Theory Qualifying Exam
Spring 2014

Directions. You have four hours. There are 4 problems, please do them all. If you cannot completely solve a problem, we will award partial credit for work that is correct and relevant to the question.

1. A Mealy machine is just like a standard deterministic finite automaton, but with a one-symbol output associated with each state transition. Formally, a Mealy machine M consists of:

   i) a finite alphabet Σ (used for inputs and outputs);
   ii) a finite set Q of states;
   iii) a next-state function $\delta : Q \times \Sigma \to Q$;
   iv) an output function $\lambda : Q \times \Sigma \to \Sigma$;
   v) an initial state $q_0 \in Q$.

   For simplicity, we take the input and output alphabets to be identical here. Then, M computes a function $f_M : \Sigma^* \to \Sigma^*$ in an evident way.

   a) Argue that f_M is invertible iff it is 1-1.

   b) Design a polynomial-time algorithm to solve the following problem: Given a Mealy machine M, decide whether f_M is invertible or not.

2. The puzzle page in a recent inflight magazine had the following task:

   Transform the first word into the last, changing only one letter at a time:

$$\begin{array}{cccc} S & I & C & K \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ W & E & L & L \end{array}$$

   (Only words in the dictionary should be used.) You don't have to solve this, but make sure you understand what is being asked.

   a) Suppose the two words are n letters long. In how many ways can you fill in the blanks, making only single letter changes as required? (Don't worry about the dictionary yet.) Find an elementary function f for which this number is $2^{\Theta(f(n))}$.

   b) Find an algorithm that solves the problem in worst-case time $2^{\Theta(n)}$. (Count a dictionary lookup as one step.)

3. Let $G = (V, E)$ be a graph with a given orientation, i.e., each edge $e = \{u, v\} \in E$ is given a direction, either from u to v, or from v to u, but not both. We are given a sequence of desired in-degrees $(d(v))_{v \in V}$, one for each vertex $v \in V$. The goal is to find an orientation of G that has the desired in-degree at every $v \in V$, and minimizes the number of re-oriented edges. If such an orientation is not possible, the algorithm should detect that.

   Give a polynomial-time algorithm to solve this problem.

4. Oblivious NP, denoted ONP, is the class of all languages L in NP such that for every input length n there is a witness that works for all the strings of length n in L.

   a) Formalize the above informal definition of ONP.

   b) Show that NP=ONP iff NP is in P/poly.