Limits of Computation : Problem Session 2 and hints for Homework 2

Rajiv Raman

February 28, 2007

1. (Problem 5.12) Let $E = \{< M > | M$ is a single-tape TM which ever writes a blank symbol over a non-blank symbol on any input$\}$. We show that A_{TM} reduces to E. Assume for the sake of conradiction that E is decidable, and let R be a TM that decides E. We can use R to construct a TM S that decides A_{TM}. The TM S works as follows :

   TM S: On input $< M, w >$
   1. Use M and w to construct TM T_w.
      TM T_w: On any input
      a. Simulate M on w. Use symbol $\sqcup'$ instead of $\sqcup$ when writing and treat it like $\sqcup$ when reading.
      b. If M accepts, write a true blank symbol.
   2. Run R on $< T_w >$ to determine whether T_w ever writes a blank.
   3. If R accepts, M accepts w, therefore *accept*. Otherwise *reject*.

2. (Problem 5.13) Use the universal turing machine described in the textbook.

3. (Problem 5.14) Let $L_{TM} = \{< M, w > | M$ on w tries moving it's head left from the leftmost cell, at some point in it's computation$\}$. Assume to the contrary that TM R decides L_{TM}. Construct a TM S that uses R to decide A_{TM}.

   TM S: On input $< M, w >$
   1. Convert M to M', where M' first moves it's input over one square to the right, and writes a new symbol \$ on the leftmost tape cell. Then M' simulates M on the input.

If M' ever sees a \$, then M' moves it's head one square to the right and remains in the same state. If M accepts, M' moves it's head all the way to the left and then moves left off the leftmost tape cell.

2. Run R on $< M', w >$.
3. If R accepts, *accept*. If R rejects, *reject*.

4. (Problem 5.15) Consider the length of the shortest computation path of a TM that would result in a left move. You can use this to design a TM that decides the language.

5. (Problem 5.17) Try to find the conditions under which dominos with a unary alphabet can form a match. Use this observation to design a TM that decides PCP over a unary alphabet.

6. (Problem 5.19) Any match for SPCP starts with a domino that has two equal strings, and therefore is a match all by itself. So we only need to check whether the input contains a domino that has two equal strings. If so, *accept*, else *reject*.

7. (Problem 5.33) We need to show two reductions. The first to show S is not turing recognizable, reduce A_{TM} to $\overline{S}$, and similarly to show $\overline{S}$ is not turing recognizable, reduce A_{TM} to S.

8. (Problem 5.35) Design a TM that decides X... or show that X is not decidable by reducing any undecidable problem to X.