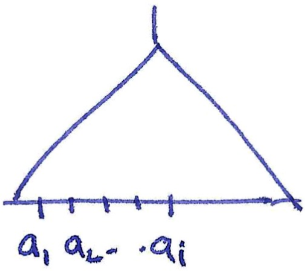
$$\sum_{b_1 \in S_1} \sum_{b_2 \in S_2} \cdots \sum_{b_n \in S_n} g(b_1, \ldots, b_n) \equiv K \pmod p$$

$S_1, S_2 \ldots S_n$ are discrete (small sets) of evaluations

$\overset{\shortparallel}{\{0,1\}}$ $\quad |S_i| = poly(n)$




might NOT be in $S_1, S_2 - S_i$

Suppose (by previous interactions), Verifier needs to check

$$\sum_{b_{i+1} \in S_{i+1}} \sum_{b_{i+2} \in S_{i+2}} \cdots \sum_{b_n \in S_n} g(\underbrace{a_1, \ldots a_i}_{\text{set}}, \underbrace{b_{i+1}, \ldots, b_n}_{\text{free}})$$

$$\equiv K' \pmod p$$

$h_{i+1} : \mathbb{F}_p \to \mathbb{F}_p$

Var. of $h_{i+1}$

$$\underset{h_{i+1}(x)}{\cancel{h_i(x)}} = \sum_{b_{i+2} \in S_{i+2}} \cdots \sum_{b_n \in S_n} g(\underbrace{a_1, \ldots, a_i}_{\text{Previously set}}, x, \underbrace{b_{i+2}, \ldots b_n}_{\text{summed over}})$$

Representation size of $h_{i+1}(x) \leq d \log_2 p = poly(n)$

poly(n) poly(n)

$$\sum_{b_{i+1}\in S_{i+1}} \cdots \sum_{b_n\in S_n} g(a_1,\dots,a_i,b_{i+1},\dots,b_n) \equiv K' \pmod p$$

Need to check

$$\| $$

$$\sum_{x\in S_{i+1}} h_{i+1}(x)$$

Give me $h_{i+1}$ $\longrightarrow$

$\hat{h}^*$ $\longleftarrow$

Verifier

$$\sum_{x\in S_{i+1}} \hat{h}^*(x) \equiv K' \pmod p \,?$$

Is $\hat{h} \equiv h_{i+1}$ ?

Is $\hat{h} - h_{i+1} \equiv 0$ ?

But $\hat{h} - h_{i+1}$ has degree $\leq d$.

If $\hat{h} - h_{i+1} \not\equiv 0$, then $\hat{h}(y) = h_{i+1}(y)$ for at most $d$ values (roots of $\hat{h} - h_{i+1}$).

For at least $p-d$ values $y$, $\hat{h}(y) \neq h_{i+1}(y)$.

Pick $a_{i+1}$ uniformly at random in $\mathbb{F}_p$.

Check if

$$\overbrace{\sum_{b_{i+2}\in S_{i+2}} \cdots \sum_{b_n} g(a_1,\dots,a_i,a_{i+1},b_{i+2},\dots,b_n)}^{\cancel{h_{i+1}(a_{i+1})}\ h_{i+1}(a_{i+1})}$$

$$\equiv \hat{h}(a_{i+1}) \pmod p \ ?$$

like $K'$

Verifier wants to check

set $\longleftrightarrow$ Summed over $\longleftrightarrow$

$$\sum_{b_{i+1} \in S_{i+1}} \cdots \sum_{b_n \in S_n} g(a_1, \ldots a_i, b_{i+1}, \ldots b_n) \equiv K \pmod{p}$$

(If $i = n$, check directly)

V: Give me the univariate polynomial $h_{i+1}$

P: Sends a polynomial $\hat{h}$

V: (1) Check that $\sum_{x \in S_{i+1}} \hat{h}(x) \equiv K \pmod{p}$. If not, reject

(2) Pick $a_{i+1}$ uar in $\mathbb{F}_p$

(3) Recursively check

$$\overbrace{\sum_{b_{i+2} \in S_{i+2}} \cdots \sum_{b_n \in S_n} g(a_1, \ldots a_i, a_{i+1}, b_{i+2}, \ldots b_n)}^{h_{i+1}(a_{i+1})} \equiv \hat{h}(a_{i+1}) \pmod{p}$$

---

Clm: Suppose Verifier has an incorrect statement (at the beginning of an iteration), then either Verifier rejects OR with probability

~~Proof~~: at least $\left(1 - \frac{d}{p}\right)$ over choice of $a_{i+1}$,

Verifier has an incorrect statement to check recursively.

Proof: If prover sends the right polynomial $h_{i+1}$ $(= \hat{h})$,

then $\underbrace{\sum_{x \in S_{i+1}} h_{i+1}(x)}_{= \sum \hat{h}(x)} \not\equiv K \pmod{p}$. Verifier rejects.

Suppose prover sends a polynomial $\hat{h}$ s.t.

$\sum_{x \in S_{i+1}} \hat{h}(x) \equiv K \pmod{p}$. Then $\hat{h} \not\equiv h_{i+1}$. Hence, these polynomials differ in at least ~~d~~ $p-d$ values. With prob $\geq \cancel{1-\frac{p}{d}}\ \left(1-\frac{d}{p}\right)$ over choice of $a_{i+1}$, $\hat{h}(a_{i+1}) \not\equiv h_{i+1}(a_{i+1}) \pmod{p}$.

Thus, the assertion to be checked recursively is false. $\blacksquare$

---

If initial assertion is false,

$$\Pr[\text{rejection}] \geq \left(1-\frac{d}{p}\right)^n \geq e^{-\frac{2dn}{p}} \geq e^{-\frac{1}{2}}$$

$$\left(\text{Assume } \frac{d}{p} \leq 1\right) \qquad \approx 0.6$$

If $p \geq 4dn$

---

Running Time of verifier $= \text{poly}(d, n, \log_2 p)$

$\#SAT \in \mathbb{IP}$

Actually $QBF \in \mathbb{IP}$

QBF is $\mathbb{PSPACE}$-complete, so this proves $\mathbb{IP} = \mathbb{PSPACE}$

$$QBF \Rightarrow \sum_{b_1 \in \{0,1\}} \prod_{b_2 \in \{0,1\}} \sum \prod \cdots g(b_1, \ldots, b_n) \cancel{\equiv K \pmod{p}} \neq 0$$

Naively, degree of this polynomial is exponential.

Linearization trick to reduce degree to polynomials

(We can replace $X_i^2$ by $X_i$.) Multilinear polynomial

---

## $\mathbb{MIP}$ and $\mathbb{PCP}$ theorem

Suppose the verifier could ~~st~~ interact with multiple provers, who cannot communicate with each other (after protocol begins).

This is the class $\mathbb{MIP}$.

<u>Thm</u>: $\mathbb{MIP} \subseteq \mathbb{NEXP}$

<u>Proof sketch</u>: A prover $P$ is a fn $f_P : \{0,1\}^* \to \{0,1\}^*$ s.t. $|f_P(x)| = \text{poly}(|x|)$. For a given size $n$, we can non-deterministically "guess" the function $\underline{f_{P,q(n)}}$ ($f_P$ restricted to $q(n)$ length inputs).

$q(n) \to$ polynomial

$$\text{Representation of } f_{P,q(n)} \leq \exp(q(n)) = 2^{\text{poly}(n)}$$

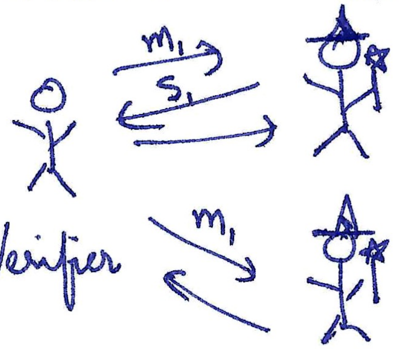
A non-deterministic exponential time machine can guess correct protocol and run it (with randomness) ∎

---

Thm [Babai-Fortnow-Lund 90] : $\mathbb{MIP} = \mathbb{NEXP}$

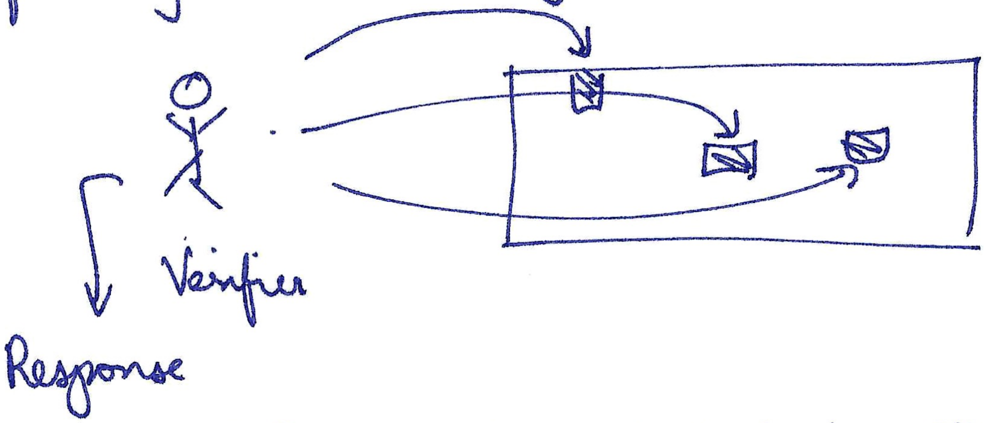
(2 provers are enough [Fortnow-Rompel-Sipser 94])

Think of $\mathbb{MIP}$ as a protocol ~~this~~ that is written down as a truth table (true protocol)




Basically verifier can check if provers responses depend on previous responses.

Second prover is used to check if first prover is following the "true" protocol.




$\mathbb{MIP} \equiv$ Exponential sized certificate into which a polynomial number of (random) queries are made.

Probabilistically Checkable Proof

$\mathbb{PCP}[r, q] \leftarrow$ class of languages decided by $2^r$ sized proof and $q$-query (randomized) verifiers

$$\mathbb{NEXP} = \bigcup_{c \in \mathbb{N}} \mathbb{PCP}[n^c, n^c]$$

$$\mathbb{NP} = \bigcup_{c \in \mathbb{N}} \mathbb{PCP}[\log^c n, \log^c n]$$

Proof of quasipolynomial size $n^{\log^c n}$

Queries are poly(log n)

Thm [AS 92, ALMSS92]:

$$\mathbb{NP} = \mathbb{PCP}[O(\log n), O(1)]$$