



INF 5110: Compiler construction

Spring 2023

Handout 3

17. 2. 2023

Handout 3: Grammars

Issued: 17. 2. 2023

The handout contains basic definitions. Additionally, for reference, to follow the slides during the lecture, the handout includes some grammars we repeatedly 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:

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

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

	rule format	languages	machines	closed
3	$A \rightarrow aA, A \rightarrow a$	regular	NFA, DFA	all
2	$A \rightarrow \alpha_1\beta\alpha_2$	CF	pushdown automata	$\cup, *, \circ$
1	$\alpha_1A\alpha_2 \rightarrow \alpha_1\beta\alpha_2$	context-sensitive	(linearly restricted automata)	all
0	$\alpha \rightarrow \beta, \alpha \neq \epsilon$	recursively enumerable	Turing machines	all, except complement

The table uses the following conventions

- terminals $a, b, \dots \in \Sigma_T$,

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

Some grammars

$$\begin{aligned} E' &\rightarrow E \\ E &\rightarrow E + \mathbf{number} \mid \mathbf{number} \end{aligned}$$

Table 1: Simple grammar for addition

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow (S)S \mid \epsilon \end{aligned}$$

Table 2: Grammar for parentheses

$$A \rightarrow (A) \mid \mathbf{a}$$

Table 3: Grammar for simplistic parentheses