**T–79.1001** **Autumn 2006**
**Introduction to Theoretical Computer Science T**
**Tutorial 12, 12 to 13 December**
**Problems**

*This is the last lecture and exercise round of the course T–79.1001; the exam on Dec 21 covers topics covered in the lectures and exercises.*

*Participants of T–79.1001 are kindly asked to give course feedback at the WWW page* `http://www.cs.hut.fi/Opinnot/Palaute/kurssipalaute.html` *between Dec 13 and Jan 3; those who take the December exam should answer after having taken the exam.*

**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) $\{w \in \{a, b, c\}^* \mid w$ contains equally many $a$'s, $b$'s and $c$'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.

**CONTINUES OVERLEAF**

**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 $S \rightarrow \varepsilon$ or $\alpha A \beta \rightarrow \alpha \omega \beta$, where $A$ is a nonterminal symbol and $\omega \neq \varepsilon$. ($S$ denotes here the start symbol of the grammar.)