Lab13-NPReduction

CS363-Computability Theory, Xiaofeng Gao, Spring 2016

∗ Please upload your assignment to FTP or submit a paper version on the next class

∗ If there is any problem, please contact: steinsgate@sjtu.edu.cn

∗ Name: Zhang Yupeng StudentId: 5130309468 Email: 845113336@qq.com

1. Find the certificate and certifier for the decision version of the following problems.

   (a) Clique: Given an undirected graph, find a subset S that there is an edge connecting every pair of nodes in S with maximum nodes.

   Solution:

   Decision version: Given an undirected graph $G = (V, E)$ and $k \geq 0$, does there exist a clique of size $\geq k$?

   Certificate: A set of vertex $S \subseteq V$.

   Certifier: Check if $|S| \geq k$ and $\forall v, v' \in V((v' = v) \vee (\exists (v, v') \in E))$

   (b) Metric k-center: Given n cities with specified distances for each pair of cities as d_{ij}, one wants to build k warehouses in different cities and minimize the maximum distance of a city to a warehouse.

   Solution:

   Decision version: Given n cities with specified distances for each pair of cities as d_{ij}, one wants to build k warehouses in different cities, can the maximum distance of a city to a warehouse be at most s?

   Certificate: k cities to build warehouses.

   Certifier: Calculate the maximum distance of one city to one warehouse and check whether it is no more than s.

   (c) Set Packing: Given a set U of n elements, a collection $S_1, \cdots, S_m$ of subsets of U, find the maximum subsets such that no two of them intersect.

   Solution:

   Decision: Given a set U of n elements, a collection $S_1, ..., S_m$ of subsets of U, can there exists no less than k subsets such that no two of them intersect.

   Certificate: A collection of subsets $\{S_1', ..., S_j'\} \subseteq \{S_1, ..., S_m\}$

   Certifier: Check whether $\{S_1', ..., S_j'\}$ contains no less than k elements and whether any of them intersect.

   (d) minimum k-cut: Given a weighted graph $G = (V, E)$, we want to find a minimum weighted set of edges whose removal would partition the graph to k connected components.

   Solution:

   Decision version: Given a weighted graph $G = (V, E)$, is there a weighted set of edges with total weight no more than s that the removal of this set would seperate the graph to k connected components.

   Certificate: A set of edges to remove.

   Certifier: Check whether the total weight of the removed edges no more than s and whether they seperate the graph to k connected components.

2. The knapsack problem is a well-known optimization problem. Given a set of n items, each item i with a weight w_i and a value v_i, determine the number of each item to include in a

collection so that the total weight is less than or equal to a given limit W and the total value is as large as possible.

Prove that the knapsack problem is NP-complete. (Hint: One solution is reducing the Subset Sum problem to it.)

Proof:

We'll reduce the Subset Sum problem to the knapsack problem in order to prove the problem.

Given an instance of Subset Sum problem with number values n_i and the objective sum S and construct the knapsack problem as follows:

Denote the weight and the value for each item i as w_i and v_i and the weight limit as W, then we have $w_i = v_i = n_i$ and $W = S$. Since the value is equal to the weight, we can know that the largest value $V_{max} = W$.

Under this construction, to find a subset of numbers that adds up to S is to find a collection of items that the total weight is less than W. Thus, we reduce the Subset Sum problem to the knapsack problem.

Since the Subset Sum problem is NP-complete, we can prove that the knapsack problem is NP-complete.

3. We know that $\mathbf{P} \subseteq \mathbf{NP} \cap \textbf{co-NP}$. Please give an example that belongs to following set. If you can, briefly explain your reason. (Should be examples different from the course slides).

   (a) **Co-NP**.

   Solution:

   For $a, b \in \mathbb{N}$, is $a = b$?

   Because $\mathbf{P} \subseteq \textbf{Co-NP}$.

   (b) **Co-NP** $\cap$ **NP-hard**.

   Solution:

   Clique.

   Because **NP-complete** $\in$ **Co-NP** $\cap$ **NP-hard**.

   (c) **Co-NP** $\cap$ **NP**, but not known to be in **P**.

   Solution:

   Pigeonhole Subset Sum: Given n positive integers with sum less than $2^n - 1$, find two disjoint nonempty subsets whose sums are equal.