Lab10-NP and Reduction

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

* Name:Hongyi Guo   Student ID: 516030910306   Email: guohongyi@sjtu.edu.cn

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 $\sum_{a\in A'} s(a) \le B$ and $\sum_{a\in A'} v(a) \ge K$?

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

   Proof. Create SUBSET SUM instance Φ: the same finite set A, size $s(a)$ as PARTITION, $B = \frac{\sum_{a\in A} s(a)}{2}$. Then if PARTITION holds,

$$\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a) = \sum_{a\in A} s(a) - \sum_{a\in A'} s(a) = \frac{\sum_{a\in A} s(a)}{2} = B$$

   So, Φ holds if and only if PARTITION holds. Therefore, PARTITION $\le_p$ SUBSET SUM.

   □

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

   Proof. Create KNAPSACK instance Φ: the same A, $s(a)$, B as SUBSET SUM. $v(a) = s(a)$, $K = B$. Then if SUBSET SUM holds, $\sum_{a\in A'} s(a) = B = K$. So, $\sum_{a\in A'} s(a) \le B$ and $\sum_{a\in A'} v(a) \ge K$ holds, which means Φ holds if and only if SUBSET SUM holds. Therefore, SUBSET SUM $\le_p$ KNAPSACK.

   □

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 $K \le |V|$, is there a subset $V' \subseteq V$ with $|V'| \ge K$ such that every two vertices in V' are joined by an edge in E?

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

   Proof. Given an instance Φ of 3SAT, suppose it has K clauses, we construct an instance (G, K) of CLIQUE that has a subset of size K if and only if Φ is satisfiable.

   G contains 3 vertices for each clause, one for each literal. Then we connect each literal to literals in other clauses except its negations.

   Then we claim G contains a clique of size K if and only if Φ is satisfiable.

   Prove $\Rightarrow$ let C be a clique set of size K. S must contain exactly one vertex in each triangle for there is no edge between any two vertices in one triangle. Then we can set the literals in C to true. Note that a literal and its negation cannot appear in C together for there is no edge between them. So truth assignment is consistent and all clauses are satisfied.

   Prove $\Leftarrow$ Given satisfying assignment, select one true literal from each triangle. This is a clique of size k because any two literals in clique are adjacent directly.

   □

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. We prove this via proving 3SAT $\leq_p$ ZERO-ONE INTEGER PROGRAMMING.

   Given an instance Φ of 3SAT, suppose it has m clauses, we construct an instance P of ZERO-ONE INTEGER PROGRAMMING with a $m \times 3$ matrix A and m-vector b.

   This is how we construct A: for the i^{th} clause in Φ, we transfer it into the i^{th} row in A.

$$A[i][j] = \begin{cases} -1, & \text{the } j^{th} \text{ element in the } i^{th} \text{ clause is } x_j \\ 1, & \text{the } j^{th} \text{ element in the } i^{th} \text{ clause is } \bar{x_j} \end{cases}$$

   And,

$$b[i] = -1 + \text{the number of } \bar{x_j}\text{'s in the } i^{th} \text{ clause}$$

   Then we claim the vector integer n-vector x with elements in the set $\{0, 1\}$ such that $Ax \leq b$ exists if and only if Φ is satisfiable.

   Suppose $Ax = c$. Due to our construction, in the i^{th} clause, if all x_j is true, then $c[i]$ =the number of $\bar{x_j}$'s in the i^{th} clause $= b[i] + 1 > b[i]$. So $c[i] \leq b[i]$ if and only if $\exists j$, clause i contains $\bar{x_j}$ and $x[i] = 0$ or clause i contains $\bar{x_j}$ and $x[i] = 1$. We let x_j indicates x_j is true(1) or false(0). At this time, the clause is satisfied too.

   For the same reason, when the clause is satisfied, $c[i] \leq b[i], \forall i$

   $\square$