COMP 9601 Assignment 2

Due on Oct 27, 2015

Warm-up. (I.) Definition: A language L is said to be complete in the class of decidable languages if L is decidable and for any decidable language L', $L' \leq_m L$.
Prove that any decidable language, except the empty set and Σ^*, is complete in the class of decidable languages.

(II.) Show that, if $P = NP$, then every language $A \in P$, except the empty set and Σ^*, is NP-complete.

(III.) A 2-PDA is a pushdown automaton that has two stacks. Show that for any language L such that $L = L(T)$ for some Turing machine T, then L can also be accepted by a 2-PDA (i.e., 2-PDA is more powerful than a PDA).

(IV.) An enumerator is a Turning machine that takes no input and keeps on outputting the strings of a certain language (say, separated by ";") on a write-only-and-once tape. An enumerator can have other read-write tapes. Let L be a Turing-recognizable language. Show that there is an enumerator E such that for any string x, x is in L if and only if x will appear in the output tape of E.

You can score up to 20 points for each question. The highest five scores will be used to calculate the total score. Make your answers **precise and concise**.

1. A Turing machine with "stay put instead of left" is similar to an ordinary Turing machine, but the transition function has the form

   $$\delta : Q \times \Gamma \to Q \times \Gamma \times \{R, S\} \ .$$

   At each point the machine can move its head right or let it stay in the same position. Let L be a language recognized by such a Turing machine. Prove that L can be accepted by an NFA.

   Hint. Let T be a Turing machine with "stay put instead of left". Suppose T in state q reads an input symbol "a" and stays there; i.e., $\delta(q, a) = (q', b, S)$ for some $q' \in Q$ and $b \in \Gamma$. T will move to the right if $\delta(q', b) = (q'', c, R)$ for some state q'' and symbol c; otherwise, T will be stationary for at least one more step. In general, one can figure out in advance a chain of stay-put moves $\delta(q, a) = (q_1, b_1, S)$, $\delta(q_1, b_1) = (q_2, b_2, S)$, ..., $\delta(q_{k-1}, b_{k-1}) = (q_k, b_k, S)$, followed by a move-to-right move $\delta(q_k, b_k) = (q'', b'', R)$.

   You also need to figure out how the NFA simulates T's behavior when it is reading a blank symbol and when the NFA should accept.

2. Let L denote the language $\{\langle M_1, M_2\rangle \mid$ Turing machine M_1 accepts the same language as Turing machine $M_2\}$. Prove that L is not Turing-decidable.

3. Let $L_\infty = \{\langle M\rangle \mid M$ accepts an infinite number of inputs$\}$. Prove that L_∞ is not Turing-recognizable.

4. Prove that deciding whether the languages of two context free grammars have non-empty intersection is not Turing-decidable. (Hint. A reduction from the Post Correspondence Problem.)

5. Prove that the problem of determining whether a given Boolean formula has at least two satisfying assignments is NP-complete.

6. Given a finite set S and a collection $C = \{C_1, \ldots, C_k\}$ of subsets of S, the 2-color problem is to find a way to color the elements of S such that each element of S is either *red* or *blue* and C_i has at least one element colored red and at least one element colored blue. Show that the 2-color problem is NP-complete.

   Hint. The following approach attempts to show that 3SAT is polynomial-time reducible to the 2-color problem. For each variable x in a given formula, we add two elements x and $\bar{x}$ into S and create a subset $\{x, \bar{x}\}$; for each clause, we create a subset containing its literals. Identify the bug in the above reduction and show a correct reduction.