CS/Math 387 Final Exam Review

Our final exam will be drawn from all of the work we have done this semester.

IMPORTANT: You will be asked to prove at least one of the following: (i) Rice's theorem or (ii) A_{TM} is not decidable. So you should have concise well-written proofs for these ready to go without having to think.

The rest of the test will cover the items listed below. Note that there are precise definitions, proofs, methods, and constructions in the list, for which it should be straightforward to study. Expect that some of the problems will be similar to the non-bonus problems assigned for group-work and for homework. Solutions to our homework problems appear on our Moodle page.

Regular languages.

1. Create a DFA, NFA, or regular expression accepting a given regular language.
2. Convert a given NFA into a DFA.
3. Given DFAs for languages A and B, construct DFAs for $A \cup B$, $A \cap B$, A^c, AB, and A^*.
4. Precisely state the pumping lemma for regular languages.
5. Use the pumping lemma for regular languages to prove a given language is not regular.

Context free languages.

1. Create a CFG or a PDA for a given language.
2. Convert an NFA into a CFG.
3. Precisely state the pumping lemma for context-free languages.
4. Use the pumping lemma for CFLs to prove a given language is not context-free.

Turing machines.

1. Give the precise mathematical definition of a TM.
2. Create the state diagram for a Turing machine accepting a specified language.
3. Prove the that acceptance question for CFGs is decidable.
4. Prove that A_{TM} is not decidable.
5. Precisely define *mapping reducibility.*
6. Apply mapping reducibility in an example.
7. Rice's theorem
   (a) Give a precise statement of Rice's theorem.
   (b) Prove Rice's theorem. (The proof was done in class on Monday of Week 6. Links to notes appear there.
   (c) Apply Rice's theorem to prove a given language is not decidable.

Complexity theory.

1. Be able to perform basic manipulations involving big-O and little-o notation.
2. Give a careful definition of the class P.
3. What is a nondeterministic TM? How does the transition function of a nondeterministic TM differ from that of a deterministic TM?
4. What is a polynomial time verifier for a language? (See Definition 7.18 in Sipser's text.)
5. There are two possible definitions of NP: one using nondeterministic TMs and the other using verifiers. (Sipser's text uses verifiers.) Be able to explain why these are the same. (The answer is in the proof of Theorem 7.20.)
6. What does it mean for a language to be NP-complete?
7. Know how to reduce 3SAT to CLIQUE.
8. Consider the classes REGULAR, CFL, P, PSPACE. Are these closed under the following operations? Give a proof or a counterexample:
   (a) union
   (b) intersection
   (c) concatenation
   (d) complement
   (e) star.
9. State Savitch's theorem.
10. Why is $\mathrm{P} \subseteq \mathrm{PSPACE}$?
11. Why is $\mathrm{NP} \subseteq \mathrm{PSPACE}$?
12. Why is PATH in P? Why is PATH in NL?
13. State the space hierarchy theorem.
14. State the time hierarchy theorem.
15. Prove that $\mathrm{TIME}(2^n) = \mathrm{TIME}(2^{n+1})$, but $\mathrm{TIME}(2^n) \subsetneq \mathrm{TIME}(2^{2^n})$.
16. Prove that $\mathrm{PSPACE} \subsetneq \mathrm{EXPSPACE}$ and $\mathrm{P} \subsetneq \mathrm{EXPTIME}$.
17. What does it mean for a language to be in IP? What are the conditions for acceptance and rejection?
18. Let
    $$\mathrm{NONISO} = \{\langle G, H\rangle : G \text{ and } H \text{ are not isomorphic graphs}\}.$$
    Explain why $\mathrm{NONISO} \in \mathrm{IP}$.
19. There will be nothing about BPPs or BPs on the exam.