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 \( \text{NL} \)-complete.

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

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 \( L \in \text{PSPACE} \).

Homework Problems

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

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

Problem 3. Define robust oracle machine \( M^A \) 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 \( M^A \). Let \( \text{F}_h \) 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 $\mathbf{P}_h = \mathbf{NP} \cap \mathbf{co-NP}$.