T-79.148 **Spring 2002**
Introduction to Theoretical Computer Science
Tutorial 11, 16 – 18 April
Problems

Homework problems:

1. Are the following statements true or false? Justify your answers.
   (a) All context-free languages are recursive ("Turing-decidable").
   (b) The complement of any context-free language is recursive.
   (c) All languages recognised ("semidecided") by nondeterministic Turing machines are recursive.
   (d) The complement of any language recognised by a deterministic Turing machine is recursively enumerable ("semidecidable", "Turing-recognisable").

2. Recall the correspondence between formal languages and decision problems ("yes/no"-mappings): given a decision problem P, the corresponding language A_P consists of those input strings x for which the answer to question P is "yes". Interpret the following fundamental results concerning computable languages in the decision problems framework. (I.e. replace the concepts of a "recursive" and "recursively enumerable" language by those of a "(totally) decidable" and "partially decidable" or "semidecidable" decision problem.)
   (a) The class of recursive languages is closed under unions, intersections, and complements.
   (b) A formal language is recursive, if and only if both the language itself and its complement are recursively enumerable.

   How would you prove the decision problem version of item (b) above by using as a proof formalism e.g. C programs instead of Turing machines?

3. Choose some explicit encoding of Turing machines into strings (e.g. the one presented on p. 96 of the lecture notes, or the one in your favourite textbook). What is the lexicographically first string c for which $L(M_c) \neq \emptyset$ (i.e. the machine encoded by c halts and accepts *some* input string). Is it the case, for this particular c, that $c \in L(M_c)$?

Demonstration problems:

4. Prove that the class of recursively enumerable languages is closed under unions and intersections. Why cannot one prove that the class is closed under complements in a similar way as in the case of recursive languages, i.e. simply by interchanging the accepting and rejecting states of the respective Turing machines?

5. (a) Prove that any decision problem that has only finitely many possible inputs is decidable.
   (b) Prove that the problem "Does the decimal expansion of π contain 100 consecutive zeros?" is decidable. What does this result tell you about (i) the decimal expansion of π, (ii) the notions of decidability and undecidability?