8\}$ $X = \{do, re, mi, fa, sol, la, si\}$ $X = \{if, then, else\}$

Mot Sur un alphabet X une suite finie, éventuellement vide de symboles de X

> $a_1 = 2246$ $a_2 = fasolsi$

Longueur d'un mot ω $|\omega|$ =Nombre de symboles de x qui contient ω

Mot vide λ tel que $|\lambda| = 0$ $\forall x,\ \lambda \notin x$

Nombre d'occurences d'un sybole $s \in x$ dans un mot *omega* $|\omega|s$

2.2 Opérations sur les mots

Concaténation de mots Soient X un alphabet et deux mots ω_1 et ω_2.
$\omega = x_1 x_2 \dots x_n, x_1 \in x \forall i \in [1, n]$

- La concaténation n'est pas commutative.
- La concaténation est associative

R λ est l'élément neutre de la concaténation

Puissance soit $w \in X$, $w^n = \lambda$ si $n = 0$ et $w^n = w.w^{n-A}$ pour $n > 0$.

R L'ensemble X^* de mots construits sur X muni de la concaténation . est un monoïde libre ayant X pour générateur.

2.3 Langage L

Un langage est un ensemble de mots. $L \subseteq X^*$. $X = \{a, b, c\} X^*$
L = ensemble de mots sur qui commencent par a = $\{w \in X^* / w = a.w', w' \in X^*$

2.4 Opérations sur les langages

Les opérateurs ensemblistes fonctionnent sur les langages, mais également une opération induite par la concaténation : le produit.

Soit X un alphabet. $L_1 \subseteq X^*, L_2 \subseteq X^*$.
- $L_1 \cup L_2 = L_1 + L_2 = \{w \in X^*/w \in L_1 ou w \in L_2\}$
- $L_1 \cap L_2 = \{w \in X^*/w \in L_1 et w \in L_2\}$
- $\overline{L_1} = X^* - L_1 = \{w \in X^*/w \notin L_1\}$
- Produit $L_1.L_2 = L_1L_2 = \{w \in X^*/\exists w_1 \in_1 \exists w_2 \in L_2$ tel que$w = w_1.w_2$

> - Le produit de langages n'est pas commutatif.
> - Le produit de langages est associatif

- $L \subseteq X^*$ Langage
  $L^* = \bigcup_{n>=0} L^n$ avec L^n tel que $L^0 = \lambda$

$$L^n \begin{cases} si \quad n = 0 & L^0 = \{\lambda\} \\ \quad\quad n > 0 & L^n = L.L^{n-1} \end{cases}$$

Chapitre 3

Grammaire

Dérivation, arbre de dérivation, ambigüité, langage engendré, classification

Definition 3.1 Moyen précis, concis, explicite pour exprimer comment sont construit les mots d'un langage. Une grammaire $G =< N, X, P, S >$

- X : alphabet, ensemble de terminaux(minuscule).
- N : ensemble de non terminaux (majuscule)
- S : axiome, $S \in N$
- P : ensemble de règle de production (réécriture) = $\{\lambda \to \beta, \lambda \in (NUX)^+, \beta \in (NUX)^*\}$

$$
\begin{aligned}
G'_1 &= < N, X, P, S > \\
N &= \{S, A\} X \{a, b, c\} \\
P &= \{S \to aAc \\
& \quad A \to bAb \\
& \quad A \to b\}
\end{aligned}
$$

3.1 Dérivation

$$
\begin{aligned}
G &= < N, X, P, S > \\
W &\in (NUX)^+ \\
w_1 &\in (NUX)^*
\end{aligned}
$$

w se dérive en w_1. Si $\exists x, y \in (NUX)^*$ tel que $w = xuy$ et $w_1 = xvy$ avec $u \to v \in p$

w se dérive en plusieurs étapes en w.

Definition 3.2 Un arbre de dérivation est un outil visuel pour exprimer la dérivation des mots.

Soit G une grammaire $=< N, X, P, S >$, le langage engendré par $G = L(G)$ est = l'ensemble de tous les mots de X^* que l'on peut engendré à a partir de l'axiome S.

$L(G) = \{w \in X^*/S \Rightarrow^* w\}$