CS:4330 Theory of Computation **Spring 2018**

HW 4: Turing Machines

Instructor: Haniel Barbosa, TA: Shantanu Agarwal *Due date: March 11, 2018*

1. (10 points) Consider the TM M_1 from Example 3.9 in the textbook. Give the sequence of configurations that M_1 enters when started on the following input strings:
   (a) 1#1
   (b) 10#11
2. (10 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, i.e. using English to describe the algorithm without specifying how the TM manages its head and tape.
3. (30 points) Show that the collection of decidable languages is closed under the operation of
   (a) concatenation
   (b) complementation
4. (25 points) A *two-stack pushdown automaton* (2-stack PDA) is exactly like a pushdown automaton except that it has two stacks from which we can push and pop at each step. More formally, a 2-stack PDA consists of a 6-tuple (Q, Σ, Γ, δ, q_0, F) where the transition function is defined as

   $$\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \times \Gamma_\epsilon \to \mathcal{P}(Q \times \Gamma_\epsilon \times \Gamma_\epsilon)$$

   If $(q', \mathsf{b}'_1, \mathsf{b}'_2) \in (q, \mathsf{a}, \mathsf{b}_1, \mathsf{b}_2)$, it means that the 2-stack PDA at state q can read the input symbol 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 to state q'. Acceptance for a 2-stack PDA is just as in a regular PDA.

   Show that a language is Turing-recognizable if and only if it can be recognized by a 2-stack PDA.
5. (25 points) Not all Turing-machine variants are as expressive as 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 expressivity power, that is, show that there is a binary language (i.e. $|\Sigma| = 2$) that is recognizable by a 100-TM but not by a DFA, and that 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 cannot be recognized by a 100-TM.)