Interactive Proof

Xin Li

lixints@cs.jhu.edu
Topics

- Standard Proof
- Interactive Proof
- Different models of IP
- Basic properties
- Examples
Proofs

- Standard proof: certificate based definition of NP.
- Prover: provides a proof written on a paper.
- Verifier: checks the proof is valid.
- Is this the only way of doing proofs?
Interactive Proofs

• Imagine you are at a court/thesis defense…

• A series of questions/answers alternate.

• The proof becomes interactive.

• Verifier asks questions while prover provides answers, until the verifier is convinced.
Interactive Proofs

• Interactive proofs are a generalization of NP.

• Question: what is the power of succinct interactive proofs?

• Example: can we provide a short interactive proof for $\psi \in \neg \text{SAT (coNP-complete)}$?

• How about $\psi \in \text{TQBF (PSPACE-complete)}$?
Interactive Proofs

• Try to formalize interactive proofs.

• Verifier: conducts an interrogation by repeatedly asking questions, decides in the end.

• Prover: provide responses.

• The message of each party can depend on all previous messages.
Example

A red pie and a green pie

Bob is color blinded so he thinks the two pies are the same.

Alice tries to convince Bob that the two pies are different, how?
Example

Bob does the following: switch the order of the two pies arbitrarily without letting Alice see the switch. Then ask Alice if he made a switch.

After Bob finishes, Alice answers Bob’s question.
Example

A red pie and a green pie

After say 100 times of this experiment, if Alice always answers correctly, then Bob should be pretty convinced that the two pies are different.

If the two pies are indeed identical, then it’s extremely unlikely Alice can answer correctly in all 100 experiments.
A closer look at the example

• Alice is more powerful than Bob.

• Alice can convince Bob the two pies are different without telling him what’s the difference (zero-knowledge proof).

• What’s the key ingredient used here?

• Bob’s “arbitrary” choice of switching without Alice knowing it.
Interactive Proofs

• Generally, assume the prover to be all powerful (can solve anything).

• Prover: deterministic or probabilistic?

• Verifier: deterministic or probabilistic?

• If allow probabilistic prover/verifier, how to define “accept” and “reject”? 
Deterministic Interactive Proofs (DIP)

• Both prover and verifier are deterministic.

• Example: DIP for $\psi \in \text{SAT}$.

• Verifier: asks prover to announce the values for variables clause by clause, checks each clause and the consistency of all variables.

• However, what’s the point of interaction? The prover can send all values at once.
Deterministic Interactive Proofs (DIP)

• Definition: Let $f, g : \{0,1\}^* \rightarrow \{0,1\}^*$ be functions and $k$ be an integer (can depend on input length).

• A $k$-round interaction of $f$ and $g$ on input $x \in \{0,1\}^*$, denoted by $<f, g>(x)$ is the sequence of strings $a_1, a_2, \ldots, a_k \in \{0,1\}^*$ defined as follows.

  $a_1 = f(x), a_2 = g(x, a_1), \ldots, a_{2i+1} = f(x, a_1, \ldots, a_{2i})$ for $2i < k$ and $a_{2i+2} = g(x, a_1, \ldots, a_{2i+1})$ for $2i + 1 < k$

• The output of $f$ in the end, denoted $\text{Out}_f <f, g>(x)$ is defined as $f(x, a_1, \ldots, a_k)$ and assume in $\{0,1\}$. 
**Deterministic Interactive Proofs (DIP)**

- **Definition (DIP):** Say a language $L$ has a $k$-round deterministic interactive proof system if $\exists$ a deterministic TM $V$ that on input $x$, $a_1, \ldots, a_i$ runs in time $\text{poly}(|X|)$, and can have a $k$-round interaction with any function $P$ s.t.

  - $x \in L \Rightarrow \exists P: \{0,1\}^* \rightarrow \{0,1\}^*$, $\text{Out}_V <V, P>(x)=1$ (completeness)
  - $x \notin L \Rightarrow \forall P: \{0,1\}^* \rightarrow \{0,1\}^*$, $\text{Out}_V <V, P>(x)=0$ (soundness)

- **The class DIP contains all languages with a $k(n)$ round deterministic interactive proof system where $k(n)$ is polynomial in $n$.**
Deterministic Interactive Proofs (DIP)

• Definition (DIP): Say a language \( L \) has a \( k \)-round deterministic interactive proof system if \( \exists \) a deterministic TM \( V \) that on input \( x, a_1, \ldots, a_i \) runs in time \( \text{poly}(|X|) \), and can have a \( k \)-round interaction with any function \( P \) s.t.

\[
\begin{align*}
  x \in L &\Rightarrow \exists P: \{0,1\}^* \rightarrow \{0,1\}^*, \text{Out}_V <V, P>(x)=1 \quad \text{(completeness)} \\
  x \notin L &\Rightarrow \forall P: \{0,1\}^* \rightarrow \{0,1\}^*, \text{Out}_V <V, P>(x)=0 \quad \text{(soundness)}
\end{align*}
\]

• Note: no limit on the power of \( P \).

• Completeness: if \( x \in L \) then it can be proved. Soundness: if \( x \notin L \) then no proof should exist, no matter how powerful the prover.
Deterministic Interactive Proofs (DIP)

- Lemma: DIP=NP.

- Trivially, any NP language has a one-round deterministic proof system.

- Show $L \in DIP \Rightarrow L \in NP$.

- If $V$ is the verifier for $x \in L$ in the deterministic proof system, then a certificate for $x \in L$ is just a transcript $(a_1, \ldots, a_k)$ that causes $V$ to accept.
Deterministic Interactive Proofs (DIP)

- Lemma: DIP=NP.

- Trivially, any NP language has a one-round deterministic proof system.

- Show $L \in DIP \implies L \in NP$.

- If $V$ is the verifier for $x \in L$ in the deterministic proof system, then a certificate for $x \in L$ is just a transcript $(a_1, \ldots, a_k)$ that causes $V$ to accept.
Deterministic Interactive Proofs (DIP)

• If $V$ is the verifier for $x \in L$ in the deterministic proof system, then a certificate for $x \in L$ is just a transcript $(a_1, \ldots, a_k)$ that causes $V$ to accept.

• The verifier of NP checks the following: $V(x) = a_1$, $V(x, a_1, a_2) = a_3$, $\ldots$, and $V(x, a_1, \ldots, a_k) = 1$.

• If $x \in L$ then such a transcript exists.

• If $x \notin L$ the no such transcript can exist (otherwise a prover $P$ exists).
IP: Probabilistic Verifier

• To make the class IP more interesting, need to change prover/verifier.

• Changing prover to be probabilistic does not change the power of IP.

• Since the prover is already all powerful.

• Changing the verifier to be probabilistic.
IP: Probabilistic Verifier

- What does it mean to change the verifier to be probabilistic?

- The verifier’s questions will be computed using a probabilistic algorithm.

- Verifier will be allowed to come to a wrong conclusion with some small probability.

- The probability is over the verifier’s random bits used.
**IP: Probabilistic Verifier**

- **To model the probabilistic verifier**, add **an m-bit random string** $r$ as the input to the function $f$ (for verifier).

- However, the function $g$ (for prover) does not see $r$ (private coin).

- The transcript is now a random variable (because of $r$).

- The output $\text{Out}_V \langle V, P \rangle(x)$ is also a random variable.
Basic Properties of IP

• Definition (IP): For an integer $k \geq 1$ (may depend on the input length), say a language $L$ is in IP[$k$] if $\exists$ a probabilistic poly-time TM $V$ that can have a $k$-round interaction with any function $P: \{0,1\}^* \rightarrow \{0,1\}^*$ s.t.

  $x \in L \Rightarrow \exists P, \Pr[\text{Out}_V \prec V, P>(x)=1] \geq 2/3$ (completeness)
  $x \not\in L \Rightarrow \forall P, \Pr[\text{Out}_V \prec V, P>(x)=1] \leq 1/3$ (soundness)

• Define $IP = \bigcup_{c \geq 0} IP(n^c)$