Lab10-NP and Reduction

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2018.

$*$ If there is any problem, please contact TA Jiahao Fan.

$*$ Name:Juncheng Wan Student ID:516021910620 Email: 578177149@qq.com

1. *PARTITION*: Given a finite set A and a size $s(a) \in \mathbb{Z}$ for each $a \in A$, is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a)$?

   SUBSET SUM: Given a finite set A, a size $s(a) \in \mathbb{Z}$ for each $a \in A$ and an integer B, is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = B$?

   KNAPSACK: Given a finite set A, a size $s(a) \in \mathbb{Z}$ and a value $v(a) \in \mathbb{Z}$ for each $a \in A$ and integers B and K, is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) \leq B$ and $\sum_{a\in A'} v(a) \geq K$?

   (a) Prove *PARTITION* $\leq_p$ *SUBSET SUM*.

   Proof. Assume that there is an instance of *PARTITION* $A' \subseteq A$. First, calculate the the whole size of A:

   $$s(A) = \sum_{a\in A} s(a) \tag{1}$$

   This will take $O(|A|)$ time. To make use of the black box of *SUBSET SUM*, input A and $\frac{1}{2}s(A)$ to the black box.

   I think that:

   A' satisfies $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a)$ iff the black box of *SUBSET SUM* finds a subset A'' satisfies $\sum_{a\in A''} s(a) = \frac{1}{2}s(A)$.

   "$\Longleftarrow$:" If there is a subset $A'' \subseteq A$ such that $\sum_{a\in A''} s(a) = \frac{1}{2}s(A)$, then $A' = A''$ as $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a) = \frac{1}{2}s(A)$.

   "$\Longrightarrow$:" If A' satisfies $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a)$, because $A' \cap (A - A') = \emptyset$ and $A' \cup (A - A') = A$, $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a) = \frac{1}{2}\sum_{a\in A} s(a) = \frac{1}{2}s(A)$. Therefore, there is a subset A'' satisfies $\sum_{a\in A''} s(a) = \frac{1}{2}s(A)$. □

   (b) Prove *SUBSET SUM* $\leq_p$ *KNAPSACK*.

   Proof. Assume that there is an instance of *SUBSET SUM* $A' \subseteq A$. For each element $a \in A$, add a new property "value" where $v(a) = s(a)$. This will cost $O(|A|)$ time.

   To make use of the black box of *KNAPSACK*, input A and set $K = B$. I think that:

   A' satisfies $\sum_{a\in A'} s(a) = B$ iff the black box of *KNAPSACK* finds a subset A'' satisfies $\sum_{a\in A''} s(a) \leq B$ and $\sum_{a\in A''} v(a) \geq K$.

   "$\Longleftarrow$:" If there is a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) \leq B$ and $\sum_{a\in A'} v(a) \geq K$, then A' satisfies that $\sum_{a\in A'} s(a) = B$. Because $B \geq \sum_{a\in A'} v(a) \geq K$ and $K = B$.

   "$\Longrightarrow$:" Because $K = B$ and $v(a) = s(a)$, there is at least one subset $A'' = A'$ can satisfy $\sum_{a\in A''} s(a) \leq B$ and $\sum_{a\in A''} v(a) \geq K$. □

2. *3SAT*: Given a set U of variables, a collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$, is there a satisfying truth assignment for C?

   CLIQUE: Given a graph $G = (V, E)$ and a positive integer $K \leq |V|$, is there a subset $V' \subseteq V$ with $|V'| \geq K$ such that every two vertices in V' are joined by an edge in E?

   Prove *3SAT* $\leq_p$ *CLIQUE*.

Proof. Assume that there is a instance Φ of *3SAT* with N variables and K clauses. I construct a graph G with the following laws:

(a) Each variable in each clause is a vertice. If two clauses have the same variable (i.e. x_1 is in $c_1 = (x_1, \bar{x}_2, x_3)$ and $c_2(\bar{x}_1, x_2, x_3)$), these two vertices should not be the same.

(b) Connect all the vertices with edges except the vertice and the "complement" of the vertice (i.e x_1 and $\bar{x}_1$). Besides, disconnect the vertices in the same clause.

To make use of the black box of *CLIQUE*, input G and K. I think that:

The 3-SAT Φ is satisfied iff the black box of *CLIQUE* finds a clique with input G and K.

"$\Longleftarrow$:" If the black box of *CLIQUE* finds K vertices that is a clique, then assign the corresponding variable the value (i.e if x_1 is choosen, assign x_1 1; if $\bar{x_1}$ is choosen, assign x_1 0).

Because for each variable x and $\bar{x}$, there is no edge between them in G, the property that a variable x can only be assigned 1 or 0 is satisfied.

It is also obvious that the maximal number of vertices is a cluque is K. Because one vertice can only connect with the vertices of other clauses and there is only K clauses. Therefore, if $G(V, E)$ and the black box of *CLIQUE* finds a subset $V' \subseteq V$ with $|V'| \geq K$, then $|V'| = K$ and each clause has exact one vertice choosen. Because each clause has exact one vertice choosen, each clause is satisfied.

"$\Longrightarrow$:" If the 3-SAT Φ is satisfied with an assign C, it means that at least one variable of each clause is satisfied. Just choose all the vertices in the clauses and in the assign C as V'. Because in the assign C there is not coexist of x and $\bar{x}$, at least one vertice in each clause is in C, and according to the construction of G each vertice x can have an edge with the other vertices in V' and not in the same clause of x, there should be a clique of size $\geq K$ (in fact $= K$). □

3. *ZERO-ONE INTEGER PROGRAMMING*: Given an integer $m \times n$ matrix A and an integer m-vector b, is there an integer n-vector x with elements in the set $\{0, 1\}$ such that $Ax \leq b$.

   Prove *ZERO-ONE INTEGER PROGRAMMING* is NP-complete. (Hint: Reduce from *3SAT*)

Proof. First I give my method of proving, then I use the hint to reduce from *3SAT*.

Abbreviate *ZERO-ONE INTEGER PROGRAMMING* **ZOIP**, *ZERO-ONE EQUATION* **ZOE**. I prove that **ZOIP** is NP-complete in the following steps.

(a) **ZOIP** is NP problem.

Given a vector x, just calculate Ax (time complexity $O(mn)$) and compare whether $Ax < b$ (time complexity $O(n)$). The total time complexity of certifier is $O(mn)$ (polynomial).

(b) **Subset Sum $\leq_p$ ZOE.**

Assume that there are n natural numbers $W = \{w_1, w_2, \cdots, w_n\}$ and an integer T. To make use of the black box of **ZOE**, I construct a zero-one matrix $A_{t \times n}$. Transform each w_i to b_i as the binary code of w_i and get $B = \{b_1, b_2, \cdots, b_n\}$. Find the largest element b_i in B and set $t = |b_i|$. For the rest of element in B, set the length to t by adding 0 in the front of the binary code. Set $\mathbf{c}$ the binary code of T and make it a column vector.

Therefore, input A and $\mathbf{c}$ to the black box of **ZOE**. If the black box of **ZOE** finds a zero-one vector $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{c}$, select the element w_i in W according to $\mathbf{x}$ (if $x_i = 1$, select w_i and if $x_i = 0$, don't select w_i). It is obvious that **Subset Sum** has a solution iff **ZOE** has a solution.

(c) **ZOE** is NP-complete.

Because **Subset Sum** is NP-complete (proved in the class **Subset Sum** $\leq_p$) **3-SAT**, **ZOE** is NP-complete.

(d) **ZOE** $\leq_p$ **ZOLP**

Assume that there is a matrix $A_{n\times m}$ and a column vector b $= (b_1, b_2, \cdots, b_n)$. **ZOE** wants to find if there is a column vector $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{b}$.

We have the following relationship:

$$A\mathbf{x} = \mathbf{b} \Longleftrightarrow \begin{pmatrix} A & 0 \\ 0 & -A \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix} \leq \begin{pmatrix} \mathbf{b} \\ \textbf{-b} \end{pmatrix} \tag{2}$$

Set:

$$\hat{A} = \begin{pmatrix} A & 0 \\ 0 & -A \end{pmatrix} \quad \mathbf{c} = \begin{pmatrix} \mathbf{b} \\ \textbf{-b} \end{pmatrix} \tag{3}$$

To make use of the black box of **ZOLP**, input $\hat{A}$ and $\mathbf{c}$. Because **ZOE** is NP-complete, **ZOLP** is NP-complete.

In addition to the above proof, I use the hint to prove that **3SAT** $\leq_p$ **ZOIP**:

Assume that there is a instance of **3SAT** with n variables and m clauses. Construct $\mathbf{x} = (x_1, x_2, \cdots, x_n, \bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n)^T$. Construct $A_{m\times 2n}$ by the laws:

(a) if variable x_j appears in clause i, set $a_{i,j} = 1$;

(b) if variable $\bar{x}_j$ appears in clause i, set $a_{i,n+j} = 1$;

(c) set the rest of A 0.

Set $\mathbf{b} = (1, 1, \cdots, 1)^T$. By using the above equation (2) and (3), we input $\hat{A}$ and $\mathbf{c}$ to the black box of $ZOIP$.

It is obvious that this construction can promise that the instance of **3SAT** has a satisfiable assignment iff the black box of **ZOIP** outputs a solution.

$\square$

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.