T-79.1001 **Spring 2006**
Introduction to Theoretical Computer Science T
Tutorial 12, 2–3 May
Problems

This is the last set of tutorial problems. The course exam is on Friday 19 May, 3–6 p.m. Check the time and place still from the departmental exam schedule, remember to register for the exam via TOPI, and note that all your Regis-assignments must be completed by the exam date — otherwise your exam paper will not be graded (really!). Please fill in also the computerised course feedback form; this earns you an extra exam point that counts towards increasing your grade in case you pass the course. (The questionnaire opens on 5 May and closes on 22 May. Further instructions available on the course's web page.)

Homework problems:

1. With respect to the following claims, list in order whether each one of them is *true (T)* or *false (F)*.
   (a) The union of any two regular languages is context-free.
   (b) Every language that can be recognised by a nondeterministic pushdown automaton can be generated by a context-free grammar.
   (c) Every language that can be recognised by a deterministic pushdown automaton can be described by a regular expression.
   (d) There exist nonrecursive (i.e. "undecidable") context-free languages.
   (e) Nondeterministic Turing machines recognise ("accept", "semidecide") exactly the recursively enumerable languages.
   (f) The language $\{a^n b^n \mid n \geq 0\}$ can be recognised on a nondeterministic finite automaton.
   (g) The complement of every recursive ("decidable") language is recursively enumerable ("Turing-recognisable", "semidecidable").
   (h) The computation of a deterministic Turing machine terminates on every input.

2. Design unrestricted grammars (general rewriting systems) that generate the following languages:
   (a) $\{w \in \{a, b, c\}^* \mid w \text{ contains equally many } a\text{'s, } b\text{'s and } c\text{'s}\}$,
   (b) $\{ww \mid w \in \{a, b\}^*\}$.

3. Design a Turing machine that recognises the language $\{a^{2k} \mid k \geq 0\}$, and based on that design an unrestricted grammar whose derivations simulate the computations of your Turing machine. (For the general construction see e.g. the textbook by Lewis & Papadimitriou (2nd Ed. 1998), pp. 230–231, the textbook

by Hopcroft & Ullman (1979), pp. 222-223, or search the Internet using keywords "from Turing machines to grammars".) Give a derivation for the sentence aa in your grammar, and explain why the string aaa cannot be derived in it.

Demonstration problems:

4. Show that all context-sensitive languages can be recognised by linear-bounded automata. (Make use of the fact that in applying the grammar's production rules, the length of the sentential form under consideration can never decrease, except in the special case of the empty string.) Deduce from this result the fact that all context-sensitive languages are recursive.

5. Show that every language generated by an unrestricted grammar can also be generated by a grammar where no terminal symbols occur on the left hand side of any production.

6. Show that every context-sensitive grammar can be put in a normal form where the productions are of the form $S \to \varepsilon$ or $\alpha A \beta \to \alpha \omega \beta$, where A is a nonterminal symbol and $\omega \neq \varepsilon$. (S denotes here the start symbol of the grammar.)