Correction of Homework 6

April 2, 2021

TangLin

Correction 1. Solution:

We need also an additional proof about size of certificate. Assume that length of binary representation of algebraic expression exp is $len(exp) = n$, we need to prove that there will be certificated whose length is $poly(n)$.
I'm going to use the fundamental theorem of algebra to prove it.
First, we observe the algebraic function with one variable. If it is an n-th order expression, we need at least $\log_2 n$ bits to describe it. Because when $f(x) = x^n$ or $f(x) = \underbrace{x \cdot x \cdot x \cdots x}_{n \text{ times}}$, it requires the least bits. That is, if $len(exp) = n$, the expression is at most $log_2 n$ degree. According to the fundamental theorem of algebra, we know that there will be at most n different values α such that $f(\alpha) = 0$. So we can test number α from 1 to $2n$, if $\forall \alpha \in \{1, 2, \cdots, 2n\} f(\alpha) = 0$, then we can assert that $f(x) = 0$.
For multi-variables function, we can get the similar results.
If $len(f(x_1, x_2, \cdots, x_m)) = n$, then assume the $degree(x_i) = p_i$, we have

$$\sum_{i=1}^{m} len(p_i) = \sum_{i=1}^{m} \log_2 p_i \leqslant n$$

If we concentrate on only variable x_k and fix all the others, we will get a single variable function, and we can test number α_k from 1 to $2p_k$. It needs at most $\log_2 p_k$ bits. So for all the m variables, we can use vector$(\alpha_1, \alpha_2, \cdots, \alpha_m)$, where$\alpha_i \in \{1, 2, \cdots, p_i\}$, to build the certificates. Every that kind of certificate need at most $\sum_{i=1}^{m} \log_2 p_i <= n$ bits.
So, we have proved that

$$M(exp\#\vec{x}) = 1 \Leftrightarrow \exists \vec{x} exp(\vec{x}) \neq 0.$$

Where $(\vec{x}) \in O(len(exp)) = O(n)$.

Correction 2. Solution:

This bug can be repaired, but the process of construction of set family need to be changed. First, we need find out degree of every vertex. We can easily prove that it can be achieved in polynomial time. We need only traverse E and record every $d(i)$, then select the min one.
For example,$G = (V, E)$, where $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_3, v_4), (v_3, v_5),$

$$d(v_1) = 3, d(v_2) = 1, d(v_3) = 3, d(v_4) = 4, d(v_5) = d(v_6) = d(v_7) = 1$$

And then sort them from smallest to largest, it can be completed in polynomial time. We got

$$v_2, v_5, v_6, v_7, v_1, v_3, v_4.$$

Then we build set for every vertex by following rule according to the sorted list of vertices:

1. For v_i, we build an empty S_i, then append i to it.
2. If vertex v_j which connects to v_i has been translated to set, then append j to S_i.
3. When traverse all the neighborhoods have been check, then return to the first step for the next vertex.
4. When all the vertices have been translated, then we get the set family.

For our example,

1. $S_2 = \{2\}$
2. $S_5 = \{5\}$
3. $S_6 = \{6\}$
4. $S_7 = \{7\}$
5. $S_1 = \{1{,}2\}$
6. $S_3 = \{3{,}1{,}5\}$
7. $S_4 = \{4{,}3{,}6{,}7\}$

What needs illustration is that, if the graph G has some disjoint subgraph or some isolated points (they can be treated as subgraphs), we can execute the translation algorithm for every subgraph.