T-79.148 **Spring 2003**
Introduction to Theoretical Computer Science
Tutorial 7, 3 – 5 Mar
Problems

Homework problems:

1. Design a right-linear grammar that generates the language

$$\{a^m b^n \mid m = n \pmod 3\}.$$

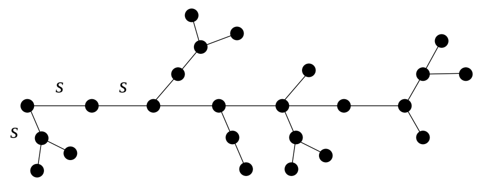
(Cf. Demonstration Problem 3/5a.)

2. Consider the following grammar generating a certain type of list structures:

$$S \to (S) \mid S, S \mid a.$$

   (a) Based on the above grammar, give a leftmost and rightmost derivation and a parse tree for the sentence "$(a, (a))$".
   (b) Prove that the grammar is ambiguous.
   (c) Design an unambiguous grammar generating the same language.

3. A *fern* consists of a stem and a number of subferns rooted on the left and right sides of the stem. For instance, the following structure is a fern:



A fern structure can be described by a string where each unit of the stem is denoted by a letter s, and each subfern is described by a similar string in parentheses, located at the point where the subfern is rooted, and prefixed by l or r depending on whether the subfern occurs on the left or right side of the main stem, respectively. At most one subfern can be rooted to the left and to the right at each point, and each subfern must contain at least one stem unit. For instance, the string representation corresponding to the above example would be:

$$r(sl(s)r(s))ssl(ssl(s)r(s))sr(ss)sl(s)r(sl(s)r(s))ssl(sr(s)s)r(s).$$

Design a context-free grammar describing the structure of ferns. (I.e. the grammar should generate all and only the valid fern strings.) Give parse trees according to your grammar for the ferns $sl(s)r(s)$ and $r(sl(s)r(s))l(s)$. Is your grammar ambiguous or unambiguous?

PLEASE TURN OVER

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 $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

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

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

   (b) Design an unambiguous grammar that is equivalent to the grammar in item (a), i.e. that generates the same language. (*Hint:* Introduce new nonterminals B and U that generate, respectively, only "balanced" and "unbalanced" **if-then-else-**sequences.)

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