TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Software Systems

OHJ-2306 **Introduction to Theoretical Computer Science** Exercises 7
prof. Tapio Elomaa Nov. 10, 2009

1. Prove that the formal language of Theorem (5.2)

$$\mathrm{NE_{TM}} = \{ \langle M \rangle \in \{0,1\}^* \mid M \text{ is a TM and } L(M) \neq \emptyset \}$$

   is semidecidable (Turing-recognizable). You may assume that you have at your disposal a Turing machine M_{OK}, which tests whether the input string is a legal code for a TM, and the nondeterministic TM M_G, which generates an arbitrary binary string w at the end of the tape .

2. Let

$$\mathrm{E_{TM}} = \{ \langle M \rangle \in \{0,1\}^* \mid M \text{ is a TM and } L(M) = \emptyset \}.$$

   Prove that $\mathrm{E_{TM}}$ is undecidable.

3. Prove, following the proof of Theorem (5.2), that the following problem is undecidable:

   Does the given Turing machine M accept the empty string ε?

4. Does it follow from Rice's theorem that the problem "does the given Turing machine have an even number of states?" is undecidable? Justify your answer.

5. As you know well, the failure of arithmetic operations (division by zero, overflow or underflow of a register, etc.) can lead to the termination of a program with a runtime error. Justify why, for example, a potential division by zero could not be checked during compilation and warned in advance.

6. Let

$$\mathrm{ALL_{CFG}} = \{ \langle G \rangle \mid G \text{ is a context-free grammar and } L(G) = \Sigma^* \}.$$

   Prove that $\mathrm{ALL_{CFG}}$ is undecidable using the technique of reduction via computation histories.