UVa - cs302: Theory of Computation Spring 2008

Final Exam Preview

The goal of the final exam is to evaluate how well you understand and can apply the key ideas from the course. This handout is a "sneak preview" of some of the questions you might encounter on the final exam (Saturday, May 3, 9am-noon). The actual questions on the final may not match these questions exactly (so it would not be wise to attempt to memorize exact answers to these questions), but thinking about these questions should be useful preparation for the final exam. Until the final starts, you may discuss these questions with anyone you want, and may consult any resources you want.

During the final exam, you will not be permitted to use any materials. Unlike previous exams, you will not be permitted to use any notes during the final exam.

Problem 1: Short Answers. (20) For each question, provide a correct, clear, precise, and concise answer from the perspective of a theoretical computer scientist.

a. [4] What is a *language*?

b. [4] Why is a Turing Machine a good model of actual computers?

c. [4] Why is a Turing Machine a bad model of actual computers?

d. ...

...

Problem 2: Zero, One, Infinity. (20)

For each question, answer **0**, **1** or **Infinity** to indicate the value of the described entity. A correct answer receives full credit without any explanation. A wrong answer with a good explanation may receive some partial credit.

a. [2] Assuming P = NP, the number of NP-Complete problems that are not in P.

b. [2] Assuming P = NP, the number of NP-Hard problems ....

c. [2] The number of strings that are in the language described by the context-free grammar G (S is the start variable):

$$S \to 0A$$
$$S \to 1A$$
$$A \to S$$

d. ...

Problem 3: Drawing Classes. (20)

a. [10] Assuming P $\neq$ NP, draw a diagram that shows the following computability and complexity classes: (1) **TIME**(n^2), (2) P, (3) Turing-decidable, (4) NP-Complete, (5) **TIME**(1), (6) Regular, ...

b. [10] Place points in your diagram representing each of the following languages: (1) $ALL = \Sigma^*$ (where $\Sigma = \{0, 1\}$), ...

Problem 4: Hardness Proofs. (30)

a. [10] Prove the language $A = \ldots$ is not regular.

b. [10] Prove the language $B = \ldots$ is not decidable.

c. [10] Without assuming the *HAMPATH* problem is NP-Hard, prove the language L_{GA} (as defined in Problem Set 6 and its comments) is NP-Hard. (You don't need to show all the details of the reduction, but you should explain enough of the general idea of the proof by convincing us that if you had sufficient time you could work out all the necessary details.)

Problem 5: Closure. (10+20)

a. In *The Limits of Quantum Computers* (your Spring Break reading), Scott Aaronson writes:

> If we really could build a magic computer capable of solving an NP-complete problem in a snap, the world would be a very different place: we could ask our magic computer to look for whatever patterns might exist in stock-market data or in recordings of the weather or brain activity. Unlike with todays computers, finding these patterns would be completely routine and require no detailed understanding of the subject of the problem. The magic computer could also automate mathematical creativity. Given any holy grail of mathematicssuch as Goldbachs conjecture or the Riemann hypothesis, both of which have resisted resolution for well over a centurywe could simply ask our computer to search through all possible proofs and disproofs containing up to, say, a billion symbols.

Is the claim that a computer that could solve NP-complete problems in polynomial time could also automate mathematicl creativity correct?

b. Consider the $\wp$ operator defined as .... Is the set of regular languages closed under $\wp$?

c. Is the set of *decidable* languages closed under $\wp$?