CS530: Introduction to Computational Complexity Instructor: Xiang-Yang Li

Homework 5

Due: Dec 2nd, 2009 *TA: YanWei Wu*

Problem 1: Let A and B be nontrivial languages over an alphabet Σ (that is, not equal to $\emptyset$ or Σ^*). State whether each of the following is KNOWN TO BE TRUE, KNOWN TO BE FALSE, or UNKNOWN. Explain carefully why. For example if you claim that a reduction exists, then you should actually define the reduction.

1. If $A \leq_P B$, then $\overline{A} \leq_P \overline{B}$.
2. If $A \leq_P B$, A is NP-complete, and B is in NP, then B is NP-complete.
3. If $A \leq_P B$, B is NP-complete, and A is in NP, then A is NP-complete.
4. If $B \in P$ and A is nontrivial (not equal to $\emptyset$ or Σ^*), then $A \cap B \leq_P A$.
5. If $B \in P$ and A is nontrivial, then $A \cup B \leq_P A$.
6. If $A \cap B$ is NP-complete, $A \in NP$ and $B \in P$, then A must be NP-complete.
7. If $A \cup B$ is NP-complete, $A \in NP$ and $B \in P$, then A must be NP-complete.
8. If A is NP-complete, $\overline{A} \in NP$ and $B \in NP$, then $\overline{B}$ must be in NP.
9. If A is NP-complete, $\overline{A} \in NP$ and $B \in NP$, then $\overline{B}$ must be in P.

Problem 2: (Sipser 7.36) Show that, if P=NP, a polynomial time algorithm exists that, given a Boolean formula ϕ, actually produces a satisfying assignment for ϕ if it is satisfiable.

(Note: NP is a class of *languages*, so if NP=P, it only implies that there is a polynomial time algorithm $\mathcal{A}$ such that, given any boolean formula ϕ with m clauses and n variables, it will decide whether ϕ is satisfiable in time $f(m, n)$. Here $f(m, n)$ is some polynomial function. And in this question you are being asked for an *algorithm* that *produces* a satisfying assignment (if one exists) for a given ϕ. Thus simply saying that, "because SAT is in NP, you are done" isn't enough.)

Problem 3: (Sipser 7.20) Let G represent an undirected graph and let

$$SPATH = \{\langle G, a, b, k\rangle | \ G \text{ contains a simple path of length at } \textbf{most } k \text{ from } a \text{ to } b\}$$

and

$$LPATH = \{\langle G, a, b, k\rangle | \ G \text{ contains a simple path of length at } \textbf{least } k \text{ from } a \text{ to } b\}$$

1. Show that SPATH $\in$ P.
2. Show that LPATH is NP-complete. You may assume the NP-completeness of UHAMPATH, the Hamiltonian path problem for undirected graphs.