Theory Qualifier
June 6, 1986

Instructions

This is a *closed book* exam. All problems are of equal weight. There are two parts.

- The *basic part* consists of four questions. Do all four of them.
- The *advanced part* consists of four questions, one from each of the following areas:
    1. recursive function theory
    2. constructive mathematics and program logics
    3. algorithm design and analysis
    4. computational complexity

  You are to answer any two of them. Indicate clearly which two you are attempting; only two will be graded.

For a *basic pass* on the theory exam, you must pass only the basic part. For an *advanced pass*, you must pass both the basic and advanced parts.

Be clear and concise. Partial credit will be given for sensible attempts at answers that are free of false steps. Unclear presentations, even of correct solutions, may be penalized. Irrelevant material may also count against you. Please read the entire exam carefully before you begin. Work first on the problems that seem interesting or tractable. You have four hours.

Good luck!

Part I — Basic

1. A *queue machine* is like a Turing machine, except that it has a queue instead of a tape. It has a finite queue alphabet Γ and a finite input alphabet $\Sigma \subseteq \Gamma$. If $x \in \Sigma^*$ is the input, the machine starts in its start state q_0 with $\$x$ in the queue, where $\$ \in \Gamma - \Sigma$. In each step, it removes a symbol from the front of the queue, and based on that symbol and the current state, pushes a string $z \in \Gamma^*$ onto the back of the queue and enters a new state, according to the transition function δ. It accepts by emptying its queue.

   (i) Give a rigorous formal definition of these machines, including a definition of acceptance. Your definition should begin: A *queue machine* is a 6-tuple
   $$M \quad = \quad (Q, \Sigma, \Gamma, \$, \delta, q_0)\ ,$$
   where...

   (ii) Prove that queue machines and Turing machines are equivalent in power.

2. The *Hasse diagram* of a finite acyclic binary relation R is the smallest binary relation R_H such that $R_H^* = R^*$, where R^* is the reflexive transitive closure of R. For example, the Hasse diagram of the set $\{0,1,2,3,4\}$ under the usual ordering is
   $$0 \to 1 \to 2 \to 3 \to 4\ .$$
   The Hasse diagram of the collection of subsets of $\{0,1,2\}$ under set inclusion is
   $$\begin{array}{ccccc}
    & & \{0,1,2\} & & \\
    & \nearrow & \uparrow & \nwarrow & \\
   \{0,1\} & & \{0,2\} & & \{1,2\} \\
   \uparrow & \nwarrow\!\!\!\!\!\!\nearrow & & \nwarrow\!\!\!\!\!\!\nearrow & \uparrow \\
   \{0\} & & \{1\} & & \{2\} \\
    & \nwarrow & \uparrow & \nearrow & \\
    & & \varnothing & &
   \end{array}$$
   Give an efficient algorithm to compute the Hasse diagram of a given acyclic binary relation. Analyze the efficiency of your algorithm.

3. Show that the following problem is *coNP*-complete: Given an expression over $+$, $\cdot$, 0, 1, and variables $x_1, \ldots, x_n$ ranging over the set $\{0,1\}$, where $+$ and $\cdot$ are interpreted as arithmetic modulo 2, is the expression identically 0?

4. Give a set over a *single-letter alphabet* in each of the following classes, or explain why such a set does not exist. In each instance, briefly indicate the proof techniques you would use to justify your answer.

   (i) regular

   (ii) cfl but not regular

   (iii) csl but not cfl

   (iv) recursive but not csl

   (v) r.e. but not recursive

   (vi) not r.e.

Part II — Advanced

1. (recursive function theory) Prove that the following problem is Π_2^0-complete: Given a *recursive* set, is it infinite?

2. (constructive mathematics and program logics) Consider the following axiomatization Γ of the equational theory of binary relations under the operations $\circ$ (relational composition), $\cup$ (union), and λ (identity):

$$\begin{array}{rcll}
\alpha \circ (\beta \circ \gamma) & = & (\alpha \circ \beta) \circ \gamma & \text{associativity of } \circ \\
\alpha \cup (\beta \cup \gamma) & = & (\alpha \cup \beta) \cup \gamma & \text{associativity of } \cup \\
\alpha \cup \beta & = & \beta \cup \alpha & \text{commutativity of } \cup \\
\alpha \circ (\beta \cup \gamma) & = & \alpha \circ \beta \cup \alpha \circ \gamma & \text{distributivity} \\
(\alpha \cup \beta) \circ \gamma & = & \alpha \circ \gamma \cup \beta \circ \gamma & \\
\alpha \cup \alpha & = & \alpha & \text{idempotence} \\
\lambda \circ \alpha & = & \alpha \circ \lambda \quad = \quad \alpha & \text{identity}
\end{array}$$

   (i) Show that the free algebra T/Γ, where T is the set of terms over a countable set C of constants, is isomorphic to an algebra of binary relations over some set U. (Hint: use $U = \{c_1 \circ \cdots \circ c_n \mid c_i \in C\}$.)

   (ii) Conclude from (i) that Γ is a complete axiomatization.

3. (algorithm design and analysis) Give an efficient algorithm to ascertain whether two given binary trees are isomorphic. Analyze the efficiency of your algorithm.

4. (computational complexity) Let *DP* be the class of sets in $\{0,1\}^*$ that can be written as the intersection of a set in *NP* and a set in *coNP*. Give a complete set for *DP* with respect to polynomial-time many-one reducibility.