Qualifying Examination

Theoretical Computer Science

Saturday, February 28, 2009

Part II: Automata and Complexity

ID Number	
Pseudonym	

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 for 3 hours 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: Answer the following questions. Each proof should not take more than a few steps/lines of reasoning. You can appeal to standard results (e.g. as covered in CS 579).

1. Show that $\mathbf{polyL} \stackrel{\text{def}}{=} \mathbf{DSPACE}((\log n)^{O(1)})$ does not have any complete problems under log-space many-one reductions.
2. Recall that $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$. Is the statement $\mathbf{P}^{\mathbf{E}} = \mathbf{E}$ true, false or is not known? Explain.

Problem 2: A *deterministic non-erasing PDA* (DNEPDA) is a deterministic pushdown automaton such that in any execution of the machine the height of the stack never decreases. More formally, a DNEPDA is $M = (Q, \Sigma, \Gamma, q_0, X_0, F, \delta)$, where Q is a finite set of states, Σ is the input alphabet, Γ is the stack alphabet, $q_0 \in Q$ is the initial state, $X_0 \in \Gamma$ is the bottom stack symbol, and $F \subseteq Q$ is the set of final/accepting states. The transition relation δ is non-erasing in the sense that $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to \mathcal{P}_f(Q \times \Gamma^+)$, where $\mathcal{P}_f(X)$ is the collection of all finite subsets of X. Thus, $(q', w) \in \delta(q, a, \gamma)$ means that $|w| > 0$ and when in state q on reading a with γ on top of the stack, the machine pops γ, pushes the symbols w, and moves to state q'. Furthermore, δ is deterministic in that for every $q \in Q, a \in \Sigma \cup \{\epsilon\}, \gamma \in \Gamma$, $|\delta(q, a, \gamma)| \leq 1$, and if $\delta(q, \epsilon, \gamma) \neq \emptyset$ then $\delta(q, \epsilon, \gamma) = \emptyset$ for all a. An input u is accepted by M if the machine reaches a final state on u, and $L(M)$ is the collection of all strings accepted by M. How does the class of languages recognized by a DNEPDA compare with regular languages and deterministic context-free languages? (Recall that deterministic context-free languages are all languages that can be accepted by a deterministic PDA.) Prove your answer.

Problem 3: We consider the grid-searching capabilities of various finite-state robots walking on the infinite two-dimensional grid. A finite state robot occupies one cell of the grid, and has a sensor that tells it whether the cell is empty, or whether there is a pebble in the cell. The robot has finitely many states, and its actions include moving left, right, up, down, or diagonally, along with putting down or picking up a pebble if so equipped. The next state, and action (put down, pick up, move) depends only on the current state and on whether or not there is a pebble in the current cell and whether or not it is holding a pebble.

1. Describe how a robot with four pebbles can be programmed to visit every cell in the grid (which is infinite in each of four directions).
2. Prove that a robot with three pebbles can also visit every cell in the grid.
3. Prove that a robot with one pebble cannot visit every cell in the grid. (Hint: first consider the case of a robot with no pebbles for partial credit, then extend the argument.)
4. What is the largest portion of the grid that can be visited by robots with two pebbles?

Problem 4: A language L is said to be in the class $\mathbf{S}_2^p$ (the second level of the "symmetric hierarchy") if there is a language $F \in \mathbf{P}$ such that for all x,

$$x \in L \implies \exists y, \forall z, |y|, |z| \leq \text{poly}(|x|) \text{ and } (x, y, z) \in F$$
$$x \notin L \implies \exists z, \forall y, |y|, |z| \leq \text{poly}(|x|) \text{ and } (x, y, z) \notin F.$$

In other words, for $L \in \mathbf{S}_2^p$, there is a polynomial time verifier which takes "claims" (y and z) from two parties such that if $x \in L$ there is a convincing claim y for that, irrespective of what z says, and if $x \notin L$ there is a convincing claim z for that irrespective of what y says.

1. (Simple) How does $\mathbf{S}_2^p$ relate to $\mathbf{\Sigma}_2^p$? Recall that $\mathbf{\Sigma}_2^p$ is the class of languages for which there is a language $F \in \mathbf{P}$ such that for all x,

$$x \in L \iff \exists y, \forall z, |y|, |z| \leq \text{poly}(|x|) \text{ and } (x, y, z) \in F.$$

2. Show that $\mathbf{P}^{\mathbf{NP}} \subseteq \mathbf{S}_2^p$.