Hardness HW 9 solution

TAMREF

tamref.yun@snu.ac.kr

Due: December 15, 2022

Rules

- You can just link your former post instead of the solution. Otherwise, it is recommended to write the proof in your language.
- You are qualified if you **read** all the problems, and answered **at least 2** of them.

Question 1. Counting Bipartite Matching is Hard

Imitating the proof of $\sharp\mathrm{BPM} \leq_{p.o.} \sharp\mathrm{BMM}$, show that counting the number of bipartite matching is $\sharp\mathbb{P}-\mathrm{complete}$.

Proof. Given a graph $G = (A \sqcup B, E)$ with $|A| = |B| = n$, add nk additional vertices and nk additional edges to graph G_k. For each $b_i \in B$, connect b_i with k additional nodes. Given a matching $N \subseteq E$ with $|N| = r$, there are $(k+1)^r$ matchings M in the extended graph satisfying $M \cap E = N$. Thus

$$\sharp\mathrm{BM}(G_k) = \sum_r m_{n-r}(k+1)^r \tag{1}$$

Where m_k is the number of bipartite matchings with cardinality k. Evaluating above for $k = 0, \cdots, n$, we obtain the polynomial $f(x)$ with $[x^i]f(x) = m_{n-i}$. Hence $m_n = \sharp\mathrm{BPM}(G)$ is easily obtained from $\sharp\mathrm{BM}$ oracles. □

Question 2. Another proof for hardness of $\sharp\mathrm{CLIQUE}$

Construct a parsimonious reduction $\mathrm{TH-POS-2SAT} \to \mathrm{CLIQUE}$. This gives an alternative proof about $\sharp\mathbb{P}-\mathrm{complete}$ness of $\sharp\mathrm{CLIQUE}$.

Proof. One can easily find that $\mathrm{TH-POS-2SAT}$ is just a Vertex Cover Problem written in the language of SAT. Motivated by this, given a $\mathrm{TH-POS-2SAT}$ instance (ϕ, t), design a graph G_ϕ. The vertices of G_ϕ corresponds to each variable, and an edge $e = uv$ presents in G_ϕ iff a clause $x_u \vee x_v$ is **not** present in ϕ. Then ϕ is satisfied if and only if there is a clique with size $\geq t$ in G_ϕ. And the variable assigned to be **false** precisely correspond to the vertex set giving the clique, proving the parsimonity of the reduction. □

Question 3. UP-hardness of Permanent modulo k

In this problem, we investigate how Valiant(1979) could separate the complexity of $\operatorname{per}(M) \mod 2^d$ and other $\operatorname{per}(M) \mod k$ for $k \neq 2^d$.

> **Definition.** A decision problem $A = \{x : \exists^p y \text{ s.t. } (x, y) \in B\} \in \mathbb{NP}$ is called $\mathbb{UP}$ if $x \in A$ implies that there is **only one** y such that $(x, y) \in B$.

Given the lemmas below, define the reduction for $\mathbb{UP}$ and deduce that computing $\operatorname{per}(M) \mod k$ is $\mathbb{UP}$-hard. According to complexityzoo, it is not known whether $\mathbb{UP}$ possess a complete problem.

(a) **(Valiant 1979, Lemma 3.1)** Given a CNF formula F, there is a function $f \in \mathbb{FP}$ such that $f(F)$ is a $\{-1, 0, 1, 2, 3\}$-matrix such that

$$\operatorname{per}(f(F)) = 4^{t(F)} \cdot \sharp\mathrm{SAT}(F)$$

- $t(F)$: twice the number of occurrences of literals − the number of clauses in F. The only important fact is that $t(F)$ is a positive integer.

(b) **(Valiant 1979, Lemma 3.2)** For any $A \in \mathbb{NP}$, there is a parsimonious reduction $A \leq_p \mathrm{3SAT}$.

Proof. Given a problem $A \in \mathrm{UP}$, take a parsimonious reduction $g : A \to \mathrm{3SAT}$, according to the lemma (b). Given an input x, $\sharp\mathrm{SAT}(g(x)) = 1$ if and only if $x \in A$. For $k \neq 2^d$, we can determine the condition above by computing $\operatorname{per}(f(g(x))) \mod k$, and the oracle is provided by the lemma (a). □

Question 4. Computing volume of a polytope is $\sharp P$-hard

Here, we define a **polytope** as an intersection of half-planes in n-dimension.

> **Definition.** Given an n-element POSET $(P, <)$, **order polytope** $\mathcal{O}_P$ of P is given by the sub-polytope of $C = [0, 1]^n$, defined by
>
> $$\mathcal{O}_P := C \cap \bigcap_{p<q} \{(x_1, \cdots, x_n) : x_p < x_q\}$$

It is known that computing the number of **linear extensions (or, topological sorts)** of **POSETs (or, DAGs)** is $\sharp P-\mathrm{complete}$. (Brightwell & Winkler, 1991). Assuming the previous fact, deduce that computing volume of the polytope is $\sharp P$-hard, even for order polytopes.

Proof. Refer to `https://infossm.github.io/blog/2022/11/08/toposort/`. $\square$