University of Virginia - cs3102: Theory of Computation Spring 2010

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 (Thursday, May 13, 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 should be worthwhile to understand these questions deeply) and there may be some questions on the final that are not presaged by this handout, but thinking about these questions should be useful preparation for the final exam.

Honor Policy. 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 must work alone and 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: Definitions. For each term, provide a clear, concise, and precise definition from the perspective of a computer scientist.

a. *language*

b. *computer*

c. *algorithm*

d. $\cdots$

Problem 2: Language Classification.

Classify each of the following languages as:

(R) Regular,
(P) Decidable by a Turing machine in polynomial time, but not regular,
(TD) Turing-Decidable, but not known to be in **P**,
(TR) Turing-Recognizable, but not Turing-decidable, or
(None) None of the above.

For full credit, your answer should include a brief argument supporting your answer (but a detailed proof is not needed).

Problem 3: Short Questions. For full credit, your answer should state clearly **Yes**, **No**, **Unknown**, or **Unknowable** and include a clear and convincing argument supporting your answer. **Unknown** means the answer is now currently known by anyone (e.g., it depends on resolving an open question like P =? NP); **Unknowable** means that it is known that the question cannot always be answered correctly by any algorithm.

a. Is $R \subset \mathbf{P}$? R is the set of regular languages and **P** is the set of languages that can be decided by a Turing machine in polynomial time.

b. Is $BIGGER_{CFG}$ decidable? $BIGGER_{CFG} = \{\langle G_1, G_2 |$ where G_1 and G_2 are context-free grammars and the language generated by G_1 contains more strings than the language generated by G_2 $\}$.

c. ...

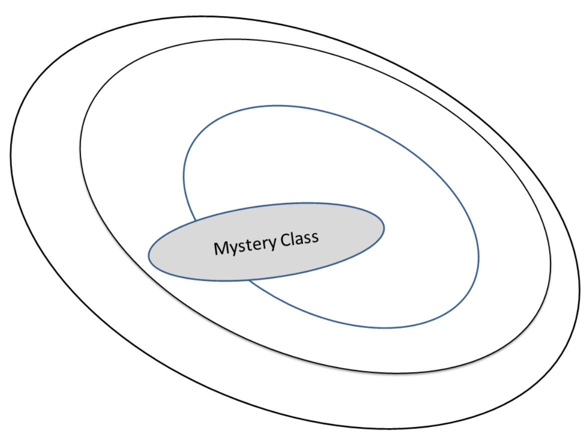
Problem 4: Hardness Proofs.

a. Prove that the language X is not regular.

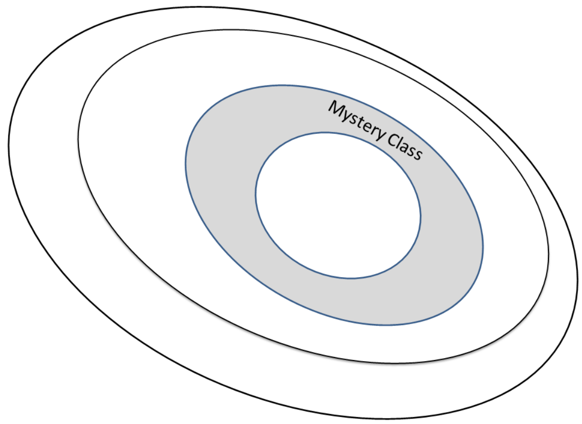
b. Prove that the language Y is not context-free.

c. Prove that the language Z is not decidable.

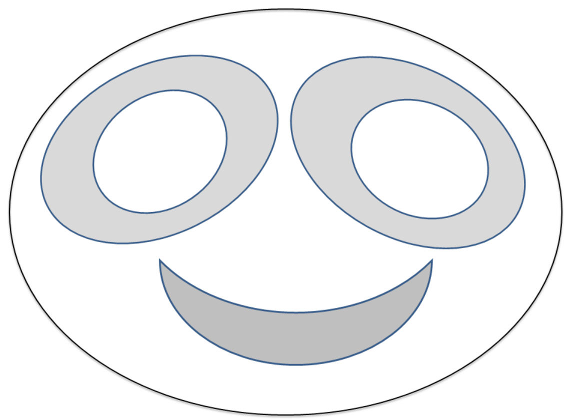
Problem 5: Mystery Class. For this question, assume $\mathbf{P} \neq \mathbf{NP}$.



a.



b.



c.

Problem 6: Machine Models.

a. Consider 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. Consider a Right-Only Cyclic Turing Machine which 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. Describe precisely the class of languages that can be recognized by a Right-Only Cyclic Turing Machine.

c. $\cdots$

Problem 7: Busy Otters.

a. Is the *BUSY-OTTER* language defined below decidable or undecidable? $\cdots$

b. Is the *BUSY-BADGER* language defined below decidable or undecidable? $\cdots$

Problem 8: NP-Completeness.

a. Prove the *ONEORTWO-SUM* language defined below is NP-Complete.

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

b. 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.

$\ldots$