T-79.148 **Spring 2002**
Introduction to Theoretical Computer Science
Tutorial 7, 12–14 March
Problems

Homework problems:

1. *(Leftover problem on preceding week's material.)* Design a right-linear grammar that generates the language $\{w \in \{a,b\}^* \mid w \text{ does not contain the substring } abaa\}$. (Cf. Problem 4/2.)

2. Design a context-free grammar describing balanced sequences of parentheses that may also contain parallel subexpressions, e.g. "(()())()" or "()()()". Based on your grammar, give the leftmost and rightmost derivations and the parse trees for the above sequences. Is your grammar ambiguous or unambiguous?

3. (a) Prove that the following grammar is ambiguous:

$$S \to aSb \mid aaSb \mid \varepsilon$$

   (b) Describe the language generated by the above grammar in words, and design an unambiguous grammar generating the same language.

Demonstration problems:

4. Prove that the class of context-free languages is closed under unions, concatenations, and the Kleene star operation, i.e. if the languages $L_1, L_2 \subseteq \Sigma^*$ are context-free, then so are the languages $L_1 \cup L_2$, L_1L_2 and L_1^*.

5. (a) Prove that the following context-free grammar is ambiguous:

$$\begin{aligned} S &\to \textbf{if } b \textbf{ then } S \\ S &\to \textbf{if } b \textbf{ then } S \textbf{ else } S \\ S &\to s. \end{aligned}$$

   (b) Design an unambiguous grammar that is equivalent to the grammar in item (a), i.e. that generates the same language. (*Hint:* Introduce a new nonterminal S' that generates only "balanced" **if-then-else**-sequences.)

6. Design a recursive-descent (top-down) parser for the grammar from Problem 6/6.