Foundation Lab 19.4.2023

Recall power of NP - 1

Informal : NP is the class of problems that have poly time verifiers.

Formal:

Definition : if language $L \in NP$, then $\forall x \in L$, if $\exists \omega$ then $V(x,\omega) = 1$ and $|V| = poly(|x|)$

Recall power of NP - 2

Informal : NP is the class of problems that have poly time verifiers.

Formal:
Definition : if language L $\in$ NP, then $\forall x \in L$, if $\exists\, \omega$ then $V(x,\omega) = 1$ and $|V| = poly(|x|)$

Informal : A language L is poly-time verifiable iff it is decided by a poly-time NTM.

Formal: Consider an NP language L

(i) if $x \in L$, if $\exists\, \omega$ s.t. $V(x,\omega) = 1$ and $|V| = poly(|x|)$

(ii) if $x \,!\!\in L$, if $!\exists\, \omega$ s.t. $V(x,\omega) = 1$ and $|V| = poly(|x|)$

Recall power of NP - 3

Informal : NP is the class of problems that have poly time verifiers.

Formal:
Definition : if language L $\in$ NP, then $\forall x \in L$, if $\exists \omega$ then $V(x,\omega) = 1$ and $|V| = \text{poly}(|x|)$

Informal : A language L is poly-time verifiable iff it is decided by a poly-time NTM.

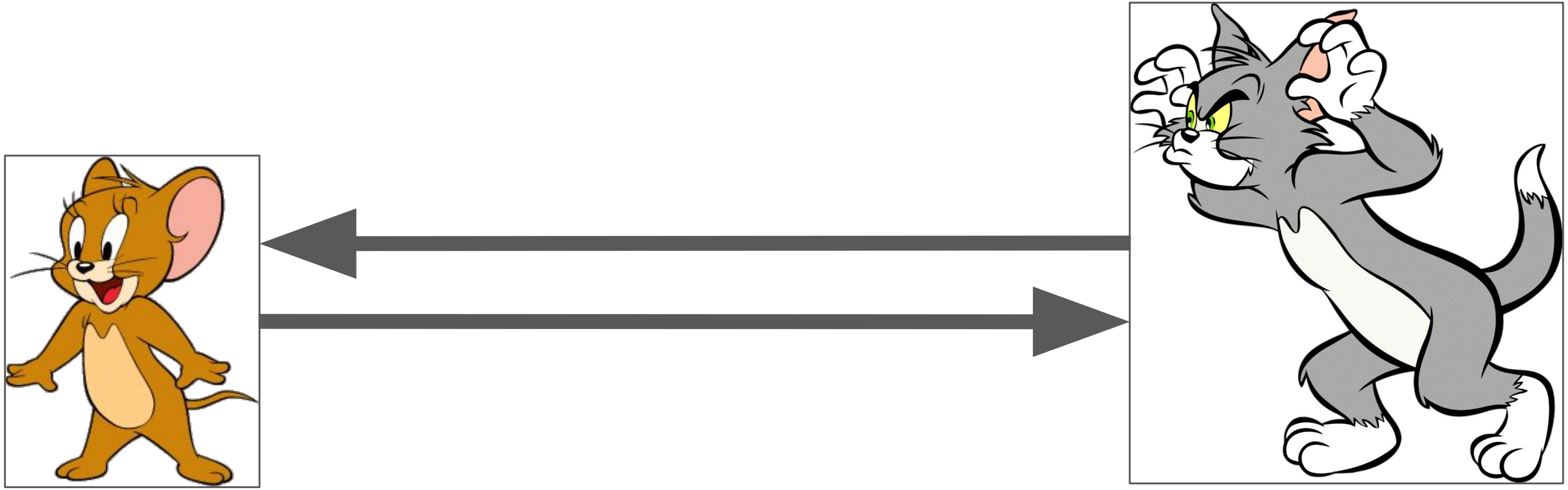
Formal: Consider an NP language L
(i) if $x \in L$, if $\exists \omega$ s.t. $V(x,\omega) = 1$ and $|V| = \text{poly}(|x|)$

(ii) if $x \,!\!\in L$, if $!\exists\ \omega$ s.t. $V(x,\omega) = 1$ and $|V| = \text{poly}(|x|)$

Example Problems : Clique, SubsetSum, SAT, 3SAT, HamPath.

Beyond NP : backdrop

Membership problem of $x \in L$



Efficient
Verifier

"Verifier" is making sure that
the prover is not cheating

All Powerful Prover

"Prover" is trying to prove that
$x \in L$

Deterministic Interactive Proof -1

$V(x) = a_1$

$a_2 = P(x, a_1)$

$V(x, a_1, a_2) = a_3$

$\vdots \qquad\qquad\qquad \vdots$

A general Proof Protocol

Interactive Proofs - 2

This usual notion proof can be generalized in the following sense:

- This usual ”non-interactive” framework can be seen as an interaction between the ”Prover(P)” and the ”Verifier(V)” where P sends a message to V, such that V performs some computation on the message, and finally returns true or false
- This notion can be naturally generalized into an interactive framework, where the proof can be considered as many rounds of interaction between P and V

Interactive Proofs - 3

To formalize interactive proofs, we need to model the P and V

Interaction of deterministic functions: Let P, V : $\{0, 1\}^* \rightarrow \{0, 1\}^*$ be functions. A k-round interaction of P, V , on input $x \in \{0,1\}^*$. denoted by $< P,V > (x)$, is sequence of following strings $a_1,a_2,\ldots,a_k \in \{0,1\}^*$ defined as follows:

$$a_1 = V(x)$$

$$a_2 = P(x,a_1)$$

$$..$$

$$a_{2i+1} = V(x,a_1,\ldots,a_{2i})$$

$$a_{2i+2} = P(x,a_1,\ldots,a_{2i+1})$$

The output of V (or P) at the end of the interaction denoted by $out_V < P,V > (x)$ is defined to be $V(x,a_1,a_2,\ldots,a_k)$

Interactive Proofs - 4

To formalize interactive proofs, we need to model the P and V

Interaction of deterministic functions: Let P, V : $\{0, 1\}^* \rightarrow \{0, 1\}^*$ be functions. A k-round interaction of P, V , on input $x \in \{0,1\}^*$. denoted by $< P,V > (x)$, is sequence of following strings $a_1,a_2,...,a_k \in \{0,1\}^*$ defined as follows:

$$a_1 = V(x)$$

$$a_2 = P(x,a_1)$$

..

$$a_{2i+1} = V(x,a_1,...,a_{2i})$$

$$a_{2i+2} = P(x,a_1,...,a_{2i+1})$$

The output of V (or P) at the end of the interaction denoted by $out_V < P,V > (x)$ is defined to be $V(x,a_1,a_2,...,a_k)$

Formalising Deterministic Interactive Proofs - 4

Deterministic proof system

We say that a k-round deterministic interactive proof system, if there is a deterministic TM V, that on input x, $a_1, a_2, \ldots, a_k$ runs in time polynomial in |x|,satisfying:

- (Completeness) $x \in L \Rightarrow \exists P : \{0, 1\}^* \to \{0, 1\}^*$ such that out_V $<V,P>(x)=1$
- (Soundness) $x \notin L \Longrightarrow \forall P : \{0,1\}^* \to \{0,1\}^*$ such that $out_V < V,P > (x) = 0$

We define class dIP, that contains all the languages with a k(n)-round deterministic interactive proof system, where k is a polynomial.

It turns out **dIP=NP**

Exercise 1:

Consider a language L is decided by a deterministic interactive proof.
Prove that $L \in NP$

The Class IP

- We saw that increasing the number of rounds of interaction from 1 to polynomially many, did not increase the power of our proof system.
- To realize full potential interaction, we need to let verifier be probabilistic.
- Verifier is allowed to have its own randomness

Formalising IP

Definition

Let $k : N \rightarrow N$ be some function with $k(n)$ computable in poly(n) time.
A language L is in IP[k], if there is a Turing machine V, such that on inputs $x, a_1, a_2, ..., a_i,$, V runs in polynomial time in $|x|$ and such that

- (Completeness) $x \in L \Rightarrow \exists P$ s.t. $\Pr[out_V \langle V,P \rangle (x) = 1] \geq 2/3$
- (soundness) $x\ ! \in L \Rightarrow \forall P\ \ \Pr[out_V \langle V,P \rangle (x)=1] \leq 1/3$

Formalising IP

Definition

Let $k : N \rightarrow N$ be some function with $k(n)$ computable in poly(n) time.
A language L is in IP[k], if there is a Turing machine V, such that on inputs $x, a_1, a_2, \ldots, a_i,$, V runs in polynomial time in $|x|$ and such that

- (Completeness) $x \in L \Rightarrow \exists P$ s.t. $\Pr[out_V \langle V,P \rangle(x) = 1] \geq 2/3$
- (soundness) $x \,!\!\in L \Rightarrow \forall P \; \Pr[out_V \langle V,P \rangle(x)=1] \leq 1/3$

Few examples: GNI, GI, QNR, QR, 3-Col, HM-Cycle

Outline

Another Example – Polynomial Identity Testing

Another Example – Polynomial Identity Testing

Polynomial

A polynomial is an equation in one-variable

$$\begin{aligned} f(x) &= x^3 - 6x^2 + 11x - 7 = \\ &\quad (x-1)(x-2)(x-3) \end{aligned}$$

Another Example – Polynomial Identity Testing

Polynomial

A polynomial is an equation in one-variable

$$\begin{aligned} f(x) &= x^3 - 6x^2 + 11x - 7 = \\ &\quad (x-1)(x-2)(x-3) \end{aligned}$$

Properties:

- A root of a polynomial f is a value x s.t. $f(x) = 0$

Another Example – Polynomial Identity Testing

Polynomial

A polynomial is an equation in one-variable

$$\begin{aligned} f(x) &= x^3 - 6x^2 + 11x - 7 = \\ &\quad (x-1)(x-2)(x-3) \end{aligned}$$

Properties:

- A root of a polynomial f is a value x s.t. $f(x) = 0$
- The degree of a polynomial $f(x)$ is the maximum exponent in f

Another Example – Polynomial Identity Testing

Polynomial

A polynomial is an equation in one-variable

$$\begin{aligned} f(x) &= x^3 - 6x^2 + 11x - 7 = \\ &\quad (x-1)(x-2)(x-3) \end{aligned}$$

Properties:

- A root of a polynomial f is a value x s.t. $f(x) = 0$
- The degree of a polynomial $f(x)$ is the maximum exponent in f
- A polynomial of degree d has at most d roots

Another Example – Polynomial Identity Testing

PIT Problem

Another Example – Polynomial Identity Testing

PIT Problem

- Prover P has a degree d polynomial f and wants to prove that

$$\forall x, f(x) = 0$$

Another Example – Polynomial Identity Testing

PIT Problem

- Prover P has a degree d polynomial f and wants to prove that

$$\forall x, f(x) = 0$$

- V is allowed to query $f(x)$ at points x of its choice – but, P knows V's strategy

Another Example – Polynomial Identity Testing

PIT Problem

- Prover P has a degree d polynomial f and wants to prove that

$$\forall x, f(x) = 0$$

- V is allowed to query $f(x)$ at points x of its choice – but, P knows V's strategy

Question: What should V do?

Another Example – Polynomial Identity Testing

PIT Problem

- Prover P has a degree d polynomial f and wants to prove that

$$\forall x, f(x) = 0$$

- V is allowed to query $f(x)$ at points x of its choice – but, P knows V's strategy

Question: What should V do?

- Suppose that V is deterministic:

Another Example – Polynomial Identity Testing

PIT Problem

- Prover P has a degree d polynomial f and wants to prove that

$$\forall x, f(x) = 0$$

- V is allowed to query $f(x)$ at points x of its choice – but, P knows V's strategy

Question: What should V do?

- Suppose that V is deterministic:
- What if you allow V to be randomized:

The Power of Randomness in Interactive Proofs

- By allowing V to be randomized we went from $d+1$ queries to 1 query

The Power of Randomness in Interactive Proofs

- By allowing V to be randomized we went from $d + 1$ queries to 1 query
- We have strong evidence that derandomizing PIT will be very hard – it implies strong complexity results that we have no idea how to prove

The Power of Randomness in Interactive Proofs

- By allowing V to be randomized we went from $d+1$ queries to 1 query
- We have strong evidence that derandomizing PIT will be very hard – it implies strong complexity results that we have no idea how to prove

Take away

Randomness and interaction are key to the power of $\mathcal{IP}$

The Power of Randomness in Interactive Proofs

- By allowing V to be randomized we went from $d+1$ queries to 1 query
- We have strong evidence that derandomizing PIT will be very hard – it implies strong complexity results that we have no idea how to prove

Take away

Randomness and interaction are key to the power of $\mathcal{IP}$

Thursday: We will prove that co-$\mathcal{NP} \subseteq \mathcal{IP}$