CS 154 - Introduction to Automata and Complexity Theory

Spring Quarter, 2009

Assignment #7 - Due date: Tuesday, 6/2/09

Homework Due Date. *The final exam is on Friday, June 5. This homework is due by end of class on Tuesday, June 2. Normally, we would allow people using the late credit on this homework to turn in the homework by Firday June 5. However, the final exam is on June 5, and so we would like to distribute the sample solutions for Homework 7 by June 4. For these reasons, we will make the late submission deadline for Homework 7 to be 4pm on Thursday June 4.*

Problem 1. [20 points] Prove that the following problem, called 4TA-SAT, is NP-complete. The problem is defined as follows:
INPUT: A boolean formula $F(X_1, X_2, \cdots X_n)$.
PROBLEM: Does F have at least 4 satisfying truth assignments?

You should reduce the 3-SAT problem to 4TA-SAT. The answers to the following questions constitute the proof of NP-completeness.
(a). Prove that 4TA-SAT is in NP.
(b). Describe a polynomial-time reduction from 3-SAT to 4TA-SAT. Your reduction should take a 3-SAT formula F and construct an instance of the 4TA-SAT problem, say the formula G.
(*Hint:* Suppose you add a variable Y to the boolean formula F, but don't actually use it in any of the clauses. Suppose F had k satisfying truth assignments originally, how many satisfying truth assignments will it have now?)
(c). Show that F is satisfiable if and only if G has at least four satisfying truth assignments.

Problem 2. [20 points] Prove that the following problem called DS (Dominating Set) is NP-complete. A dominating set in a graph $G(V, E)$ is a set of vertices $S \subseteq V$ such that each vertex in V is either in S or has an edge to some vertex in S. The definition of the problem is the following:
INPUT: Graph $G(V, E)$ and integer k.
PROBLEM: Does G have a dominating set of size at most k?

We suggest you use the following reduction from 3-SAT to DS. The reduction function takes as input a 3-SAT formula $F(X_1, X_2, \ldots, X_n)$ with clauses $C_1, \cdots, C_m$ such that $C_i = Z_{i1} \vee Z_{i2} \vee Z_{i3}$, where Z_{ij} denotes a literal. The output of the reduction is a graph $G(V, E)$ and $k = n$. The graph is constructed as follows. For each clause C_i there is a vertex c_i. For each variable X_i there are three vertices x_i, $\overline{x}_i$ and y_i. The three vertices for each variable are connected to each other to form a triangle. The vertex for a clause is connected by an edge to the three vertices which correspond to its literals.

Problem 3. [20 points] The 2-SAT problem is the following: given a boolean formula $F(X_1, \ldots, X_n)$ with exactly *two* literals per clause, does there exist a satisfying truth assignment for F? Similarly, in the 3-SAT problem the input formula must have exactly *three* literals per clause. You are required to show that there is a polynomial-time reduction from the 2-SAT problem to the 3-SAT problem.

(a). Provide a polynomial-time reduction which will convert a 2-SAT formula F into a 3-SAT formula G. Briefly explain why your reduction works in polynomial time.
(b). For your reduction, show that the input F is satisfiable if and only if the output G is satisfiable.
(c). Does all this prove that 2-SAT is NP-hard? Explain.

Problem 4. [20 points] We define a ZERO-ONE INTEGER PROGRAMMING (ZIP) problem as follows.

> INPUT: We are given an integer t and a set of m linear inequalities over a collection of n variables $X_1, \ldots, X_n$,
>
> $$\begin{array}{rcl} a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n & \geq & b_1 \\ a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n & \geq & b_2 \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ a_{m1}X_1 + a_{m2}X_2 + \cdots + a_{mn}X_n & \geq & b_m \end{array}$$
>
> where the a_{ij}'s and b_i's are integer values specified as a part of the input.
>
> PROBLEM: Assign each X_i a value that is either 0 or 1 such that the m inequalities are all satisfied and $\sum_{i=1}^{n} X_i = t$.

Show that ZIP is NP-hard using a reduction from VC (the VERTEX COVER problem).

Recall that in a graph $G(V, E)$ with $V = \{1, \ldots, n\}$, a vertex cover is a set of vertices $C \subseteq V$ such that for each edge $(i, j) \in E$, we have that $\{i, j\} \cap C \neq \emptyset$. The VERTEX COVER problem is the following: given a graph $G(V, E)$ and a positive integer k, does G contain a vertex cover of size k? We have seen in the textbook that VC is NP-hard.

(**Hint:** Your reduction should start with a VC instance $< G(V, E), k >$ and create an instance of ZIP as follows: for each vertex $i \in V$, introduce a variable X_i; further, introduce linear inequalities involving the variables $X_1, \ldots, X_n$ to capture the structure of the vertex cover problem.)

Problem 5. [20 points] The SUBGRAPH ISOMORPHISM problem is defined as follows: given two graphs G_1 and G_2, does G_1 contain a copy of G_2 as a subgraph?

Show that SUBGRAPH ISOMORPHISM is NP-complete. You must use a reduction from one of the problems proved to be NP-complete in the class or lecture notes.

Reading Assignment: We are done with Chapter 9. Make sure you understand the PCP reduction to CFG ambiguuity in Section 9.4. In the last week of classes, we will be talking about Chapter 10.