T-79.1001 **Spring 2006**
Introduction to Theoretical Computer Science T
Tutorial 10, 11–12 April
Problems

Note that because of the Easter break, there is no lecture on Thu 13 April, and correspondingly no tutorial sessions on 18–19 April.

Homework problems:

1. Choose some explicit encoding of Turing machines into strings (e.g. the one presented on p. 98 of the lecture notes, or the one in your favourite textbook). Determine 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 value of c, that $c \in L(M_c)$? Give a simple verbal description of the language $L(M_c)$.

2. With respect to the enumeration discussed in the previous problem, prove that the following language

$$\bar{D} = \{c \in \{0,1\}^* \mid c \in L(M_c)\}$$

is recursively enumerable ("semidecidable", "Turing-recognisable"). The proof does not need to be spelled out in full detail: it suffices to argue "convincingly" that the necessary Turing machine(s) can be constructed. (*Hint:* Make use of the existence of universal Turing machines.) [*NB.* The complementary language D is shown to be *not* recursively enumerable in Lemma 6.5 of the lecture notes.] Give examples of Turing machines M and M' for whose codes c, c' it is the case that $c \in \bar{D}$, $c' \notin \bar{D}$ (i.e. $c' \in D$).

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

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. by simply interchanging the accepting and rejecting states of the respective Turing machines?

5. Show that Turing machines that have only *one* internal state in addition to their accepting and rejecting states are capable of recognising exactly the same languages as the standard machines with arbitrarily many states. For simplicity, you may assume that the simulating machines have multiple tapes and may also keep their tape heads stationary (direction code 'S') in a transition. How many internal states would be needed to effect the simulation on single-tape machines?

6. (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?