Lecture 17: Interactive Proofs

Arijit Bishnu

23.04.2010

Outline

Outline

A Relook at NP

Certificate definition of NP

A language $L \subseteq \{0,1\}^*$ is in NP if $\exists$ a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a poly-time TM M s.t. $\forall x \in \{0,1\}^*$,

$$x \in L \iff \exists u \in \{0,1\}^{p(|x|)} \text{ such that } M(x,u) = 1$$

A Relook at NP

Certificate definition of NP

A language $L \subseteq \{0,1\}^*$ is in NP if $\exists$ a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a poly-time TM M s.t. $\forall x \in \{0,1\}^*$,

$$x \in L \iff \exists u \in \{0,1\}^{p(|x|)} \text{ such that } M(x,u) = 1$$

Breaking up the certificate definition of NP

A language $L \subseteq \{0,1\}^*$ is in NP if $\exists$ a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a poly-time TM M s.t. $\forall x \in \{0,1\}^*$,

(Completeness) $x \in L \Rightarrow \exists u \in \{0,1\}^{p(|x|)}$ such that $M(x,u) = 1$

(Soundness) $x \notin L \Rightarrow \forall u \in \{0,1\}^{p(|x|)}$ such that $M(x,u) = 0$

A Relook at NP

Another Interpretation

A Relook at NP

Another Interpretation

- A prover (the boss! and boss can be wrong!) has given a short certificate in one shot which the verifier checks for correctness.

A Relook at NP

Another Interpretation

- A prover (the boss! and boss can be wrong!) has given a short certificate in one shot which the verifier checks for correctness.
- Now, we give the verifier a little bit more power, it can repeatedly interrogate the prover and listen to the prover's responses and then finally decide.

An Example

3SAT

- Let us look at 3SAT under this interactive prover-verifier model.

An Example

3SAT

- Let us look at 3SAT under this interactive prover-verifier model.
- The verifier proceeds clause by clause and asks the prover for the values of the literals.

An Example

3SAT

- Let us look at 3SAT under this interactive prover-verifier model.
- The verifier proceeds clause by clause and asks the prover for the values of the literals.
- The verifier keeps track of all these values and checks if under no conflicting assignment, all the clauses turned out to be true.

An Example

3SAT

- Let us look at 3SAT under this interactive prover-verifier model.
- The verifier proceeds clause by clause and asks the prover for the values of the literals.
- The verifier keeps track of all these values and checks if under no conflicting assignment, all the clauses turned out to be true.
- The prover could have given the entire information at one go as both the actions of the prover and verifier are deterministic.

An Example

3SAT

- Let us look at 3SAT under this interactive prover-verifier model.
- The verifier proceeds clause by clause and asks the prover for the values of the literals.
- The verifier keeps track of all these values and checks if under no conflicting assignment, all the clauses turned out to be true.
- The prover could have given the entire information at one go as both the actions of the prover and verifier are deterministic.
- What if the prover and verifier are probabilistic?

Some Definitions

- By interaction, we mean that the verifier and the prover are two deterministic functions.

Some Definitions

- By interaction, we mean that the verifier and the prover are two deterministic functions.
- At each round of interaction, these functions compute the next question or response as a function of the input and the questions and responses of the previous rounds.

Definition

Definition: Interaction of Deterministic Functions

Let $f, g : \{0,1\}^* \to \{0,1\}^*$ be functions and $k \geq 0$ be an integer allowed to depend on the input size. A k-round interaction of f and g on the input $x \in \{0,1\}^*$, denoted by $\langle f, g \rangle(x)$ is the sequence of strings $a_1, \ldots, a_k \in \{0,1\}^*$ defined as follows:

$$\begin{aligned} a_1 &= f(x) \\ a_2 &= g(x, a_1) \\ &\ldots \\ a_{2i+1} &= f(x, a_1, \ldots, a_{2i}) \quad \text{for } 2i < k \\ a_{2i+2} &= g(x, a_1, \ldots, a_{2i+1}) \text{ for } 2i+1 < k \end{aligned}$$

The output of f at the end of the interaction denoted $\mathcal{O}_f\langle f, g \rangle(x)$ is defined to be $f(x, a_1, \ldots, a_k) \in \{0,1\}$.

Another Definition of a Class

Definition: Deterministic Proof Systems

We say that a language L has a k-round deterministic interactive proof system if there is a deterministic TM V that on input $x, a_1, \ldots, a_i$ runs in time polynomial in $|x|$, and can have a k-round interaction with any function P such that

$$\text{(Completeness)} \quad x \in L \Rightarrow \exists P : \{0,1\}^* \rightarrow \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1$$

$$\text{(Soundness)} \quad x \notin L \Rightarrow \forall P : \{0,1\}^* \rightarrow \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 0$$

The class dIP contains all languages with a $k(n)$-round deterministic interactive proof system where $k(n) = \text{poly}(n)$.

A Relation Among Classes

Theorem

dIP = NP.

A Relation Among Classes

Theorem

dIP = NP.

A relook at the definition of dIP

We say that a language L has a k-round deterministic interactive proof system if there is a deterministic TM V that on input $x, a_1, \ldots, a_i$ runs in time polynomial in $|x|$, and can have a k-round interaction with any function P such that

$$\begin{aligned}
\text{(Completeness)} \quad & x \in L \Rightarrow \exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1 \\
\text{(Soundness)} \quad & x \notin L \Rightarrow \forall P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 0 \\
\text{(Contrapositive of Soundness)} \quad & \exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1 \Rightarrow x \in L
\end{aligned}$$

The class dIP contains all languages with a $k(n)$-round deterministic interactive proof system where $k(n) = \text{poly}(n)$.

The Proof of dIP = NP

Proof

The Proof of dIP = NP

Proof

- NP $\subseteq$ dIP is trivial.

The Proof of dIP = NP

Proof

- NP $\subseteq$ dIP is trivial.
- To prove dIP $\subseteq$ NP, fix a $L \in$ dIP and show that $L \in$ NP.

The Proof of dIP = NP

Proof

- NP $\subseteq$ dIP is trivial.
- To prove dIP $\subseteq$ NP, fix a $L \in$ dIP and show that $L \in$ NP.
- As $L \in$ dIP, there exists a deterministic TM V that acts as a verifier for L. A certificate that $x \in L$ is a transcript $(a_1, a_2, \ldots, a_k)$ that makes V accept x.

The Proof of dIP = NP

Proof

- NP $\subseteq$ dIP is trivial.
- To prove dIP $\subseteq$ NP, fix a $L \in$ dIP and show that $L \in$ NP.
- As $L \in$ dIP, there exists a deterministic TM V that acts as a verifier for L. A certificate that $x \in L$ is a transcript $(a_1, a_2, \ldots, a_k)$ that makes V accept x.
- To verify the transcript, V can be used by the machine of NP to make successive checks $V(x) = a_1$, $V(x, a_1, a_2) = a_3$, ... and the number of checks is $O(k)$ which is polynomial in the input size. If $x \in L$, such a transcript always exists. This ensures the short certificate for NP.

The Proof of dIP = NP

Proof

- NP $\subseteq$ dIP is trivial.
- To prove dIP $\subseteq$ NP, fix a $L \in$ dIP and show that $L \in$ NP.
- As $L \in$ dIP, there exists a deterministic TM V that acts as a verifier for L. A certificate that $x \in L$ is a transcript $(a_1, a_2, \ldots, a_k)$ that makes V accept x.
- To verify the transcript, V can be used by the machine of NP to make successive checks $V(x) = a_1$, $V(x, a_1, a_2) = a_3$, ... and the number of checks is $O(k)$ which is polynomial in the input size. If $x \in L$, such a transcript always exists. This ensures the short certificate for NP.
- This proves the completeness condition of the class NP discussed earlier.

The Proof of dIP = NP

Proof

- NP $\subseteq$ dIP is trivial.
- To prove dIP $\subseteq$ NP, fix a $L \in$ dIP and show that $L \in$ NP.
- As $L \in$ dIP, there exists a deterministic TM V that acts as a verifier for L. A certificate that $x \in L$ is a transcript $(a_1, a_2, \ldots, a_k)$ that makes V accept x.
- To verify the transcript, V can be used by the machine of NP to make successive checks $V(x) = a_1$, $V(x, a_1, a_2) = a_3$, ... and the number of checks is $O(k)$ which is polynomial in the input size. If $x \in L$, such a transcript always exists. This ensures the short certificate for NP.
- This proves the completeness condition of the class NP discussed earlier.
- Next we use the contrapositive of the soundness condition of dIP to show the contrapositive of the soundness condition of

The Proof of dIP = NP

Proof

The Proof of dIP = NP

Proof

- The contrapositive of the soundness condition of dIP says that $\exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V \langle V, P \rangle(x) = 1 \Rightarrow x \in L$.

The Proof of dIP = NP

Proof

- The contrapositive of the soundness condition of dIP says that $\exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1 \Rightarrow x \in L$.
- Consider the transcript generated by $\mathcal{O}_V\langle V, P\rangle(x) = 1$.

The Proof of dIP = NP

Proof

- The contrapositive of the soundness condition of dIP says that $\exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1 \Rightarrow x \in L$.
- Consider the transcript generated by $\mathcal{O}_V\langle V, P\rangle(x) = 1$.
- If such a transcript exists, define a prover function P such that $P(x, a_1) = a_2$, $P(x, a_1, a_2, a_3) = a_4$, ... This prover function is polynomial in the input size.

The Proof of dIP = NP

Proof

- The contrapositive of the soundness condition of dIP says that $\exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1 \Rightarrow x \in L$.
- Consider the transcript generated by $\mathcal{O}_V\langle V, P\rangle(x) = 1$.
- If such a transcript exists, define a prover function P such that $P(x, a_1) = a_2$, $P(x, a_1, a_2, a_3) = a_4$, ... This prover function is polynomial in the input size.
- This prover function satisfies $\mathcal{O}_V\langle V, P\rangle(x) = 1$, which implies $x \in L$.

The Proof of dIP = NP

Proof

- The contrapositive of the soundness condition of dIP says that $\exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1 \Rightarrow x \in L$.
- Consider the transcript generated by $\mathcal{O}_V\langle V, P\rangle(x) = 1$.
- If such a transcript exists, define a prover function P such that $P(x, a_1) = a_2$, $P(x, a_1, a_2, a_3) = a_4$, ... This prover function is polynomial in the input size.
- This prover function satisfies $\mathcal{O}_V\langle V, P\rangle(x) = 1$, which implies $x \in L$.
- Thus the contrapositive of the soundness condition of NP is proved.

The Proof of dIP = NP

Proof

- The contrapositive of the soundness condition of dIP says that $\exists P : \{0,1\}^* \to \{0,1\}^* \mathcal{O}_V\langle V, P\rangle(x) = 1 \Rightarrow x \in L$.
- Consider the transcript generated by $\mathcal{O}_V\langle V, P\rangle(x) = 1$.
- If such a transcript exists, define a prover function P such that $P(x, a_1) = a_2$, $P(x, a_1, a_2, a_3) = a_4$, ... This prover function is polynomial in the input size.
- This prover function satisfies $\mathcal{O}_V\langle V, P\rangle(x) = 1$, which implies $x \in L$.
- Thus the contrapositive of the soundness condition of NP is proved.

- We did not gain much by making the verifier deterministic.

Outline

Probabilistic Verifier and a New Class

- What about making the verifier probabilistic, the probability is over the choice of the verifier's coin?

Probabilistic Verifier and a New Class

- What about making the verifier probabilistic, the probability is over the choice of the verifier's coin?
- The verifier is polynomial time and we allow the prover to be of high computing power.

Probabilistic Verifier and a New Class

- What about making the verifier probabilistic, the probability is over the choice of the verifier's coin?
- The verifier is polynomial time and we allow the prover to be of high computing power.

Example

Let us consider a two player game.

Definition

Definition: Interaction of Probabilistic Functions

Let $f, g : \{0,1\}^* \to \{0,1\}^*$ be functions, where f is probabilistic taking an additional m-bit input r, and $k \geq 0$ be an integer allowed to depend on the input size. A k-round interaction of f and g on the input $x \in \{0,1\}^*$, denoted by $\langle f, g \rangle(x)$ is the sequence of strings $a_1, \ldots, a_k \in \{0,1\}^*$ defined as follows:

$$\begin{aligned} a_1 &= f(x, r) \\ a_2 &= g(x, a_1) \\ a_3 &= f(x, r, a_1, a_2) \\ &\ldots \\ a_{2i+1} &= f(x, r, a_1, \ldots, a_{2i}) \quad \text{for } 2i < k \\ a_{2i+2} &= g(x, a_1, \ldots, a_{2i+1}) \text{ for } 2i + 1 < k \end{aligned}$$

The output is defined as usual.
The interaction $\langle f, g \rangle(x)$ is a random variable over $r \in_R \{0,1\}^m$. The output $\mathcal{O}_f \langle f, g \rangle(x)$ is also a random variable.

Another Class

Definition: Probabilistic Verifiers and the class IP

For an integer $k \geq 1$, we say that a language L is in $\mathsf{IP}[k]$ if there is a probabilistic poly-time TM V that can have a k-round interaction with a function $P : \{0,1\}^* \to \{0,1\}^*$ such that

$$\text{(Completeness)} \quad x \in L \Rightarrow \exists P \Pr[\mathcal{O}_V\langle V, P\rangle(x) = 1] \geq 2/3$$
$$\text{(Soundness)} \quad x \notin L \Rightarrow \forall P \Pr[\mathcal{O}_V\langle V, P\rangle(x) = 1] \leq 1/3$$

where all probabilities are over choices of r.
We define $\mathsf{IP} = \bigcup_{c \geq 1} \mathsf{IP}[n^c]$.

Lemma

The class IP is unchanged if we replace the completeness parameter $2/3$ by $1 - 2^{-n^s}$ and the soundness parameter $1/3$ by 2^{-n^s} for any fixed constant $s > 0$.

Lemma

The class IP is unchanged if we replace the completeness parameter $2/3$ by $1 - 2^{-n^s}$ and the soundness parameter $1/3$ by 2^{-n^s} for any fixed constant $s > 0$.

Proof

Omitted.

Graph (Non)Isomorphism

Graph Isomorphism (GI)

Given two labeled graphs G_1 and G_2, we want to find whether there exists a permutation π of the labels of the nodes of G_1 such that $\pi(G_1) = G_2$, where $\pi(G)$ is the labeled graph obtained by applying π on the labels of its vertices.

Graph (Non)Isomorphism

Graph Isomorphism (GI)

Given two labeled graphs G_1 and G_2, we want to find whether there exists a permutation π of the labels of the nodes of G_1 such that $\pi(G_1) = G_2$, where $\pi(G)$ is the labeled graph obtained by applying π on the labels of its vertices.

- GI $\in$ NP $\subseteq$ IP and one can prove the following.

Graph (Non)Isomorphism

Graph Isomorphism (GI)

Given two labeled graphs G_1 and G_2, we want to find whether there exists a permutation π of the labels of the nodes of G_1 such that $\pi(G_1) = G_2$, where $\pi(G)$ is the labeled graph obtained by applying π on the labels of its vertices.

- GI $\in$ NP $\subseteq$ IP and one can prove the following.

Theorem

If GI $\in$ NP-complete, then the polynomial hierarchy collapses, i.e. $\Sigma_2^p = \Pi_2^p$.

Graph Non-Isomorphism

Graph Non-Isomorphism

Graph Nonisomorphism (GNI) is the complement problem of GI.

Graph Non-Isomorphism

Graph Non-Isomorphism

Graph Nonisomorphism (GNI) is the complement problem of GI.

An interactive proof protocol for GNI

Graph Non-Isomorphism

Graph Non-Isomorphism

Graph Nonisomorphism (GNI) is the complement problem of GI.

An interactive proof protocol for GNI

V: Choose $i \in \{1, 2\}$ randomly. Randomly permute the vertices of G_i to get a new graph G'. Send G' to P.

Graph Non-Isomorphism

Graph Non-Isomorphism

Graph Nonisomorphism (GNI) is the complement problem of GI.

An interactive proof protocol for GNI

V: Choose $i \in \{1, 2\}$ randomly. Randomly permute the vertices of G_i to get a new graph G'. Send G' to P.

P: Identify $j \in \{1, 2\}$ such that G_j was used to produce G'. Send j to V.

Graph Non-Isomorphism

Graph Non-Isomorphism

Graph Nonisomorphism (GNI) is the complement problem of GI.

An interactive proof protocol for GNI

V: Choose $i \in \{1, 2\}$ randomly. Randomly permute the vertices of G_i to get a new graph G'. Send G' to P.

P: Identify $j \in \{1, 2\}$ such that G_j was used to produce G'. Send j to V.

V: Accept if $i = j$; reject otherwise.

A Proof Sketch that the protocol satisfies IP

- If $G_1 \ncong G_2$, then $\exists$ a prover s.t. $\Pr[V \text{ accepts}] = 1$.

A Proof Sketch that the protocol satisfies IP

- If $G_1 \ncong G_2$, then $\exists$ a prover s.t. $\Pr[V \text{ accepts}] = 1$.
- The all powerful prover can certainly check which of G_1 or G_2 is isomorphic to G' as $G_1 \ncong G_2$.

A Proof Sketch that the protocol satisfies IP

- If $G_1 \ncong G_2$, then $\exists$ a prover s.t. Pr[V accepts] $= 1$.
- The all powerful prover can certainly check which of G_1 or G_2 is isomorphic to G' as $G_1 \ncong G_2$.
- If $G_1 \cong G_2$, the prover can only randomly guess which one of G_1 or G_2 is isomorphic to G', so for every prover Pr[V accepts] $\leq 1/2$. One can reduce the probability my repetitions.

IP and PSPACE

Theorem

IP = PSPACE.

IP and PSPACE

Theorem

IP = PSPACE.

Proof

Omitted.