Theory Qual

Spring 2018

Directions. You have 4 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) Show by examples that both are possible, for non-regular languages L: L^* is regular, and L^* is not regular.

   (b) One version of the pumping lemma for regular languages goes as follows.

   > PL: Let L be a regular language. Then, any sufficiently long $x \in L$ can be factored as uvw, where i) $|v| > 0$ and ii) $uv^k w \in L$, for all $k \geq 0$.

   Find a nonregular language L that satisfies PL.

2. Let X be a finite set with n elements, and f a function taking X to X.

   (a) We seek a largest $S \subseteq X$ on which f is 1-1, that is, for all $x, y \in S$ with $x \neq y$, we have $f(x) \neq f(y)$. Design a linear time algorithm to find such a set.

   (b) Let us impose the additional constrant that f maps S to itself. Here is one procedure for finding a largest such S.

   > Identify all elements of X that have no preimage.
   >
   > (If this set is empty, the algorithm stops and returns $S = X$.)
   >
   > Remove these and recursively process the smaller set X'.

   Suppose that f is chosen randomly and uniformly from among the n^n possibilities. Show that the expected number of rounds is $O(\log n)$.

3. This problem is concerned with graph isomorphism. Assume all graphs have vertex set $\{1, \ldots, n\}$ for some n.

   (a) Let L denote the set

   $$\{(G_0, G_1, \varphi) \mid \varphi : V(G_0) \to V(G_1) \text{ is the lexicographically first isomorphism from } G_0 \text{ to } G_1\}$$

   Give an interactive proof system for L.

   (b) Describe the steps you would take to adapt this interactive proof to an AM protocol.

4. Given a graph $G = (V, E)$, an *edge dominating set* is a subset of edges, $F \subset E$, such that for every edge $e \in E \setminus F$ there exists at least one edge $e' \in F$ such that e and e' share a vertex. In other words, F covers E. The goal in this problem is to find the smallest such F.

   (a) Prove that every maximal matching is an edge dominating set, and that there always exists a minimum edge dominating set that is a maximal matching.

   (b) Give a polynomial time algorithm for finding a minimum maximal matching when the graph G is a tree.

   (c) Give a polynomial time 2-approximation for finding a minimum edge dominating set in general graphs.