COL872
Problem Set 1

Mallika Prabhakar (2019CS50440)
Sayam Sethi (2019CS10399)
Satwik Jain (2019CS10398)

January 2023

Contents

1 Question 1

Question 1: Doubly Efficient Interactive Proofs

Question. *Give a doubly-efficient protocol for the disjoint sets problem where the verifier runs in time $\tilde{O}(n)$, and the prover runs in time $\tilde{O}(n^2)$,*

Proof. We have two set of sets: $S = \{S_i\}_{i\in[n]}$ and $T = \{T_i\}_{i\in n}$, and each $S_i, T_i \subseteq [d]$.
Numbers till n can be represented using **log n** bits. So we will try to represent the problem as a polynomial in 2log n bits(log n bits each for S and T). To do this,
For set of sets S, for each $j \in [d]$ which can exist in a set, we can create a Boolean algebra expression which takes as input the log n bits which create a number **i** and outputs 1 if $j \in S_i$ and 0 otherwise. Let us name this expression as S^j. Similarly we can create boolean algebra expression T^j.
Now taking a **NAND** of these two expression $(1 - S^jT^j)$, for the input bits $I = i_1i_2...i_{logn}$ and $K = k_1k_2...k_{logn}$, we will get an output of 0 if the element j is present in both the sets S_i and T_k and 1 otherwise. In this way we can check for one element. Now we can multiply all the NAND expressions and the resulting expression

$$f(I, K) = \prod_{j=1}^{d}(1 - S^j(I)T^j(K))$$

will return 0 if any one of the numbers in set [d] is present in both of S_i and T_k i.e.,$S_i \cap T_k \neq \phi$ or the intersection of the sets S_i and T_k is empty and return 1 otherwise.
Now to get the answer, Δ, we can simply sum up the outputs of $f(I, K)$ over all possible values of I and K.

$$\Delta = \sum_{I,K\in\{0,1\}^{logn}} f(I, K)$$

Now we can apply the sum check protocol over this. Suppose the value of Δ is p. Now the prover will prove that $\delta = p$ to the verifier. as the total number of variables is 2log n, We will have 2log n rounds in the sum check protocol. In each iteration of the protocol, the verifier has to evaluate a univariate polynomial of degree atmost **d** at each iteration. which will take O(d) iterations via the Horner's rule. So the expected running time of the Verifier is O(d * 2log n) = O(log^3 n) as d= log^2 n which can be represented as $\tilde{O}(1)$(as poly logarithmic factors are omitted).
Now we will find the running time of the prover. In the j^{th} sum, we have (1+deg(g))2^{v-j} terms where g is the univariate polynomial formed. As there are 2log n rounds, for a v round protocol, the prover has to do $\sum_{j=1}^{2logn}(1 + deg(g))2^{v-j}$ evaluations. As each univariate polynomial will have a degree of atmost d, this comes out to be O(2^vd) evaluations. For this problem we have $v = 2log$ n, So we will have O(n^2log^2 n) evaluations and we have know that each evaluation takes O(log^2 n) time for a O(log n) variate polynomial, so the total time for the prover will be O(n^2log^4 n) which is $\tilde{O}(n^2)$. □

2 Question 2

Zero-knowledge protocols for group-theoretic problems

2.1 Question 2.1

HVZK PoK for DDH

Question. *Consider the following language:*

$$\mathcal{L}_{DDH} = \{(g, h, u, v) : \exists a \in \mathbb{Z} s.t. u = g^a, v = h^a \tag{1}$$

Construct an honest-verifier zero-knowledge proof-of-knowledge protocol for $\mathcal{L}_{DDH}$. The protocol must have perfect completeness, the knowledge error must be $1/q$, and it should satisfy the honest-verifier zero-knowledge property.

Proof. In an honest verifier setting, the zero-knowledge, proof-of-knowledge can be defined as follows:

- Consider an honest verifier $\mathcal{V}$ and a prover $\mathcal{P}$.
- **Common input:** g, u, h, v where all of them are group elements.
- **Prover's private input:** a s.t. $u = g^a, v = h^a$
- **Claim to prove:** Prover knows a

The protocol:

1. **P:** prover picks random t_1, t_2, t_3 and sends $x = g^{t_1}$, $y = h^{t_2}$ and $z = (g \cdot h)^{t_3}$ to the verifier
2. **V:** verifier sends uniformly random c_1, c_2, c_3 with respect to the pairs $(g, u), (h, v), (g \cdot h, u \cdot v)$
3. **P:** prover calculates $w_1 = t_1 + c_1 \cdot a$, $w_2 = t_2 + c_2 \cdot a$ and $w_3 = t_3 + c_3 \cdot a$ and sends w_1, w_2, w_3 to the verifier
4. **V:** verifier checks if $x \cdot u^{c_1} = g^{w_1}$, $y \cdot v^{c_2} = h^{w_2}$ and $z \cdot (u \cdot v)^{c_3} = (g \cdot h)^{w_3}$

□

2.2 Question 2.2

ZK protocol for DDH

Question. *quess*

Proof. □

2.3 Question 2.3

HVZK PoK for k-out-of-t DDH

Question. *quess*

Proof. □

3 Question 3

Question 3: Impossibility of two-round zero-knowledge protocols with auxiliary information

Question. *quess*

Proof. (Proof Idea: We use the auxiliary input as the string that is sent to the prover. Thus, the simulator (even if it is non-black box) cannot "generate" a valid response unless L itself is in BPP)
The default interaction between the prover and the verifier is defined as:

1. $V_1(x, r) = m_1$ is sent to the prover
2. $P_2(x, m_1) = m_2$ is set to the verifier
3. $V_3(m_1, m_2) = m_3 \in \{0, 1\}$ is the result of the interaction (rejected or accepted)

Now consider the following verifier $V^* = (V_1^*, V_3^*)$ that takes in auxiliary information z and interacts with the prover as follows:

$$V_1^*(z)(x, r) = z, V_3^*(z)(x, r, m_2) = V_3(m_1, m_2) \tag{2}$$

Since by our assumption, the given protocol is two-round zero-knowledge in the presence of auxiliary information, we will have a simulator for V^*. Let any such simulator be S^*. We now propose the following algorithm $\mathcal{A}$ for checking if $x \in \mathcal{L}_{yes}$,

Algorithm $\mathcal{A}$

1. Sample a random r and compute $m_1 = V_1(x, r)$
2. Obtain the transcript (m_1, m_2, m_3) on running $S^*(x, m_1)$.
3. If S^* fails to simulate, output 0. Else output m_3.

Figure 1: Algorithm for checking $x \in \mathcal{L}_{yes}$

We first note that $\mathcal{A}$ is a ppt algorithm since it uses V_1, S^* in sequence which are both ppt algorithms. Now, consider the probability of $\mathcal{A}$ in deciding $x \in \mathcal{L}_{yes}$ when x is a yes instance (assuming that completeness of the protocol is c and soundness error is s),

$$\begin{aligned} \Pr[\mathcal{A} \text{ succeeds}] &= \Pr[S^* \text{ succeeds}] \cdot \Pr[m_3 = 1] \\ &= (1 - \mu(n)) \cdot \Pr[m_3 = 1] \\ &\geq (1 - \mu(n)) \cdot (c - s) \end{aligned} \tag{3}$$

Therefore, we can see that $\mathcal{A}$ can decide $x \in \mathcal{L}_{yes}$ with probability $> 1/3$ in ppt. Therefore, $\mathcal{L}$ is in BPP. □