交互证明系统

We have seen interactive proofs, in various disguised forms, in the definitions of $\mathbf{NP}$, OTM, Cook reduction and $\mathbf{PH}$.

We will see that interactive proofs have fundamental connections to cryptography and approximate algorithms.

It was not until 1985 that the idea of computation through interaction was formally studied by two groups.

1. L. Babai, with a complexity theoretical motivation;
2. S. Goldwasser, S. Micali and C. Rackoff, with a cryptographic motivation.

An interactive proof system consists of a Prover and a Verifier.

1. Prover's goal is to convince Verifier of the validity of an assertion through dialogue.
2. Verifier's objective is to accept/reject the assertion based on the information it gathers from the dialogue.

Prover's answers are adaptive. Adaptivity is the difference between a prover and an oracle.

Synopsis

1. Introduction
2. Interactive Proof with Private Coins
3. Interactive Proof with Public Coins
4. Set Lower Bound Protocol
5. $\mathbf{IP} = \mathbf{PSPACE}$
6. Public Coins versus Private Coins
7. $\mathbf{MIP} = \mathbf{NEXP}$
8. Multilinearity Testing
9. $\mathbf{IP}$ vs $\mathbf{MIP}$
10. Parallel Repetition Theorem
11. Programme Checking

Introduction

Basic Principle

A verifier's job must be easy (polynomial time on input length), otherwise there is no need for any dialogue.

A prover can be as powerful as it takes, as long as the answers it produces are short (polynomial size on input length).

A verifier is not supposed to ask too many questions. Its best bet is ask random questions.

A prover is supposed to provide an answer no matter what.

Deterministic Verifier

A k-round interaction of f and g on input $x \in \{0,1\}^*$, denoted by $\langle f, g\rangle(x)$, is the sequence $a_1, \ldots, a_k \in \{0,1\}^*$ defined as follows:

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

The output of f at the end, noted $\mathrm{out}_f\langle f, g\rangle(x)$, is $f(x, a_1, \ldots, a_k) \in \{0,1\}$.

$f, g : \{0,1\}^* \to \{0,1\}^*$ are TM's, and $k(n)$ is a polynomial.

Deterministic Proof Systems

We say that a language L has a k-round deterministic proof system if there is a TM $\mathbb{V}$ that runs in $\mathrm{poly}(|x|)$ time, and can have a $k(|x|)$-round interaction with any TM $\mathbb{P}$ such that the following statements are valid:

$$\begin{aligned} \text{Completeness.} \quad & x \in L \Rightarrow \exists \mathbb{P} : \{0,1\}^* \to \{0,1\}^*.\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 1, \\ \text{Soundness.} \quad & x \notin L \Rightarrow \forall \mathbb{P} : \{0,1\}^* \to \{0,1\}^*.\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 0. \end{aligned}$$

Every **NP** language has a one-round deterministic proof system.

Suppose L has a k-round deterministic proof system. There is a P-time TM $\mathbb{V}$ such that

$$x \in L \text{ iff } \exists \mathbb{P} : \{0,1\}^* \to \{0,1\}^*.\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 1 \text{ iff}$$

$$\exists a_1, a_2, \ldots, a_k.\mathbb{V}(x) = a_1 \wedge \mathbb{V}(x, a_1, a_2) = a_3 \wedge \ldots \wedge \mathbb{V}(x, a_1, \ldots, a_k) = 1.$$

The verification time is polynomial.

Interaction + Randomness + Small Error

We shall only be interested in verifiers who ask clever questions.

"...in the context of interactive proof systems, asking random questions is as powerful as asking clever questions."

Goldreich

The Power of Randomness

B has one red sock and one green sock.

How can he convince A, who is color blind, that the socks are of different color?

Interactive Proof with Private Coins

S. Goldwasser, S. Micali, C. Rackoff. The Knowledge Complexity of Interactive Proofs. 1985.

  

Private Coins Model

The verifier generates an l-bits r by tossing coins:

$$r \in_{\mathrm{R}} \{0,1\}^l.$$

The verifier of course knows r:

$$a_1 = f(x,r),\ \ a_3 = f(x,r,a_1,a_2),\ \ \ldots.$$

The prover cannot see r:

$$a_2 = g(x,a_1),\ \ a_4 = g(x,a_1,a_2,a_3),\ \ \ldots.$$

Both the interaction $\langle f,g\rangle(x)$ and the output $\mathrm{out}_f\langle f,g\rangle(x)$ are random variables over $r \in_{\mathrm{R}} \{0,1\}^l$.

$\mathbf{IP}$, Interactive Proofs with Private Coins

Suppose k is a polynomial. A language L is in $\mathbf{IP}[k(n)]$ if there's a P-time PTM $\mathbb{V}$ that can have a $k(|x|)$-round interaction with any TM $\mathbb{P}$ and renders valid the following.

Completeness.

$$x \in L \Rightarrow \exists \mathbb{P} : \{0,1\}^* \to \{0,1\}^*.\Pr[\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 1] \geq 2/3.$$

Soundness.

$$x \notin L \Rightarrow \forall \mathbb{P} : \{0,1\}^* \to \{0,1\}^*.\Pr[\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 1] \leq 1/3.$$

The class $\mathbf{IP}$ is defined by $\bigcup_{c\geq 1} \mathbf{IP}[cn^c]$.

BPP Verifier, **PSPACE** Prover

1. A verifier is a $\mathbf{BPP}$ machine.
2. We may assume that a prover is a $\mathbf{PSPACE}$ machine.
   - There is an optimal prover.
   - A single $\mathbf{PSPACE}$ prover suffices for all $x \in L$.

An almighty prover knows Verifier's algorithm.

- Prover enumerates all answers $a_2, a_4, \ldots$, and uses Verifier's algorithm to calculate the percentage of the random strings that make verifier to accept.

$\mathbf{IP}$, Interactive Proofs with Private Coins

$L \in \mathbf{IP} \Leftrightarrow$ there is an interactive proof system of a verifier $\mathbb{V} \in \mathbf{BPP}$ and a prover $\mathbb{P} \in \mathbf{PSPACE}$ that interact for a polynomial round and renders valid the following.

Completeness.

$$x \in L \Rightarrow \Pr[\mathrm{out}_{\mathbb{V}}(\mathbb{V}, \mathbb{P}) = 1] \geq 2/3.$$

Soundness.

$$x \notin L \Rightarrow \Pr[\mathrm{out}_{\mathbb{V}}(\mathbb{V}, \mathbb{P}) = 1] \leq 1/3.$$

$\mathbf{IP} \subseteq \mathbf{PSPACE}$

Proposition. $\mathbf{IP} \subseteq \mathbf{PSPACE}$.

Both a $\mathbf{PSPACE}$ machine and a $\mathbf{BPP}$ machine can be simulated in polynomial space.

Robustness of $\mathbf{IP}$

Fact. $\mathbf{IP}$ remains unchanged if we replace the completeness parameter $2/3$ by $1-2^{-n^s}$ and soundness parameter $1/3$ by 2^{-n^s}.

Proof.
Repeat the protocol $O(n^s)$ times. Majority rule. Chernoff bound. $\square$

Since there is an optimal prover, it doesn't matter if a protocol is repeated sequentially or in parallel.

Robustness of $\mathbf{IP}$

Fact. Allowing prover to use a private coin does not change $\mathbf{IP}$.

By average principle we can construct from a probabilistic prover a deterministic prover that is as good as the former.

Perfect Completeness

An interactive proof system has perfect completeness if its completeness parameter is 1.

An interactive proof system has perfect soundness if its soundness parameter is 0.

Perfect Soundness is Very Strong

1. $\mathbf{IP}$ with Perfect Completeness $= \mathbf{IP}$.
2. $\mathbf{IP}$ with Perfect Soundness $= \mathbf{NP}$.

---

1. $\mathbf{IP} \subseteq \mathbf{PSPACE}$. A problem in $\mathbf{IP}$ is Karp reducible to $\mathtt{TQBF}$. $\mathtt{TQBF}$ has an interactive proof system with perfect completeness (using the Sumcheck protocol).

2. If $x \in L$, there exists a 'yes' certificate. If $x \notin L$, the verifier always says 'no'.

Graph Non-Isomorphism

Let $\mathtt{GI}$ be the Graph Isomorphism problem; it is not known to be in $\mathbf{P}$.

Let $\mathtt{GNI} = \overline{\mathtt{GI}}$, it is not known to be in $\mathbf{NP}$.

The nodes of a graph are represented by the numbers $1, 2, \ldots, n$.

The isomorphism of G_0 to G_1 is indicated by $\pi(G_0) = G_1$, where π is a permutation of the nodes of G_0.

Graph Non-Isomorphism Protocol

PROTOCOL: Graph Non-Isomorphism

V: Pick $i \in_{\mathrm{R}} \{0, 1\}$. Generate a random permutation graph H of G_i. Send H to P.

P: Identify which of G_0, G_1 was used to produce H and send the index $j \in \{0, 1\}$ to V.

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

Graph Non-Isomorphism

Theorem. $\mathtt{GNI} \in \mathbf{IP}$.

Proof.

If $G_0 \simeq G_1$, the prover's guess is as good as anyone's guess.

If $G_0 \not\simeq G_1$, the prover can force the verifier to accept. □

1. O. Goldreich, S. Micali, A. Wigderson. Proofs that Yield Nothing but Their Validity and a Methodology of Cryptographic Protocol Design. FOCS 1986.

Quadratic Non-Residuosity

A number a is a quadratic residue modulo p if there is some number b such that $a \equiv b^2 \pmod{p}$.

- $\mathtt{QR} = \{(a, p) \mid p \text{ is prime and } \exists b.a \equiv b^2 \pmod{p}\}$ is in $\mathbf{NP}$.

Let $\mathtt{QNR} = \overline{\mathtt{QR}}$. The problem $\mathtt{QNR}$ is not known to be in $\mathbf{NP}$.

Quadratic Non-Residuosity Protocol

Input.

1. An odd prime number p and a non-zero number a.

Goal.

1. The prover tries to convince the verifier that $a \in \mathtt{QNR}$.
2. The verifier should reject with good probability if $a \notin \mathtt{QNR}$.

V: Pick $r < p$ and $i \in \{0, 1\}$ randomly. If $i = 0$ then send r^2 *mod* p to P; otherwise send ar^2 *mod* p to P.

P: Identify which case it is and send a number $j \in \{0, 1\}$ to V accordingly.

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

Quadratic Non-Residuosity

Theorem. $\texttt{QNR} \in \mathbf{IP}$.

If a is a quadratic residue, then ar^2, like r^2, is a random quadratic residue modulo p. In this case prover can only guess.

If a is not a quadratic residue, then ar^2, unlike r^2, is a random non-quadratic residue modulo p. In this case prover can force verifier to accept.

The argument is with the multiplicative field $([p-1], \cdot)$.

---

1. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proofs. STOC 1985.

Interactive Proof for Permanent

Suppose $A = (a_{j,k})_{1 \leq j,k \leq n}$ is an $n \times n$ matrix. According to the expansion in cofactors,

$$\mathtt{perm}(A) = \sum_{i=1}^{n} a_{1i} \cdot \mathtt{perm}(A_{1,i}).$$

Computing the permanent of an $n \times n$ matrix reduces to computing the permanents of n matrices of dimension $(n-1) \times (n-1)$.

We design an interactive proof system for $\mathtt{perm}(A)$ using arithmetic method.

Interactive Proof for Permanent

We look for an $(n-1)\times(n-1)$-matrix $D_A(x)$ such that $D_A(i) = A_{1,i}$.

- $(D_A(x))_{j,k}$ is a univariate polynomial of degree $n-1$, and
- $\mathtt{perm}(D_A(x))$ is a univariate polynomial of degree $(n-1)^2$.

Vandermonde matrix is nonsingular. Verifier can calculate $(D_A(x))_{j,k}$ by solving the following.

$$\begin{pmatrix} 1 & 1 & \ldots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & k & \ldots & k^{n-2} & k^{n-1} \\ 1 & k+1 & \ldots & (k+1)^{n-2} & (k+1)^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & n & \vdots & n^{n-2} & n^{n-1} \end{pmatrix} \begin{pmatrix} b_0 \\ \vdots \\ b_k \\ b_{k+1} \\ \vdots \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} a_{(j+1)(k+1)} \\ \vdots \\ a_{(j+1)(k+1)} \\ a_{(j+1)k} \\ \vdots \\ a_{(j+1)k} \end{pmatrix}$$

Interactive Proof for Permanent

PROTOCOL: Permanent

Condition: Both parties know a number k and a matrix A.
Prover's goal is to show that $k = \mathtt{perm}(A)$.
Verifier should reject with good probability if $k \neq \mathtt{perm}(A)$.

P: Send to V a polynomial $g(x)$ of degree $(n-1)^2$, which is supposedly $\mathtt{perm}(D_A(x))$.

V: Check if $k = \sum_{i=1}^{n} a_{1i} \cdot g(i)$. If not, reject; otherwise pick up $b \in_{\mathrm{R}} \mathrm{GF}(p)$ and ask P to prove $g(b) = \mathtt{perm}(D_A(b))$.

One has to deal with an exponential number of monomials to calculate $g(x)$. However verifier can calculate the matrix $D_A(x)$.

Interactive Proof for Permanent

Let $L_{\texttt{perm}}$ be the language

$$\left\{\langle A, p, k\rangle \mid p > n^4,\ k = \texttt{perm}(A),\ A \text{ is an } n \times n \text{ matrix over } \mathrm{GF}(p)\right\}.$$

Theorem. $L_{\texttt{perm}} \in \mathbf{IP}$.

Proof.

If $n \leq 3$, use brutal force; otherwise use the permanent protocol.

Verifier accepts with probability 1 if $k = \texttt{perm}(A)$.

The error rate is bounded by $\frac{1}{3}$. [see next slide.] $\square$

Interactive Proof for Permanent

Suppose $k \neq \mathtt{perm}(A)$ and the prover sends a fake $g(x)$.

- $g(x) - \mathtt{perm}(D_A(x))$ has at most $(n-1)^2$ roots.
- The probability of choosing a b such that $g(b) = \mathtt{perm}(D_A(b))$ is $\leq \frac{(n-1)^2}{p}$.
- If $g(b) \neq \mathtt{perm}(D_A(b))$ and Prover's foul play has not been caught, he is left with the task to prove $g(b) = \mathtt{perm}(D_A(b))$.

If Prover manages to get away with all his previous foul plays, he gets caught in the end.

The probability of the verifier reaching a wrong answer is less than

$$\frac{(n-1)^2}{p} + \frac{(n-2)^2}{p} + \ldots + \frac{4^2}{p} < \frac{n^3}{p} < \frac{1}{n} < \frac{1}{3}.$$

Interactive Proof with Public Coins

“We can formulate a decision problem under uncertainty as a new sort of game, in which one opponent is ‘disinterested’ and plays at random, while the other tries to pick a strategy which maximizes the probability of winning – a ‘game against Nature’.”

1. Christos Papadimitriou. Games Against Nature. FOCS 1983.

László Babai. Trading Group Theory for Randomness. STOC 1985.

Interactive Proofs with Public Coins

In a public coins system, the verifier's message is identical to the outcome of the coins tossed at the current round.

- Whatever verifier computes, prover can do the same.
- Verifier's actions except for its final decision are oblivious of prover's messages.

Arthur-Merlin Game

Arthur-Merlin Game $=$ Interactive Proof with Public Coins

- Arthur/Nature is the verifier who tosses public coins, and
- Merlin is the prover.

Suppose $k : \mathbf{N} \to \mathbf{N}$ is a polynomial. Obviously

$$\mathbf{AM}[k(n)],\ \mathbf{MA}[k(n)] \subseteq \mathbf{IP}[k(n)].$$

Notational Convention

$$\mathbf{MA},\ \mathbf{AM},\ \mathbf{AMA},\ \mathbf{MAMAMA},\ \ldots$$

Switching Lemma. $\mathbf{MA} \subseteq \mathbf{AM}$.

Suppose $L \in \mathbf{MA}$. The completeness is not affected since

$$x \in \mathbf{L} \quad \Rightarrow \quad \exists a.\Pr_r[\mathbb{V}(x,a,r)=1] \geq 1-\epsilon \quad \Rightarrow \quad \Pr_r[\exists a.\mathbb{V}(x,a,r)=1] \geq 1-\epsilon.$$

Perfect Completeness would survive. Soundness is affected though.

$$x \notin \mathbf{L} \quad \Rightarrow \quad \forall a.\Pr_r[\mathbb{V}(x,a,r)=1] \leq \epsilon \quad \Rightarrow \quad \Pr_r[\exists a.\mathbb{V}(x,a,r)=1] \leq 2^{|a|}\epsilon.$$

Since a is of polynomial size, verifier can reduce the error rate by

- repeating the protocol for a polynomial number of time and
- applying majority rule after getting all the answers.

Collapse Theorem

Theorem (Babai, 1985). $\mathbf{AM}[k(n)-1] = \mathbf{MA}[k(n)] = \mathbf{AM}[k(n)]$ for $k(n) > 2$.

Both $\mathbf{AM}(k(n)-1) \subseteq \mathbf{AM}(k(n))$ and $\mathbf{AM}(k(n)-1) \subseteq \mathbf{MA}(k(n))$ are obvious.

Suppose $L \in \mathbf{AM}(k(n))$ has an interactive proof system that has a fragment of type $\mathbf{AMAMA}$.

Let x be the input, and m be the length of Merlin's answer.

Let $(a_1, b_1, a_2, b_2, a_3)$ be part of an interactive proof. We switch the 2nd and the 3rd actions.

$$(a_1 a_2^1 \ldots a_2^t, b_1' b_2^1 \ldots b_2^t, ia_3'),$$

followed by randomly selecting $i \in_{\mathrm{R}} [t]$ to continue. The number of round is reduced by 2.

Before switching, if Arthur sends a_1 to Merlin and Merlin responds with b, then the expected value of Arthur's decision is

$$A_x(b) \stackrel{\mathrm{def}}{=} \mathrm{E}_{a_2}[\mathbb{A}(x, \ldots, b, a_2, \ldots)].$$

The expected value of Arthur's decision after $a_1.b_1$ before switching is $A_x = A_x(b_1)$. Clearly $A_x \geq A_x(b)$ for all b.

After the switching, the expected value of Arthur's decision is

$$\mathrm{E}_{a_2^1,\ldots,a_2^t}\left[\max_{b_1' \in \{0,1\}^m}\left\{\mathrm{E}_{i \in_R [t]}[\mathbb{A}(x, \ldots, b_1', a_2^i, \ldots)]\right\}\right], \tag{1}$$

which by the uniform distribution is the same as

$$\mathrm{E}_{a_2^1,\ldots,a_2^t}\left[\max_{b_1' \in \{0,1\}^m}\left\{\frac{1}{t}\sum_{i=1}^{t}\mathbb{A}(x, \ldots, b_1', a_2^i, \ldots)\right\}\right].$$

After switching the probability that the expected value increases by at least δ is

$$\begin{aligned}
& \mathtt{Pr}_{a_2^1,\ldots,a_2^t}\left[\max_{b_1'\in\{0,1\}^m}\left\{\mathtt{E}_{i\in_{\mathtt{R}}[t]}[\mathbb{A}(x,\ldots,b_1',a_2^i,\ldots)]\right\}-A_x>\delta\right] \\
\leq\quad & \mathtt{Pr}_{a_2^1,\ldots,a_2^t}\left[\exists b_1'\in\{0,1\}^m.\left|\sum_{i=1}^t \mathbb{A}(x,\ldots,b_1',a_2^i,\ldots)-tA_x\right|>\frac{\delta}{A_x}(tA_x)\right] \\
\leq\quad & 2^m\cdot\mathtt{Pr}_{a_2^1,\ldots,a_2^t}\left[\left|\sum_{i=1}^t \mathbb{A}(x,\ldots,b_1'',a_2^i,\ldots)-tA_x(b_1'')\right|>\frac{\delta}{A_x(b_1'')}\left(tA_x(b_1'')\right)\right] \\
\leq\quad & 2^m\cdot\left(2\cdot e^{-\frac{1}{3}\cdot(tA_x(b_1''))\cdot\frac{\delta^2}{A_x(b_1'')^2}}\right)\ \leq\ 2^{m+1}\cdot e^{-\frac{1}{3}t\frac{\delta^2}{A_x(b_1'')}}\ \leq\ 2^{m+1}\cdot e^{-\frac{1}{3}t\delta^2} \\
<\quad & 2^{-h}.
\end{aligned}$$

The fifth inequality is valid by setting $t=O((m+h)/\delta^2)$.

If $x\notin L$, the error probability $\leq$ (1) $<(1-p)(A_x+\delta)+p<A_x+\delta+2^{-h}\leq\frac{3}{8}$ by taking $\delta=2^{-h}=1/8$, assuming the error probability is $\frac{1}{8}$ before switching.

Arthur-Merlin Hierarchy Collapses

Theorem (Babai, 1985). $\mathbf{AM}[k] = \mathbf{AM}[2]$ for all constant $k > 2$.

By Babai Theorem the following abbreviation makes sense.

$$\mathbf{AM} \stackrel{\text{def}}{=} \mathbf{AM}[2].$$

Speedup Theorem for Unbounded Interaction

Theorem (Babai and Moran, 1988). $\mathbf{AM}[2k(n)] = \mathbf{AM}[k(n)]$ if $k(n) \geq 2$.

The overall error probability is bounded by

$$A_x + \frac{k}{4} \cdot \left(\delta + 2^{m+1} \cdot e^{-\frac{1}{3} t\delta^2}\right) < \frac{1}{2},$$

by taking $\delta = \frac{1}{4k}$ and $t = 48k^4 m$.

$\mathbf{NP} \subseteq \mathbf{MA} \subseteq \mathbf{AM}$ can be interpreted as saying that $\mathbf{MA}$ and $\mathbf{AM}$ are randomized analogues of $\mathbf{NP}$.

- In $\mathbf{AM}$ the randomness is announced first.
- In $\mathbf{MA}$ the randomness comes afterwards.

If $\mathbf{BPP} = \mathbf{P}$, then $\mathbf{MA} = \mathbf{NP}$. Under plausible complexity conjecture, $\mathbf{AM} = \mathbf{NP}$.

1. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proofs. STOC '85.
2. L. Babai and S. Moran. Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes. JCSS, 1988.

The authors of the two papers shared the first Gödel Prize (1993).

Set Lower Bound Protocol

Set lower bound protocol [2] is based on Carter and Wegman's universal hash function.

1. J. Carter and M. Wegman. Universal Classes of Hash Functions. Journal of Computer and System Sciences. 143-154, 1979. (FOCS 1977)
2. S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. STOC 1986.

Pairwise Independent Hash Function

Let $\mathcal{H}_{n,k}$ be a collection of hash functions from $\{0,1\}^n$ to $\{0,1\}^k$.

We say that $\mathcal{H}_{n,k}$ is pairwise independent if the following hold:

- For each $x \in \{0,1\}^n$ and each $y \in \{0,1\}^k$,

$$\Pr_{h \in_{\mathrm{R}} \mathcal{H}_{n,k}}[h(x) = y] = \frac{1}{2^k}.$$

- For all $x, x' \in \{0,1\}^n$ with $x \neq x'$ and all $y, y' \in \{0,1\}^k$,

$$\Pr_{h \in_{\mathrm{R}} \mathcal{H}_{n,k}}[h(x) = y \wedge h(x') = y'] = \frac{1}{2^{2k}}.$$

Efficient Pairwise Independent Hash Function

Theorem. For every n, let $\mathcal{H}_{n,n}$ be $\{h_{a,b}\}_{a,b\in\mathrm{GF}(2^n)}$, where for all a, b the function $h_{a,b} : \mathrm{GF}(2^n) \to \mathrm{GF}(2^n)$ is defined by

$$h_{a,b}(x) = a \cdot x + b.$$

Then the collection $\mathcal{H}_{n,n}$ is efficient pairwise independent.

We get $\mathcal{H}_{n,k}$ from $\mathcal{H}_{n,n}/\mathcal{H}_{k,k}$ by projection/embedding.

Motivation

Assume $S \subseteq \{0,1\}^m$ and $2^{k-2} < K \leq 2^{k-1}$.

Suppose $|S| \geq K$ and $y \in_{\mathrm{R}} \{0,1\}^k$. By pairwise independence,

$$\Pr_{h \in_{\mathrm{R}} \mathcal{H}_{m,k}}[y \in h(S)] \geq \sum_{x \in S} \Pr_h[h(x) = y] - \sum_{x < x'} \Pr_h \left[\begin{array}{l} h(x) = = y, \\ h(x') = y \end{array} \right] = \frac{|S|}{2^k} \cdot \left(1 - \frac{|S|-1}{2} \cdot \frac{1}{2^k}\right) > \frac{13}{16}.$$

By taking $\kappa = k/(2 - \log 3)$ one gets

$$\Pr_{h_1,\ldots,h_\kappa \in_{\mathrm{R}} \mathcal{H}_{m,k}} \left[y \notin \bigcup_{i=1}^{\kappa} h_i(S) \right] \leq \left(\frac{3}{4}\right)^{\kappa} < 2^{-k}.$$

Hence

$$\Pr_{h_1,\ldots,h_\kappa \in_{\mathrm{R}} \mathcal{H}_{m,k}} \left[\exists y \in \{0,1\}^k . y \notin \bigcup_{i=1}^{\kappa} h_i(S) \right] < 1.$$

Conclude that $\bigcup_{i=1}^{\kappa} h_i(S) = \{0,1\}^k$ for some $h_1, \ldots, h_\kappa \in \mathcal{H}_{m,k}$.

Motivation

Suppose $|S| \leq \frac{K}{p(k)}$ for a polynomial $p(k) \geq 2\kappa$. For all $h_1, \ldots, h_\kappa$,

$$\left| \bigcup_{i=1}^{\kappa} h_i(S) \right| \leq \sum_{i=1}^{\kappa} |h_i(S)| \leq \frac{K}{p(k)}\kappa \leq \frac{1}{4} \cdot 2^k = \frac{1}{4} \cdot \left| \{0,1\}^k \right|.$$

Set Lower Bound Protocol.

M: Send $h_1, \ldots, h_\kappa$ to Arthur.

A: Pick $y \in_{\mathrm{R}} \{0,1\}^k$. Send y to Merlin.

M: Send i, x to Arthur, together with a certificate that $x \in S$.

Arthur accepts if $h_i(x) = y$ and the certificate validates $x \in S$; otherwise it rejects.

The protocol has perfect completeness. Its soundness parameter is $\frac{1}{4}$.

The protocol can be simplified if perfect completeness is compromised.

Set Lower Bound Protocol

Input.

1. Numbers K, k such that $2^{k-2} < K \leq 2^{k-1}$.
2. $S \subseteq \{0,1\}^m$ such that the membership in S can be certified.

Goal.

1. Prover tries to convince verifier that $|S| \geq K$.
2. Verifier should reject with good probability if $|S| \leq \frac{K}{2}$.

Let $\ell = \log k + 3$. We transform in P-time the question "$|S| \geq K$ or $|S| \leq K/2$?" to

$$\text{“}|S^\ell| \geq K^\ell \text{ or } |S^\ell| \leq K^\ell/2^\ell\text{ ?”}.$$

Then apply the protocol defined on the previous slide.

$\mathtt{GNI}$ is in $\mathbf{AM}$

Let S be

$$\{\langle H, \pi\rangle \mid H \simeq G_0 \text{ or } H \simeq G_1, \text{ and } \pi \text{ is an automorphism}\}.$$

Observe that

$$\text{if } G_0 \not\simeq G_1 \text{ then } |S| = 2n!$$

and

$$\text{if } G_0 \simeq G_1 \text{ then } |S| = n!.$$

Now apply the set lower bound protocol.

---

1. Suppose $\langle H, \pi\rangle$ is coded up by binary string of length m. Then $S \subseteq \{0,1\}^m$.
2. Checking the membership of S can be done in P-time.

Can GI be NP-Complete?

Theorem. If GI is $\mathbf{NP}$-complete, then $\Sigma_2^p = \Pi_2^p$.

---

1. R. Boppana, J. Håstad, and S. Zachos. Does co-NP Have Short Interactive Proofs? Information Processing Letters, 25:127-132, 1987.

Proof of Boppana-Håstad-Zachos Theorem

If $\mathtt{GI}$ is $\mathbf{NP}$-complete, then $\mathtt{GNI}$ is $\mathbf{coNP}$-complete. It follows that

- there is a reduction function f such that for every formula $\varphi(x, y)$ of $2n$ variables and for every fixed value x, $\forall y.\varphi(x, y)$ if and only if $f(\forall y.\varphi(x, y)) \in \mathtt{GNI}$.

Consider an arbitrary $\sum_2 \mathtt{SAT}$ formula $\psi = \exists x \in \{0, 1\}^n.\forall y \in \{0, 1\}^n.\varphi(x, y)$. Now

$$\psi \text{ iff } \exists x \in \{0, 1\}^n.g(x) \in \mathtt{GNI},$$

where g is a P-time function that maps x onto $f(\forall y.\varphi(x, y))$.

$\mathtt{GNI}$ has a two round Arthur-Merlin proof system with perfect completeness and soundness error $< 2^{-n}$. Let

- $\mathbb{A}$ be Arthur's algorithm, and
- m be the length of Arthur's questions and Merlin's answers.

Proof of Boppana-Håstad-Zachos Theorem

We claim that ψ is true if and only if

$$\forall q \in \{0,1\}^m . \exists x \in \{0,1\}^n . \exists a \in \{0,1\}^m . \mathbb{A}(g(x), q, a) = 1, \tag{2}$$

which would show $\sum_2 \subseteq \prod_2$. Notice that ψ is equivalent to

$$\exists x \in \{0,1\}^n . \forall q \in \{0,1\}^m . \exists a \in \{0,1\}^m . \mathbb{A}(g(x), q, a) = 1. \tag{3}$$

---

(3)⇒(2). If (2) holds, that is $\forall q \in \{0,1\}^m . \exists x \in \{0,1\}^n . \exists a \in \{0,1\}^m . \mathbb{A}(g(x), q, a) = 1$, there is some x_0 such that for at least 2^{m-n} number of $q \in \{0,1\}^m$,

$$\exists a \in \{0,1\}^m . \mathbb{A}(g(x_0), q, a) = 1.$$

This implies that the error rate for the input $g(x_0)$ is $\geq \frac{1}{2^n}$ if ψ does not hold, which would contradict to our assumption. So (2) $\Rightarrow \Psi$. Conclude (2) $\Rightarrow \Psi \Rightarrow$ (3) $\Rightarrow$ (2).

$\mathbf{IP} = \mathbf{PSPACE}$

C. Lund, L. Fortnow, H. Karloff, and N. Nisan.

- Algebraic Methods for Interactive Proof Systems. FOCS 1990.

A. Shamir.

- IP = PSPACE. FOCS 1990.

L. Babai, L. Fortnow, and L. Lund.

- Nondeterministic Exponential Time has Two-Prover Interactive Protocols. FOCS 1990.

We only have to prove $\texttt{TQBF} \in \mathbf{IP}$.

We start by looking at an interactive proof system for a decision version of $\overline{\texttt{SAT}}$.

Counting the Number of Satisfying Assignments

Let $\#\phi$ be the number of the satisfying assignments of ϕ.

- ϕ is a tautology iff $\#\phi = 2^n$ iff

$$\left(\sum_{b_1,\ldots,b_n \in \{0,1\}} \phi(b_1, \ldots, b_n) \right) = 2^n.$$

Let $\#\mathtt{SAT}_D$ be $\{\langle \phi, K \rangle \mid \phi \text{ is a 3CNF and } K = \#\phi\}$.

- This is a decision version of $\#\mathtt{SAT}$.
- An interactive proof system for $\#\mathtt{SAT}_D$ solves $\overline{\mathtt{SAT}}$ as well.

Arithmetization

Suppose $\phi = \phi_1 \wedge \ldots \wedge \phi_m$ is a 3CNF with n variables.

Let $X_1, \ldots, X_n$ be variables over a finite field $\mathrm{GF}(p)$, where p is a prime in $(2^n, 2^{2n}]$.

Arithmetization refers to for example the following conversion:

$$x_i \vee \overline{x_j} \vee x_k \;\mapsto\; 1 - (1 - X_i)X_j(1 - X_k).$$

We let 1 represent the truth value and 0 the false value.

We write $p_j(X_1, \ldots, X_n)$ for the arithmetization of ϕ_j.

We write $p_\phi(X_1, \ldots, X_n)$ for $\prod_{j \in [m]} p_j(X_1, \ldots, X_n)$, the arithmetization of ϕ.

- $|p_\phi(X_1, \ldots, X_n)| = \mathrm{poly}$. But if we open up the brackets in $p_\phi(X_1, \ldots, X_n)$, we would generally get an expression of exponential size.

Arithmetization

Clearly

$$\#\phi = \sum_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} p_\phi(b_1, \ldots, b_n) \leq 2^n.$$

Suppose $g(X_1, \ldots, X_n)$ is a degree d polynomial, K an integer.

We show how the prover can provide an interactive proof for

$$K = \sum_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} g(b_1, \ldots, b_n). \tag{4}$$

Notice that

$$\sum_{b_2 \in \{0,1\}} \sum_{b_3 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} g(X_1, b_2, \ldots, b_n) \tag{5}$$

is a univariate polynomial $h(X_1)$ whose degree is bounded by d.

- It takes exponential time to calculate (5).
- Prover can produce the small size polynomial $h(X_1)$ equal to (5). Using $h(X_1)$ the equality (4) becomes $K = h(0) + h(1)$.

Sumcheck Protocol

PROTOCOL: Sumcheck

A: If $n = 1$, check $g(0) + g(1) = K$. If the equality is valid, accept; otherwise reject. If $n \geq 2$, ask M to send some polynomial equal to (5).

M: Send some polynomial $s(X_1)$ to A.

A: Reject if $s(0) + s(1) \neq K$; otherwise send a random $a \in_{\mathrm{R}} \mathrm{GF}(p)$ to M. Recursively use the protocol to check

$$s(a) = \sum_{b_2 \in \{0,1\}} \sum_{b_3 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} g(a, b_2, \ldots, b_n).$$

Sumcheck is a public coins protocol with perfect completeness.

Sumcheck Protocol

Claim. If (4) is true, then $\Pr[\mathbb{A} \text{ accepts}] = 1$.

Claim. If (4) is false, then $\Pr[\mathbb{A} \text{ rejects}] \geq \left(1 - \frac{d}{p}\right)^{n-1}$.

Proof.

Assume (4) is false.

Case $n = 1$. Arthur rejects with probability 1.

Case $n > 1$.

- If Merlin returns $s(X_1) \neq h(X_1)$, then $s(X_1) - h(X_1)$ has at most d roots.
- Since Arthur picks up a randomly, $\Pr[s(a) \neq h(a)] \geq 1 - d/p$.

If $s(a) \neq h(a)$, Arthur rejects inductively with probability $\geq \left(1 - \frac{d}{p}\right)^{n-2}$. □

Interactive Proof for $\#\mathtt{SAT}_D$

Theorem (Lund, Fortnow, Karloff, Nisan, 1990). $\#\mathtt{SAT}_D \in \mathbf{IP}$.

Use the Sumcheck protocol.

Arithmetization for TQBF

Given a quantified Boolean formula

$$\psi = \forall x_1 \exists x_2 \forall x_3 \ldots \exists x_n . \phi(x_1, \ldots, x_n),$$

the arithmetization of $\psi \Leftrightarrow \top$ could be

$$\prod_{b_1 \in \{0,1\}} \sum_{b_2 \in \{0,1\}} \prod_{b_3 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} p_\phi(b_1, \ldots, b_n) \neq 0. \tag{6}$$

The problem is that the degree of (6) could be too high.

Arithmetization for TQBF

The idea is to use linearization operators

$$\begin{aligned} L_{X_i}(p) &= (1 - X_i)p_0 + X_i p_1, \\ \forall_{X_i}(p) &= p_0 p_1, \\ \exists_{X_i}(p) &= 1 - (1 - p_0)(1 - p_1) \end{aligned}$$

to obtain a multilinear polynomial, where

$$\begin{aligned} p_0 &= p(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n), \\ p_1 &= p(X_1, \ldots, X_{i-1}, 1, X_{i+1}, \ldots, X_n). \end{aligned}$$

---

1. A. Shen. IP=PSPACE: Simplified Proof. J.ACM, 1992.

Reduce the inequality (6) in $O(n^2)$ time to the equality:

$$\forall X_1 L_{X_1} \exists X_2 L_{X_1} L_{X_2} \ldots \forall X_{n-1} L_{X_1} ... L_{X_{n-1}} \exists X_n L_{X_1} ... L_{X_n}. p_\phi(X_1, \ldots, X_n) \; = \; 1. \tag{7}$$

Then apply the modified sumcheck protocol to check if (7) is valid.

Sumcheck Protocol:

1. Merlin sends $s_1(X_1)$ to Arthur, meant to be the openup of the red-expression in (7).
2. Arthur rejects if $s_1(0) \cdot s_1(1) \neq 1$. Otherwise he chooses $r_1 \in_R \mathrm{GF}(p)$ and asks Merlin to prove
$$(L_{X_1} \exists X_2 L_{X_1} L_{X_2} \ldots \exists X_n L_{X_1} .. L_{X_n}. p_\phi(X_1, \ldots, X_n)) \{r_1/X_1\} = s_1(r_1). \tag{8}$$
3. Merlin sends $s_2(X_1)$ to Arthur, meant to be the openup of the blue-expression in (8).
4. Arthur rejects if $(1 - r_1) \cdot s_2(0) + r_1 \cdot s_2(1) \neq s_1(r_1)$. Otherwise he chooses $r_1' \in_R \mathrm{GF}(p)$ and asks Merlin to prove blue-expression$\{r_1'/X_1\} = s_2(r_1')$.
5. ...

$\mathbf{IP} = \mathbf{PSPACE}$

Theorem (Shamir 1990). $\mathbf{IP} = \mathbf{PSPACE}$.

Using Sumcheck protocol one sees that $\mathtt{TQBF}$ is in $\mathbf{IP}$.

Theorem. $\mathbf{IP} = \bigcup_{c\geq 1} \mathbf{AM}[cn^c]$.

We have defined an Arthur-Merlin game for `TQBF`, which is $\mathbf{PSPACE}$-complete. Hence

$$\mathbf{IP} = \mathbf{PSPACE} \subseteq \bigcup_{c\geq 1} \mathbf{AM}[cn^c] \subseteq \mathbf{IP}.$$

The theorem does not say that a constant round private coin interactive system can be simulated by a constant round public coin interactive system.

Remark on the Proof of $\mathbf{IP} = \mathbf{PSPACE}$

- The proof of $\mathbf{IP} = \mathbf{PSPACE}$ does not relativize.
  1. Fortnow and Sipser proved in 1988 that $\exists O.\, \mathbf{coNP}^O \not\subseteq \mathbf{IP}^O$.
  2. If $\mathbf{IP} = \mathbf{PSPACE}$ had a proof that would relativize, then $\mathbf{coNP} \subseteq \mathbf{IP}$ would have a proof that would relativize.
- $\mathbf{IP} = \mathbf{PSPACE}$ implies that every problem in $\mathbf{IP}$ has an interactive proof with perfect completeness.

There e-mail announcements were made within a month of 1989.

1. N. Nisan. “Co-SAT Has Multi-Prover Interactive Proofs”, Nov. 27.
2. C. Lund, L. Fortnow, H. Karloff, and N. Nisan. “The Polynomial Time Hierarchy Has Interactive Proofs”, Dec. 13.
3. A. Shamir. “IP=PSPACE”, Dec. 26.

   

L. Babai. E-mail and the unexpected power of interaction. In Proc. The Fifth Annual Structure in Complexity Theory Conference, 1990.

Public Coins versus Private Coins

Theorem (Goldwasser-Sipser). $\mathbf{IP}[k(n)] = \mathbf{AM}[k(n)]$ for all polynomial $k(n) > 2$.

Goldwasser and Sipser. Private Coins versus Public Coins in Interactive Proof Systems. STOC 1986.

 

The key to the proof of Goldwasser-Sipser Theorem is that Merlin can apply the set lower bound protocol to convince Arthur that the chance for Prover to make Verifier believe is big if $x \in L$.

Proof of Goldwasser-Sipser Theorem

Let L be accepted by a $2k$ round private coin interactive proof system $(\mathbb{V}, \mathbb{P})$.

Let h be the length of $\mathbb{V}$'s questions and $\mathbb{P}$'s answers, ℓ be the length of random strings.

Without loss of generality suppose $2kh < \ell$.

Let x be the input.

We will design an $O(k)$ round Arthur-Merlin game $(\mathbb{A}, \mathbb{M})$ that accepts L.

- $(\mathbb{A}, \mathbb{M})$ simulates every round of $(\mathbb{V}, \mathbb{P})$ by 3 rounds.
- Merlin will convince Arthur that in each round of $(\mathbb{V}, \mathbb{P})$ there are many random strings that eventually force $\mathbb{V}$ to say "yes".

Proof of Goldwasser-Sipser Theorem

Let $\gamma_i = a_1, b_1, \ldots, a_i, b_i$ denote the initial i round dialogue between $\mathbb{V}$ and $\mathbb{P}$.

Let γ_0 be the empty string ϵ.

Let $\mathrm{Yes}_x(\gamma_i)$ be the set of all the random strings $r \in \{0,1\}^\ell$ that make $\mathbb{V}$ say "yes" by dialogues with the initial i rounds being γ_i.

By definition,

$$|\mathrm{Yes}_x(\gamma_i)| = \sum_{a \in \{0,1\}^h} |\mathrm{Yes}_x(\gamma_i, a)|. \tag{9}$$

Since $\mathbb{P}$ is optimal,

$$|\mathrm{Yes}_x(\gamma_i, a)| = \max_{b \in \{0,1\}^h} |\mathrm{Yes}_x(\gamma_i, a, b)|. \tag{10}$$

Proof of Goldwasser-Sipser Theorem

Suppose $K_i \leq \ldots \leq K_0 = 2^\ell$ have been defined such that $K_\imath \leq |\mathsf{Yes}_x(\gamma_\imath)|$ for all $\imath \in \{0, \ldots, i\}$.

Classify into ℓ groups the elements $a \in \{0,1\}^h$ satisfying $|\mathsf{Yes}_x(\gamma_i, a)| > 0$. For $j \in \{0\} \cup [\ell - 1]$,

$$V_j = \left\{ a \in \{0,1\}^h \mid 2^j \leq |\mathsf{Yes}_x(\gamma_i, a)| < 2^{j+1} \right\}.$$

Since $|\mathsf{Yes}_x(\gamma_i)| \geq K_i$, there is some j such that $|\{r \in \mathsf{Yes}_x(\gamma_i, a) \mid a \in V_j\}| \geq K_i/\ell$. Hence

$$|V_j| \quad > \quad \frac{K_i}{2^{j+1}\ell}. \tag{11}$$

For every $a \in V_j$, one has

$$|\mathsf{Yes}_x(\gamma_i, a)| \quad \geq \quad 2^j. \tag{12}$$

For each a in (12), Prover's answer $b \in \{0,1\}^h$ satisfies $|\mathsf{Yes}_x(\gamma_i, a, b)| \geq 2^j$. Let $K_{i+1} = 2^j$.

Two step verification: the membership check of the first step is broken into two parts, the second part is carried out in the second step with additional messages from Merlin.

Proof of Goldwasser-Sipser Theorem

Protocol input at round $i+1$: input x, private coin interactive proof system $(\mathbb{V},\mathbb{P})$, the i-round dialogue $\gamma_i = a_1, b_1, \ldots, a_i, b_i$, and $K_i \leq \ldots \leq K_0 = 2^\ell$.

M: Send j and $h_1, \ldots, h_\kappa$ to A.

A: Send $\alpha \in_R \{0,1\}^g$ to M.

M: Send $s \in \{0,1\}^\ell$, $f \in [\kappa]$ and a_{i+1}, b_{i+1}, γ to A. Also send $h'_1, \ldots, h'_{\kappa'}$ to A.

A: If $\gamma_i, a_{i+1}, b_{i+1}, \gamma$ is inconsistent with x, s, or $\mathbb{V}(x, s, \gamma_i, a_{i+1}, b_{i+1}, \gamma) = 0$, or $h_f(a_{i+1}) \neq \alpha$, reject; otherwise send $\beta \in_R \{0,1\}^{j+2}$ to M.

M: Send $t \in \{0,1\}^\ell$, $f' \in [\kappa']$ and γ' to A.

A: If $\gamma_i, a_{i+1}, b_{i+1}, \gamma'$ is inconsistent with x, t, or $\mathbb{V}(x, t, \gamma_i, a_{i+1}, b_{i+1}, \gamma') = 0$, or $h'_{f'}(t) \neq \beta$, reject; otherwise go to round $i+2$ with $K_{i+1} = 2^j$ and $\gamma_{i+1} = \gamma_i, a_{i+1}, b_{i+1}$.

$h_1, \ldots, h_\kappa : \{0,1\}^h \to \{0,1\}^g$ and $h'_1, \ldots, h'_{\kappa'} : \{0,1\}^\ell \to \{0,1\}^{j+2}$ are pairwise independent Hash functions; $\kappa = g/(2-\log 3)$ and $2^{g-2} \leq \frac{K_i}{2^{j+1}\ell} < 2^{g-1}$; and $\kappa' = (j+2)/(2-\log 3)$.

Proof of Goldwasser-Sipser Theorem

The protocol has perfect completeness. If $x \in L$, then for all $i \in [k]$,

$$|\mathrm{Yes}_x(\gamma_i)| \geq K_i.$$

Suppose $(\mathbb{V}, \mathbb{P})$ has an error probability $\frac{1}{p(\ell)^{k+1}}$ for a large polynomial p. Suppose $x \notin L$. Now

$$|\mathrm{Yes}_x(\epsilon)| \leq \frac{1}{p(\ell)^{k+1}} \cdot 2^{\ell} = \frac{1}{p(\ell)^{k+1}} \cdot K_0.$$

Assume that the following holds:

$$|\mathrm{Yes}_x(\gamma_i)| < \frac{1}{p(\ell)^{k+1-i}} \cdot K_i.$$

We will prove that (13) is valid with probability greater than $1 - \frac{1}{3k}$.

$$|\mathrm{Yes}_x(\gamma_{i+1})| \quad < \quad \frac{1}{p(\ell)^{k+1-(i+1)}} \cdot K_{i+1}. \tag{13}$$

Proof of Goldwasser-Sipser Theorem

Consider $S' = \left\{ a \;\middle|\; |\mathsf{Yes}_x(\gamma_i, a)| \geq \frac{1}{p(\ell)^{k+1-(i+1)}} \cdot K_{i+1} \right\}$. By (9) and the inductive hypothesis, then

$$|S'| \cdot \frac{1}{p(\ell)^{k+1-(i+1)}} \cdot K_{i+1} \leq |\mathsf{Yes}_x(\gamma_i)| < \frac{1}{p(\ell)^{k+1-i}} \cdot K_i.$$

According to the above inequality and (11),

$$|S'| < \frac{1}{p(\ell)} \cdot \frac{1}{K_{j+1}} \cdot K_i < \frac{1}{p(\ell)} \cdot \frac{1}{2^j} \cdot K_i < 2\ell \cdot \frac{1}{p(\ell)} \cdot |V_j|.$$

Since p is large enough,

$$\Pr_{r \in_{\mathrm{R}} \{0,1\}^\ell}[a \in S'] < \frac{|S'|}{|V_j|} < \frac{2\ell}{p(\ell)} \leq \frac{1}{3k}.$$

$\Pr_{r \in_{\mathrm{R}} \{0,1\}^\ell}[a \in S']$ is the probability that the following holds

$$|\mathsf{Yes}_x(\gamma_i, a)| \geq \frac{1}{p(\ell)^{k+1-(i+1)}} \cdot K_{i+1}.$$

Since $|\mathsf{Yes}_x(\gamma_{i+1})| = |\mathsf{Yes}_x(\gamma_i, a)|$, the probability that (13) is valid is at least $1 - \frac{1}{3k}$.

Proof of Goldwasser-Sipser Theorem

The following inequality is valid with probability greater than $\left(1-\frac{1}{3k}\right)^k \geq 1-\frac{1}{3k}\cdot k = \frac{2}{3}$.

$$\bigwedge_{i=0}^{k}\left(|\mathsf{Yes}_x(\gamma_i)| < \frac{1}{p(\ell)^{k+1-i}}\cdot K_i\right).$$

The set lower bound protocol has an error probability $\frac{1}{4}$.

The error probability of Goldwasser-Sipser Protocol is less than $\frac{1}{3}+\frac{2}{3}\cdot\frac{1}{4}=\frac{1}{2}$.

Conclude that

$$\mathbf{IP}[2k(n)] \subseteq \mathbf{AM}[6k(n)] \subseteq \mathbf{AM}[2k(n)].$$

Corollary. If $k(n) > 2$, then

$$\mathbf{IP}[k(n)] = \mathbf{IP}[k(n)]^+ = \mathbf{AM}[k(n)]^+ = \mathbf{AM}[k(n)],$$

where $\mathbf{AM}[k(n)]^+$ is the subset of $\mathbf{AM}[k(n)]$ with perfect completeness.

Corollary. $\mathbf{AM} = \mathbf{AM}^+ \subseteq \Pi_2^p$

$\mathbf{AM} \stackrel{?}{=} \mathbf{IP}$

Theorem. If $\mathbf{PSPACE} \subseteq \mathbf{P}_{/\mathrm{poly}}$ then $\mathbf{IP} = \mathbf{MA} = \mathbf{AM}$.

We only have to prove that $\mathbf{PSPACE} \subseteq \mathbf{MA}$. If $\mathbf{PSPACE} \subseteq \mathbf{P}_{/\mathrm{poly}}$, then the prover in the `TQBF` protocol can be replaced by a P-size circuit family $\{C_n\}_{n \in \mathbf{N}}$.

Define a game in which Merlin simply sends the description of $C_{|x|}$ to Arthur.

Arthur can now make use of $C_{|x|}$ without the necessity for any further interaction.

Theorem. If $\mathbf{coNP} \subseteq \mathbf{AM}$, then $\mathbf{PH} = \mathbf{AM}$.

Proof.

Clearly $\Sigma_1^p = \mathbf{NP} \subseteq \mathbf{MA}^+ \subseteq \mathbf{AM}^+ = \mathbf{AM}$, and $\Pi_1^p = \mathbf{coNP} \subseteq \mathbf{AM}$ by the hypothesis. Prove that $\Sigma_i^p, \Pi_i^p \subseteq \mathbf{AM}$ implies $\Sigma_{i+1}^p, \Pi_{i+1}^p \subseteq \mathbf{AM}$ for all $i > 0$. $\square$

Corollary. If $\texttt{GI}$ is $\mathbf{NP}$-complete, then $\mathbf{PH} = \mathbf{AM}$.

Proof.

If $\texttt{GI}$ is $\mathbf{NP}$-complete, then $\texttt{GNI}$ is $\mathbf{coNP}$-complete. We have proved that $\texttt{GNI} \in \mathbf{AM}$, hence $\mathbf{coNP} \subseteq \mathbf{AM}$. $\square$

Multi-Prover Interactive Proof System

We know that $\mathbf{NP} \subseteq \mathbf{MA} \subseteq \mathbf{AM} \subseteq \mathbf{PH} \subseteq \mathbf{PSAPCE} = \mathbf{IP}$.

We do not know if interactive proof systems are more powerful than P-time NDTMs.

It turns out that multi-prover interactive proof systems are strictly more powerful.

---

1. M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-Prover Interactive Proofs: How to Remove Intractability Assumptions. STOC 1988.
2. L. Babai, L. Fortnow, and L. Lund. Nondeterministic Exponential Time Has Two Prover Interactive Protocols. Computational Complexity, 1991 (FOCS'90).
3. L. Fortnow, J. Rompel, and M. Sipser. On the Power of Multi-Prover Interactive Protocols. Theoretical Computer Science, 1994.

Theorem. $\mathbf{MIP} = \mathbf{NEXP}$.

Provers may decide on any strategy before the game. Once the game started, a prover is only allowed to communicate with the verifier.

The verifier can talk to any prover.

The verifier can force a prover to answer in a nonadaptive fashion.

Multi-Prover Interactive Proof System

Suppose k, t are polynomial, n is the input length.

A $t(n)$-round $k(n)$-prover interactive proof system consists of a verifier $\mathbb{V}$ and $k(n)$ provers $\mathbb{P}_1, \ldots, \mathbb{P}_{k(n)}$, the verifier is a P-time PTM, and a prover is a TM (function). The number of messages exchanged between $\mathbb{V}$ and $\mathbb{P}_1, \ldots, \mathbb{P}_{k(n)}$ is $2t(n)$.

After the $t(n)$-round interaction, the verifier makes a decision $\mathbb{V}(x, r, \gamma_1 \sharp \gamma_2 \sharp \ldots \sharp \gamma_{k(n)})$, where r is a random string, and for each $i \in [k(n)]$, γ_i is the dialogue between $\mathbb{V}$ and $\mathbb{P}_i$.

Languages Accepted by Multi-Prover Interactive Proof System

L is accepted by a $t(n)$-round $k(n)$-prover interactive proof system $(\mathbb{V},\ldots)$ if for any x,

1. if $x \in L$, some $\mathbb{P}_1,\ldots,\mathbb{P}_{k(|x|)}$ exist such that

$$\Pr_{r\in\{0,1\}^{q(|x|)}}[\mathbb{V}(x,r,\gamma_1\sharp\gamma_2\sharp\ldots\sharp\gamma_{k(|x|)})=1] \geq 1-\frac{1}{2^{|x|}};$$

2. if $x \notin L$, then for any $\mathbb{P}_1,\ldots,\mathbb{P}_{k(|x|)}$,

$$\Pr_{r\in\{0,1\}^{q(|x|)}}[\mathbb{V}(x,r,\gamma_1\sharp\gamma_2\sharp\ldots\sharp\gamma_{k(|x|)})=1] < \frac{1}{2^{|x|}},$$

where q is a polynomial and $q(|x|)$ is the length of verifier's random string.

$L \in \mathbf{MIP}$ if L is accepted by a polynomial round $k(n)$-prover interactive proof system, where $k(n)$ is polynomial.

Lemma. $L \in \mathbf{MIP}$ if and only if L is accepted by a P-round 2-prover interactive proof system.

We design a two prover interactive proof system $(\mathbb{P}_1, \mathbb{P}_2, \mathbb{V})$ that simulates $(\mathbb{Q}_1, \ldots, \mathbb{Q}_k, \mathbb{U})$.

1. $\mathbb{V}$ interacts with $\mathbb{P}_1$ to simulate the interaction of $(\mathbb{Q}_1, \ldots, \mathbb{Q}_k, \mathbb{U})$.
2. To force nonadaptivity $\mathbb{V}$ chooses randomly $i \in [k]$ and ask $\mathbb{P}_2$ to repeat the dialogue of $\mathbb{P}_i$.

Error probability less than $1 - \frac{1}{k} + \frac{1}{2^n}$. Repeat the protocol k^2 times to reduce it to below $\frac{1}{2^n}$.

---

1. M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-Prover Interactive Proofs: How to Remove Intractability Assumptions. STOC 1988.

Probabilistic Oracle Turing Machine

A P-time Probabilistic OTM $\mathbb{M}^?$ accepts L if the following statements are valid:

1. If $x \in L$, then some oracle O exists such that $\mathtt{Pr}[\mathbb{M}^O(x) = 1] \geq 1 - \frac{1}{2^n}$.
2. If $x \notin L$, then for any oracle O, it holds that $\mathtt{Pr}[\mathbb{M}^O(x) = 1] < \frac{1}{2^n}$.

Lemma. $L \in \mathbf{MIP}$ if and only if L is accepted by a P-time POTM.

($\Rightarrow$). Question $(i, j, d, \gamma_1 \sharp \ldots \sharp \gamma_k)$, the d-th bit of $\mathbb{P}_i$'s answer in the j-th round.

($\Leftarrow$). To force non-adaptivity, the verifier asks at most one question to every prover.

---

1. L. Fortnow, J. Rompel, M. Sipser. On the power of multi-prover interactive protocols. In: The Third Annual Conference on Structure in Complexity Theory. Also in Theoretical Computer Science, **134**, 1994.

Multi-Prover Interactive Proof System = POTM = 2-Prover Interactive Proof System

---

$L \in \mathbf{MIP}$ if and only if there is a polynomial round interactive proof system $(\mathbb{V}^?, \_)$, where $\mathbb{V}^?$ is a P-time POTM, such that the followings are valid.

1. If $x \in L$, then $\exists O, \mathbb{P}. \Pr_{r \in \{0,1\}^{q(n)}}[\mathbb{V}_{\mathbb{P}}^{O}(x, r, \gamma_1 \sharp \gamma_2) = 1] \geq 1 - \dfrac{1}{2^n}$.
2. If $x \notin L$, then $\forall O, \mathbb{P}. \Pr_{r \in \{0,1\}^{q(n)}}[\mathbb{V}_{\mathbb{P}}^{O}(x, r, \gamma_1 \sharp \gamma_2) = 1] < \dfrac{1}{2^n}$.

The question is really about how many rounds are necessary.

Proposition. $\mathbf{MIP} \subseteq \mathbf{NEXP}$.

Suppose a POTM $\mathbb{Q}^?$ accepts L in n^c time, and the input length is n.

$\mathbb{Q}^?$ asks at most n^c questions, the length of the answers is bounded by n^c. There are at most 2^{n^c} combinations of the answers.

A NDTM guesses an oracle O and simulates $\mathbb{Q}^O$ on all random strings.

If there is a computation path that accepts x by the ratio $1 - \frac{1}{2^n}$, accept; o.w. reject.

---

1. L. Fortnow, J. Rompel, M. Sipser. On the power of multi-prover interactive protocols. In: The Third Annual Conference on Structure in Complexity Theory. Also in Theoretical Computer Science, **134**, 1994.

Perfect Completeness

Theorem. $L \in \mathbf{MIP}$ iff there is a polynomial round interactive proof system $(\mathbb{V}, \_, \_)$ rendering true the followings:

1. Completeness. If $x \in L$, then $\exists\, \mathbb{P}_1, \mathbb{P}_2. \Pr_{r \in \{0,1\}^{q(n)}}[\mathbb{V}_{\mathbb{P}_1,\mathbb{P}_2}(x, r, \gamma_1 \sharp \gamma_2) = 1] = 1$.
2. Soundness. If $x \notin L$, then $\forall\, \mathbb{P}_1, \mathbb{P}_2. \Pr_{r \in \{0,1\}^{q(n)}}[\mathbb{V}_{\mathbb{P}_1,\mathbb{P}_2}(x, r, \gamma_1 \sharp \gamma_2) = 1] < \frac{1}{2^n}$.

Multiprover Interactive System for $\mathbf{NEXP}$

Suppose a $T(n)$ time NDTM $\mathbb{N}$ accepts $L \in \mathbf{NP}$, and the input x is of length n.

By Cook-Levin reduction, one obtains a 3-CNF $\psi(z)$ such that $\mathbb{N}(x) = 1$ iff

$$\psi(z) = \bigwedge_{c \in [O(T(|x|))]} \psi_c(z_c^1, z_c^2, z_c^3),$$

is true, where x is part of z.

The variables in z may be encoded by a string of length $t = \log|z| = O(\log(|x|))$.
We use variables $v_1, \ldots, v_t$ to represent z.

An assignment to $v = v_1, \ldots, v_t$ codes up a variable in z. An assignment to z is a function

$$A : \{0,1\}^t \to \{0,1\}.$$

A string of length $s = O(\log(|x|))$ codes up c. Use variables $u = u_1, \ldots, u_s$ to represent c.

For $u \in \{0,1\}^s$ and $h \in [3]$, define the constant $s_{h,u}$ as follows:

$$s_{h,u} = \begin{cases} 1, & \text{if the } h\text{-th variable } z_u^h \text{ in } \psi_u \text{ appears positive,} \\ 0, & \text{otherwise.} \end{cases}$$

Define the indicator variable $\chi_{h,u} : \{0,1\}^t \to \{0,1\}$ as follows:

$$\chi_{h,u}(v) = \begin{cases} 1, & \text{if the } h\text{-th variable } z_u^h \text{ in } \psi_u \text{ is coded up by } v, \\ 0, & \text{otherwise.} \end{cases}$$

A truth assignment A satisfies $\psi_u(z_u^1, z_u^2, z_u^3)$ iff

$$\bigvee_{v^1,v^2,v^3 \in \{0,1\}^t} \bigwedge_{h \in [3]} \chi_{h,u}(v^h)(s_{h,u} - A(v^h)) = 0. \tag{14}$$

A satisfies $\psi(z)$ iff

$$\bigvee_{u \in \{0,1\}^s} \bigvee_{v^1,v^2,v^3 \in \{0,1\}^t} \bigwedge_{h \in [3]} \chi_{h,u}(v^h)(s_{h,u} - A(v^h)) = 0. \tag{15}$$

We arithmetize (14) and (15).

- For instance $t = 3$ and the first variable of ψ_4 is $u_5 = 101$. Then $\chi_{1,4}(v^1) = v_1^1(1 - v_2^1)v_3^1$.
- The arithmetization of A is a multilinear function.

- $s_{1,u}$, $s_{2,u}$, $s_{3,u}$ can be computed from u. Let $\theta_h(u_1, \ldots, u_s, s_{h,u})$ be the polylog size 3-CNF for this computation. Let $O_h(u_1, \ldots, u_s, s_{h,u})$ be the arithmetization.
- $\chi_{1,u}(v^1)$, $\chi_{2,u}(v^2)$, $\chi_{3,u}(v^3)$ can be computed respectively from u, v^1, u, v^2, u, v^3.
  Let $\vartheta_h(u_1, \ldots, u_s, v^h)$ be the polylog 3-CNF that codes up this computation.
  Let $Q_h(u_1, \ldots, u_s, v^h)$ be the arithmetization.

Now arithmetize $\chi_{h,u}(v^h)(s_{h,u} = A(v^h))$ by

$$Q_h(u_1, \ldots, u_s, v^h) \cdot O_h(u_1, \ldots, u_s, s_{h,u}) \cdot \chi_{h,u}(v^h)(s_{h,u} - A(v^h)).$$

Let $\Psi_{u,v^1,v^2,v^3}(A(v^1), A(v^2), A(v^3))$ denote the above arithmetic expression.

It is tempting to apply the Sumcheck Protocol to test

$$\sum_{u\in\mathbf{F}_p^s}\sum_{v^1,v^2,v^3\in\mathbf{F}_p^t}\prod_{h\in[3]}\Psi_{u,v^1,v^2,v^3}(A(v^1),A(v^2),A(v^3)) \quad = \quad 0. \tag{16}$$

There is however a problem.

Set $\ell = s+3t$. The verifier randomly selects $r_1,\ldots,r_\ell$ from $\mathbf{F}_p$.

For every $d = u,v^1,v^2,v^3 \in \{0,1\}^\ell$, set $r_{u,v^1,v^2,v^3} = \prod_{d_i=1} r_i$.

If some $\prod_{h\in[3]}\Psi_{u,v^1,v^2,v^3}(A(v^1),A(v^2),A(v^3))$ is unequal to 0, the following

$$\sum_{u\in\mathbf{F}_p^s}\sum_{v^1,v^2,v^3\in\mathbf{F}_p^t} r_{u,v^1,v^2,v^3}\cdot\prod_{h\in[3]}\Psi_{u,v^1,v^2,v^3}(A(v^1),A(v^2),A(v^3)) = 0 \tag{17}$$

is valid with probability at least $\left(1-\frac{1}{p}\right)^\ell$. This is clear if we see $r_1,\ldots,r_\ell$ as variables over $\mathbf{F}_p$.

We are now in a position to define the verifier as an POTM.

---

1. Test if the oracle A defined on $\mathbf{F}_p$ is multilinear. If successful, the probability that $\Psi_{u,v^1,v^2,v^3}(A(v^1), A(v^2), A(v^3))$ is a low degree polynomial is great.
2. Suppose (17) is a d-degree polynomial, apply the Sumcheck Protocol.

During the interaction the verifier randomly assigns $b_1, \ldots, b_s, a_1^1, \ldots, a_t^1, a_1^2, \ldots, a_t^2, a_1^3, \ldots, a_t^3$ to $u_1, \ldots, u_s, v_1^1, \ldots, v_t^1, v_1^2, \ldots, v_t^2, v_1^3, \ldots, v_t^3$. The prover answers with the polynomial $g_1, \ldots, g_{s+3t}$, and the verifier carries out consistency check after each round.

If the first $s+3t-1$ consistency checks are successful, the verifier, after getting the final answer a_s^3, gets $A(a^1)$, $A(a^2)$, $A(a^3)$ by querying the oracle A. It then does the last consistency check:

$$g_{s+3t}(a_t^3) = r' \cdot \prod_{h \in [3]} \Psi_{b,a^1,a^2,a^3}(A(a^1), A(a^2), A(a^3)).$$

Let $\mathbf{F}_p$ be such that $p = O(\log|x|)$. Both the testing time and the size of random strings are `polylog`.

The failure probability is

$$\left(1 - \frac{O(1)}{O(\log|x|)}\right)^{O(\log|x|)} \cdot \left(1 - \frac{O(1)}{O(\log|x|)}\right)^{O(\log|x|)} \cdot (\text{success rate of linearity resting}).$$

Repeat the protocol $O(\log|x|)$ times, the error probability is decreased to $\frac{1}{\mathtt{polylog}(|x|)}$.

Repeat the above proof to a $2^{\mathtt{poly}}$ time NDTM, one derives that $\mathbf{NEXP} \subseteq \mathbf{MIP}$.

Theorem. $\mathbf{MIP}$ is the same as public-key 2-prover $\mathbf{MIP}$.

Multilinearity Testing

Multi-linear Function

A function $f: \mathbf{F}_p^s \to \mathbf{F}_p$ is multilinear if it is linear on every linear subspace $\mathfrak{l}$ of $\mathbf{F}_p^s$.

When we test a function, we test its geometric shape.

Basic Idea

Consider s-ary functions $f : \mathbf{F}_p^s \to \mathbf{F}_p$.

For $a_h \in \mathbf{F}_p$ let $f_{x_h=a_h}$ be the function $f(x_1, \ldots, a_h, \ldots, x_s)$ on the $(s-1)$-dimensional space $\left(\mathbf{F}_p^s\right)^{x_h=a_h} = \{(a'_1, \ldots, a'_s) \in \mathbf{F}_p^s \mid a'_h = a_h\}$.

A function is x_h-linear if it is a linear function by fixing all input parameters except x_h.

---

f is multilinear if and only if for any $(a_1, \ldots, a_s) \in \mathbf{F}_p^s$ the function $f_{x_1=a_1}$ is multilinear and $f(x_1, a_2, \ldots, a_s)$ is x_1-linear.

Approximate Muliti-Linearity

The Hemming distance $\mathrm{dist}(f, g)$ of $f, g : \mathbf{F}_p^s \to \mathbf{F}_p$ is $\Pr_{x \in_{\mathrm{R}} \mathbf{F}_p^s}[f(x) \neq g(x)]$.

Let ML be the set of the multilinear functions of type $\mathbf{F}_p^s \to \mathbf{F}_p$.

We measure the dissimilarity of f to any multilinear function by

$$\Delta_{ML}(f) = \min_{\mathfrak{l} \in ML} \mathtt{dist}(f, \mathfrak{l}).$$

Sample Points

A triple (a, b, c) is x_h-colinear if a, b, c are on a line parallel to the x_h-axis; and (a, b, c) is colinear if it is x_h-colinear for some x_h.

Suppose (a, b, c) is an x_h-colinear triple.

If there is an x_h-linear function $g : \mathbf{F}_p^s \to \mathbf{F}_p$ such that $f(a) = g(a)$, $f(b) = g(b)$ and $f(c) = g(c)$, then (a, b, c) is called f-linear.

We shall measure the non-multilinearity of f by

$$\tau(f) = \frac{|\text{all colinear triples that are not } f\text{-linear}|}{|\text{all colinear triples}|}.$$

Main Lemma. Non-multilinearity and Hemming distance are related as follows:

$$\tau(f) \geq \frac{3\Delta_{ML}(f)\,(1-\Delta_{ML}(f))}{s} - \frac{3}{p}.$$

Suppose $\mathfrak{l}$ is a multilinear function such that $\mathtt{dist}(f,\mathfrak{l}) = \Delta_{ML}(f)$.

Call colinear triple (a,b,c) chromatic if $f(a)=\mathfrak{l}(a)$ and $f(b)=\mathfrak{l}(b)$ and $f(c)=\mathfrak{l}(c)$, or $f(a)\neq\mathfrak{l}(a)$ and $f(b)\neq\mathfrak{l}(b)$ and $f(c)\neq\mathfrak{l}(c)$. Call it non-chromatic otherwise.

Proof of the Main Lemma

Let E be the union of the following events:

1. E_1: $f(a) = \mathfrak{l}(a)$ and $f(b) \neq \mathfrak{l}(b)$,
2. E_2: $f(b) = \mathfrak{l}(b)$ and $f(c) \neq \mathfrak{l}(c)$,
3. E_3: $f(c) = \mathfrak{l}(c)$ and $f(a) \neq \mathfrak{l}(a)$.

By symmetry $\mathtt{Pr}[E] = 3 \cdot \mathtt{Pr}[E_1]$.

It is not difficult to see that the probability $\mathtt{Pr}[E]$ is the same as

$$\mathtt{Pr}_{(a,b,c)\text{ colinear}}[(a, b, c)\text{ is non-chromatic}].$$

Proof of the Main Lemma

Select a, b randomly, by definition $\mathtt{Pr}_{a,b\in\mathbf{F}_p^s}[E_1] = \Delta_{ML}(f)\,(1 - \Delta_{ML}(f))$.

How do we choose two random points a, b that are on a line parallel to an axis?

Proof of the Main Lemma

1. Select $d, e \in_{\mathrm{R}} \mathbf{F}_p^s$ randomly.
2. Choose randomly $s' \in_{\mathrm{R}} [s]$.
   - Let the first $s'-1$ bits of a, b be the first $s'-1$ bits of d, and the last $s-s'$ bits of a, b be the last $s-s'$ bits of e;
   - Let the s'-th bit of a be the s'-th of d, and the s'-th bit of b be the s'-th bit of e.

If $f(d) = \mathfrak{l}(d)$ and $f(e) \neq \mathfrak{l}(e)$, then there are s equalities/inequalities:

$$f(d) = \mathfrak{l}(d), f(d_1, \ldots, d_{s-1}, e_s) \stackrel{?}{=} \mathfrak{l}(d_1, \ldots, d_{s-1}, e_s), \ldots, f(e) \neq \mathfrak{l}(e).$$

$$\Pr[E] = 3 \cdot \Pr[E_1] \geq 3 \cdot \frac{\Delta_{ML}(f)\,(1 - \Delta_{ML}(f))}{s}. \tag{18}$$

According to the inequality (18), there are enough non-chromatic colinear triples.

Proof of the Main Lemma

We need to exclude from (18) the colinear triples that are f-linear.

Suppose (a, b, c) is x_h-colinear and f-linear.

1. $f(a) = g(a)$, $f(b) = g(b)$ and $f(c) = g(c)$ for some x_h-linear function g.
2. g (consequently f) and $\mathfrak{l}$ cannot coincide on two of a, b, c but differ on the other.
3. If the event E_3 occurs, it must be that $f(c) = \mathfrak{l}(c)$ and $f(a) \neq \mathfrak{l}(a)$ and $f(b) \neq \mathfrak{l}(b)$.
4. Thus $g(c) = \mathfrak{l}(c)$.

For fixed a, b there is only one such triple.

Assume that (a, b, c') were another such triple. Then $f(a) = g'(a)$, $f(b) = g'(b)$ and $f(c') = g'(c')$ for some x_h-linear function g', and $g'(c') = \mathfrak{l}(c')$.

It follows from $g(a) = f(a) = g'(a)$ and $g(b) = f(b) = g'(b)$ that $g(c') = g'(c') = \mathfrak{l}(c')$.

One would then derive the contradictory equality $\mathfrak{l}(a) \neq f(a) = g(a) = \mathfrak{l}(a)$.

Proof of the Main Lemma

There are $\binom{p}{3}$ different triples on a line parallel to an axis.

By the discussion in the above, if there is some c on the line such that $f(c) = \mathfrak{l}(c)$, there are $\binom{p-1}{2}$ possibilities to pick a, b from the line such that $f(a) \neq \mathfrak{l}(a)$ and $f(b) \neq \mathfrak{l}(b)$.

So the colinear triples that are non-chromatic and f-linear account for at most $\binom{p-1}{2}/\binom{p}{3} = 3/p$ percent of all the colinear triples.

Conclude that

$$\tau(f) \geq \mathtt{Pr}[E] - \frac{3}{p} \geq \frac{3\Delta_{ML}(f)\,(1 - \Delta_{ML}(f))}{s} - \frac{3}{p}.$$

Main Theorem. Suppose $p > 20s$. If $\Delta_{ML}(f) \geq \frac{1}{10}$, then $\tau(f) > \frac{1}{10s}$.

Suppose $p \geq 20s$ and $1/10 \leq \Delta_{ML}(f) \leq 9/10$.

In this case the minimal value of $\Delta_{ML}(f)(1 - \Delta_{ML}(f))$ is $9/100$. Using the inequality of Main Lemma one gets that

$$\tau(f) > \frac{1}{9s}. \tag{19}$$

Proof of the Main Theorem

Suppose $p \geq 20s$ and $\Delta_{ML}(f) > 9/10$. We shall prove by induction on s that

$$\tau(f) > \left(1 - \frac{1}{p}\right)^{s-1} \frac{1}{9s}. \tag{20}$$

$s = 1$. In this case $\tau(f) > 1/9$ must be valid, which implies that (20) is valid.

Proof.

If $\tau(f) \leq 1/9$, the probability of a random colinear triple being f-linear would be $8/9$.

By the average principle, there must be two points a, b on a line parallel to some axis such that at least $8/9$ of the colinear triples of the form (a, b, c) on that line are f-linear.

It then follows that the Hemming distance between f and some multilinear function is below $1/9$, contradicting to the assumption $\Delta_{ML}(f) > 9/10$. □

Proof of the Main Theorem

$s > 1$. For any assignment $x_1 = a_1$, the function $f_{x_1=a_1}$ is defined on $\left(\mathbf{F}_p^s\right)^{x_1=a_1}$. A colinear triple is either in $\left(\mathbf{F}_p^s\right)^{x_1=a_1}$ or on a line orthogonal to $\left(\mathbf{F}_p^s\right)^{x_1=a_1}$. Define

$$\tau_{a_1} = \frac{|T'_{a_1}|}{|T_{a_1}|},$$

where T_{a_1} is the set of x_1-colinear triples with one point in $\left(\mathbf{F}_p^s\right)^{x_1=a_1}$, and T'_{a_1} is the subset of T_{a_1} whose elements are not f-linear.

Proof of the Main Theorem

Consider the probabilistic inequalities:

$$\Delta_{ML}(f_{x_1=a_1}) < \frac{1}{10}, \tag{21}$$

$$\tau_{a_1} < \frac{1}{3}. \tag{22}$$

Assume that some $b_1 \neq a_1$ also renders true the following:

$$\Delta_{ML}(f_{x_1=b_1}) < \frac{1}{10}, \tag{23}$$

$$\tau_{b_1} < \frac{1}{3}. \tag{24}$$

We prove that this is impossible.

Proof of the Main Theorem

Suppose $\mathtt{dist}(f_{x_1=a_1}, \mathfrak{l}_{a_1}) < 1/10$ and $\mathtt{dist}(f_{x_1=b_1}, \mathfrak{l}_{b_1}) < 1/10$, where $\mathfrak{l}_{a_1}, \mathfrak{l}_{b_1}$ are multilinear.

Define the multilinear function

$$\begin{aligned}
\mathfrak{l}_{a_1,b_1}(x_1,\ldots,x_s) &= \mathfrak{l}_{a_1}(x_2,\ldots,x_s) + \frac{x_1-a_1}{b_1-a_1}(\mathfrak{l}_{b_1}(x_2,\ldots,x_s) - \mathfrak{l}_{a_1}(x_2,\ldots,x_s)) \\
&= \frac{b_1-x_1}{b_1-a_1}\cdot\mathfrak{l}_{a_1}(x_2,\ldots,x_s) + \frac{x_1-a_1}{b_1-a_1}\cdot\mathfrak{l}_{b_1}(x_2,\ldots,x_s).
\end{aligned}$$

We claim that

$$\mathtt{dist}(f, \mathfrak{l}_{a_1,b_1}) < \frac{9}{10}, \tag{25}$$

which contradicts to $\Delta_{ML}(f) > 9/10$.

Proof of the Main Theorem

We argue that (21), (22), (23) and (24) imply that $\mathtt{dist}(f, \mathfrak{l}_{a_1,b_1}) < \frac{9}{10}$.

Choose randomly two points $r = (r_1, r_2, \ldots, r_s)$ 和 $r' = (r'_1, r_2, \ldots, r_s)$ on a line parallel to x_1-axis.

1. If $r_1 \in \{a_1, b_1\}$, then by (21) and (23), $\mathtt{Pr}[f(r) = \mathfrak{l}_{a_1,b_1}(r)] \geq 9/10$, which implies (25).
2. Suppose $r_1 \notin \{a_1, b_1\}$ and without loss of generality $r'_1 \notin \{a_1, b_1\}$.
    - Suppose the line defined by r and r' intersects with the hyperplane $\left(\mathbf{F}_p^s\right)^{x_1=a_1}$ at r^{a_1} and respectively the hyperplane $\left(\mathbf{F}_p^s\right)^{x_1=b_1}$ at r^{b_1}.
    - By (22)/(24), the probability of $(r^{a_1}, r, r')/(r^{b_1}, r, r')$ being not f-linear is $< \frac{1}{3}$.
    - By (21)/(23), $f_{x_1=a_1}(r^{a_1}) \neq \mathfrak{l}_{a_1}(r^{a_1})/f_{x_1=b_1}(r^{b_1}) \neq \mathfrak{l}_{b_1}(r^{b_1})$ holds with probability $< \frac{1}{10}$.
    - Hence $\mathtt{Pr}[f(r) = \mathfrak{l}_{a_1,b_1}(r)] > 1 - 1/3 - 1/3 - 1/10 - 1/10 > 1/10$. This is (25).

Proof of the Main Theorem

There is at most one a_1 that satisfies both (21) and (22). For all $b_1 \neq a_1$ (virtually all points) we may carry out the following case analysis.

---

1. $1/10 \leq \Delta_{ML}(f_{x_1=b_1}) \leq 9/10$. Fix $x_1 = b_1$. A total of $\frac{1}{p} \cdot \frac{s-1}{s} \cdot |T|$ colinear triples. It follows from induction on (19) that the non-$f_{x_1=b_1}$-colinear triples are bounded in number by
$$\frac{(s-1)|T|}{sp} \cdot \frac{1}{9(s-1)}. \tag{26}$$

2. $\Delta_{ML}(f_{x_1=b_1}) > 9/10$. By induction on (20) one derives that the non-$f_{x_1=b_1}$-linear triples are bounded in number by
$$\frac{(s-1)|T|}{sp} \cdot \left(1 - \frac{1}{p}\right)^{s-2} \frac{1}{9(s-1)}. \tag{27}$$

Proof of the Main Theorem

3 $\Delta_{ML}(f_{x_1=b_1}) < \frac{1}{10}$, and $\tau_{b_1} \geq 1/3$. A hyperplane is orthogonal to one of s axis. An axis is orthogonal to p hyperplanes. On average $\tau_{b_1} T_{b_1}$ is at least

$$\frac{1}{3} \cdot \frac{|T|}{sp}. \tag{28}$$

The number of choices for b_1 is $p-1$. Summarizing (26), (27) and (28),

$$\begin{aligned} \tau(f) &\geq (p-1)\min\left\{\frac{(s-1)}{sp} \cdot \frac{1}{9(s-1)}, \frac{(s-1)}{sp} \cdot \left(1-\frac{1}{p}\right)^{s-2} \frac{1}{9(s-1)}, \frac{1}{3} \cdot \frac{1}{sp}\right\} \\ &\geq \left(1-\frac{1}{p}\right)^{s-1} \cdot \frac{1}{9s}. \end{aligned}$$

Using $p > 20s$ one gets $\tau(f) > (1-\frac{1}{20s})^{s-1} \cdot \frac{1}{9s} > \frac{1}{10s}$.

Algorithm

1. Choose randomly a colinear triple (a, b, c), and query f for $f(a)$, $f(b)$, $f(c)$.
2. If $f(a), f(b), f(c)$ are f-linear, report success.
3. Repeat the above two steps $10s$ times. If none reports failure, accept.

---

1. If f is multilinear, the algorithm always accepts.
2. If $\Delta_{ML}(f) \geq 0.1$, the probability the algorithm refuses is greater than $1/2$.
3. $O(s^2 \log(s))$ long random strings are sufficient.

Parallel Repetition Theorem

Theorem. For any game $\mathfrak{G}$ and $n > 1$, the following is valid:

$$v(\mathfrak{G}^n) \quad \leq \quad \left(1 - \frac{(1 - v(\mathfrak{G}))^3}{6000}\right)^{\frac{n}{\log|\mathfrak{G}|}}.$$

If $v(\mathfrak{G}) \leq 1 - \frac{1}{p}$, then

$$\left(1 - \frac{1}{6000p^3}\right)^{\frac{n}{\log(|\mathfrak{G}|)}} \leq e^{-\frac{6000p^3}{\log|\mathfrak{G}|}\cdot n}.$$

1. R. Raz. A parallel repetition theorem. SIAM J. Comput., 27(3):763–803, 1998. Prelim version STOC ’ 95.
2. T. Holenstein. Parallel Repetition: Simplifications and the No-Signaling Case. Theory of Computing. Volume 5, 141–172, 2009.

IP vs MIP

Is there a hierarchy result

$$\mathbf{IP}[n] \subsetneq \mathbf{IP}[n^2] \subsetneq \mathbf{IP}[n^3] \subsetneq \ldots ?$$

Or is there a collapsing theorem

$$\mathbf{IP} = \mathbf{IP}[n]\,?$$

For **MIP** the issue has been resolved.

1. L. Fortnow, J. Rompel and M. Sipser. On the power of multi-prover interactive protocols. In: Proceedings of the Third Annual Conference on Structure in Complexity Theory, June 1988, pp. 156-161. Also in Theoretical Computer Science, **134**:545-557, 1994.
2. J. Cai, A. Condon and R. Lipton. On bounded round multi-prover interactive proof systems. In Proceedings, Fifth Annual Conference on Structure in Complexity Theory, 45-54, 1990.
3. J. Cai, A. Condon and R. Lipton. PSPACE is provable by two provers in one round. In Proceedings of Structures in Complexity Theory Conference, 1991.
4. J. Kilian. Strong Separation Models of Multi Prover Interactive Proofs, DIMACS Workshop on Cryptography, October 1990.
5. U. Feige. On the success probability of the two provers in one round proof systems. In Proceedings of Structures in Complexity Theory Conference, 1991.
6. D. Lapidot and A. Shamir. A one-round, two-rover, zero-knowledge protocol for NP, in Combinatorica, 15 (1995), pp. 203–214.
7. U. Feige and L. Lovász. Two-prover one-round proof systems, their power and their problems. In STOC 1992, ACM, New York, 733–744, 1992.

Theorem. $\mathbf{MIP} = \mathbf{MIP}[2,1]$.

In retrospect the proof can be greatly simplified using Parallel Repetition Theorem and the idea of Cai, Condon and Lipton.

> *The verifier can ask all questions in one go. How come?*

$(\mathbb{V}, \mathbb{P}_1, \mathbb{P}_2)$

Suppose $(\mathbb{V}, \mathbb{P}_1, \mathbb{P}_2)$ is a polynomial round 2-prover interactive proof system.

1. Recall that $\mathbb{V}$'s questions are random strings.
2. $\mathbb{V}$ may decide at random which prover to interact in the next round.

$x \in L$ iff for any input x of length n the following are valid:

1. If $x \in L$, then $\mathbb{P}_1, \mathbb{P}_2$ exist such that

$$\Pr_{r_1^1,\ldots,r_t^1,r_1^2,\ldots,r_t^2\in\{0,1\}^q}[\mathbb{V}_{\mathbb{P}_1,\mathbb{P}_2}(x, r_1^1 \ldots r_t^1, r_1^2 \ldots r_t^2) = 1] \geq 1 - \frac{1}{2^n}.$$

2. If $x \notin L$, then for any $\mathbb{P}_1, \mathbb{P}_2$,

$$\Pr_{r_1^1,\ldots,r_t^1,r_1^2,\ldots,r_t^2\in\{0,1\}^q}[\mathbb{V}_{\mathbb{P}_1,\mathbb{P}_2}(x, r_1^1 \ldots r_t^1, r_1^2 \ldots r_t^2) = 1] < \frac{1}{2^n}.$$

We define a one round 2-prover interactive system $(\mathbb{V}^*, \mathbb{P}_1^*, \mathbb{P}_2^*)$ that simulates $(\mathbb{V}, \mathbb{P}_1, \mathbb{P}_2)$.

$(\mathbb{V}^*, \mathbb{P}_1^*, \mathbb{P}_2^*)$

Protocol.

1. $\mathbb{V}^*$ sends two random strings $r^1 = r^1_1, \ldots, r^1_t$ and $r^2 = r^2_1, \ldots, r^2_t$ to $\mathbb{P}_1^*$.
2. $\mathbb{P}_1^*$ replies with $a^1 = a^1_1, \ldots, a^1_t$ and $a^2 = a^2_1, \ldots, a^2_t$.
3. $\mathbb{V}^*$ chooses random $s_1, s_2 \in [t]$, and sends $r^1_1, \ldots, r^1_{s_1}$ and $r^2_1, \ldots, r^2_{s_2}$ to $\mathbb{P}_2^*$.
4. $\mathbb{P}_2^*$ replies with $b^1_1, \ldots, b^1_{s_1}$ and $b^2_1, \ldots, b^2_{s_2}$.

$\mathbb{V}^*$ accepts, written $\mathbb{V}^*_{\mathbb{P}_1^*,\mathbb{P}_2^*}(x, r^1, r^2) = 1$, if the following statements are valid.

1. $\mathbb{V}(x, r^1, a^1, r^2, a^2) = 1$.
2. $a^1_1, \ldots, a^1_{s_1} = b^1_1, \ldots, b^1_{s_1}$ and $a^2_1, \ldots, a^2_{s_2} = b^2_1, \ldots, b^2_{s_2}$.

The completeness parameter of $\mathbb{V}^*$ is at least as good as $\mathbb{V}$.

Soundness of $(\mathbb{V}^*, \mathbb{P}_1^*, \mathbb{P}_2^*)$

Suppose $x \notin L$.

The answer space of $\mathbb{P}_2^*$ is $T = \left\{c^1, \ldots, c^s \mid s \in [t] \wedge c^1, \ldots, c^s \in \{0,1\}^q\right\}$.

$$\mathbb{P}_2^*(x, \_, \_) : T^2 \to T^2.$$

The projections are denoted by $\mathbb{P}_2^*(x, \_, \_)_1, \mathbb{P}_2^*(x, \_, \_)_2 : T^2 \to T$.

Soundness of $(\mathbb{V}^*, \mathbb{P}_1^*, \mathbb{P}_2^*)$, plurality functions $M_{r^2}^1, M_{r^1}^2 : T \to T$

$$\begin{aligned} M_{r^2}^1(r) &= \text{the string that occurs most frequently in } \{\mathbb{P}_2^*(x, r, r')_1 \mid r' \text{ is a prefix of } r^2\}, \\ M_{r^1}^2(r) &= \text{the string that occurs most frequently in } \{\mathbb{P}_2^*(x, r', r)_2 \mid r' \text{ is a prefix of } r^1\}. \end{aligned}$$

(r^1, r^2) is rational if one of the followings is valid:

1. $\mathbb{V}(x, r^1, M_{r^2}^1(r^1), r^2, M_{r^1}^2(r^2)) = 0$;
2. Some $s \in [t-1]$ exists such that $M_{r^2}^1(r_1^1, \ldots, r_s^1)$ is not a prefix of $M_{r^2}^1(r_1^1, \ldots, r_{s+1}^1)$;
3. Some $s \in [t-1]$ exists such that $M_{r^1}^2(r_1^2, \ldots, r_s^2)$ is not a prefix of $M_{r^1}^2(r_1^2, \ldots, r_{s+1}^2)$.

Lemma. $\Pr_{r^1, r^2 \in_{\mathrm{R}} \{0,1\}^{qt}}[(r^1, r^2) \text{ is rational }] \geq 1 - 1/2^n$.

there are many rational pairs

Proof.

By definition $M_{r^2}^1$ and $M_{r^1}^2$ are provers. They can simulate $\mathbb{P}_1, \mathbb{P}_2$ but cannot do better. If (r^1, r^2) is not rational, $\mathbb{V}$ accepts x, the probability of which is less than $1/2^n$. □

Soundness of $(\mathbb{V}^*, \mathbb{P}_1^*, \mathbb{P}_2^*)$, more about the rational pairs

Suppose r' is a prefix of r^1 of length g and r'' is a prefix of r^2 of length h.

Abbreviate the answer $\mathbb{P}_2^*(x, r', r'')$ to $\mathbb{P}_2^*(g, h)$.

Consider the $t \times t$ grid. The value of the point (g, h) is $\mathbb{P}_2^*(g, h)$.

Soundness of $(\mathbb{V}^*, \mathbb{P}_1^*, \mathbb{P}_2^*)$

Lemma. If (r^1, r^2) is rational and $\mathbb{V}^*_{\mathbb{P}_1^*,\mathbb{P}_2^*}(x, r^1, r^2) = 1$, then

$$|\{(g, h) \mid \mathbb{P}_2^*(g, h) \text{ is not a prefix of } \mathbb{P}_1^*(x, r^1, r^2)\}| \geq t - 1.$$

Suppose $\mathbb{P}_1^*(x, r^1, r^2) = (a^1, a^2)$. For $g \in [t]$, let $V_g = \{(g, h) \mid h \in [t]\}$. Consider the problem:

Is there any $g \in [t]$ such that $t/2$ points in V_g have a value $\neq M^1_{r^2}(r^1_1, \ldots, r^1_g)$?

If the answer is positive, then at least $t/2$ points are not prefix of $\mathbb{P}_1^*(x, r^1, r^2)$.

Soundness of $(\mathbb{V}^*, \mathbb{P}_1^*, \mathbb{P}_2^*)$

If the answer is negative, then for all $g \in [t]$, at least half of the points in V_g have values whose first components are $M^1_{r^2}(r^1_1, \ldots, r^1_g)$.

Since (r^1, r^2) is rational and $\mathbb{V}^*_{\mathbb{P}_1^*, \mathbb{P}_2^*}(x, r^1, r^2) = 1$, either $M^1_{r^2}(r^1_1, \ldots, r^1_t) \neq a^1$, or $M^1_{r^2}(r^1_1, \ldots, r^1_{g'})$ is not a prefix of $M^1_{r^2}(r^1_1, \ldots, r^1_{g'+1})$ for some $g' \in [t-1]$, and in the latter case either $M^1_{r^2}(r^1_1, \ldots, r^1_{g'})$ or $M^1_{r^2}(r^1_1, \ldots, r^1_{g'+1})$ is not a prefix of a^1.

In summary some g exists such that $M^1_{r^2}(r^1_1, \ldots, r^1_g)$ is not a prefix of a^1.

Conclude that there are at least $t/2$ points in V_g whose values are not prefix of $\mathbb{P}_1^*(x, r^1, r^2)$.

Using the same argument there is a horizontal line H_h containing at least $t/2$ points whose values are not prefix of $\mathbb{P}_1^*(x, r^1, r^2)$.

Since V_g intersects H_h at no more than one point, $|V_g \cup H_h| \geq t - 1$.

Suppose $\mathbb{V}^*$ sends a rational pair (r^1, r^2) to $\mathbb{P}_1^*$.

1. If $\mathbb{V}^*_{\mathbb{P}_1^*,\mathbb{P}_2^*}(x, r^1, r^2) = 0$, then $\mathbb{V}^*$ refuses x.
2. If $\mathbb{V}^*_{\mathbb{P}_1^*,\mathbb{P}_2^*}(x, r^1, r^2) = 1$, by the previous lemma, if $\mathbb{V}^*$ sends to $\mathbb{P}_2^*$ the points in R, then $\mathbb{V}^*$ refuses x, since $\mathbb{P}_2^*(g, h)$ is not a prefix of (a^1, a^2). The probability that this happens is at least $|R|/t^2 \geq (t-1)/t^2$.

The probability that $\mathbb{P}_1^*$ receives a rational pair (r^1, r^2) is at least $1 - 1/2^n$.

So the probability that $\mathbb{V}^*$ refuses is $(1 - \frac{1}{2^n})\frac{t-1}{t^2} - \frac{1}{2^n} > \frac{1}{t^2}$.

Conclusion: $\mathbb{V}^*$ accepts x with probability no more than $1 - \frac{1}{t^2}$.

Parallel Repetition Theorem

Programme Checking

"Checking is concerned with the simpler task of verifying that a given program returns a correct answer on a given input rather than on all inputs. Checking is not as good as verification, but it is easier to do. It is important to note that unlike testing and verification, checking is done each time a program is run."

---

1. M. Blum and S. Kannan. Designing Programs that Check Their Work. J. ACM, 1995.

Checker

A checker for a task T is a P-time probabilistic OTM $\mathbb{C}$ that, given a claimed program P for T and an input x, the following statements are valid:

- If $\forall y.P(y) = T(y)$, then $\Pr[\mathbb{C}^P(x) \text{ accepts } P(x)] \geq \frac{2}{3}$.
- If $P(x) \neq T(x)$, then $\Pr[\mathbb{C}^P(x) \text{ accepts } P(x)] < \frac{1}{3}$.

The checker $\mathbb{C}$ may apply P to a number of randomly chosen inputs before making a decision. So even if $P(x) = T(x)$, the checker may still reject $P(x)$.

Checker for Graph Nonisomorphism

Suppose P is a program for `GNI`:

- $P(G_1, G_2)$ returns 'yes' if $G_1 \ncong G_2$ and 'no' if otherwise.

A program checker $\mathbb{C}$ for `GNI` can be designed as follow:

1. $P(G_1, G_2) =$'no'.
   - Run $P(G_1^1, G_2^1)$, $P(G_1^1, G_2^2)$, ..., $P(G_1^1, G_2^n)$, where G_1^1 is the graph obtained from G_1 by replacing the first node by a complete graph of $n+1$ nodes, ...
   - Accept if an isomorphism is found, and reject otherwise.
2. $P(G_1, G_2) =$'yes'.
   - Run the IP protocol for `GNI` using P as the prover for k times.

Clearly the checker $\mathbb{C}$ runs in P-time.

Checker for Graph Nonisomorphism

Theorem. If P is a correct program for $\texttt{GNI}$, then $\mathbb{C}$ always says "P's answer is correct". If P's answer is incorrect, then the probability that $\mathbb{C}$ says "P's answer is correct" is less than 2^{-k}.

Perfect completeness.

Languages that have Checkers

If L has an interactive proof system where the prover can be efficiently implemented using L as an oracle, then L has a checker.

Theorem. $\mathtt{GI}$, $\sharp\mathtt{SAT}_D$ and $\mathtt{TQBF}$ have checkers.

Random Self-Reducibility

Checkers can be designed by exploring the fact that the output of a program at an input is related to the outputs of the program on some other inputs.

- The simplest such relationship is random self-reducibility.

A problem is randomly self-reducible if solving the problem on any input x can be reduced to solving the problem on a sequence of random inputs $y_1, y_2, \ldots$, where each y_i is uniformly distributed among all inputs.

An Example

Consider a linear function $f(x) = \sum_{i=1}^{n} a_i x_i : \mathrm{GF}(2^n) \to \mathrm{GF}(2^n)$.

- Given any x, pick some y randomly.
- Compute $f(y)$ and $f(y+x)$.
- Compute $f(x)$ by $f(y) + f(y+x)$.

Lipton Theorem

Theorem (Lipton, 1991). There is a randomized algorithm that, given an oracle that computes the permanent on $1 - \frac{1}{3n}$ fraction of the $n \times n$ matrices on $\mathrm{GF}(p)$, can compute the permanents of all matrices on $\mathrm{GF}(p)$ correctly with high probability.

Proof of Lipton Theorem

Let A be an input matrix. Pick a matrix $R \in_{\mathrm{R}} \mathrm{GF}(p)^{n\times n}$. Let

$$B(x) = A + xR.$$

Clearly $\mathtt{perm}(B(x))$ is a degree n univariate polynomial.

For $a \neq 0$, $B(a)$ is a random matrix. So the probability that the oracle computes $\mathtt{perm}(B(a))$ correctly is at least $1 - \frac{1}{3n}$.

Proof of Lipton Theorem

1. Randomly generate $n+1$ distinct nonzero points $a_1, \ldots, a_{n+1}$.
2. Ask the oracle to compute $\mathtt{perm}(B(a_i))$ for all $i \in [n+1]$.
   - According to union bound, with probability at most $\frac{n+1}{3n}$, the oracle may compute at least one of $\mathtt{perm}(B(a_i))$'s incorrectly.
   - So with probability at least $1 - \frac{n+1}{3n} \approx \frac{2}{3}$, the oracle can compute all $\mathtt{perm}(B(a_i))$'s correctly.
3. Finally calculate $\mathtt{perm}(A) = \mathtt{perm}(B(0))$.
   - $\mathtt{perm}(B(x))$ is a univariate polynomial of degree n.
   - Construct the polynomial using interpolation.

Lipton's algorithm provides a checker for the permanent problem.

interaction + randomness + error