Introduction to Language Theory and Compilation: Exercises

Session 3: Introduction to grammars

Definition

A grammar is a tuple $G = \langle V, T, P, S \rangle$ where

- V is the set of *nonterminals* (or *variables*);
- T is the set of *terminals*;
- P is the set of *production rules*. In the general case:

$$P \subseteq \underbrace{(V \cup T)^* V (V \cup T)^*}_{\text{at least one variable}} \times (V \cup T)^*$$

- $S \in V$ is the *start symbol*.

ULB

Example

Let G be a grammar whose production rules in P are:

$$\begin{array}{lcl} S & \rightarrow & aSbS \\ S & \rightarrow & bSaS \\ S & \rightarrow & e \end{array}$$

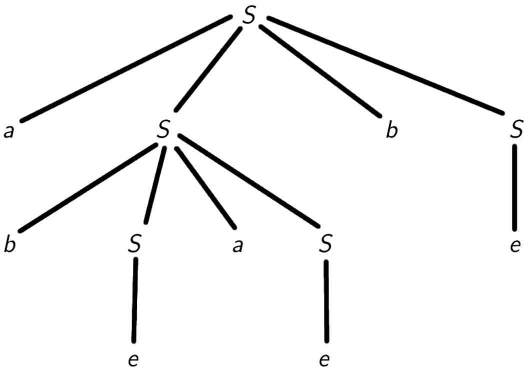
The rules can also be written in a more compact form:

$$S \rightarrow aSbS \mid bSaS \mid e$$

S is the only variable (or nonterminal) and is also the start symbol. We have $T = \{a, b, e\}$.

Parse tree (or derivation tree)

The string `abeaebe` can be parsed using G and is thus part of the language defined by the grammar.



Class 0: Unrestricted grammars

No restrictions on the structure of production rules.

$\Longrightarrow$ A bag of production rules.

Chomsky hierarchy – class 1/3

Class 1: Context-sensitive grammars Every production rule must have the following structure:

$$\alpha \to \beta$$

with $|\alpha| \leqslant |\beta|$. As an exception, the following rule may be part of the grammar as well:

$$S \to \varepsilon$$

where S is the start symbol. This rule is only allowed if S never appears on the right hand of any production rule.

$\Longrightarrow$ A starting production rule S and each production rule is composed of a pair of ordered sets (left,right) where $|\ \textit{left}\ | \leqslant |\ \textit{right}\ |$.

ULB

Chomsky hierarchy – class 2/3

Class 2: Context-free grammars (CFG) Every rule must obey the following structure:

$$A \rightarrow \alpha$$

$\Longrightarrow$ A starting production rule S and each production rule is composed of a pair of ordered sets (left,right) where $\mid left \mid = 1$

Chomsky hierarchy – class 3/3

Class 3: Regular grammars Two subclasses:

Right linear grammars Rules must obey this structure:

$$A \to wB \quad \text{or} \quad A \to w \qquad (w \in T^*)$$

Left linear grammars Rules must obey this structure:

$$A \to Bw \quad \text{or} \quad A \to w \qquad (w \in T^*)$$

$\Longrightarrow$ A starting production rule S and each production rule is composed of a pair of ordered sets (left,right) where $\mid \textit{left} \mid = 1$ and *right* may contain **some terminals** but at most **one starting/ending variable**.

Chomsky – classes summary

Class 0: Unrestricted grammars A bag of production rules

Class 1: CS grammars A starting production rule S and each production rule is composed of a pair of ordered sets (left,right) where $\mid left \mid \leqslant \mid right \mid$.

Class 2: CF grammars A starting production rule S and each production rule is composed of a pair of ordered sets (left,right) where $\mid left \mid = 1$

Class 3: RE grammars A starting production rule S and each production rule is composed of a pair of ordered sets (left,right) where $\mid left \mid = 1$ and *right* may contain some terminals but at most one starting/ending variable

ULB

Exercise 1

Informally describe the languages generated by the following grammars and also specify what kind of grammars they are:

(a)
$$
\begin{array}{rcl}
S & \rightarrow & abcA \\
 & & Aabc \\
A & \rightarrow & \varepsilon \\
Aa & \rightarrow & Sa \\
cA & \rightarrow & cS
\end{array}
$$

(b)
$$
\begin{array}{rcl}
S & \rightarrow & 0 \\
 & & 1 \\
 & & 1S
\end{array}
$$

(c)
$$
\begin{array}{rcl}
S & \rightarrow & a \\
 & & *SS \\
 & & +SS
\end{array}
$$

Exercise 2

Let G be the following grammar:

S	$\rightarrow$	AB
A	$\rightarrow$	Aa
A	$\rightarrow$	bB
B	$\rightarrow$	a
B	$\rightarrow$	Sb

1. Is G a regular grammar?
2. Give the *parse tree* for
   a) *baabaab*
   b) *bBABb*
   c) *baSb*
3. Give the *leftmost* and *rightmost derivations* for *baabaab*

ULB

Exercise 3

3. Write a context-free grammar that generates all strings of *a*s and *b*s (in any order) such that there are more *a*s than *b*s. Test your grammar on the input *baaba* by giving a derivation.

Exercise 4

4. Write a context-sensitive grammar that generates all strings of as, bs and cs (in any order) such that there are as many of each. Give a derivation of $cacbab$ using your grammar.