COMS W3261 CS Theory: Problem Assignment 3
Assigned October 31, 2016; due on Courseworks November 11, 2016

Instructions

• Problems 1-5 are each worth 20 points.
• Submit your solutions in pdf format on Courseworks/COMSW3261/Assignments by 11:59 pm November 11, 2016. Late assignments will not be accepted.
• You can discuss these problems with others but your answers must be in your own words.

Problems

1. Consider the Turing machine

\[ M = ( \{ q_0, q_1, q_f \}, \{ 0, 1 \}, \{ 0, 1, B \}, \delta, q_0, B, \{ q_f \} ) \]

Informally but clearly describe the language \( L(M) \) if \( \delta \) consists of the following sets of rules:

(a) \( \delta(q_0, 0) = (q_1, 1, R); \delta(q_1, 1) = (q_0, 0, R); \delta(q_1, B) = (q_f, B, R); \)

(b) \( \delta(q_0, 0) = (q_1, 1, R); \delta(q_1, 1) = (q_2, 0, L); \delta(q_2, 1) = (q_0, 1, R); \delta(q_1, B) = (q_f, B, R); \)

2. Show that the set of recursively enumerable languages is closed under the operations of

(a) intersection; (b) concatenation.

3. An enumerator is a deterministic Turing machine that, in addition to its standard work tape, has an extra special “printer tape,” where the head can only move right (so it cannot change what it has written in the past). An enumerator \( E \) always starts with an all-blank work tape, runs, and prints a sequence of 0,1-strings on the printer tape (say the strings are separated by a special symbol #). (The enumerator may never, halt, in which case it may print an infinite list of strings on the printer tape.) We use \( L(E) \) to denote the set of strings an enumerator \( E \) ever prints out on the printer tape. Prove that \( L = L(E) \) for some enumerator \( E \) if and only if \( L \) is recursively enumerable.

4. A write-once Turing machine is a restricted TM model where each tape cell can be altered at most twice (including the input portion of the tape). This means that the symbol stored in each cell can only change at most twice during the execution of the Turing machine (so writing the same symbol into the current cell does not count as one alteration). Prove that this variant of the standard Turing machine model accepts the same class of languages as ordinary Turing machines. (This is not part of the problem but can you prove the same statement for write-once Turing machines?)

5. Use reductions and the diagonalization language to show that the following two languages are not recursively enumerable:

(a) \( L = \{ w_i \mid \text{the Turing machine } M_i \text{ does not halt on } w_i \} \). (Note that the original diagonalization language only requires that \( M_i \) does not accept \( w_i \)).

(b) \( L = \{ M \mid M \text{ is a Turing machine that does not halt on the empty string } \epsilon \} \).