TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Software Systems

OHJ-2306 **Introduction to Theoretical Computer Science** — Examination
prof. Tapio Elomaa — Mar. 11, 2013

Write your name and student number to each separate answer sheet. Neither calculators nor any other extra material is allowed in the exam.

All students are required to answer questions 1 and 2. In addition, you may choose to answer any two questions from among 3–6. The maximum score for questions 1 and 2 is 8 points and for questions 3–6 7 points. In total 30 points.

Give careful and detailed answers to the questions!

ANSWER QUESTIONS 1 AND 2

---

1. Let w^R denote the reverse of string w; i.e. if

$$w = a_1 a_2 \dots a_n,$$

then

$$w^R = a_n \dots a_2 a_1.$$

A string is a *palindrome* if $w = w^R$ (for example, "Step on no pets"). Let us examine the language of palindromes over the alphabet $\{ a, b \}$:

$$\mathrm{PAL} = \left\{ w \in \{ a, b \}^* \mid w = w^R \right\}.$$

   (a) Is PAL a regular language?
   (b) Is PAL a context-free language?

2. Show that the halting problem of Turing machines

$$\mathrm{HALT}_{\mathrm{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM and halts on input } w \}$$

is undecidable. You may assume that the universal language U over the binary alphabet is not decidable. Is $\mathrm{HALT}_{\mathrm{TM}}$ semi-decidable? Justify your answer.

ANSWER TWO OF QUESTIONS 3–6

3. Let L be the language

$$L = \{ w \in \{ a, b, c \}^* \mid w \text{ contains equally many } as, bs, \text{ and } cs \}.$$

   (a) Give a standard Turing machine (draw its state diagram) recognizing language L.

   (b) Give an unrestricted grammar generating the language L.

4. Define incompressibility of binary strings and the required related concepts. Show that incompressible strings of every length exist.

5. Let abbreviation TR stand for Turing-recognizable. Prove the following claims (A and B are binary languages):

   (a) If A is TR-complete, $B \in$ TR, and $A \leq_m B$, then also B is TR-complete.

   (b) If A is TR-complete, then $\overline{A}$ is co-TR-complete.

6. (a) Define the time complexity class NP.

   (b) Define in detail a NP-complete language.

   (c) How can we prove that language A is NP-hard based on the knowledge that another language B is NP-complete?