CS300 Homework #6

TA: Jiman Hwang
molehair@kaist.ac.kr

Total 100 points | Due: December 4 at 18:00:00 KST

1 True / False (30 points)

Give **true** or **false** for each of the following statements. Briefly justify your answer in less than four sentences.

1) If 3-CNF-SAT $\in$ **P**, then CLIQUE $\in$ **P**. [10 points]

sol) True. Since both 3-CNF-SAT and CLIQUE are **NP-complete**, CLIQUE $\le_p$ 3-CNF-SAT. Thus, CLIQUE $\in$ **P**.

2) For decision problems $L_1, L_2 \in$ **NP**, if **P** $\neq$ **NP**, $L_1 \le_p L_2$ and $L_2 \le_p L_1$, then L_1 and L_2 are **NP-complete**. [10 points]

sol) False. Let $L = L_1 = L_2$ be any decision problem in **P**. Then, $L_1 \le_p L_2$ and $L_2 \le_p L_1$ but L_1 and L_2 are not **NP-complete**.

3) For decision problems $L_1, L_2 \in$ **NP-complete**, if $L_1 \notin$ **P**, then $L_2 \notin$ **P**. [10 points]

sol) True. Contrapositive: $L_2 \in$ **P** $\to L_1 \in$ **P**. Since $L_1 \le_p L_2$, the statement is true.

2 Closest string (70 points)

Given two binary strings $x = x_1 \cdots x_n$, $y = y_1 \cdots y_n$, let $d(x, y)$ denote the number of different bit pairs. That is,

$$d(x,y) = \sum_{i=1}^{n} (x_i \oplus y_i) \quad \text{with } x_i \oplus y_i = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{otherwise} \end{cases}$$

For example, $d(0011, 0001) = 1$ and $d(0011, 1100) = 4$. We define the Closest String Problem as follow.

Definition Closest String Problem(CSP)
Given a natural number k and m strings $s_1, s_2, \cdots, s_m$ where $s_i \in \{0,1\}^n$ for $i = 1, \cdots, m$, is there a string $t \in \{0,1\}^n$ such that $d(t, s_i) \le k$ for all i?

1) Assuming $m = O(n^c)$ for some constant c, prove that CSP $\in$ **NP**. [10 points]

sol) Given the certificate $t \in \{0,1\}^n$, computing $d(t, s_i)$ takes $O(n)$. Repeating this for $i = 1, \cdots, m$ takes $O(mn)$, taking polynomial time with respect to n.

We'll show that CSP $\in$ **NP-hard** by proving 3-CNF-SAT $\leq_p$ CSP, thus CSP $\in$ **NP-complete**.

Given a boolean formula $\phi = C_1 \wedge \cdots \wedge C_m$ of variables $x_1, \cdots, x_n$ in 3-CNF, generate the strings $s_1, \cdots, s_{6n+m}$ each of length $2n$-bit as follow.

For $i = 1, \cdots, 2n$

$$s_i = b_1 b_2 \cdots b_{2n} \quad \text{with } b_j = \begin{cases} 0 & j = i \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

For $i = 2n+1, \cdots, 4n$, $s_i = \overline{s_{i-2n}}$ where $\overline{s}$ is bit-wise negation of s.

For $i = 4n+1, \cdots, 5n$

$$s_i = b_1 b_2 \cdots b_n c_1 c_2 \cdots c_n \quad \text{with } b_j = c_j = \begin{cases} 0 & j = i - 4n \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

For $i = 5n+1, \cdots, 6n$, $s_i = \overline{s_{i-n}}$ where $\overline{s}$ is bit-wise negation of s.

For $i = 6n+1, \cdots, 6n+m$,

$$s_i = b_1 b_2 \cdots b_n c_1 c_2 \cdots c_n \quad \text{with}$$

$$b_j = \begin{cases} 1 & \text{if } C_{i-6n} \text{ includes the literal } x_j \\ 0 & \text{otherwise} \end{cases} \qquad c_j = \begin{cases} 1 & \text{if } C_{i-6n} \text{ includes the literal } \neg x_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

For each string $s = b_1 \cdots b_n c_1 \cdots c_n$, b_j corresponds to x_j, and c_j does $\neg x_j$.

For example, given $\phi = (x_1 \vee x_2 \vee \neg x_4) \wedge (x_1 \vee \neg x_3 \vee x_4)$, we generate

$$\begin{array}{lllll} s_1 = 01111111 & s_9 = 10000000 & s_{17} = 01110111 & s_{21} = 10001000 & \\ s_2 = 10111111 & s_{10} = 01000000 & s_{18} = 10111011 & s_{22} = 01000100 & s_{25} = 11000001 \\ \vdots & \vdots & s_{19} = 11011101 & s_{23} = 00100010 & s_{26} = 10010010 \\ s_8 = 11111110 & s_{16} = 00000001 & s_{20} = 11101110 & s_{24} = 00010001 & \end{array}$$

Note that this process runs in polynomial time.

At first, we show that if 3-CNF-SAT produces the positive answer, then so does CSP.

2) Prove that if ϕ has a satisfying assignment, then there exists $t \in \{0,1\}^{2n}$ such that $d(t, s_i) \leq n+1$ for all i. [15 points]
(Hint: Convert the satisfying assignment $b_1, \cdots, b_n$(1 for `TRUE`, 0 for `FALSE`) into $t = b_1 \cdots b_n c_1 \cdots c_n$ with $c_j = \neg b_j$)

sol) Let $b_1, \cdots, b_n$ be the satisfying assignment and $t = b_1 \cdots b_n c_1 \cdots c_n$ with $c_j = \neg b_j$. Note that t has exactly n 1s and n 0s.
For any s_i where $i = 1, \cdots, 2n$, t and s_i have at least $(n-1)$ 1s in common positions, thus $d(t, s_i) \leq n+1$.
For any s_i where $i = 2n+1, \cdots, 4n$, t and s_i have at least $(n-1)$ 0s in common positions, thus $d(t, s_i) \leq n+1$.
For any s_i where $i = 4n+1, \cdots, 5n$, t and s_i have at least $(n-1)$ 1s in common positions, thus $d(t, s_i) \leq n+1$.
For any s_i where $i = 5n+1, \cdots, 6n$, t and s_i have at least $(n-1)$ 0s in common positions, thus $d(t, s_i) \leq n+1$.
For any s_i where $i = 6n+1, \cdots, 6n+m$, t and s_i have at least a 1 in a common position(satisfying assignment) and at least $(n-2)$ 0 in common positions(there are exactly 3 1s in each s_i). Thus $d(t, s_i) \leq n+1$.

Next, we show that if CSP produces the positive answer, then so does 3-CNF-SAT.

For the problems 3) - 5), assume that there exists $t \in \{0,1\}^{2n}$ such that $d(t, s_i) \leq n+1$ for all i.

3) Prove that t has exactly n 1s and n 0s. [15 points]
(Hint: Use (1) to show that t contains at least n 1s)

sol) Note that t includes at least a $1(\because s_1, \cdots, s_{2n})$. We assert that t includes at least n 1s. Suppose t has at most $(n-1)$ 1s. It follows that t has at least $(n+1)$ 0s. Then, there exists s_i with $i \in [1, 2n]$ such that at least $(n+1)$ 0s in t correspond to $(n+1)$ 1s in s_i, and the 0 in s_i corresponds to a 1 in t, followed by $d(t, s_i) \geq n+2$.

$$\begin{aligned} s_i &= 1 \cdots 111011 \\ t &= \underbrace{0 \cdots 0}_{\geq n+1} \underbrace{1 \cdots 1}_{\leq n-1} \end{aligned}$$

This is contradiction, thereby, t has at least n 1s. Similarly, we can use $s_{2n+1}, \cdots, s_{4n}$ to show that t has at least n 0s. Therefore, t has exactly n 1s and n 0s.

4) Prove that if $t = b_1 \cdots b_n c_1 \cdots c_n$(binary string), then $b_j \neq c_j$ for j. [15 points]
(Hint: Use (2) to show that there does not exist k such that $b_k = c_k = 1$)

sol) Suppose $b_k = c_k = 1$ for some k. Then, $d(t, s_{4n+k}) \geq n+2$ because n 0s in t correspond to n 1s in s_{4n+k}, and the two 0s in s_{4n+k} correspond to $b_k = c_k = 1$ in t.

$$\begin{aligned} s_{4n+1} &= 01 \cdots 1\ 01 \cdots 1 \\ t &= 1? \cdots ?\ 1? \cdots ? \quad \text{(when } k = 1\text{, for example)} \end{aligned}$$

This is contradiction, thereby, there doesn't exist k such that $b_k = c_k = 1$. Similarly, we can use $s_{5n+1}, \cdots, s_{6n}$ to show that there doesn't exist k such that $b_k = c_k = 0$. Therefore, $b_j \neq c_j$ for all j.

5) Prove that ϕ has a satisfying assignment. [15 points]
(Hint: Use (3) to show that t and s_{6n+i} where $i = 1, \cdots, m$ have at least a 1 in a common position.)

sol) We assert that t and s_{6n+i} where $i = 1, \cdots, m$ have at least a 1 in a common position. Suppose t and s_{6n+i} have no 1 in common position. Then, $d(t, s_{6n+i}) = n+3$ because s_{6n+i} has exactly three 1s that correspond to three 0s in t, and t has n 1s that correspond to n 0s in s_{6n+i}.

$$\begin{aligned} s_{6n+i} &= 1110000 \cdots 0000 \\ t &= 0000 \underbrace{1 \cdots 1}_{n} 0 \cdots 0 \end{aligned}$$

This is contradiction, thereby, t and s_{6n+i} where $i = 1, \cdots, m$ have at least a 1 in common position.
Since t is expressed as $t = b_1 \cdots b_n c_1 \cdots c_n$ with $c_j = \neg b_j$, $(b_1, \cdots, b_n)$ is an assignment for ϕ(`TRUE` for 1, `FALSE` for 0). Moreover, each s_{6n+i} where $i = 1, \cdots, m$ corresponds to the clause C_i. As t and s_{6n+i} have at least a 1 in common position, the clause C_i has a literal, assigned as `TRUE`.