Course: CSC707, Automata, Computability and Computational Theory
Homework 3: Self-reduction and Approximability.
Submission: Use Wolfware
File Format: LaTeX and PDF

Due Date: 2:00 A.M. (EST), Thursday, February 18, 2010

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

---

1. **Self-reducibility**: Let G denote a finite, simple, undirected graph. Let k denote an arbitrary, positive integer.
   (a) Reduce the problem of finding a coloring of G with k or fewer colors to the problem of deciding whether such a coloring exists. **Solution:** Given a graph G and a decision version of algorithm, D, that solves the problem of k-coloring in a graph
      i. mark all the vertices in G white
      ii. call $D(G, k)$, if answer "no", output empty and stop;
      iii. construct a clique of size k, and color the vertices in the clique with different colors (except white)
      iv. pick any vertex v is colored white, connect v with $k-1$ vertices in the clique
      v. call $D(G, k)$, if answer "no", connect v with other $k-1$ vertices in the clique, repeat step iv.
      vi. if answer "yes", mark v with the color of the vertex in the clique that is not connected with v.
      vii. repeat step iii until all the vertices are not colored white

      After these steps, every vertex in G will be colored.
   (b) Reduce the problem of finding a vertex cover of size k in polynomial time to the problem of deciding whether such a cover exists in polynomial time.

      Given a graph G and a decision version of algorithm, D, that solves the problem of vertex cover in a graph
      i. call $D(G, k)$, if answer "no", output empty and stop;
      ii. construct G': add vertex v_{new} to G; pick vertex v that has not been marked, add an edge (v, v_{new}) to G
      iii. call $D(G, k)$

iv. if answer “yes”, mark v as the vertex in the vertex cover, $G = G'$ and repeat step ii

v. if answer “no”, remove v and v_{new} from G and repeat step ii

After these steps, every vertex in vertex cover of G will be marked.

(c) The Independent Set problem is defined as follows. (Problem 4 from HW2)
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 $T(n, k)$.

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

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

Solution:

i. Use D to find the size of the maximum independent set:
Given a graph G,
for k = n to 1
    call $D(G, k)$
    if $D(G, k)$ answer “yes” , then output k and stop

Since in the worse case, we call D $n = |V|$ times, the time complexity is $T'(n, k) = O(n * T(n, k))$

ii. Use D in a self-reduction to solve the search version of the independent set problem:
Given a graph $G = (V, E)$

call $D(G, k)$ if $D(G)$ answer “no”, then output empty and stop
else
  while G is not empty and $k > 0$
    pick any $v \in V$
    construct G' by removing v from V
    call $D(G', k)$
    if $D(G', k)$ answer “yes”, $G = G'$
    else add v to IS, $k = k - 1$,
        construct G'' by removing v and its neighbor vertices
        $G = G''$
  output IS and stop

Since in the worse case, we need to call D $n = |V|$ times, and construct G' or G'' requires $O(|V| + |E|)$, the time complexity is $T'(n, k) = O(n * T(n, k) + |V| + |E|)$

2. **Approximability**:

   (a) If $P \neq NP$, then for any constant $c \geq 1$, there is no polynomial-time c-approximation algorithm for a general Traveling Salesman Problem. (Hint: Show that HAM-CYCLE can be solved in polynomial time. Also, see the additional hint in the lecture slides.)

   Solution:

   (a) i. We first prove that HAM-CYCLE problem can be solved by using the c-approximation algorithm for a general Traveling Salesman Problem. Given a undirected and unweighted graph G, we can create a complete graph G' of G by adding missing edges and assign weight to each edge. For the edges in G, we assign 1 for the weight. For the edges not in G, we assign $c * |V| + 1$ for the weight. This reduction can be done in polynomial time ($O(|V| * |E|)$). Using the c-approximation algorithm for TSP, we can get a TSP tour of weight W. If $W > c * |V|$, then we know this tour pick at least one edge that is not in G (since every edge in G is assigned weight of 1) and G does not have HAM-CYCLE. Otherwise, G has a HAM-CYCLE.

   ii. Since the reduction from HAM-CYCLE to TSP is in polynomial time, if there exists a polynomial-time c-approximation algorithm for a general Traveling Salesman Problem, HAM-CYCLE can be solved in polynomial time. We already know that HAM-CYCLE $\in NP$. If it can be solved in polynomial time, then $P = NP$, which contradicts with out assumption that $P \neq NP$. Thus, there is no polynomial-time c-approximation algorithm for a general Traveling Salesman Problem.