CSE 431: Introduction to Theory of Computation Spring 2005

PROBLEM SET 7

Due Friday, May 27, 2005, in class

Reading assignment: Sipser's book, Sections 7.4, 7.5, 8.1, 8.2.

Instructions: Same as for Problem set 1.

Each question is worth 10 points. Please be as clear and concise as possible in your arguments and answers. The optional problem is for extra credit.

---

1. A Boolean circuit is said to be *monotone* if it contains only AND and OR gates, but no NOT gates.
   (a) Consider the language $MonotoneCircuitSat = \{\langle C\rangle \mid C$ is a monotone circuit that has a satisfying assignment$\}$.
   Do we know whether $MonotoneCircuitSat$ is NP-complete? Justify your answer.
   (b) Consider the language $MMSA = \{\langle C, k\rangle \mid C$ is a monotone circuit that has a satisfying assignment with at most k 1's$\}$.
   Do we know whether $MMSA$ is NP-complete? Justify your answer. (Hint: If you are stuck, review the VERTEX-COVER problem that is discussed in Theorem 7.34 of the text.)

2. This problem is inspired by the single-player game *Minesweeper*, generalized to an arbitrary graph. Let G be an undirected graph, where each node either contains a single, hidden *mine* or is empty. The player chooses nodes, one by one. If the player chooses a node containing a mine, the player loses. If the player chooses an empty node, the player learns the number of neighboring nodes containing mines. (An neighboring node is one connected to the chosen node by an edge.) The player wins if and when all empty nodes have been so chosen.

   In the problem *Mine-Consistency*, we are given a graph G, along with numbers labeling some of G's nodes. The goal is to determine whether a placement of mines on the remaining nodes is possible, so that any node u which is labeled k has exactly k neighboring nodes containing mines. Formulate *Mine-Consistency* as a language, and prove that it is NP-complete.

   Hint: One possibility is a reduction from 3SAT. The reduction to SUBSET-SUM in the text (Theorem 7.37) might inspire you in the right direction.

3. Problem 7.34, Sipser's book ($3COLOR$ is NP-complete)

4. Problem 8.18, Sipser's book ($DFACHAIN$ is in PSPACE)

5. * **(Optional Problem)** A bipartite graph is a graph whose vertices can be partitioned into two disjoint parts each of which is an independent set. Formally a bipartite graph $H = (X, Y, E)$ has vertex set $X \cup Y$ for disjoint sets X, Y and each edge in its edge set E has one endpoint in X and one in Y. A k-bipartite clique of H is a pair of subsets $S \subseteq X$ and $T \subseteq Y$ with $|S| = |T| = k$ such that $(s, t) \in E$ for each $s \in S$ and $t \in T$ (informally all "cross-edges" exist between S and T). Define the language

   $$\text{BIPARTITE-CLIQUE} = \{\langle H, k\rangle \mid H \text{ is a bipartite graph that has a } k\text{-bipartite clique}\}\ .$$

   Prove that BIPARTITE-CLIQUE is NP-complete.
   (Hint: This problem is reasonably tricky and the "most obvious" reduction from $CLIQUE$ that one would be tempted to first try does *not* work. But a different reduction from $CLIQUE$ exists.)