6.045J/18.400J: Automata, Computability and Complexity Nancy Lynch, Nati Srebro

Homework 8

Due: April 21, 2004 *Susan Hohenberger*

Readings: Sections 7.4, 7.5

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 TRUE or FALSE, and explain why.

1. If $A \leq_P B$, then $\overline{A} \leq_P \overline{B}$.
2. If A is NP-complete, $\overline{A} \in NP$ and $B \in NP$, then $\overline{B} \in NP$.
3. If $A \cap B$ is NP-complete, $A \in NP$ and $B \in P$, then A is NP-complete.

Problem 2: For each of the following pairs of sets A and B, show that $A \leq_P B$.

1. $A = SAT$, and $B = SAT\text{-}UNSAT = \{\langle\phi\rangle \mid \phi$ is a satisfiable Boolean formula that is not a tautology (that is, it has at least one non-satisfying assignment)$\}$;
2. $A = CLIQUE$, and $B = HALF\text{-}CLIQUE = \{\langle G\rangle \mid G$ is a graph with an even number of nodes containing an $n/2$-clique, where n is the number of nodes in $G\}$. (Recall that $\langle G, k\rangle$ is an element of $CLIQUE$ if and only if G is an undirected graph with a k-clique.)

Problem 3: (Sipser 7.32) In the proof of the Cook-Levin theorem, we defined a window to be a 2 by 3 rectangle of cells. Show why the proof would have failed if we had used 2 by 2 windows instead.

Problem 4: (Sipser 7.29) 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* and above 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.)