Universitetet i Oslo
Institutt for Informatikk
PMA
Martin Steffen

INF 5110: Compiler construction

Spring 2018 — **Handout 3** — 15. 1. 2018

Handout 3: Grammars, top-down parsing

Issued: 15. 1. 2018

The handout contains basic definitions. Additionally, for reference, to follow the slides during the lecture, the handout includes some grammars we repeteadly used for illustration, especially for the LR-parsing principles and the construction of the NFA/DFA

Some definitions

Definition 1 (CFG) A *context-free grammar* G is a 4-tuple $G = (\Sigma_T, \Sigma_N, S, P)$:

1. 2 disjoint finite alphabets of terminals Σ_T and
2. non-terminals Σ_N
3. 1 start-symbol $S \in \Sigma_N$ (a non-terminal)
4. productions P = finite subset of $\Sigma_N \times (\Sigma_N + \Sigma_T)^*$

Definition 2 The language of a CFG G, written as $\mathcal{L}(G)$, is defined as follows:

$$\mathcal{L}(G) = \{s \mid \mathit{start} \Rightarrow^* s \text{ and } s \in \Sigma_T^*\}$$

Definition 3 (Ambiguous grammar) A grammar is *ambiguous* if there exists a word with *two different* parse trees.

	rule format	languages	machines	closed
3	$A \to aB$, $A \to a$	regular	NFA, DFA	all
2	$A \to \alpha_1 \beta \alpha_2$	CF	pushdown automata	$\cup$, $*$, $\circ$
1	$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$	context-sensitive	(linearly restricted automata)	all
0	$\alpha \to \beta$, $\alpha \neq \epsilon$	recursively enumerable	Turing machines	all, except complement

The table uses the following conventions

- terminals $a, b, \ldots \in \Sigma_N$,

- non-terminals $A, B, \ldots \in \Sigma_T$
- general words $\alpha, \beta \ldots \in (\Sigma_T \cup \Sigma_N)^*$

Some grammars

$$\begin{array}{rcl} E' & \to & E \\ E & \to & E + \textbf{number} \mid \textbf{number} \end{array}$$

Table 1: Simple grammar for addition

$$\begin{array}{rcl} S' & \to & S \\ S & \to & (\, S \,)\, S \mid \epsilon \end{array}$$

Table 2: Grammar for parentheses

$$A \to (\, A \,) \mid \mathbf{a}$$

Table 3: Grammar for simplistic parentheses