Problem Set 4 Due: October 27, 2009

Homework: (Total 100 points) Do the following exercises.

Problem 1. [10 points] In this problem you will establish some closure properties for languages of Turing machines.
(a). Are recursive languages closed under *intersection*? Justify your answer.
(b). Are recursively enumerable languages closed under *intersection*? Justify your answer.

Problem 2. [20 points] Consider the following problem concerning Turing machines with a tape alphabet Γ.

> Given a Turing machine M, input string w, and a symbol $X \in \Gamma$, decide whether M, when running on input w, will ever write the symbol X on its tape.

Show that this problem is undecidable. Is this problem recursively enumerable? (*Hint:* Can you give a reduction from the universal language L_u? It may help to look at the solution to Exercise 9.3.7(a), although Exercise 9.2.1 is more directly relevant.)

Problem 3. [20 points] (Exercise 9.3.6(c) in the textbook, on page 400 of the third edition, or page 391 of the second edition)
Show that the following problem is decidable.

> Given a Turing machine M and an input string w, decide whether M, when started with input w, never scans any tape cell more than once.

(*Hint:* Look at the solution to Exercise 9.3.6(a). The very first transition of the Turing machine can either move the head to the left or to the right – consider these two cases separately.)

Problem 4. [15 points] Prove that the following language L is non-recursive:

$$L = \{<M_1, M_2> \mid L(M_1) \subseteq L(M_2)\}.$$

The strings in L encode two Turing machines M_1 and M_2 such that the language of M_1 is a subset of the language of M_2.

To prove this result you may use the fact that the language L_{all} is not recursively enumerable, where

$$L_{all} = \{<M> \mid L(M) = \Sigma^*\}.$$

Problem 5. [20 points] Consider the following two languages over the alphabet $\Sigma = \{0, 1\}$.

$$L_H = \{<M, w>|\text{ TM } M \text{ halts on input } w\}$$

$$L_{NR} = \{<\widehat{M}>|\ L(\widehat{M}) \text{ is } \textit{non-regular}\}$$

The language L_H corresponds to the following decision problem: given a Turing machine M and an input string w, does M halt on input w? Similarly, the language L_{NR} corresponds to the following decision problem: given a Turing machine $\widehat{M}$, is $L(\widehat{M})$ a non-regular language?

Your goal is to show that L_{NR} is undecidable, assuming that L_H is undecidable. We propose the following reduction from L_H to L_{NR}.

> Given a Turing machine M and input w, construct a Turing machine $\widehat{M}$ which behaves as follows on being given input $\widehat{w}$.
>
> 1. $\widehat{M}$ simulates the behavior of M on input w
> 2. if M halts on w, then $\widehat{M}$ checks to see if its input $\widehat{w}$ is a palindrome, halting in a final state if $\widehat{w}$ is indeed a palindrome and halting in a non-final state if $\widehat{w}$ is not a palindrome.

(a). [5 points] In the case where M does not halt on w, what is $L(\widehat{M})$?
(b). [5 points] In the case where M does halt on w, what is $L(\widehat{M})$?
(c). [10 points] Show that L_{NR} is undecidable.

Problem 6. [15 points] Consider the following language:

$$L = \{<M_1, M_2>|\text{ for Turing machines } M_1 \text{ and } M_2,\ L(M_1) \cap L(M_2) = \emptyset\}.$$

Show that L is not recursively enumerable. To prove this result you may use the fact that the language L_e is not recursively enumerable, where

$$L_e = \{<M>|\ L(M) = \emptyset\}.$$