# I216E: Computational Complexity and Discrete Mathematics

Answers and Comments on Report 2

HOANG, Duc Anh (1520016)

November 17, 2016

Ph.D Student @ Uehara Lab
Japan Advanced Institute of Science and Technology
hoanganhduc@jaist.ac.jp

# Problem 1

We define an equivalence relation $\equiv_m^p$ as follows.

$$A \equiv_m^p B \leftrightarrow A \leq_m^p B \text{ and } B \leq_m^p A.$$

Prove that the relation $\equiv_m^p$ is surely an equivalence relation. Precisely, you need to show that it is reflexive, symmetric and transitive.

# Problem 1 (Answer)

We show that the relation $\equiv_m^p$ is

1. **reflexive:** $A \equiv_m^p A$.
    - It is clear because $A \leq_m^p A$. (Take function $h(x) = x$ for any $x \in A$.)
2. **symmetric:** if $A \equiv_m^p B$ then $B \equiv_m^p A$.
    - By definition, $A \equiv_m^p B$ if and only if $A \leq_m^p B$ and $B \leq_m^p A$.
    - Also by definition, $B \leq_m^p A$ and $A \leq_m^p B$ imply $B \equiv_m^p A$.
3. **transitive:** if $A \equiv_m^p B$ and $B \equiv_m^p C$ then $A \equiv_m^p C$.
    - By definition, $A \equiv_m^p B$ if and only if $A \leq_m^p B$ and $B \leq_m^p A$.
      Similarly, $B \equiv_m^p C$ if and only if $B \leq_m^p C$ and $C \leq_m^p B$.
    - $A \leq_m^p B$ and $B \leq_m^p C$ imply $A \leq_m^p C$.
      $C \leq_m^p B$ and $B \leq_m^p A$ imply $C \leq_m^p A$.
      Finally, $A \leq_m^p C$ and $C \leq_m^p A$ imply $A \equiv_m^p C$.

Hence, the relation $\equiv_m^p$ is an equivalence relation.

# Problem 2

It is nonsense that defining the class $coP$ because $coP = P$. Prove $coP = P$.

## Problem 2 (Answer)

Recall that for a language $L \subseteq \Sigma^*$, $\bar{L} = \{x \in \Sigma^* : x \notin L\}$. We define the class $coP$ is the set of all languages $L$ such that $\bar{L} \in P$. Now, if for any language $L \in P$, $\bar{L} \in P$, then we can conclude that $coP = P$ because

- For any $L \in P$, by our assumption, $\bar{L} \in P$; then by definition $\bar{\bar{L}} = L \in coP$, hence $P \subseteq coP$.
- For any $L \in coP$, by definition $\bar{L} \in P$; then by our assumption $\bar{\bar{L}} = L \in P$, hence $coP \subseteq P$.

It is sufficient to show that for any language $L \in P$, $\bar{L} \in P$. By definition, $L \in P$ if and only if the function $f : \Sigma^* \to \{0, 1\}$ defined by

$$f(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases} \tag{1}$$

is polynomial-time computable. Then, the following function $g : \Sigma^* \to \{0, 1\}$ is also polynomial-time computable

$$g(x) = \begin{cases} 0 & \text{if } x \in L \text{ (i.e. } x \notin \bar{L}) \\ 1 & \text{if } x \notin L \text{ (i.e. } x \in \bar{L}) \end{cases} \tag{2}$$

This means that $\bar{L} \in P$.

# Problem 3

Prove KNAP $\leq_m^p$ BIN.

# Problem 3

Prove KNAP $\leq_m^p$ BIN.

**Knapsack Problem (KNAP)**

- Input: $n+1$ tuple of natural numbers $\langle a_1, a_2, \ldots, a_n, b\rangle$.
- Question: Is there a set of indices $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = b$?

**Bin Packing Problem (BIN)**

- Input: $n+2$ tuple of natural numbers $\langle a_1, a_2, \ldots, a_n, b, k\rangle$.
- Question: Is there a partition of the set of indices $U = \{1, \ldots, n\}$ into $U_1, \ldots, U_k$ such that $\sum_{i \in U_j} a_i \leq b$ for each $j$?

JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY 1990

**To prove that KNAP $\leq_m^p$ BIN**

We need to define a function $h$ such that

(a) $h$ is a total function, i.e., for each instance $x$ of KNAP, $h(x)$ is defined (as an instance of BIN);

(b) $x$ is a YES-instance of KNAP if and only if $h(x)$ is a YES-instance of BIN;

(c) $h$ is polynomial-time computable.

Two steps need to be done:

- **Step 1:** Define $h$.
- **Step 2:** Check if $h$ satisfies (a), (b) and (c).

# Problem 3 (Answer) – Step 1

- **Step 1: Define $h$.** Let $\langle a_1, a_2, \ldots, a_n, b\rangle$ be an instance of KNAP. We construct an instance $\langle A_1, \ldots, A_{n+2}, B, K\rangle$ of BIN as follows. Let $A = \sum_{i=1}^{n} a_i$. We define $A_i = a_i$ for $i \in \{1, 2, \ldots, n\}$, $A_{n+1} = |2A - b|$, $A_{n+2} = A + b$, $B = 2A$ and $K = 2$. In other words,

$$h(\langle a_1, a_2, \ldots, a_n, b\rangle) = \langle a_1, \ldots, a_n, |2A - b|, A + b, 2A, 2\rangle.$$

- **Step 2: Check if $h$ satisfies (a), (b) and (c).** (a) and (c) are clear from the definition of $h$. It remains to check if (b) holds. That is, we need to check: $x$ is a YES-instance of KNAP if and only if $h(x)$ is a YES-instance of BIN.
  - Let $x = \langle a_1, a_2, \ldots, a_n, b\rangle$ be a YES-instance of KNAP, i.e., there exists a set of indices $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i\in S} a_i = b$. In this case, note that $2A - b \geq 2\sum_{i\in S} a_i - b = b \geq 0$, which means $|2A - b| = 2A - b$. We show that

$$h(x) = \langle a_1, \ldots, a_n, 2A - b, A + b, 2A, 2\rangle$$

    is a YES-instance of BIN.
    Choose $U_1 = S \cup \{n+1\}$ and $U_2 = U \setminus U_1$, where $U = \{1, 2, \ldots, n+2\}$. Hence,

$$\sum_{i\in U_1} A_i = \sum_{i\in S} a_i + (2A - b) = b + (2A - b) = 2A = B.$$

$$\sum_{i\in U_2} A_i = \sum_{i\notin S} a_i + (A + b) = (A - b) + (A + b) = 2A = B.$$

- Let $h(x) = \langle a_1, \ldots, a_n, |2A - b|, A + b, 2A, 2\rangle$ be a YES-instance of BIN, i.e., there exists a partition $U_1, U_2$ of $U = \{1, 2, \ldots, n+2\}$ such that $\sum_{i \in U_j} A_i \le B = 2A$ for $j \in \{1, 2\}$. We show that

$$x = \langle a_1, a_2, \ldots, a_n, b\rangle$$

is a YES-instance of KNAP.

Since $h(x)$ is a YES-instance of BIN, it must happen that $2A - b \ge 0$. Assume for the contradiction that $2A - b < 0$, i.e., $b > 2A$. Then $A_{n+2} = A + b > 3A$, which means that one cannot partition $U$ into $U_1, U_2$ such that $\sum_{i \in U_j} A_i \le B = 2A$ for $j \in \{1, 2\}$, i.e., $h(x)$ is a NO-instance Hence, $|2A - b| = 2A - b$.

Since $\sum_{i \in U} A_i = 4A$, it must happen that $\sum_{i \in U_1} A_i = \sum_{i \in U_2} A_i = 2A$. Moreover, since $A_{n+1} + A_{n+2} = (2A - b) + (A + b) > 2A$, without loss of generality, we can assume that $n + 1 \in U_1$ and $n + 2 \in U_2$. Hence, one can pick $S = U_1 \setminus \{n + 1\}$, and we have

$$\sum_{i \in S} a_i = \sum_{i \in U_1} A_i - (2A - b) = 2A - (2A - b) = b.$$