University of Virginia - cs3102: Theory of Computation Spring 2010

Final Exam **9am-noon, 13 May 2010**

Honor Policy. For this exam, you must **work alone**. No resources may be used during the final. The only things you can use are this handout, your brain, and a writing tool. You may not aid or accept aid from other students.

Directions. Answer all questions including all sub-parts (question 9 is an optional, ungraded question). You may use the backs of pages for your scratch work, but we will only grade answers that are written in the answer spaces, or that are found following clearly marked arrows from these spaces. The space provided for each answer is designed to be big enough to easily fit a full credit, correct answer. If you feel like you need more space to write your answer, then either you have extremely large handwriting, or your answer is incorrect, inelegant, or you are providing more detail than is needed for full credit. In general, the questions are organized by topic and the sub-parts of each question are intended to get progressively more challenging.

Your Name:

UVa Email Id:

Problem 1: Definitions. (10) For each term, provide a clear, concise, and precise definition from the perspective of a computer scientist.

a. (2) *language*

b. (2) *algorithm*

c. (3) **P** (the complexity class)

d. (3) **NP-Complete**

Problem 2: Short Answers. (10) Classify each of the following statements as:

> **True** (the statement is correct),
> **False** (the statement is wrong),
> **Unknown** (it is not known to anyone if the statement is True or False)

If the statement is **False** or **Unknown**, your answer should include a brief explanation why. If the statement is **True**, you do not need to include any explanation.

a. (2) If A is a language in **NP**, A is Turing-decidable.

b. (2) If D is a regular language, $D \in \mathbf{P}$.

c. (2) The language

$$LISKOV = \{w | w \text{ is a string Barbara Liskov said during her talk at UVa }\}$$

   is context-free.

d. (2) There exists some deterministic TM that can decide $3SAT$ in polynomial time.

e. (2) The language $ADD = \{1^x + 1^y = 1^z | x + y = z\}$ is in **NP-Hard**.

Problem 3: Language Classes. (10)

a. (5) Is the language $\{a^n b a^n | n \geq 0\}$ regular? (A full credit answer *must include a clear and convincing proof* supporting your answer.)

b. (5) Is the language $\{a^n b a^n | n \geq 0\}$ context-free? (A full credit answer *must include a clear and convincing proof* supporting your answer.)

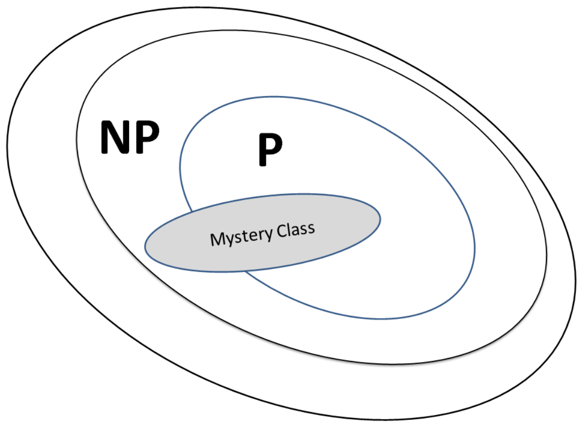
Problem 4: Decidability. (10) Prove the language *RECIPROCAL* (defined below) is undecidable.

> *RECIPROCAL* = $\{\langle M, w\rangle | M$ is a description of a TM and $w \in \Sigma^*$ and the result of running M on w is the same as the result of running M on w^R (the reverse of the string w) $\}$

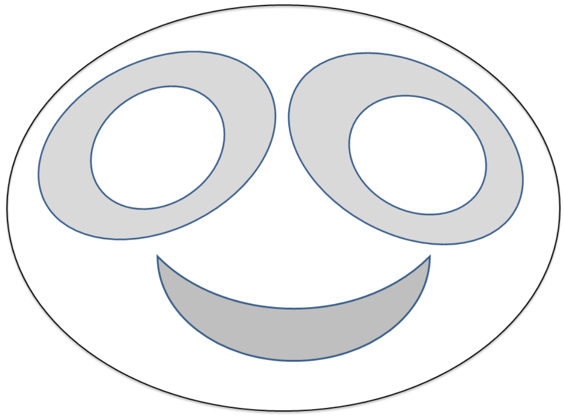
That is, a string $\langle M, w\rangle$ is in *RECIPROCAL* if either (1) M accepts w and M accepts w^R or (2) M rejects w and M rejects w^R or (3) M does not halt on w and M does not halt on w^R. (A full credit answer *must include a clear and convincing proof* supporting your answer.)

Problem 5: Mystery Class. (10) For this question, you should assume that "Proof by Cake Deconstruction" has been accepted as a valid proof technique and used to prove $\mathbf{P} \neq \mathbf{NP}$.

Define a complexity class that corresponds to the "Mystery Class" depicted below:



Your class should (1) include infinitely many languages that are in **NP** but not in **P**, (2) include infinitely many languages that are in **P**, (3) exclude infinitely many languages that are in **P**, and (4) exclude infinitely many languages that are in **NP** but not in **P**, and (5) exclude all languages that are not in **NP**. A full credit answer will define the class precisely in terms of a type of machine and its properties.



(This is just a smiley face, but feel free to scribble on it.)

Problem 6: Machine Models. (15)

a. (5) A *Right-Only Turing Machine* which is identical to a standard Turing Machine except the tape head can only move right. Describe precisely the class of languages that can be recognized by a *Right-Only Turing Machine*.

b. (5) A *Right-Only Cyclic Turing Machine* is like a *Right-Only Turing Machine* except instead of having a one-way infinite tape, the tape is arranged in a cycle. So, when the TM moves right from the last input square, it ends up on the leftmost input square. Prove that a *Right-Only Turing Machine* can recognize some languages that are not context free but is *l*ess powerful than a standard Turing Machine.

c. (5) Is the language $HALTS_{RCTM}$ defined below decidable or undecidable? Include a convincing argument supporting your answer.

$$HALTS_{RCTM} = \{\langle M, w\rangle | M \text{ is a description of a Right-Only Cyclic Turing Machine and } w \in \Sigma^* \text{ and } M \text{ halts on input } w\ \}$$

d. (bonus +5) Is a Right-Only Turing Machine *more powerful* than a NDPDA? (A convincing argument is required for credit.)

Problem 7: NP-Completeness.

a. (10) Prove the *ONEORTWO-SUM* language defined below is in NP-Complete.

$$\textit{ONEORTWO-SUM} = \{\langle x_1, x_2, \ldots, x_n, k\rangle \mid \text{there exists a set of coefficients, } c_1, \ldots, c_n \text{ where each } c_i \in \{1, 2\} \text{ and } \textstyle\sum_{1 \leq i \leq n} c_i x_i = k\}$$

One of the answers to the question of how P=NP will be resolved in Gasarch's survey (http://www.cs.umd.edu/~gasarch/papers/poll.pdf) is:

> **61. Anonymous3:** (Names, schools, dates changed to protect the innocent) On Dec 14, 1991 it was shown that P = NP by undergraduate Mary Lou Koslowsky on her Algorithms final exam at The University of Southern North Dakota. Her ingenious but somewhat hastily written proof, establishing that *3-SAT* could be reduced to *2-SAT* in $O(n^3)$ time, received only 2 points of credit out of a possible 25 and the comment "Wrong." She left computer science and became a pharmacist, working now at Osco Drugs in Lake Wobogon, where all problems have above average complexity.

For this question, you may assume that *2-SAT* is in **P** (which is in fact known to be true).

b. (5) If Ms. Koslowsky's proof was indeed correct, what would it mean to show that *3-SAT* can be reduced to *2-SAT* in $O(n^3)$ time?

c. (5) Suppose instead that Ms. Koslowsky proved that *2-SAT* can be reduced to *3-SAT* in $O(n^3)$ time. What would this mean?

Problem 8: Double Jeopardy. (15)

a. (5) Prove $n!$ is not in $O(2^n)$.

b. (10) Prove $3DSAT$ is in **NP-Hard**.

$3DSAT = \{\phi | \phi$ is a satisfiable formula in 3-dcnf $\}$ where 3-dcnf is like 3-cnf, except clauses can be connected using either $\vee$ (or) or $\wedge$ (and).

3-dcnf = strings of the form:

$((v_{0,0} \vee v_{0,1} \vee v_{0,2}) \wedge \ldots \wedge (v_{i_k,0} \vee v_{i_k,1} \vee v_{i_k,2})) \vee (((v_{i_{k+1},0} \vee v_{i_{k+1},1} \vee v_{i_{k+1},2}) \wedge \ldots \wedge (v_{i_m,0} \vee v_{i_m,1} \vee v_{i_m,2} \vee v_{i_m,3}))) \vee \cdots \vee (((v_{i_r,0} \vee v_{i_r,1} \vee v_{i_r,2}) \wedge \ldots \wedge (v_{i_n,0} \vee v_{i_n,1} \vee v_{i_n,2} \vee v_{i_n,3})))$

where $v_{i,j} \in \{x_i | i \geq 0\} \cup \{\overline{x_i} | i \geq 0\}$

Problem 9: Questions. (Ungraded)

This question is optional and ungraded.

a. Is there any reason your performance on this exam will not adequately reflect what you have learned in this class?

b. Is there anything else you think I show know to determine your final grade fairly?

End of exam! Enjoy your summer.