Technische Universität München (I7)
Prof. J. Křetínský / Dr. M.Raskin

Summer term 2019
22.05.2019

Solution

Computational Complexity – Homework 8

Discussed on 24.05.2019.

Exercise 8.1

Give an interactive proof protocol for graph isomorphism and show that your protocol satisfies the completeness and soundness requirements.

Can you give a zero-knowledge one, too?

For this exercise, it is enough to use perfect zero knowledge: each execution of a protocol leaves a complete execution log; for a perfect zero-knowledge protocol there is a polynomial-time algorithm generatig logs with the exact same distribution.

Solution: A simple protocol is one in which prover provides an isomorphism and then verifier checks that it really is an isomorphism. This is not zero-knowledge since verifier learns what the isomorphism is (in addition to its mere existence).

The following is a zero knowledge protocol:

(a) Input: Two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ (represented as adjacency matrices).

(b) If $h : \mathcal{G}_1 \cong \mathcal{G}_2$, then prover uniformly at random chooses a permutation of nodes π and computes the graph $H := \pi(\mathcal{G}_1)$. It is thus the case that H is isomorphic to both graphs with isomorphisms $h_1 : H \cong \mathcal{G}_1$ and $h_2 : H \cong \mathcal{G}_2$. (If the graphs are not isomorphic, then prover can behave arbitrarily).

(c) Prover sends H to verifier.

(d) Verifier uniformly at random chooses a bit $b \in \{0, 1\}$ and sends b to prover.

(e) Prover sends h_b to verifier.

(f) Verifier accepts if h_b is a permuation of the nodes of H and also $h_b(H) = \mathcal{G}_b$, otherwise rejects.

If the graphs are isomorphic and the parties follow the protocol, then it is clear that verifier will *always* accept. If the graphs are non-isomorphic, then it is impossible for prover to pick an H that is isomorphic to both graphs and so verifier will accept with probability at most 0.5 (if verifier is unlucky and chooses b such that H is isomorphic to $\mathcal{G}_b$ and prover returns the associated isomorphism).

Thus by repeating the protocol twice and accepting iff both repeats are accepting, verifier will always accept when the graphs are isomorphic and incorrectly accept only with probability bounded above by $0.25 < 1/3$.

In order to formally show that the protocol is zero-knowledge, we show that in the case when $\mathcal{G}_1 \cong \mathcal{G}_2$ there exists a probabalistic Turing Machine running in expected polynomial time (in the size of the two graphs) that generates transcripts of the conversation with the same probability distribution that would result from interacting with prover. This is the case even if verifier cheats (diverges from the protocol) so long as verifier remains a probabalistic turing machine operating in polynomial time.

Let $V(\mathcal{G}_1, \mathcal{G}_2, H)$ be a probabalistic polynomial time Turing machine that verifier uses to generate the bit $b \in \{0, 1\}$ that is sent to prover after prover has sent H to verifier.

A PTM operating as follows will then generate transcripts with the same probability distribution as genuine interactions (where verifier uses the machine V to generate its challenge bit) and will terminate in expected polynomial time:

(a) Input: Graphs $\mathcal{G}_1$ and $\mathcal{G}_\in$ that are isomorphic.

(b) Uniformly at random pick a bit $b' \in \{0, 1\}$ and then pick a permutation π and compute $H := \pi(\mathcal{G}_{b'})$.

(c) Run the machine V to compute $b := V(\mathcal{G}_1, \mathcal{G}_2, H)$.

(d) If $b = b'$, then output the transcript: (H, b, π^{-1})

(e) If $b \neq b'$, then go back to the beginning and start again

(The expected number of repeats is 2 since b' is chosen independently of b and so the probability of $b = b'$ on a single iteration is $1/2$).

Exercise 8.2

The class **IP** requires that for *some* prover probability of acceptance of a good word is at least $2/3$ and for *all* provers probability of acceptance of a bad word is at most $1/3$.

What class we get if we only require *existence* of a prover in both cases?

What class we get if we require the property of *all* provers in both cases?

What class we get if we swap the quantifiers?

Solution: 1. All languages, including undecidable ones

2. **BPP**

3. $co - \mathbf{IP} (= \mathbf{IP} = \mathbf{PSPACE})$

Exercise 8.3

Let p be a prime number. An integer a is then a quadratic residue modulo p if there is some integer b s.t. $a \equiv b^2 \pmod{p}$.

(a) Show that QR $:= \{(a, p) \in \mathbb{Z}^2 \mid a \text{ is a quadratic residue modulo } p\}$ is in **NP**.

(b) Set QNR $:= \{(a,p) \in \mathbb{Z}^2 \mid a$ is $\underline{\text{not}}$ a quadratic residue modulo $p\}$.

Complete the following sketch to an interactive proof protocol for QNR and show its completeness and soundness:

i) Input: integer a and prime p.

ii) The verifier chooses $r \in \{0, 1, \ldots, p-1\}$ and $b \in \{0, 1\}$ uniformly at random, keeping both secret.

   i. If $b = 0$, the verifier sends $r^2 \mod p$ to the prover.

   ii. If $b = 1$, the verifier sends $ar^2 \mod p$ to the prover.

iii) ...

Exercise 8.4

Is there an **IP** protocol consisting of $O(n)$ copies of some interaction, such that any constant number of rounds is not enough?

Exercise 8.5

Show that *perfect soundness* collapses the class **IP** to **NP**, where perfect soundness means soundness with error probability 0.

Solution: We already know that **NP** $\subseteq$ **IP** (prover provides the certificate and verifier checks it).

Now suppose that $\mathcal{L} \in$ **IP** and that this is witnessed by a k-round interactive proof that is *perfectly sound.* By the definition of **IP** there must be a deterministic polynomial time Turing machine $V_i(x, u, x_1, y_1, \ldots, x_i)$ giving verifier's response at the ith step of the protocol on input x where u is a polynomial length string chosen uniformly at random, x_j is Prover's jth response, and y_j is the message sent by verifier at the jth step. When $i = k$ this response will be either 'accept' or 'reject'. Due to perfect soundness verifier will never accept $x \notin \mathcal{L}$.

We can thus construct a deterministic polynomial time Turing machine $M(x, u_1 \cdots u_k, x_1, \ldots, x_k)$ that behaves as follows:

(a) For $i = 1$ to k compute $y_i := V_i(x, u_i, x_1, y_1, \ldots, x_i)$.

(b) Accept if y_k is accept, otherwise reject.

Regardless of the choice of $u_1, \ldots, u_k, x_1, \ldots, x_k$ perfect soundness ensures that M will reject every $x \notin \mathcal{L}$. Since the interactive proof will result in verifier accepting $2/3$ of the time, there must in particular exist some choice of string $u_1, \ldots, u_k$ and some choice of responses $x_1, \ldots, x_k$ by prover that results in verifier accepting. There must thus exist some certificate resulting in M accepting x. Thus M witnesses $\mathcal{L} \in$ **NP**.