Problem Set 2
CS 579: Computational Complexity

Assigned: February 20, 2014 Due on: Feb 27, 2014

Instructions: Please do not turn in solutions to the practice problems.

Practice Problems

Practice Problem 1. A directed graph $G = (V, E)$ is *strongly connected* if for every pair of vertices (x, y) there is a directed path from x to y and a directed path from y to x. Consider STRONGLY CONNECTED, the language of graphs G that are strongly connected. Prove that STRONGLY CONNECTED is **NL**-complete.

Practice Problem 2. Recall that $\mathbf{NP}^{\mathbf{NP} \cap \mathbf{co\text{-}NP}}$ is the collection of all problems A such that there is a nondeterministic oracle Turing machine $M^?$ running in polynomial time and a language $B \in \mathbf{NP} \cap \mathbf{co\text{-}NP}$ with $A = L(M^A)$. Prove that $\mathbf{NP}^{\mathbf{NP} \cap \mathbf{co\text{-}NP}} = \mathbf{NP}$. *Aside:* Note that $\mathbf{NP}^{\mathbf{NP}} = \mathbf{NP}^{\mathbf{co\text{-}NP}} = \Sigma_2^{\mathbf{P}}$, which may not be **NP**.

Practice Problem 3. A language A is *polynomial-time downward self-reducible* if there is a polynomial-time oracle machine M such that:

- $L(M^A) = A$. That is, when given an oracle for A, M decides A (self-reducibility).
- On input x, M only queries the oracle on strings *smaller* than x (downward reducibility).

The second restriction is necessary to make the property interesting — otherwise, on input x, M could just directly ask the oracle if $x \in A$.

Prove that if L is polynomial time downward self-reducible then $L \in \mathbf{PSPACE}$.

Homework Problems

Problem 1. A *strong* nondeterministic Turing Machine has, in addition to its q_{accept} and q_{reject} states, a special state $q_?$. Such a Turing Machine *accepts* its input if all computation paths lead to q_{accept} and $q_?$ states, and it *rejects* its input if all computation paths lead to q_{reject} and $q_?$ states. Moreover, on every input, there is at least one computation path leading to q_{accept} or q_{reject}. Show that the class of languages decided by a strong nondeterministic Turing Machine in polynomial time is exactly $\mathbf{NP} \cap \mathbf{co\text{-}NP}$.

Problem 2. Prove that $\mathbf{NP} \neq \mathbf{DSPACE}(n)$. *Hint:* Let us say a complexity class $\mathcal{C}$ is closed under reductions, if whenever $A \leq_{\mathbf{P}} B$ and $B \in \mathcal{C}$ then $A \in \mathcal{C}$. Show that **NP** is closed under reductions while $\mathbf{DSPACE}(n)$ is not.

Problem 3. Define robust oracle machine $M^?$ deciding a language L to be one such that $L(M^A) = L$ for all oracles A. That is, the answers are always correct, no matter what the oracle is; but, the running time of the machine may vary depending on the oracle. In addition, if M^A runs in polynomial time then we say that oracle A *helps* the robust machine $M^?$. Let $\mathbf{P}_h$ be the class of languages decided in polynomial time by deterministic robust oracle machines that can be helped, and $\mathbf{NP}_h$ be the class for nondeterministic machines.

1. Prove that $\mathbf{NP}_h = \mathbf{NP}$.
2. Prove that $\mathbf{P}_h = \mathbf{NP} \cap \textbf{co-NP}$.