CS 2111 Final Exam
Fall 1996

This test is closed book. You will get reasonable partial credit for attempting the correct technique but recognizing that you get stuck at some point. You will get significantly less partial credit for incorrect solutions that you claim are correct than for partial solutions where you state that you don't see how to finish the proof.

Feel free to ask questions. Concentrate on communicating the main ideas in your answers. Don't dwell unnecessarily on details. Manage your time.

Keep your answers to the first six questions to at most a few sentences. You are welcome to quote well known theorems.

1. (10 points) You have a database consisting of a single binary relation $R \subset A \times A$. The query language for the database is existential second order logic using the relation R. Are there any queries that can not be expressed in this query language? Explain.

2. (10 points) A language A is in the class Parity-P if there is a nondeterministic polynomial-time Turing machine M with the property that if $x \in L$ then M accepts x on an odd number of computation paths, and if $x \notin L$ then M accepts x on an even (possibly 0) number of computation paths. Consider the following problem, Parity-SAT. The input for this problem is a Boolean formula ϕ in conjunctive normal form. The output is 1 if an odd number of the possible assignments to the variables satisfy ϕ, and is 0 otherwise. Do you think you could prove that Parity-SAT is log-space complete for Parity-P? Explain.

3. (10 points) Are there any languages that are not accepted by a family of Boolean circuits? Explain.

4. (10 points) Explain why if $RP \cap coRP \neq P$ then $P \neq NP$.

5. (10 points) Assume that someone was able to prove that one-way functions did not exist. Would that necessarily resolve the $P = NP$ problem? Explain.

6. (10 points) Martha brought in a New York times article at the end of class yesterday trumpeting some amazing new mathematical theorem proved by an automated theorem proving program. The article implied that mathematicians may someday be unnecessary since all theorems may someday be provable by a computer. It this possible? That is, is it possible to write a program that has the ability to either prove or disprove every mathematical statement. Explain.

7. (30 points) Show that the following problem is NP-complete. The input is a graph G with n vertices, and the question is whether G has a simple cycle spanning $\lfloor \sqrt{n} \rfloor$ vertices. Use the fact that the Hamiltonian cycle problem is NP-complete. Recall that the the input for the Hamiltonian cycle problem is a graph G with n vertices, and the question is whether G has a simple cycle that spans all of the vertices in G.

8. (30 points) Show that the following problem is undecidable. The input is a two Turing machine M and N. The problem is to determine if $L(M) \subset L(N)$. Use a reduction from the standard halting problem, determining whether a Turing machine M halts on an input x.

9. (30 points) Define a robust oracle machine $M^?$ deciding a language B to be one such that $L(M^A) = B$ for all oracles A. That is, the answers are always correct independently of the oracle (although the number of steps may vary from oracle to oracle). If furthermore M^A works in polynomial time, we say that oracle A helps the robust machine $M^?$. Let P_h be the class of languages decidable in polynomial time by deterministic robust oracle machines that can be helped. Show that $P_h = NP \cap coNP$.

10. (30 points) Consider a programming language H that has no recursion, or goto statements. It only has one type of loop, a FOR loop. A FOR loop has the property that the number of iterations of the loop is fixed and known before the loop is executed. Hence, all programs in H much halt in finite time on all inputs. Show that there is a recursive language that can not be accepted by an H program.

11. (30 points) Show that if $TIME(n^2) \subset SPACE(n)$ then $TIME(n^4) \subset SPACE(n^2)$.

12. (30 points) Show that RE is closed under Kleene star. Recall that if L is a language then the Kleene star of L, denoted L^* is the collection of strings x such that x is equal to the concatenation of k strings $x_1 x_2 \dots x_k$ with each $x_i \in L$ for $1 \leq i \leq k$.