Qualifying Examination

Theoretical Computer Science

Friday, February 15, 2008

Part II: Formal Languages and Complexity Theory

ID Number	
Pseudonym	

Problem	Maximum Points	Points Earned	Grader
1	50		
2	50		
3	50		
4	50		
Total	200		

Instructions:

1. This is a closed book exam.
2. The exam is for 3 hours and has four problems of 50 points each. Read all the problems carefully to see the order in which you want to tackle them.
3. Write clearly and concisely. You may appeal to some standard algorithms/facts from text books unless the problem explicitly asks for a proof of that fact or the details of that algorithm.
4. If you cannot solve a problem, to get partial credit write down your main idea/approach in a clear and concise way. For example you can obtain a solution assuming a clearly stated lemma that you believe should be true but cannot prove during the exam. However, please do not write down a laundry list of half baked ideas just to get partial credit.

May the force be with you.

Problem 1. A *probabilistic finite automaton (PFA) M* is a finite automaton that on reading an input symbol tosses a coin to decide the next state. A string w is said to be accepted with probability ρ, if M reachs an accepting state with probability ρ on the input w. For a fixed λ, $L_\lambda(M)$ is the collection of all strings w that are accepted with probability $> \lambda$.

(A) **[10 points]** Formally define a PFA M and the language $L_\lambda(M)$.

(B) **[40 points]** Suppose M has the property that every string is accepted either with probability $< \lambda - \epsilon$ or with probability $> \lambda + \epsilon$. Prove that $L_\lambda(M)$ is regular. *Hint:* Use the Myhill-Nerode theorem.

Problem 2. Let M^A denote an oracle TM M with oracle A. We will say A is *polynomial time Turing reducible* to B (denoted as $A \leq_T B$) if there is a deterministic polynomial time machine M such that $A = L(M^B)$. A complexity class $\mathcal{C}$ is said to be *closed* under $\leq_T$ if whenever $A \leq_T B$ and $B \in \mathcal{C}$ then $A \in \mathcal{C}$.

(A) **[25 points]** Show that NP is closed under $\leq_T$ iff NP is closed under complementation.

(B) **[25 points]** An oracle machine M is said to be *positive* iff whenever $A \subseteq B$ then $L(M^A) \subseteq L(M^B)$. We will say A is *polynomial time positive Turing reducible* to B (denoted as $A \leq_T^{\text{pos}} B$) if there is deterministic polynomial time positive oracle machine M such that $A = L(M^B)$. Prove that NP is closed under $\leq_T^{\text{pos}}$, where closure under $\leq_T^{\text{pos}}$ is defined in the same way as closure under $\leq_T$.

Problem 3. A Turing Machine computing a function is a *deterministic* machine with a read-only input tape, finitely many read-write work tapes, and a write-only output tape. The value of the function on an input w is the string written on the output tape when the machine halts on input w. The time taken by the machine is the number of steps of the computation. The space used by such a machine is all the tape cells (including those on the output tape) that are either read from or written to during the computation. Define PF to be the class of all functions computable in polynomial time, and PSPACEF to be the class of all functions computable in polynomial space. Finally, recall that P and PSPACE are the class of *decision* problems solvable in polynomialtime and polynomial space, respectively.

(A) **[15 points]** Prove that if PF = PSPACEF then P = PSPACE.

(B) **[35 points]** Prove that if P = PSPACE then PF = PSPACEF.

Problem 4. Given a graph $G = (V, E)$ and two nodes s, t the s, t longest path problem is to find a *simple* path from s to t of the longest possible length. Suppose a famous researcher has shown that unless $P = NP$ the s, t longest path problem does not admit an α-approximation for some fixed constant $\alpha > 1$. Using this, show that unless $P = NP$ the s, t longest path problem does not admit an α^2-approximation.