Theory Qualifier
January 20, 1987

Instructions

Four hours, closed book. All questions are of equal weight.

There are two parts, a *basic part* and an *advanced part*, each consisting of four questions. Of the four advanced questions, there is 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

For a *basic pass*, you must answer all four basic questions satisfactorily. For an *advanced pass*, you must answer all four basic questions and any two of the four advanced questions satisfactorily. Indicate clearly which two advanced questions you are attempting; only two will be graded.

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

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

Partial credit will be awarded where appropriate. Unclear presentations and irrelevancies will be penalized. *Please read all questions carefully before answering.*

GOOD LUCK!

Part I — Basic

1. A *one-counter automaton* is an automaton with a finite set of states Q, a two-way read-only input head, and a separate counter that can hold any nonnegative integer. The input $x \in \Sigma^*$ is enclosed in endmarkers $\vdash, \dashv \notin \Sigma$, and the input head may not go outside the endmarkers. The machine starts in its start state s with its counter empty and with its input head pointing to the left endmarker $\vdash$. In each step, it can test its counter for zero, and based on this information, its current state, and the symbol its input head is currently reading, it can either add one to its counter or subtract one, move its input head either left or right, and enter a new state. It accepts by entering a distinguished final state f.

   (a) Give a rigorous formal definition of these machines, including a definition of acceptance. Your definition should begin: A *one-counter automaton* is a 7-tuple
   $$M = (Q, \Sigma, \vdash, \dashv, s, f, \delta)$$
   where...

   (b) Prove that the membership problem for deterministic one-counter automata is decidable: given M and x, does M accept x?

2. A *mixed graph* is a graph in which some of the edges are directed and some are undirected. Prove that if a given mixed graph G has no directed cycle, then it is always possible to orient the remaining undirected edges so that the resulting directed graph has no directed cycle. Give an efficient algorithm for obtaining such an orientation if one exists, and analyze its complexity.

3. Which of the following problems are decidable and which are not?

   (a) whether a given one-tape TM on a given input ever writes anything on its tape

   (b) whether a given TM on a given input ever writes the symbol # on its tape

   (c) whether a given CFG generates an infinite set

   (d) whether a given CFG generates a set in P

   (e) whether a given LBA accepts an even length string

   (f) whether two given CFGs are equivalent

   (g) whether there are infinitely many TMs equivalent to a given one

(h) whether a given sentence of first-order number theory is true

4. Suppose you are the coach of a crew team consisting of n rowers with weights $w_1, \ldots, w_n$ and strengths $s_1, \ldots, s_n$. You wish to select a starting squad of size k with combined weight less than W and combined strength greater than S. Show that this problem is *NP*-complete. (You may use that the following version of the *knapsack problem* is *NP*-complete: given a finite set A and size $s(a)$ for each $a \in A$, can A be partitioned into two equal size sets A' and $A - A'$ such that

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)?)$$

Part II — Advanced

1. (recursive function theory) Show that for any acceptable Gödel numbering of the partial recursive functions (i.e. one containing universal and s_n^m functions), there exist total recursive functions *comp* and *const* such that

$$\varphi_{comp(x,y)} = \varphi_x \circ \varphi_y$$

$$\varphi_{const(x)}(y) = x.$$

2. (constructive mathematics and program logics) Suppose we wish to add certain infinitary joins and meets to a PRL-like system. We are interested in making certain statements of the form, "$\bigvee_{phi \in S} \varphi$" and "$\bigwedge_{phi \in S} \varphi$", where S is a possibly infinite set of formulas. Discuss how this could be done in a way consistent with the philosophy of constructive mathematics. Propose introduction and elimination rules for the infinitary constructs, and use your rules to extract the constructive content of the infinitary DeMorgan law

$$\bigvee_{\varphi \in S} \neg\varphi \rightarrow \neg(\bigwedge_{\varphi \in S} \varphi).$$

3. (algorithm design and analysis) For a list $(x_1, \ldots, x_n)$ of n elements drawn from an ordered set, define d to be the number of pairs of elements in the list that are out of order. That is,

$$d_{ij} = 0 \text{ if } x_i \leq x_j, \quad 1 \text{ if } x_i > x_j$$

$$d = \sum_{1 \leq i < j \leq n} d_{ij}.$$

An *exchange sort* is a sorting algorithm that only exchanges adjacent elements of the list. Give an exchange sort that runs in time $O(n+d)$ and makes exactly d exchanges. Provide a complete analysis of your algorithm. Show that these bounds are tight for exchange sorting.

4. (computational complexity) Show that the membership problem for nondeterministic one-counter automata (see basic question 1) is complete for *NLOGSPACE*.