T-79.1001 **Spring 2006**
Introduction to Theoretical Computer Science T
Tutorial 7, 21–22 March
Problems

Homework problems:

1. Convert the following grammar for certain type of list structures,

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow N \mid \varepsilon \\ N &\rightarrow S, N \mid S \end{aligned}$$

into Chomsky normal form.

2. Determine, using the CYK algorithm ("dynamic programming method", Sipser p. 241, Lewis & Papadimitriou p. 155), whether the strings *abba*, *bbaa* and *bbaab* are generated by the grammar

$$\begin{aligned} S &\rightarrow AB \mid BA \mid a \mid b \\ A &\rightarrow BA \mid a \\ B &\rightarrow AB \mid b \end{aligned}$$

In the positive cases, give also the respective parse trees.

3. Design pushdown automata recognising the following languages:
   (a) $\{w \in \{a,b\}^* \mid w = w^R\}$;
   (b) The language generated by grammar

$$S \rightarrow (S) \mid S, S \mid a$$

   (Cf. Tutorial 6, Problem 1.)

Demonstration problems:

4. Design an algorithm for testing whether a given a context-free grammar $G = (V, \Sigma, P, S)$, generates a nonempty language, i.e. whether any terminal string $x \in \Sigma^*$ can be derived from the start symbol S.

5. Design a pushdown automaton corresponding to the grammar $G = (V, \Sigma, P, S)$, where

$$\begin{aligned} &V = \{S, (,), {}^*, \cup, \emptyset, a, b\} \\ &\Sigma = \{(,), {}^*, \cup, \emptyset, a, b\} \\ &P = \{S \rightarrow (SS), S \rightarrow S^*, S \rightarrow (S \cup S), \\ &\qquad S \rightarrow \emptyset, S \rightarrow a, S \rightarrow b\} \end{aligned}$$

6. Design a grammar corresponding to the pushdown automaton $M = (Q, \Sigma, \Gamma, \Delta, s, F)$, where

$$\begin{aligned} Q =& \{s, q, f\},\ \Sigma = \{a, b\},\ \Gamma = \{a, b, c\},\ F = \{f\}, \\ \Delta =& \big\{ \big((s, e, e), (q, c)\big), \big((q, a, c), (q, ac)\big), \big((q, a, a), (q, aa)\big) \\ & \big((q, a, b), (q, e)\big), \big((q, b, c), (q, bc)\big), \big((q, b, b), (q, bb)\big) \\ & \big((q, b, a), (q, e)\big), \big((q, e, c), (f, e)\big) \big\} \end{aligned}$$