Qualifying Examination

Theoretical Computer Science

Thursday, March 17, 2015

Part II: Automata and Complexity

Name	

Problem	Maximum Points	Points Earned	Grader
1	25		
2	25		
3	25		
4	25		
Total	100		

Instructions:

1. This is a closed book exam.
2. The exam is from 10:30am–6:30pm and has four problems of 25 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: Let A and B be two DFAs with n states each. Prove that if $L(A) \neq L(B)$ then there is a string w of length at most $2n$ that belongs to the symmetric difference of $L(A)$ and $L(B)$.

Problem 2: Recall that Mahaney's theorem states that if L is sparse [1] and **NP**-hard then $\mathbf{P} = \mathbf{NP}$. This observation can be strengthened under the *Exponential Time Hypothesis* (ETH) as follows.

The ETH states that there exists $c > 0$ such that $\mathsf{3SAT} \notin \mathbf{DTIME}(2^{cn})$. We will say that L is *almost sparse* iff $\forall \epsilon > 0 \exists n_\epsilon \forall n \geq n_\epsilon.\ |L \cap \{0,1\}^n| \leq 2^{n^\epsilon}$. Prove that if ETH holds, and L is almost sparse and **NP**-hard then $\mathbf{P} = \mathbf{NP}$.

Problem 3: This problem is related to average case hardness against circuits. We say that a boolean function $g : \{0,1\}^* \to \{0,1\}$ is (σ, ϵ)-*inapproximable* if for all (non-uniform) circuit families $\{C_k\}_{k \in \mathbb{N}}$ of size at most $\sigma(k)$, for all sufficiently large k,

$$\Pr_{x \leftarrow \{0,1\}^k}[C_k(x) = g(x)] \leq \epsilon(k).$$

Let $f : \{0,1\}^* \to \{0,1\}$ be a boolean function. Then we define $F : \{0,1\}^* \to \{0,1\}$ and $G : \{0,1\}^* \to \{0,1\}$ as follows.

For each $n \in \mathbb{N}$, let $F(x_1, \ldots, x_n, r) = \langle \alpha, r \rangle$, where $x_i, r \in \{0,1\}^n$, and $\alpha := f(x_1) \cdots f(x_n) \in \{0,1\}^n$. Here $\langle \cdot, \cdot \rangle$ stands for inner product in $GF(2)^n$ (i.e., inner product modulo 2). (If the input string is not of length of the form $n^2 + n$, then it is truncated to the longest such length.)

G is defined as $G(x) = F(x, 1^n)$, where $|x| = n^2$. (If the input string is not of length of the form n^2, then it is truncated to the longest such length.)

1. Suppose F is a (σ, ϵ)-inapproximable function. Show that G is (σ', ϵ')-inapproximable for as large a value of σ' and as small a value of ϵ' as you can.

   (It is not important to fine-tune the parameters.)

   [Hint: Suppose you are given two oracles A and B as follows. Oracle A, on input $x \in \{0,1\}^{n^2}$, returns $G(x)$. Oracle B, when invoked (without any input), returns $(z, f(z))$ for a random $z \leftarrow \{0,1\}^n$. How can you use them to compute F? What happens if A is only approximately correct? Can you avoid the need for B?]

2. Conversely, suppose G is a (σ, ϵ)-inapproximable function. Then show that F is (σ', ϵ')-inapproximable for as large a value of σ' and as small a value of ϵ' as

[1] L is sparse if there is a polynomial $p(n)$ and n_0 such that for all $n > n_0$, $L \cap \{0,1\}^n \leq p(n)$.

you can. [Hint: Again, start with oracles A and B, where now, A computes F and B is as before.]

Problem 4:

1. Show that for each L in $DSPACE(n^2)$ there is a function f, computable in $O(n^2)$ time, such that for all $x \in \{0,1\}^*$, $x \in L$ iff $f(x) \in U$. (In other words, f is a reduction of L to U.)

2. Using (a), or otherwise, show that $DSPACE(n^2) \neq P$. [Hint: Argue that if $DSPACE(n^2) = P$, then P would collapse somewhat.]