Theoretical Computer Science

Winter semester 2021/22

Prof. Dr. Georg Schied

Assignment 4

Deadline: Wednesday, 3 November 2021

- 10 out of 20 points must be achieved in order to pass.

Exercise 4.1 Exercise 4.2 – obligatory (6 points)

Let the following context-free grammar *G* with start symbol *S* be given:

$S \rightarrow BA$

$A \rightarrow cA \mid dA \mid \varepsilon$

$B \rightarrow aBbb \mid c$

a) Which of the following words are derivable? If possible, give a *derivation* for each.

(1) **cdc**

(2) **acbb**

(3) **accba**

b) Give a *derivation tree* for the string **aacbbbbcd**.

c) Give a *rightmost derivation* for the string **aacbbbbcd**.

d) (optional, +2 bonus points) Which *language* is generated by the grammar *G*?

Exercise 4.3

Define a context-free grammar for the language of the following regular expression:

$a(b|cc)^* (a|\varepsilon)$

Exercise 4.4 – obligatory (6 points)

a) Define a context-free grammar for the language *L*

$L = \{ dc^{2n}ba^n \mid n \geq 0 \} \cup \{ ac^kb \mid k > 0 \}$.

b) Specify a context-free grammar for logical expressions. The expressions can be build from the variable `x`, `y`, `z`, and the logical operators `!`, `&&`, and `||` (similar to Java). Any (sub-)expression can be encompassed by parenthesis. Here are some examples:

```
x
!z
!y || x
(x)
!(z && y || z)
(x || !( y && !!z))
```

Tip: You can learn from the grammar for arithmetic expressions.

Exercise 4.5 (tricky!)

Let $\Sigma = \{0,1\}$. Specify context-free grammars for the following languages:

a) $L_1 = \{ 0^n 1^m \mid n \leq m \}$

b) $L_2 = \{ 0^n 1^{n+m} 0^m \mid n > 0, m > 0 \}$

c) $L_3 = \{ w \in \Sigma^* \mid |w|_0 = |w|_1 \}$, i.e. the language of all strings that contain the same number of zeros and ones.

Exercise 4.6 - obligatory (3 points)

Let the following context-free grammar *G* with start symbol *S* be given:

S → SaS | B

B → b | c

a) Draw a *parse tree* for the string **`bacab`**.

b) Show that grammar *G* is *ambiguous*.

Exercise 4.7

The Ada 2012 programming language reference manual defines the syntax of if statements using EBNF:

If_Statement → **if** *Condition* **then** *Sequence_of_Statements*
(**elsif** *Condition* **then** *Sequence_of_Statements*)*
[**else** *Sequence_of_Statements*]
end if ;

Sequence_of_Statements → *Statement* +

Terminal symbols are printed as boldface text. *Condition* and *Statement* are non-terminal symbols defined elsewhere. Square brackets enclose optional items.

Draw a *syntax diagram* (*railroad diagram*) for this section of the Ada grammar.

Exercise 4.8

Transform the following *EBNF grammar* into a *context-free grammar*.

S → (aSb)* [Ac]

A → (ab | Ad) bb

Exercise 4.9 - obligatory (5 points)

The following *EBNF grammar* is given, where [...] indicates optional parts:

S → aAbA*[cA]a

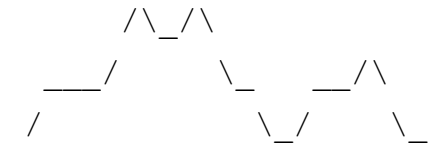
A → d^+e

a) Represent the grammar as a *syntax diagram* (*railroad diagram*).

b) Convert the EBNF grammar to an equivalent *context-free grammar*.

Exercise 4.10 (tricky!)

Mountain panoramas like the following



can be described as strings consisting of the characters `/`, `_`, and `\`. The example given above is represented by the string `/___//\_/\\_\_/__/\\_` .

Each panorama has to fulfill the following properties:

1. The panorama starts and ends at the same base level.
2. No part of the panorama might be below this base level.
3. The empty string is a valid panorama and also "planes" _____ of arbitrary length are allowed.

Define the syntax of all possible mountain panoramas using EBNF.