CS 273: Intro to Theory of Computation, Fall 2007
Problem Set 10 (due Thursday, November 29th)

This homework contains 5 problems, one of which is bonus. It is due 12:30 in class or by noon at Elaine Wilson's office (3229 Siebel).

1. **Decidability vs. Turing Recognizable**

   For each of the following languages, determine if it is decidable, Turing recognizable, or neither, and briefly explain why. Since all decidable languages are recognizable, you need not mention that decidable languages are recognizable.

   (a) The set of encodings of Turing machines that halt after at most 20 steps from a blank tape.

   (b) The set of encodings of Turing machines which halt, leaving at most 20 tape squares full at the end of their computation from a blank tape.

   (c) The set of encodings of Turing machines which halt after at most 20 steps on *all* inputs.

   (d) The set of encodings of Turing machines which halt, and can never move their heads left.

2. **Closure Property**

   Prove that the class of *Turing-recognizable* languages is closed under the union operation. In other words, if A_1 and A_2 are Turing-recognizable languages, so is $A_1 \cup A_2$.

   For the definition for Turing-recognizable languages, refer to Definition 3.5 on page 142 of your textbook. Also see Theorems 1.25 and 4.22 in your textbook to get an idea of how to do the proof. For this proof, suppose you have two Turing machines, M_1 and M_2 that recognize A_1 and A_2, respectively. Design a Turing machine M that recognizes $A_1 \cup A_2$.

3. **Reduction (easy)**

   Suppose we know that the language

   $$ALL_{CFG} = \{\langle G \rangle \mid G \text{ is a CFG such that } L(M) = \Sigma^*\}$$

   is undecidable (you saw a proof of this in Tuesday's lecture and this is proved in Sipser p.197).

   Use this fact to show that the language

   $$EQREG_{CFG} = \{\langle G, A \rangle \mid G \text{ is a CFG and } A \text{ is a DFA and } L(G) = L(A)\}$$

   is undecidable.

   Prove this using reductions (by reducing ALL_{CFG} to $EQREG_{CFG}$).

4. **Reduction**

   Show that the following language is undecidable.

$$SUBSET_{TM} = \{\langle M, N\rangle \mid M \text{ and } N \text{ are TMs and } L(M) \subseteq L(N)\}.$$

   Hint: Prove that we can reduce A_{TM} to $SUBSET_{TM}$. That is, show that if there exists a decider for $SUBSET_{TM}$, then there must also exist a decider for A_{TM}.

5. **Decidability (bonus)**

   Fix an alphabet Σ. For any $n \in \mathbb{N}$

$$\begin{aligned} L_n = \{\langle M\rangle \mid\ & M \text{ is a Turing machine over } \Sigma \text{ and} \\ & M \text{ writes to at most } n \text{ squares on the tape} \\ & \text{during the course of the computation from a blank tape.}\} \end{aligned}$$

   Show that L_n is decidable, for any $n \in \mathbb{N}$.

   Note that M need not halt in the computation and yet be in L, as long as it uses fewer than n squares on the tape during the course of its computation. For example, a code for a Turing machine which repeatedly writes 1 to the first cell of the tape is in L_2, even though it does not halt. A code for a Turing machine which places $n + 1$ characters on the tape and then halts is not in L_n, even though it halts.