CS 1511/2110 Midterm 1
Spring 2017

Directions

1. The test is closed book and closed notes.
2. There are 8 part B questions. Answer at most 6 part B questions. Please try to limit your answers to one sentence. Part B questions are worth 10 points per question.
3. There are 4 part A questions. Answer at most 2 part A questions. Part A questions are worth 20 points per question.
4. Time will likely be an issue for most students. So use time wisely. Initially concentrate on the main ideas, and then fill in details with any remaining time.
5. In particular, for the part A questions, usually it is then a good idea for the start of your answer to define relevant terms, give an overview of the proof strategy/technique that you will use, and to explain the key ideas are. After this, you may launch into details.

PART B Questions

1. (a) Let COOL be the complexity class of all languages accepted by a Turing machine that has the property of being cool. Let FROSTY be the complexity class of languages accepted by a Turning machine that has the property of being frosty. Explain how you could show that COOL is a subset of FROSTY. Of course you can't have a complete proof with knowing what the formal definitions of "cool" and "frosty", which I am not giving you. We've done lots of proofs like this, and I'm just trying to test whether you understand the structure of these proofs.

   (b) Explain how to show that a language L is complete for a complexity class C under type R reductions.

2. (a) Define $H(X)$ the entropy of a discrete random variable X.

   (b) Define $H(X|Y)$, the conditional entropy of a discrete random variable X conditioned on another discrete random variable Y.

   (c) Define $I(X;Y)$, the mutual information between two discrete random variables X and Y.

3. (a) Explain how an algorithm can enumerate all the statements that are provable consequences (say by modus ponens) of some axioms.

   (b) Define what it means for an axiomatization to be sound.

   (c) Define what it means for an axiomatization to be complete.

   (d) Explain why a sound and complete axiomatization for number theory implies that there is an algorithm to determine whether a number theoretic statement is true.

4. (a) Is there an algorithm to decide whether a first-order number theoretic formula, where all the standard arithmetic operations are allowed (e.g. addition, multiplication, subtraction, division, exponentiation), and the only relation is equality, is true? Explain why.

   (b) Is there an algorithm to decide whether a first-order number theoretic formula, where now the only arithmetic operation is addition, and the only relation is equality, is true? Explain why.

5. (a) Define the complexity class Σ_3^p.

   (b) Give an example of a language that is complete for Σ_3^p.

6. (a) We showed in class that the Circuit Value Problem/Language is complete for the complexity class P under what type of reduction?

   (b) In this proof we showed how to construct a circuit C and input I to C, from a Turing Machine M, an n bit input x to M, and an integer k. Explain what I was in this construction.

   (c) Conceptually C consisted of a top part, to which I is fed into, and a bottom part which receives the output of the top part and that produces the final bit. Conceptually the top of C consisted of $n^k \cdot n^k$ subcircuits. Conceptually what was the bit that the subcircuit in row i column j is computing.

   (d) Conceptually what is the bottom part of the circuit trying to accomplish.

7. (a) Draw a Venn diagram explaining the known inclusion relationships for the complexity classes:

   EXPSPACE, P, EXPTIME, LOGSPACE, PSPACE, P/poly, Σ_1^p, Π_1^p, Σ_2^p, Π_2^p

   (b) State which inclusions are known to be proper.

   (c) State why you know these inclusions are proper.

8. Explain how to convert a combinatorial circuit C (with 1 bit of output) to a Boolean formula F such that C will output 1 on an input I if and only if F is made true by setting the variables in F using I, and such that F is not more than polynomially larger than C. For simplicity you can assume assume that only types of gates in C are AND and NOT. F can use AND, OR, and NOT operations. F need not be in any special form.

PART A Questions

1. Prove $SPACE(n^3)$ is a strict subset of $SPACE(n^5)$. That is, show that there is a language in $SPACE(n^5)$ that is not in $SPACE(n^3)$.

2. Prove TQBF is complete for PSPACE. Start by defining TQBF.

3. Prove that there are Boolean functions $f : \{0,1\}^n \to \{0,1\}$ that can be computed with n^4 gates but not with n^2 gates.

4. Prove that if $NP \subseteq P/poly$ then $\Pi_2^p \subseteq \Sigma_2^p$.