Qualifying Examination

Theoretical Computer Science

Firday, March 11, 2016

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 9.30am–4.30pm 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: For any $k \in \mathbb{N}$, let w_k denote the concatenation of all k-bit long strings (in lexicographic order) separated by #'s, i.e., $w_k = 0^k\#0^{k-1}1\#0^{k-2}10\#0^{k-2}11\#\cdots\#1^k$.

- Prove that the language $L = \{w_k \mid k \in \mathbb{N}\}$ is not regular.
- Prove that $L \in \mathbf{DSPACE}(\log\log n)$. If you cannot prove that $L \in \mathbf{DSPACE}(\log\log n)$, find the smallest asymptotic space complexity that L lies in.

Aside: One can also show that if a language $L \in \mathbf{DSPACE}(o(\log\log n))$ then L is regular. Thus, first interesting complexity class that contains non-regular languages is $\mathbf{DSPACE}(\log\log n)$.

Problem 2: Recall that $\mathbf{NTIME}(f(n))$ is the collection of problems solvable in $f(n)$ time on nondeterministic Turing machines. Prove that $\mathbf{P} \neq \mathbf{NTIME}(n)$ as a corollary of the following two sub-problems.

- Prove that if $\mathbf{NTIME}(n) \subseteq \mathbf{P}$ then $\mathbf{P} = \mathbf{NP}$.
- Prove that if $\mathbf{P} \subseteq \mathbf{NTIME}(n)$ then $\mathbf{P} \neq \mathbf{NP}$.

You can get full credit by proving $\mathbf{P} \neq \mathbf{NTIME}(n)$ via a different approach.

Problem 3: A hash function family from a domain W to a range Z (where typically, $|Z| \leq |W|$) is formally defined by a function $\mathcal{H} : R \times W \to Z$. Each choice of $r \in R$ corresponds to a member of the family given by $\mathcal{H}(r, \cdot)$. In typical applications of such a hash function family, r will be chosen uniformly at random from R.

1. A hash function family $\mathcal{H} : R \times W \to Z$ is said to be a *2-universal hash function family* if, for every possible pair of distinct inputs $w_1, w_2 \in W$ ($w_1 \neq w_2$) and every possible pair of outputs $z_1, z_2 \in Z$ (possibly equal),

$$\Pr_{r \leftarrow R}[\mathcal{H}(r, w_1) = z_1 \text{ and } \mathcal{H}(r, w_2) = z_2] = 1/|Z|^2.$$

   Let $z^* \in Z$ be an arbitrary fixed element in Z. Prove that there is a constant $c > 0$ such that for any set $T \subseteq W$, if $|Z|/4 \leq |T| \leq |Z|/2$, then

$$\Pr_{r \leftarrow R}[\exists! w \in T \text{ s.t. } \mathcal{H}(r, w) = z^*] \geq c.$$

   ($\exists!$ stands for "there exists a unique.") That is, prove that, with probability at least c there exists a *unique* element in T that maps to z^* (with the probability taken over $r \leftarrow R$).

Hint: For distinct $w, w^* \in W$, what can you say about $\Pr_{r \leftarrow R}[\mathcal{H}(r, w^*) = z^*]$ and $\Pr_{r \leftarrow R}[\mathcal{H}(r, w) = z^* \mid \mathcal{H}(r, w^*) = z^*]$. Using this, for each element $w^* \in T$, lower-bound

$$\Pr_{r \leftarrow R}[\mathcal{H}(r, w^*) = z^*] \cdot \Pr_{r \leftarrow R}\left[\neg\Big(\bigvee_{w \in W \setminus \{w^*\}} \mathcal{H}(r, w) = z^*\Big) \mid \mathcal{H}(r, w^*) = z^*\right].$$

2. Recall that for every language $L \in \mathbf{NP}$, there is a polynomial time computable relation R_L (between strings of polynomially related lengths) such that $L = \{x \mid \exists w \text{ s.t. } R_L(x, w) = 1\}$. Let $U_L = \{x \mid \exists! w \text{ s.t. } R_L(x, w) = 1\}$ be the subset of L with *unique witnesses.*

   Suppose that for each $L \in \mathbf{NP}$, $U_L \in \mathbf{P}$. Then show that $\mathbf{NP} \subseteq \mathbf{RP}$.[1]

   You may use the result from the above problem here. You can assume that for each n and $k \leq n$ there is a 2-universal hash function $\mathcal{H}_{n,k} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^k$, where d is polynomial in n and $\mathcal{H}$ is polynomial time computable.

   Hint: Given a language $L \in \mathbf{NP}$, construct a set of languages $L_k \in \mathbf{NP}$ and a set of probabilistic transformations, such that if $x \in L$ then, for at least one k, x is transformed to an element in U_{L_k} with constant probability.

Problem 4: Consider the following two definitons of log-space counting problems. A function $f : \{0,1\}^* \to \mathbb{N}$ is in $\#L_a$ if there is a non-deterministic Turing machine M_f that on input x of length n uses $O(\log n)$ space and is such that the number of accepting paths of $M_f(x)$ equals $f(x)$. A function $f : \{0,1\}^* \to \mathbb{N}$ is in $\#L_b$ if there is a relation $R(.,.)$ that is decidable in log-space and a polynomial p such that if $R(x, y)$ then $|y| \leq p(|x|)$ and such that $f(x)$ equals $|\{y \mid R(x, y)\}|$.

- Prove that all functions in $\#L_a$ can be computed in polynomial time.
- Prove that $\#L_b$ equals $\#P$. Recall that $\#P$ is the class of functions $f : \{0,1\}^* \to \mathbb{N}$ such that there is a non-deterministic polynomial TM M_f such for all x, $f(x)$ equals the number of accepting computation paths of M_f on x.

[1] Recall that **RP** is the class of languages which have probabilistic polynomial time algorithms with zero probability of false positives. That is, for $L \in \mathbf{RP}$, there is a polynomial time algorithm A such that if $x \in L$, $A(x) = 1$ with probability at least a constant $c > 0$ and for $x \notin L$, $A(x) = 0$ with probability 1.