CS 172: Computability and Complexity, Spring 2010 | **S. A. Seshia & O. Etesami**

HW 5: Turing Machines and Variants

Assigned: March 4, 2010 | *Due in class: March 11, 2010 by 5 pm*

Note: Take time to write clear and concise solutions. Confused and long-winded answers may be penalized. Consult the course webpage for course policies on collaboration.

1. (6 points)

   Let $\Sigma = \{0, 1\}$. Let $L = \{ww^R \mid w \in \Sigma^*\}$ (i.e., L is the set of all palindromes of even length).

   Design a Turing Machine that decides L. You only need to give a *high-level description* of it (in the manner that Sipser does in Chapter 3).

2. (6 points) Prove that if L is Turing-recognizable, then so is L^*. (L^* is the Kleene-star operation on languages defined as $\{x_1 \cdots x_k \mid k \geq 0, \text{ and for all } 1 \leq i \leq k, x_i \in L\}$.)

3. (8 points) Recall the *2-stack pushdown automaton* from Midterm 1.

   In this question we will be concerned with 2-stack PDA. Recall that a 2-stack PDA consists of a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where the transition function is defined as

   $$\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \times \Gamma_\epsilon \rightarrow 2^{Q \times \Gamma_\epsilon \times \Gamma_\epsilon}$$

   If $(q', b'_1, b'_2) \in \delta(q, a, b_1, b_2)$, it means that the 2-stack PDA can read the input character a, pop b_1 from the first stack, pop b_2 from the second stack, push b'_1 onto the first stack, push b'_2 onto the second stack, and go from state q to state q'. The acceptance condition for a 2-stack PDA is just as in the PDA.

   Prove that a language is Turing-recognizable if and only if it can be recognized by a 2-stack automaton.

4. (10 points) Not all Turing-machine variants are as expressive as the standard TM!

   Define a 100-TM to be a one-tape Turing machine where the tape is infinite, but the size of the tape alphabet is at most 100, and the number of states is at most 100.

   Show that 100-TMs are incomparable with DFAs in terms of power, that is, show that there is a binary language (i.e., one with $|\Sigma| = 2$) that is recognizable by a DFA but not by a 100-TM, and there is a binary language that is recognizable by a 100-TM but not by a DFA.

   (Note that to compare 100-TMs with other models of computation, we have restricted ourselves to binary languages, since any language with alphabet size greater than 100 trivially cannot be recognized by a 100-TM.)