Circuits Complexity

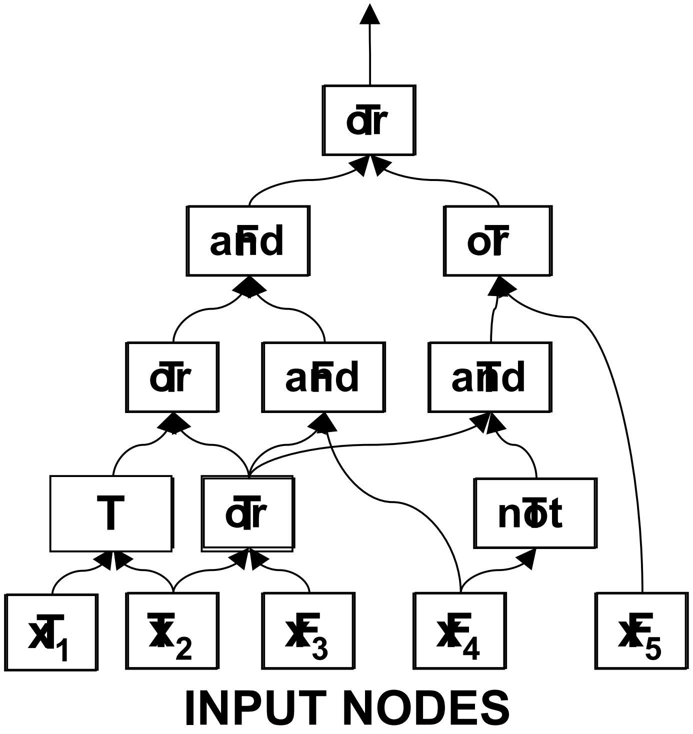
Konstantin Ushakov

JASS'06, St. Petersburg, Russia

Boolean circuit

Boolean circuit C_n:

- finite acyclic directed graph
- each node is labeled as
  - input node (x_i, $1 \leq i \leq n$)
  - logical gate {$\wedge$, $\vee$, $\neg$}
  - “ $\wedge$ ” and “ $\vee$ ” gates have indegree 2
  - “$\neg$” gates have indegree 1
  - at least one output gate



- $S(C_n)$: size of the circuit = number of edges
- $D(C_n)$: depth of the circuit = length of the longest path from input to output (not counting “not” gates)

Circuits properties

- Circuits generation:
  - circuit families must be generated by computer
    - such circuit families can be considered as a good computational model
    - Theorem: a language L $\subseteq \{0, 1\}^*$ has uniform polynomial circuits iff L lies in P
  - circuit families can be described in abstract way
- Circuits properties
  - any Boolean function can be implemented by a circuit
  - any language can be decided by a circuit family of size $O(n2^n)$

Circuits and computers

- **OR**, **AND** and **NOT** can be easily implemented in the chip
- in all computers all operations are implemented using circuits
- once invented the circuit can be placed in the hardware and used forever
- what if we invent a small circuit that solves **SAT** for input of size 1024?

Outline

- **P/poly**
- Circuits and **SAT**
- **Size**[n^k]
- Circuit Complexity of PP

P/poly

- L $\ni$ **P/poly** if there exists $\{C_i\}_{i \ni N}$ and polynomial p:
  - $\forall$ i $|C_i| \leq p(i)$
  - x $\ni$ L iff $C_{|x|}(x) = 1$
- L $\ni$ **P/poly** ←→ there exists a polynomial time computable relation R:

$$\exists\{y_i\}_{i\in\mathrm{N}}\forall x(x\in L \Leftrightarrow R(x, y_{|x|}) = 1)$$

- This two definitions are equivalent by the theorem from the first talk

P, NP and P/poly

- **P** $\subseteq$ **P/poly**
- **P** ≠ **P/poly** (example in the lecture 1)
- **NP** $\subseteq$ **P/poly** ?
- Theorem (Karp-Lipton):
  if **SAT** has polynomial circuits, then the polynomial hierarchy collapses to the second level.
- Theorem (Karp-Lipton):
  if NP has polynomial circuits, then

$$\mathbf{PH} = \Sigma_{\mathbf{2}}\mathbf{P} \cap \Pi_{\mathbf{2}}\mathbf{P}$$

- Theorem (Karp-Lipton):

  NP $\subseteq$ **P/poly** iff there exists a sparse **NP-hard** language in terms of Cook reduction

$\sum_i$ REMINDER

Polynomial Hierarchy:

- i=0: $\prod_0 \mathbf{P} = \sum_0 \mathbf{P} = \Delta_0 \mathbf{P} = \mathbf{P}$
- i>0:
  - $\Delta_{i+1} \mathbf{P} = \mathrm{P}^{\Sigma_i \mathbf{P}}$
  - $\sum_{i+1} \mathbf{P} = \mathrm{NP}^{\Sigma_i \mathbf{P}}$
  - $\prod_{i+1} \mathbf{P} = \mathrm{coNP}^{\Sigma_i \mathbf{P}}$

Cumulative polynomial hierarchy **: PH=** $\bigcup_{i\geq 0} \sum_i \mathbf{P}$

We know:

$\sum_1 \mathbf{P} = \mathbf{NP}$, $\prod_1 \mathbf{P} = \mathbf{coNP}$

Proof plan

Proof.

- We show: $\sum_3$ **P** = $\sum_2$ **P**
- Take L : $\sum_\mathbf{3}$ **P-complete** language

$$L = \{x \mid \exists y \forall z (x, y, z) \in R\},$$

where R is polynomially balanced relation decidable in **NP**

- Why L lies in $\sum_2$ **P**?
- We need to prove that

$$L = \{x | \exists y \forall z (x,y,z) \in Q\},$$

where Q is polynomially balanced relation decidable in **P**

Proof: our knowledge

- L : $\sum_3$ **P-complete** language:

  $$L = \{x \mid \exists y \forall z (x, y, z) \in R\},$$

  where R is polynomially balanced relation decidable in **NP**

What we know:

- R lies in **NP** → it can be reduced to **SAT** (**NP-complete**):
  - F is a reduction
  - R(x, y, z) ←→ F(x, y, z) is satisfiable
- **SAT** has a polynomial circuit
  - **C** = (C_0, ….) **:** polynomial circuits that solves **SAT**
  - $\mathbf{C_n}$ = (C_0, …., C_n) **:** initial segment of length n
  - $\mathbf{C_m}$ is a correct initial segment iff $\boldsymbol{C}_m$ correctly decides **SAT** for formulas of size <=m

Proof: correct initial segment

- Self-reducibility of **SAT:**

  for every formula **G** and for every variable $\boldsymbol{x}$:

  $\mathbf{G} = \mathbf{G}[x := \mathbf{true}]$ or $\mathbf{G}[x := \mathbf{false}]$
- $\boldsymbol{w}$ – Boolean formula:

  Test($\boldsymbol{C_n}$, $\boldsymbol{w}$):
  - $\boldsymbol{w}$ has variable x:

    $C_{|w|}(w) =$

    $= C_{|w[x:=\mathbf{true}]|}(w[x := \mathbf{true}])$ or $C_{|w[x:=\mathbf{false}]|}(w[x := \mathbf{false}])$
  - $C_{|w|}(\mathbf{true}) = \mathbf{true}$
  - $C_{|w|}(\mathbf{false}) = \mathbf{false}$
- $\boldsymbol{C_n}$ – correct initial segment if and only if

$$\forall w\ (|w| \leq n)\ \text{Test}(C_n, w)$$

Proof: gathering ideas

- We prove:
  x is in L iff $\exists \boldsymbol{C}_m \exists y \forall z$ (all of length at most m) :
  - $\boldsymbol{C_m}$ (F(x, y, z)) = **true**
  - $\mathbf{C_m}$ is a correct initial segment of length m
- What m should we take?
  - x : $\exists$ **p:** $\forall y \forall z(|F(x,y,z)| < \mathbf{p}(|x|))$:
    - F is a reduction from R to **SAT**
    - R is polynomially balanced
    - F is a polynomial
    - → **p** is a polynomial
  - $m=\mathbf{p}(|x|)$

Proof ideas: finish

- We prove: x is in L ←→ $\exists$ ***C***$_m$ $\exists$ y : $\forall$ z $\forall$ *w* (all of length at most **p**(|x|))
  - ***C***$_m$ works correct on ***w***
  - ***C$_m$*** (F(x, y, z)) = **true**

$x \in L$

$\Rightarrow \exists y \forall z R(x,y,z)$

$\Rightarrow \exists y \forall z F(x,y,z) \in SAT$

$\Rightarrow \exists \{C_i\}_{i=1}^{m}$ – correct initial segment

C_m – correct initial segment

$\Rightarrow F(x,y,z) \in SAT$

$\Rightarrow \exists y \forall z R(x,y,z)$

$\Rightarrow x \in L$

- Reminder: if R is polynomially balanced, polynomial-time decidable, then

$$L = \{x \mid \exists y_1 \forall y_2 : (x, y_1, y_2) \in R\} \in \Sigma_2 P$$

□

Second Theorem

- Theorem (Karp-Lipton):

  if **SAT** has polynomial circuits, then the polynomial hierarchy collapses to the second level.

- **Corollary**:

  if NP has polynomial circuits, then $\mathbf{PH} = \sum_{\mathbf{2}} \mathbf{P} \cap \prod_{\mathbf{2}} \mathbf{P}$

  Proof: PH is closed under complement.

□

TODO

- Theorem (Karp-Lipton):
  if **SAT** has polynomial circuits, then the polynomial hierarchy collapses to the second level.
- Theorem (Karp-Lipton):
  if NP has polynomial circuits, then

$$\mathrm{PH} = \Sigma_2\mathrm{P} \cap \Pi_2\mathrm{P}$$

- Theorem (Karp-Lipton):
  NP $\subseteq$ **P/poly** iff there exists a sparse **NP-hard** language in terms of Cook reduction

Size[n^k]

- **Size**[$f(n)$] **:** class of languages accepted by Boolean circuit families of size $O(f(n))$
- **Size**[n^k] **:** class of languages accepted by Boolean circuit families of size $O(n^k)$

- **Lemma:** $\sum_4$ **P** □ **Size**[n^k] for any k
  Proof: later…

- Corollary 1: **PH** □ **Size**[n^k]

- **NB**: it does not follow that $\sum_4$ **P** □ **P/poly**:
  Why?
  Size[poly(n)] (the union of **Size**[n^k] over all k) equals **P/poly**

$\sum_2$ **P** ∩ $\prod_2$ **P** □ **Size**[n^k]

Reminder: PH □ **Size**[n^k]:

Theorem: $\sum_2$ **P** ∩ $\prod_2$ **P** □ **Size**[n^k] for any k

Proof:

assume: $\sum_2$ **P** ∩ $\prod_2$ **P** ⊆ **Size**[n^k] for some k

→ there exists a polynomial circuit that accepts **NP**

→ the polynomial hierarchy collapses on $\sum_2$ **P** ∩ $\prod_2$ **P**

→ **PH** = $\sum_2$ **P** ∩ $\prod_2$ **P** ⊆ **Size**[n^k] ?!

□

Proof of the lemma

__**Lemma:**__ $\sum_4$ **P** □ **Size**[n^k] for any k

Proof:

- f **:** function that depends only on the first c*k*log(n) bits of input
  - such function can be encoded by polynomial number of bits
  - number of possible f functions is $2^{2^{c*k*\log(n)}} = 2^{n^{c*k}}$
- number of possible circuits of size n^k is at most $2^{n^k/2 + n}$
- **M**={ f | ∀c (circuit of size n^k) ∃x (input of length n): f(x) ≠ c(x)} ($2^{n^{c*k}} > 2^{n^k/2 + n}$ → **M** is not empty)
- let "≤" be any order on **M** (for instance lexicographical order)
- ***f*** is the smallest function in **M**
- L = {x | ***f***(x) = 1}

Proof of the lemma

- L = {x | ***f***(x) = 1}

$$y \in L \Leftrightarrow \begin{cases} f(y) = 1 \\ \forall c \exists x : f(x) \neq c(x) \\ \forall f' : (\forall c \exists x : (f'(x) \neq c(x)) \Rightarrow f \leq f' \end{cases}$$

- rewriting:

$$y \in L \Leftrightarrow \exists f \, \forall c \forall f' \exists x \exists c' \forall x' :$$
$$f(x) \neq c(x) \wedge ((f \leq f') \vee f'(x') = c'(x')) \wedge f(y) = 1$$

- L is from $\sum_4$ **P** and it can’t be accepted by a circuit of size n^k

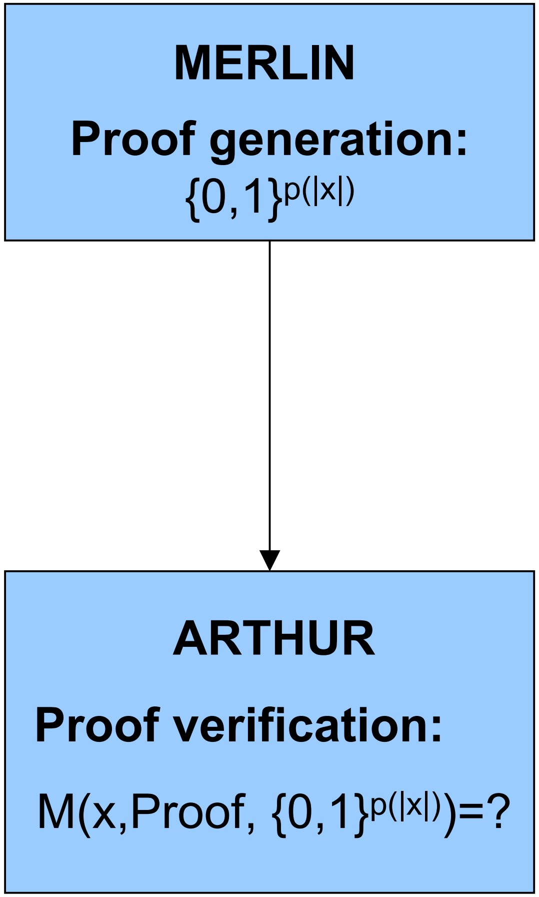
Proof's bugs

- What is wrong with the proof?
- Lemma: $\sum_4$ **P** □ **Size**[n^k] for any k
- What we proved: L is from $\sum_4$ **P** and it can't be accepted by a circuit of size n^k
- Proof completion:
  - Take a circuit **c** of size $C*n^{k-1}$
  - $\exists$ n_0: $C*n_0^{k-1} < n_0^{k.}$
  - $\exists$ x(|x|=n_0): on input x **c** works incorrect
  - L□ **Size**[n^{k-1}]

□

MA protocol

- **MA protocols**: L ∋ **MA** if there exist polynomials p and q and Turing machine M, working polynomial time on all inputs, that for every x:
  - **x** is from L ←→ Merlin can think of a proof : Arthur will accept is with high probability
  - **x** is not from L ←→ every proof created by Merlin will be rejected with high probability



MERLIN

Proof generation:
$\{0,1\}^{p(|x|)}$

ARTHUR

Proof verification:

$M(x,Proof, \{0,1\}^{p(|x|)})=?$

REMINDER

- **MA protocols**: L $\ni$ **MA** if there exist polynomials p and q and Turing machine M, working polynomial time on all inputs, that for every x:

$$x \in L \Rightarrow \exists y \in \{0,1\}^{p(|x|)} : \Pr_{z \in \{0,1\}^{q(|x|)}} \{M(x,y,z)=1\} > 3/4,$$

$$x \notin L \Rightarrow \forall y \in \{0,1\}^{p(|x|)} : \Pr_{z \in \{0,1\}^{q(|x|)}} \{M(x,y,z)=1\} < 1/4$$

- Toda Theorem: **PH** $\subseteq$ **P$^{\mathbf{PP}}$**
- **P$^{\#\mathbf{P}}$** = {f: ∑*→**N** ∪ {0} | ∃ time polynomial **NTM** M_f such that for every x f(x)=$acc_{M_f}(x)$}, where $acc_{M_f}(x)$ is the number of ACCEPT paths of machine M_f.
- **P$^{\mathbf{PP}}$= P$^{\#\mathbf{P}}$:** lemma in the proof of Toda's theorem
- **P$^{\#\mathbf{P}}$** has interactive protocol with prover from **P$^{\#\mathbf{P}}$**

Circuit complexity of PP

- Lemma 1: if **PP** ⊆ **P/poly** we have $\mathbf{P}^{\mathbf{PP}}$ ⊆ **MA**.
  Proof: later.

- Lemma 2: **MA** ⊆ **PP**.
  Proof: lection 7.

- Theorem: **PP** □ **Size**$[n^k]$ for every k.

Proof: **PP** □ **Size**[n^k] for every k

REMINDER:

- Toda Theorem: **PH** ⊆ **P**$^{\mathbf{PP}}$
- Lemma 1: if **PP** ⊆ **P/poly** we have **P**$^{\mathbf{PP}}$ ⊆ **MA**
- Lemma 2: **MA** ⊆ **PP**.

Proof:

- Assume k: **PP** ⊆ **Size**[n^k] → PP ⊆ **P/poly**
- **PH** ⊆ **P**$^{\mathbf{PP}}$ (by Toda theorem)
  ⊆ **MA** (Lemma 1)
  ⊆ **PP** (Lemma 2)
- **We know: PH** □ **Size**[n^k]
- → **PP** □ **Size**[n^k]

□

PP $\subseteq$ **P/poly** → $\mathbf{P}^{\mathbf{PP}}$ $\subseteq$ **MA.**

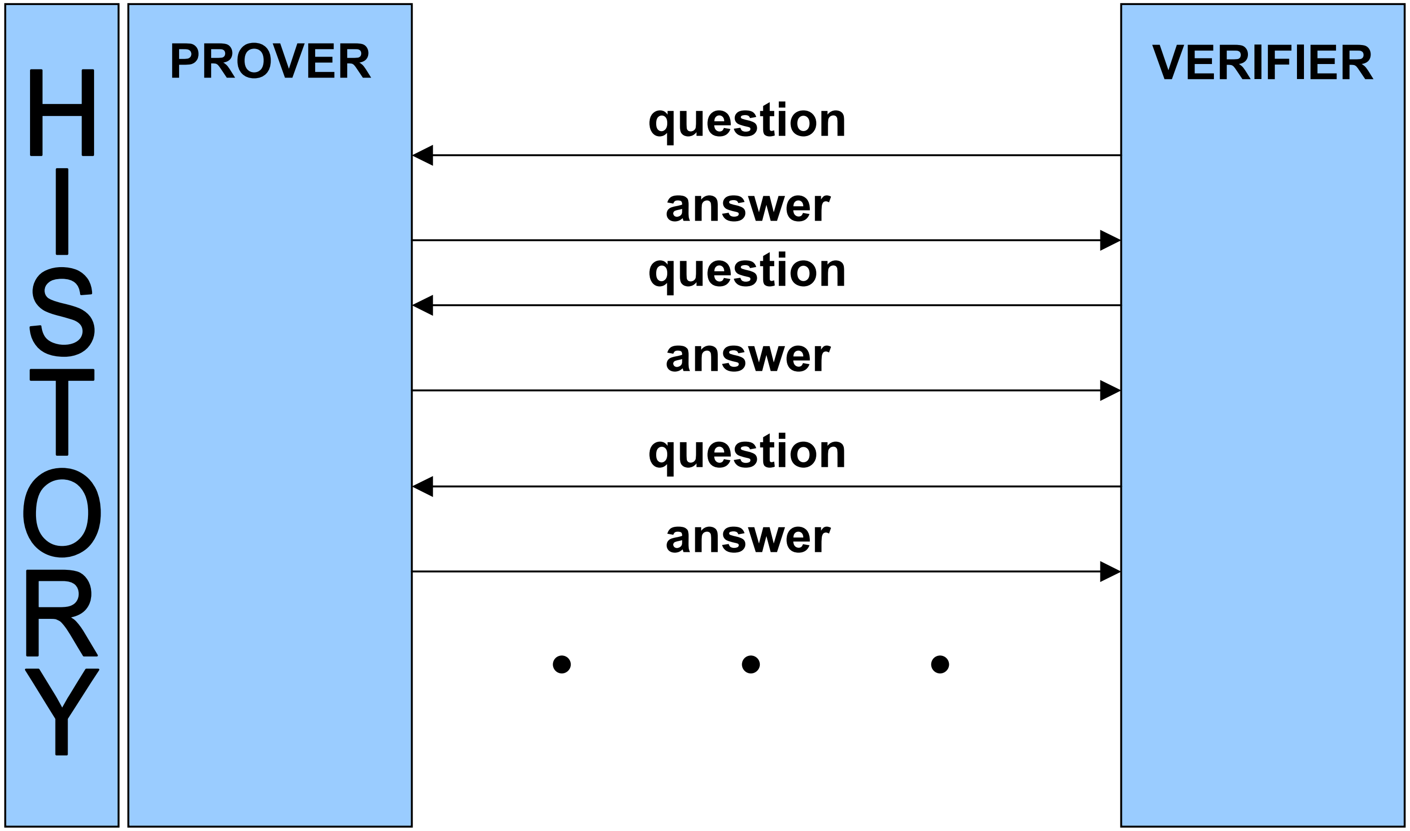
Lemma 1: **PP** $\subseteq$ **P/poly** → $\mathbf{P}^{\mathbf{PP}}$ $\subseteq$ **MA**.

Proof:

- Take M **:** polynomial time oracle Turing machine from $\mathbf{P}^{\mathbf{PP}}$
- M **:** asks questions to oracle from **PP** of at most polynomial length
- $\mathbf{P}^{\mathbf{\#P}}$ = $\mathbf{P}^{\mathbf{PP}}$ $\subseteq$ **P/poly :**
  - **PP** has polynomial circuits
  - this circuits can be considered as a **hint** string for machine from **P/poly**
- $\mathbf{P}^{\mathbf{\#P}}$ has interactive protocol with prover from $\mathbf{P}^{\mathbf{\#P}}$
- we modify the protocol:
  - prover does not remember communication history
  - verifier sends communication history with every request to the prover
- now the prover acts as a simple $\mathbf{P}^{\mathbf{\#P}}$ machine

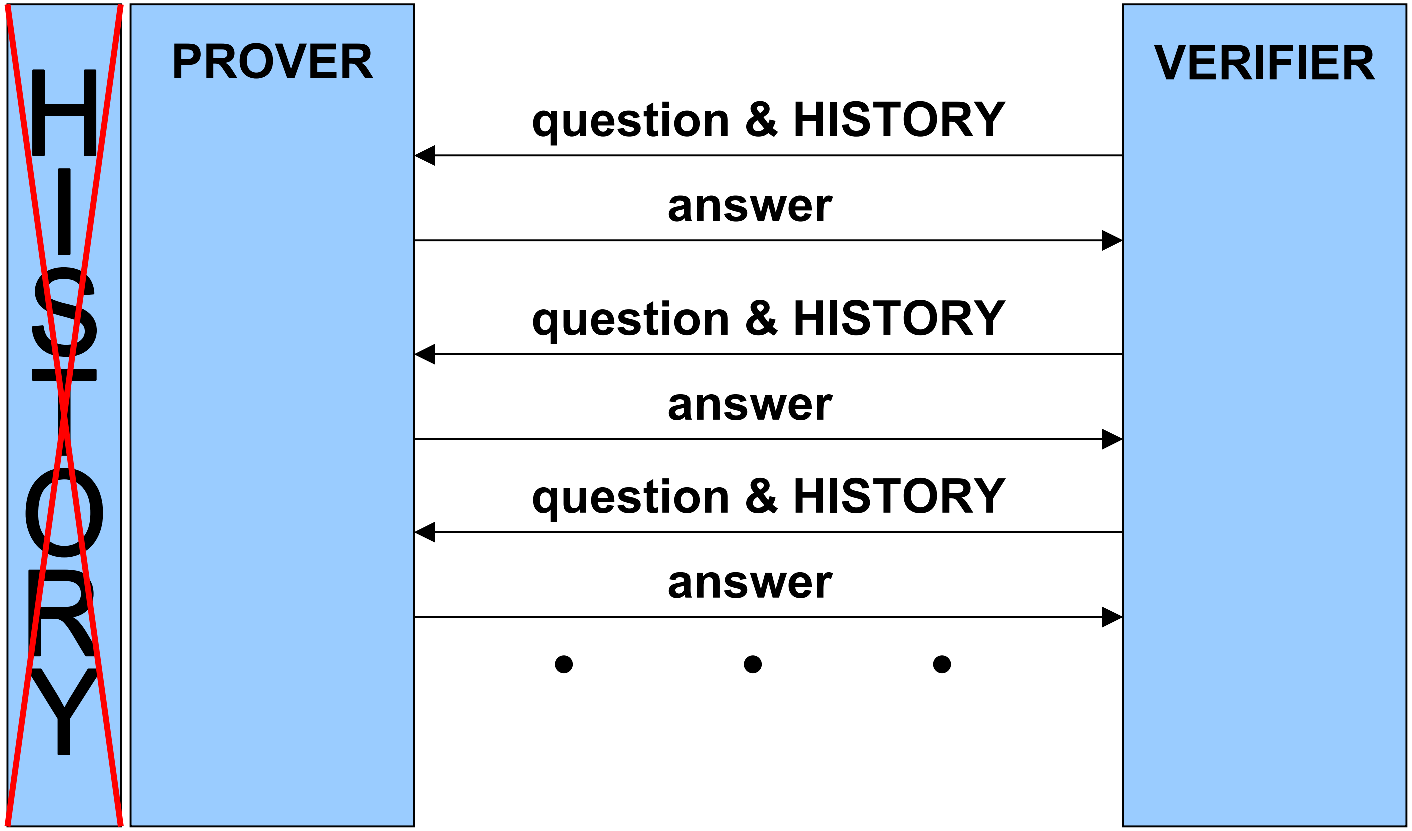
$PP \subseteq P/poly \rightarrow P^{PP} \subseteq MA$.

$P^{\#P}$ has interactive protocol with prover from $P^{\#P}$



$PP \subseteq P/poly \rightarrow P^{PP} \subseteq MA$.

MODIFIED PROTOCOL



Lemma's Proof

- We modified the prover → it acts like a simple $\mathbf{P^{\#P}}$ machine
- We know: $\mathbf{P^{\#P}} = \mathbf{P^{PP}} \subseteq \mathbf{P/poly}$
- **MA** protocol modifications
  - Arthur simulates verifier
  - instead of calling the prover Arthur uses circuits sent by the prover in the beginning of the communications
- all requests of the verifier have length poly(n) → circuits are the valid replacement for the prover
- $\mathbf{P^{\#P}} \subseteq \mathbf{P/poly} \rightarrow \mathbf{P^{PP}} = \mathbf{P^{\#P}} \subseteq \mathbf{MA}$

□

Conclusion

- P/poly & **Size**[n^k]
- P/poly as a computational model

- **SAT** has polynomial circuit → PH collapses on the second level

- **PP** □ **Size**[n^k] for every k.

Thanks for the Patience

QUESTIONS TIME