CSE 431 Spring 2009
Assignment #6

Due: Friday, May 22, 2009

Reading assignment: Read Sections 7.4 and 9.3 of Sipser's text.

Problems:

1. Show that A_{TM} is NP-hard.
2. Show that if $P = NP$ then every language $A \in P$, except $A = \emptyset$ and $A = \Sigma^*$, is NP-complete.
3. Let $U = \{\langle M, x, \#^t \rangle \mid M \text{ is an NTM that accepts input } x \text{ within } t \text{ steps}\}$. Show that U is NP-complete.
4. Let $01ROOT = \{\langle p \rangle \mid p \text{ is a polynomial in } n \text{ variables with integer coefficients such that } p(x_1, \ldots, x_n) = 0 \text{ for some assignment } (x_1, \ldots, x_n) \in \{0,1\}^n\}$.
   (a) Show that $01ROOT \in NP$.
   (b) Show that $3SAT \leq_P 01ROOT$. (HINT: First figure out how to convert each clause into a polynomial that evaluates to 0 iff the clause is satisfied. Then create a polynomial q that evaluates to 0 if and only if all of its inputs are 0. Finally, figure out how to combine the individual polynomials for the clauses using the polynomial q.
5. Show that if $P = NP$ then there is a polynomial time algorithm that produces a satisfying assignment when given a satisfiable Boolean formula. (Note: The algorithm you are asked to provide computes a function, but NP contains languages, not functions. The $P = NP$ assumption implies that SAT is in P, so testing satisfiability is solvable in polynomial time. However, the assumption doesn't say how this test is done and the test may not tell what such an assignment might be. You must show that you can find one anyway. Hint: Use the satisfiability tester repeatedly to find the assignment bit by bit.
6. (Extra credit) Recall that a 2-CNF formula is a CNF formula in which each clause has at most 2 literals and that $2\text{-}SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable 2-CNF formula}\}$. Show that $2SAT \in P$.