COMS W3261: Theoretical Computer Science.
Instructor: Tal Malkin

Problem Set 10

Due: Thur, 04/16/09.

Reading[1]: 4.2, 5.1, 6.3

1. Recall that we defined (in class and in a homework problem) the languages

$$\mathrm{X}_{TM} = \{\langle M\rangle : M \text{ does not accept } \langle M\rangle\}$$

$$\mathrm{NE}_{TM} = \{\langle M\rangle : L(M) \text{ is not empty}\}.$$

   We proved that X_{TM} is not decidable, and not even recognizable, and that NE_{TM} is recognizable.

   (a) Prove that NE_{TM} is undecidable, by showing that $\mathrm{X}_{TM} \leq_T \mathrm{NE}_{TM}$. Specifically, let S be a decider for NE_{TM}, show a decider T for X_{TM}.

   (b) Why can't the same proof be used to prove that NE_{TM} is not recognizable? Specifically, if S is a recognizer for NE_{TM}, why wouldn't T (that you built in part (a)) be a recognizer for X_{TM}?

2. Show that no Java program could be written which, given as input an arbitrary Java program, can determine whether or not there is any imput on which the given program prints out "AN ERROR HAS OCCURED". (You may assume Java programs are equivalent to Turing Machines).

3. Define the language

$$\mathrm{SMALL} = \{\langle M\rangle : L(M) = \{0000, 0101, 1110\}\}.$$

   Prove that SMALL is undecidable by showing $A_{TM} \leq_T$ SMALL.

4. (Extra Credit:) For a class of languages $\mathcal{C}$, we define $\mathcal{C}_{TM} = \{\langle M\rangle : L(M) \in \mathcal{C}\}$.

   (a) Let $\mathcal{C}$ be a class of languages such that $\emptyset \in \mathcal{C}$, and such that there exists a TM-recognizable language $L_N \notin \mathcal{C}$.
   Prove that $\mathcal{C}_{TM}$ is undecidable, by showing that $\mathrm{A}_{TM} \leq_T \mathcal{C}_{TM}$.
   Note: in class (and Theorem 5.3 in text) we proved this for the special case of $\mathcal{C}$ being the class of regular languages.

   (b) Let $\mathcal{C}$ be a class of languages, and L_N, L_Y be two TM-recognizable languages, such that $L_Y \in \mathcal{C}$, and $L_N \notin \mathcal{C}$. Prove that $\mathcal{C}_{TM}$ is undecidable, by showing that $\mathrm{A}_{TM} \leq_T \mathcal{C}_{TM}$.
   Note: Obviously, the previous part (4a) is a special case of this one. Another special case is the previous problem (3).

[1] Note that the book introduces Turing reductions in the beginning of chapter 5, by describing them intuitively and just calling them 'reductions'. It defines them and starts calling them 'Turing reductions' in section 6.3.