Theory Qualifier
January 19, 1993

Instructions

Please read these instructions and all questions C A R E F U L L Y before proceeding.

You will have $2\frac{1}{2}$ hours. The exam is closed-book. All questions are of equal weight. You must attempt to answer all of them to earn an unconditional pass. All algorithms should be accompanied by a proof of correctness and complexity analysis unless otherwise directed.

The questions will be graded according to the following criteria in order of importance:

1. correctness and completeness
2. clarity, precision, and conciseness
3. optimality of result

Partial credit will be awarded where appropriate, so be sure to show all work. Unclear or irrelevant arguments will be penalized.

Good luck!

1. (Formal languages and automata) For strings x and y over a finite alphabet Σ, define $x \sqsubseteq y$ if x is a (not necessarily contiguous) substring of y; that is, if x can be obtained from y by deleting letters. For example,

$$abc \quad \sqsubseteq \quad ababac \quad \sqsubseteq \quad cabacbaac\ .$$

Show that any subset of Σ^* closed downward under $\sqsubseteq$ is regular. (*Hint.* Use *Higman's Lemma*: any set $X \subseteq \Sigma^*$ has a finite subset X_0 such that for all $y \in X$ there exists an $x \in X_0$ such that $x \sqsubseteq y$. Equivalently, the set of $\sqsubseteq$-minimal elements of any $X \subseteq \Sigma^*$ is finite. You need not prove Higman's Lemma.)

2. (Computability and undecidability) Let M_i be the i^{th} Turing machine in some standard enumeration of Turing machines. Show that neither the set

$$\{i \mid L(M_i) \text{ is a regular set}\}$$

nor its complement is recursively enumerable.

3. (Complexity, completeness and reducibility) Both the classes of Boolean formulas in 2CNF and monotone Boolean formulas (those with no negations) have polynomial-time satisfiability algorithms. Show that for both classes it is *NP*-complete to determine whether a given formula has a satisfying assignment with at most k true variables. (*Hint.* Use *Vertex Cover*: given an undirected graph, is there a set A of k vertices such that all edges have at least one endpoint in A? Kill two birds with one stone.)

MORE—PLEASE GO ON TO THE NEXT PAGE

4. (Design and analysis of algorithms) Let k be a fixed positive integer. Let $\mathcal{P}_k$ denote the set of undirected graphs that have at most n^k perfect matchings, where n is the number of vertices of the graph.

   (a) Give a polynomial-time algorithm that on input $G \in \mathcal{P}_k$ enumerates all perfect matchings in G.

   (b) One of the following problems is *NP*-complete and the other is solvable in polynomial time. Say which is which, and give an algorithm for the one in polynomial time.

      (i) Given $G \in \mathcal{P}_k$ with an even number of vertices, determine whether G contains a Hamiltonian circuit.

      (ii) Given $G \in \mathcal{P}_k$ with an odd number of vertices, determine whether G contains a Hamiltonian circuit.

5. (Data structures) We want to implement a quick membership test for a subset Σ of a set of strings Ω using a hash table. Ω contains at most $2^{16} - 1 = 65535$ strings, but there is no constraint on the length of the strings. Suppose that we have a hash function

$$h : \Omega \quad \rightarrow \quad \{0, 1, \ldots, 2^{32} - 1\}$$

such that there are no collisions on elements of Ω. If we had 2^{32} bits of contiguous storage available, we could implement the membership test by hashing each value $x \in \Sigma$ and setting the appropriate bit to 1. Unfortunately, 2^{32} bits is over half a gigabyte, and we have only 2^{21} bits (just over 262 Kbytes) of contiguous storage available. Can we compress the table to fit into this space and still solve the membership problem efficiently? If so, how?

END OF EXAM