Helsinki University of Technology
Laboratory for Theoretical Computer Science
Pekka Orponen (tel. 5246), Tommi Syrjänen (tel. 5082)

T-79.148 Introduction to Theoretical Computer Science (2 cr)
Exam Mon 3 February 2003, 12–3 p.m.

Write down on each answer sheet:
- Your name, department, and study book number
- The text: "T-79.148 Introduction to Theoretical Computer Science 3.2.2003"
- The total number of answer sheets you are submitting for grading

In case you are taking the exam as an "old" student, i.e. you have participated in the course already in Spring 2002 or earlier, write also visibly on each answer sheet the text "OLD" and the semester in which you attended the course (e.g. "OLD SPRING 2002"). In this case your exam will be graded according to the rules of Spring 2002, otherwise according to the rules of Autumn 2002. According to the new rules, you must e.g. complete all your Regis computerised assignments before taking the exam; also the way of calculating various bonus exam points differs from earlier installments. You can only take the exam as an "old" student if you have really attended the course before the present semester.

1. (a) Give a regular expression that describes the language

$$\{w \in \{0,1\}^* \mid w \text{ begins and ends with a different symbol}\}.$$

8p.

   (b) Design a deterministic finite automaton that recognises the language in part (a). *7p.*

2. (a) Design a context-free grammar for the language

$$L = \{w \in \{a,b\}^* \mid w \text{ contains equally many } a\text{'s and } b\text{'s}\}.$$

   Draw the corresponding parse trees for the sentences *aabb*, *abab* and *baab*. *5p.*

   (b) Is the grammar you designed in part (a) unambiguous or ambiguous? (Justify your answer.) *5p.*

   (c) Prove (precisely!) that the language L in part (a) is not regular. *5p.*

3. Design a deterministic single-tape Turing machine that recognises ("decides") the language L considered in the previous problem. (Present the Turing machine preferably as a state diagram rather than a transition table.) Show the computation sequences ("runs") of your machine on inputs *abab* and *bba*. *15p.*

4. (a) Define the notions of a recursive ("decidable") and recursively enumerable ("semidecidable") language, and explain their relation to issues in computer programming. *8p.*

   (b) Give an example of a language that is recursively enumerable, but not recursive. (You should provide a precise definition for the language, but need not prove any of its claimed properties.) Explain the significance of your example from the point of view of computer programming. *7p.*

Total 60p.