Course: CSC707, Automata, Computability and Computational Theory
Homework 2: Complexity Theory, polynomial time reduction, P vs NP, NP-hard and NP-complete problems.
Submission: Use Wolfware
File Format: LaTeX and PDF

Due Date: 2:00 A.M. (EST), Thursday, February 11, 2010

1. Provide any feedback/questions you may have on this homework (**optional**).
2. Using LaTeX is required.

---

1. Let $\sum$ denote $\{0, 1\}$. Let $\propto$ denote polynomial-time reducibility. A set $L \subseteq \sum^*$ is P-complete if $L \in P$ and $M \propto L$ for all M, where $M \subseteq \sum^*$ and $M \in P$.
   (a) Show that $\emptyset$ and $\sum^*$ are not P-complete.
   (b) Show that if $L \in P$ and $\emptyset \neq L \neq \sum^*$, then L is P-complete.
2. Show that the Vertex Cover remains NP-complete even when all the vertices in the graph are restricted to have even degree.
   Vertex Cover is defined as follows:
   INSTANCE: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.
   QUESTION: Is there a subset $V' \subseteq V$ such that $|V'| \leq k$, and for each edge $\{u, v\} \in E$ at least one of u and v belongs to V'?
3. Show that the Set Cover problem is NP-complete using the reduction from Vertex Cover.
   Set Cover problem is defined as follows:
   INSTANCE: A set X of n elements, a family F of subsets of X, and a positive integer k.
   QUESTION: Is there a set k or fewer subsets from F whose union is X?
   For example, if $X = \{1, 2, 3, 4\}$ and $F = \{\{1, 2\}, \{2, 3\}, \{4\}, \{2, 4\}\}$, a solution does NOT exist for $k = 2$ but does exist for $k = 3$ (e.g., $\{\{1, 2\}, \{2, 3\}, \{4\}\}$.
4. The Independent Set problem is defined as follows.
   Set Cover problem is defined as follows:
   INSTANCE: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.
   QUESTION: Does G contain an indepenent set of size k or more, i.e., a subset $V' \subseteq V$ and $|V'| \geq k$ such that no two vertices in V' are joined by an edge in E?
   Suppose you are given a graph, $G = (V, E)$, and an integer k as input with $|V| = n$. And suppose you are given an algorithm, D, that solves the decision version of the Independent Set problem in time $T(n, k)$.

(a) Use D to find the size of the maximum independent set, and state the time complexity involved.

(b) Use D in a self-reduction to solve the search version of the independent set problem, and state the time complexity involved.