Helsinki University of Technology  
Laboratory for Theoretical Computer Science  
Pekka Orponen

T-79.148 Introduction to Theoretical Computer Science (2 cr)  
Exam 11 Feb 2002, 4–7 p.m. T1+T2

Write down on each answer sheet:
- Your name, department, and study book number

1. Construct deterministic finite automata accepting each of the following languages:
   (a) \( \{ w \in \{a,b\}^* \mid w \text{ contains both } aa \text{ and } bb \text{ as substrings} \}; \)
   (b) \( \{ w \in \{a,b\}^* \mid w \text{ does not contain the substring } aba \}; \)
   (c) \( \{ w \in \{a,b\}^* \mid w \text{ contains a number of } a\text{'s that is an exact multiple of three} \}. \)

2. Construct in a systematic way a deterministic finite automaton for the language denoted by the regular expression \( (ba)^* b \cup (b^* a) \). (Note: Also the intermediate stages of the construction must be presented – the final result alone does not suffice.)

3. Construct a context-free grammar that generates the language
   \[ L = \{ a^i b^j c^k \mid i, j, k \geq 0, \ i + j = k \}. \]
   Prove, furthermore, that the language \( L \) is not regular.

4. Construct a Turing machine that decides the language
   \( \{ w \in \{a,b\}^* \mid w \text{ contains equally many } a\text{'s and } b\text{'s} \}. \)

5. Define formally the following notions:
   (a) Turing-decidable and undecidable languages. Give an example of a language that is recursively enumerable, but not Turing-decidable. (A precise definition is required, but no proofs.)
   (b) Language classes P and NP; NP-complete languages. Give an example of an NP-complete language. (A precise definition is required, but no proofs.)

Each problem 6 points, total 30 points.