1 Review of Interactive Proofs

Recall the definition of an interactive proof system from last time:

Definition. An interactive proof system for a language $L$ is a pair of interactive TMs, Prover ($P$) and Verifier ($V$). $V$ is ppt (probabalistic polynomial time), and $P$ is computationally unbounded, with the following two conditions:

- **Completeness:** For all $x \in L$, $V(x)$ accepts (after interacting with $P(x)$) with probability $\geq \frac{2}{3}$.
- **Soundness:** For all $x \notin L$, and for all TMs $P^*$ (thought of as a malicious prover), $V(x)$ accepts (after interacting with $P^*(x)$) with probability $\leq \frac{1}{3}$.

Then we defined the class

$$\text{IP} = \{ L : L \text{ has an interactive proof system.} \}$$

Last time, we saw that TAUTOLOGY $\in \text{IP}$. In fact, it turns out that $\text{IP} = \text{PSPACE}$.

2 Zero-Knowledge proofs

Definition. An interactive proof system for $L$ is zero-knowledge if, in addition to completeness and soundness, it satisfies:

- **Zero-knowledge:** For all ppt TMs $V^*$ (thought of as a malicious verifier), whatever $V^*$ can compute about input $x$ after interacting with $P$ can be computed in ppt without interacting with $P$.

This definition is not quite formal, because it is unclear what “can compute” means. To formalize it, we say that every $V^*$ has a corresponding simulator which has no access to the prover $P$ and yet produces a transcript that “looks like” the interaction between $V^*$ and $P$. (Again, this is intuitive rather than formal.)

As an example, here is a zero-knowledge proof for graph isomorphism. The input is $G_0, G_1$, two graphs.

- **Prover:** Pick $\pi$, a permutation of the vertices, at random.
• **Prover:** If $(G_0, G_1) \in \text{GraphIso}$, then there exists a $\rho$ such that $G_0 = \rho(G_1)$. Set $H = \pi(G_0) = \pi \circ \rho(G_1)$. Send $H$ to Verifier.

• **Verifier:** Set $b \in \{0, 1\}$ at random. Send $b$ to Prover.

• **Prover:** If $b = 0$, set $\alpha = \pi$. Else, set $\alpha = \pi \circ \rho$. Send $\alpha$ to Verifier.

• **Verifier:** Check that $H = \alpha(G_b)$.

Note that Verifier runs in probabilistic polynomial time. To prove this works, we need to show completeness, soundness, and zero-knowledge. **Completeness:** If the graphs are isomorphic, then $P$ sends over a correct permutation no matter what, and Verifier always accepts. **Soundness:** If the graphs are not isomorphic and we have some other malicious prover $P$, then Verifier will catch Prover by choosing the wrong bit at least half of the time. By repeating the procedure, we get error of at most $\frac{1}{4}$. **Zero-knowledge:** Design a simulator as follows:

1. Guess $b'$, and pick a random $\alpha$ such that $H = \alpha(G_b')$.
2. Send $H$ to $V^*$.
3. Receive $b$. If $b = b'$, respond with $\alpha$, and output whatever $V^*$ outputs. Else, go to (1).

We can then define the complexity class

$$ZK = \{L : L \text{ has a zero-knowledge interactive proof system.}\}$$

It is immediate that $ZK \subseteq \text{IP}$, that is, $ZK \subseteq \text{PSPACE}$.

**Theorem.** $\text{NP} \subseteq \text{ZK}$.

**Proof.** It suffices to give a zero-knowledge proof-system for 3-colorability.

• **Prover:** Take as input the graph $G$ and the three-coloring $R, G, B$. Randomly permute $R, G$, and $B$. Send “commitments” $c_1, \ldots, c_n$ to the color of every vertex in $G$ over to Verifier, so that Prover cannot later change her mind about what color the vertices are. (This is done with some sort of cryptography.)

• **Verifier:** Choose a random edge $e \in E(G)$, and send it to Prover.

• **Prover:** “Open” the commitments $c_i, c_j$, and send them over to Verifier.

• **Verifier:** Accept if $c_i, c_j$ open to distinct colors, in set $R, G, B$.

Here is the analysis:

**Completeness:** If the graph is 3-colorable, then the edge will correctly open to two distinct colors, so Verifier will always accept.

**Soundness:** If the graph is not actually 3-colorable, then a malicious prover will have to color some two adjacent vertices the same color. If the verifier gets lucky and picks this edge, then the prover will not be able to open the commitments to distinct colors. The verifier will reject. So the probability of acceptance is at most $1 - \frac{1}{|E(G)|}$. By repeating the entire procedure, we can amplify this to $\epsilon > 0$.

**Zero-knowledge:** Design a simulator as follows: guess $e$, and pick random distinct colors for its endpoints, and color everything else green, and then commit to those colors. If we guessed the right edge, then great. Else, try again.
3 Another complexity class diagram

An arrow from $A$ to $B$ means $A$ is contained in $B$. All containments are open problems.

\[ \text{IP} = \text{PSPACE} \]

\[ \text{NP} \]

\[ \text{BPP} \]

\[ \text{coNP} \]

\[ \text{RP} \]

\[ \text{coRP} \]

\[ \text{ZPP} = \text{RP} \cap \text{coRP} \]

\[ \text{P} \]

4 Wrap-up

- Consider taking other theory courses.
  - Spring: CS151 crypto; CS155 probabilistic methods in computer science.
  - Fall: CS157 algorithms; multiprocessor synchronization; computational biology.

- Come to Shafi Goldwasser’s Talk: Next Wednesday, 4pm, this room. Shafi co-invented zero-knowledge proofs.

- Final Exam: Tuesday 5pm, Markovitz Auditorium, Meeting St. A cheat sheet allowed. One letter-size sheet, handwritten by you, front and back. Write as densely as you like.

Topics covered: Everything on the homework, i.e. Regular languages, DFAs, pumping lemma, CFGs, TMs, decidability, undecidability, Turing-recognizability, P, NP, NP-completeness, PSPACE-completeness, Probabilistic TMs and approximation algs. There might be some simple True/False questions on the other material (interactive / zero-knowledge proofs).