B501 Assignment 6
Enrique Areyan

Due Date: Friday, April 13, 2012
Due Time: 11:00pm

For the following questions, $\Sigma = \{0, 1\}$

1. (10 points) Let $T = \{\langle M \rangle | M$ is a TM that accepts w^R whenever it accepts $w\}$. Use reduction to show that T is undecidable. (w^R is the reverse of w)

   Solution: To proof that T is undecidable, we could use the reduction $A_{TM} \leq_m T$. So, a decider for T will yield a decider for A_{TM}, but we know that A_{TM} is undecidable. Therefore, let's work under the assumption that T is decidable and build a decider for A_{TM}. Let R be the decider for T. Consider the following machine:

   $M1$ on input $w1$: "
   1. if input is not in the set: $\{01, 10\}$, *reject*.
   2. if input is 01, *accept*.
   3. if input is 10, run machine M on input w and *accept* if M accepts. if M halts and reject, then *reject*."

   The following machines H decides A_{TM}:

   H on input $\langle M, w \rangle$: "
   1. Run machine R on input $\langle M1 \rangle$ and *accept* if R accepts, or *reject* if R rejects."

   $L(M1) = \{01, 10\}$ if M accepts w and $\{01\}$ otherwise. Machine R decides T, so it will know when to accept or reject $\langle M1 \rangle$. Using this capability, machine H decides A_{TM} which we know to be undecidable. Therefore, there exists no such machine R and the language T is undecidable.

   The mapping reduction lies in this proof, i.e.,

   $$\langle M, w \rangle \in A_{TM} \iff \langle M1 \rangle \in T$$

2. (10 points) A useless state in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states. Formulate this problem as a language and use reduction to show that it is undecidable.

   Solution: We want to show that the following language is undecidable:

   $$S = \{\langle M, q \rangle | M \text{ is a TM and } q \text{ is a useless state in } M\}$$

Suppose that there is a decider R for language S. In this case it will be easier to reduce E_{TM} to S. We can do so as follow: first, we reason that a TM is in E_{TM} if and only if its accept state is useless. More succinctly, $\langle M \rangle \in E_{TM} \iff q_{accept}$ of M is useless. So, if we have a machine to determine if a TM has useless state, we can easily decide E_{TM} with the following decider D:

D on input x: "
1. Run machine R on input $\langle M, q_{accept} \rangle$. If R accepts, then *accept*.
2. If R rejects, then *rejects*"

Machine D decides E_{TM} which we know to be undecidable by theorem 5.4. Therefore, machine R does not exists and S is undecidable.

The mapping reduction lies in this proof, i.e.,

$$\langle M \rangle \in E_{TM} \iff \langle M, q_{accept} \rangle \in S$$

3. (30 points) For each of the following languages, determine whether it is decidable and prove your statement. You can use Rice's theorem.

   (a) $\{\langle M \rangle |$ TM M visits the 10th cell of its tape while processing input string '01'$\}$

   Solution: This language is decidable. Here is an idea for a decider:
   Given $\langle M, 01 \rangle$, simulate M on 01, i.e., simulate two steps of the machine. While simulating M on the universal turing machine we keep track of the cell position currently being visited by the head of machine M. We essentially keep a counter of the times we move to the right and decrease the number if we move to the left, until we reach zero in which case we know we are on the first position. If we ever find the counter to indicate that we are in cell 10, then *accept*, else we *reject*.

   (b) $\{\langle M \rangle | M$ is a TM and '111' $\in L(M)\}$

   Solution: Using Rice's theorem: first, it is obvious that there exists machines that accept 111 and others that don't. In particular, we can always construct a machine that accept only 111 and reject everything else and another that accepts everything except 111 (this is a finite string, so constructing these machines is equivalent to constructing a DFA and we know that DFA $\subset$ TM).

Now, suppose we have two machines M_1 and M_2 such that $L(M1) = L(M2)$. The following holds:

$$\langle M_1\rangle \text{ possesses property } P \iff \langle M_1\rangle \text{ such that } 111 \in L(M_1) \iff$$

$$111 \in L(M_2) \iff \langle M_2\rangle \text{ possesses property } P$$

Rice's theorem hold and we can safely conclude that the above language is not decidable.

(c) $All_{TM} = \{\langle M\rangle | M \text{ is a TM and } L(M) = \Sigma^*\}$

Solution: Using Rice's theorem: first, it is obvious that there exits machines that are in All_{TM} and other machines that are not in All_{TM}., i.e., the property that $L(M) = \Sigma^*$ is not trivial.

Now, suppose we have two machines M_1 and M_2 such that $L(M1) = L(M2)$. The following holds:

$$\langle M_1\rangle \in All_{TM} \iff L(M_1) = \Sigma^* = L(M_2) \iff \langle M_2\rangle \in All_{TM}$$

Rice's theorem hold and we can safely conclude that All_{TM} is not decidable.