

# Circuits Complexity

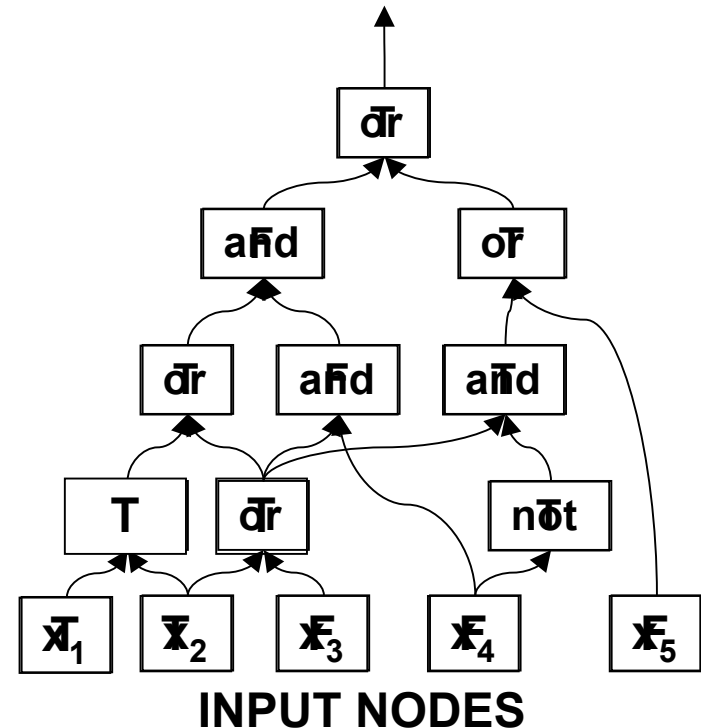
Konstantin Ushakov

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

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

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

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

# P, NP and P/poly

- **P**  $\subseteq$  **P/poly**
- **P**  $\neq$  **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

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

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

# $\Sigma_i$ REMINDER

## Polynomial Hierarchy:

- $i=0$ :  $\Pi_0 \mathbf{P} = \Sigma_0 \mathbf{P} = \Delta_0 \mathbf{P} = \mathbf{P}$
- $i>0$ :
  - $\Delta_{i+1} \mathbf{P} = \mathbf{P}\Sigma_i \mathbf{P}$
  - $\Sigma_{i+1} \mathbf{P} = \mathbf{NP}\Sigma_i \mathbf{P}$
  - $\Pi_{i+1} \mathbf{P} = \mathbf{coNP}\Sigma_i \mathbf{P}$

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

**We know:**

$$\Sigma_1 \mathbf{P} = \mathbf{NP}, \quad \Pi_1 \mathbf{P} = \mathbf{coNP}$$

# Proof plan

Proof.

- We show:  $\Sigma_3 \mathbf{P} = \Sigma_2 \mathbf{P}$
- Take  $L : \Sigma_3 \mathbf{P}$ -**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  $\Sigma_2 \mathbf{P}$ ?
- We need to prove that

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

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

# Proof: our knowledge

- $L : \Sigma_3$  **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**  $\rightarrow$  it can be reduced to **SAT** (**NP-complete**):
  - $F$  is a reduction
  - $R(x, y, z) \leftrightarrow F(x, y, z)$  is satisfiable
- **SAT** has a polynomial circuit
  - $\mathbf{C} = (C_0, \dots)$  : polynomial circuits that solves **SAT**
  - $\mathbf{C}_n = (C_0, \dots, C_n)$  : initial segment of length  $n$
  - $\mathbf{C}_m$  is a **correct initial segment** iff  $\mathbf{C}_m$  **correctly decides SAT** for formulas of size  $\leq m$

# Proof: correct initial segment

- Self-reducibility of **SAT**:

for every formula **G** and for every variable **x**:

$$\mathbf{G} = \mathbf{G}[\mathbf{x} := \mathbf{true}] \text{ or } \mathbf{G}[\mathbf{x} := \mathbf{false}]$$

- **w** – Boolean formula:

**Test**(**C<sub>n</sub>**, **w**):

- **w** has variable **x**:

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

$$= C_{|w[x:=\mathbf{true}]|}(w[x := \mathbf{true}]) \text{ or } C_{|w[x:=\mathbf{false}]|}(w[x := \mathbf{false}])$$

- $C_{|w|}(\mathbf{true}) = \mathbf{true}$

- $C_{|w|}(\mathbf{false}) = \mathbf{false}$

- **C<sub>n</sub>** – correct initial segment if and only if

$$\forall w (|w| \leq n) \text{ Test}(\mathbf{C}_n, w)$$

# Proof: gathering ideas

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

# Proof ideas: finish

- We prove:  $x$  is in  $L \iff \exists \mathbf{C}_m \exists y : \forall z \forall w$  (all of length at most  $p(|x|)$ )

- $\mathbf{C}_m$  works correct on  $w$

- $\mathbf{C}_m(F(x, y, z)) = \mathbf{true}$

$x \in L$

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

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

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

$C_m$  – correct initial segment

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

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

$\Rightarrow x \in L$

- **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  $\mathbf{PH} = \Sigma_2 \mathbf{P} \cap \Pi_2 \mathbf{P}$   
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

$$PH = \Sigma_2 P \cap \Pi_2 P$$

- 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:**  $\sum_4 \mathbf{P} \subseteq \mathbf{Size}[n^k]$  for any  $k$   
**Proof:** later...
- **Corollary 1:**  $\mathbf{PH} \subseteq \mathbf{Size}[n^k]$
- **NB:** it does not follow that  $\sum_4 \mathbf{P} \subseteq \mathbf{P/poly}$ :  
**Why?**  
**Size[poly( $n$ )]** (the union of **Size[ $n^k$ ]** over all  $k$ ) equals **P/poly**

# $\Sigma_2 \mathbf{P} \cap \Pi_2 \mathbf{P} \sqsubseteq \mathbf{Size}[n^k]$

Reminder:  $\mathbf{PH} \sqsubseteq \mathbf{Size}[n^k]$ :

Theorem:  $\Sigma_2 \mathbf{P} \cap \Pi_2 \mathbf{P} \sqsubseteq \mathbf{Size}[n^k]$  for any  $k$

**Proof:**

assume:  $\Sigma_2 \mathbf{P} \cap \Pi_2 \mathbf{P} \subseteq \mathbf{Size}[n^k]$  for some  $k$

→ there exists a polynomial circuit that accepts  $\mathbf{NP}$

→ the polynomial hierarchy collapses on  $\Sigma_2 \mathbf{P} \cap \Pi_2 \mathbf{P}$

→  $\mathbf{PH} = \Sigma_2 \mathbf{P} \cap \Pi_2 \mathbf{P} \subseteq \mathbf{Size}[n^k]$  ?!

□

# Proof of the lemma

**Lemma:**  $\Sigma_4 \text{P} \sqsubseteq \text{Size}[n^k]$  for any  $k$

**Proof:**

- $f$  : function that depends only on the first  $c \cdot k \cdot \log(n)$  bits of input
  - such function can be encoded by polynomial number of bits
  - number of possible  $f$  functions is  $2^{2^{c \cdot k \cdot \log(n)}} = 2^{n^{c \cdot k}}$
- number of possible circuits of size  $n^k$  is at most  $2^{n^{k/2} + n}$
- $\mathbf{M} = \{ f \mid \forall c \text{ (circuit of size } n^k) \exists x \text{ (input of length } n): f(x) \neq c(x) \}$   
( $2^{n^{c \cdot k}} > 2^{n^{k/2} + n} \rightarrow \mathbf{M}$  is not empty)
- let “ $\leq$ ” be any order on  $\mathbf{M}$  (for instance lexicographical order)
- $f$  is the smallest function in  $\mathbf{M}$
- $L = \{ x \mid f(x) = 1 \}$

# Proof of the lemma

- $L = \{x \mid f(x) = 1\}$

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

- 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  $\Sigma_4^P$  and it can't be accepted by a circuit of size  $n^k$

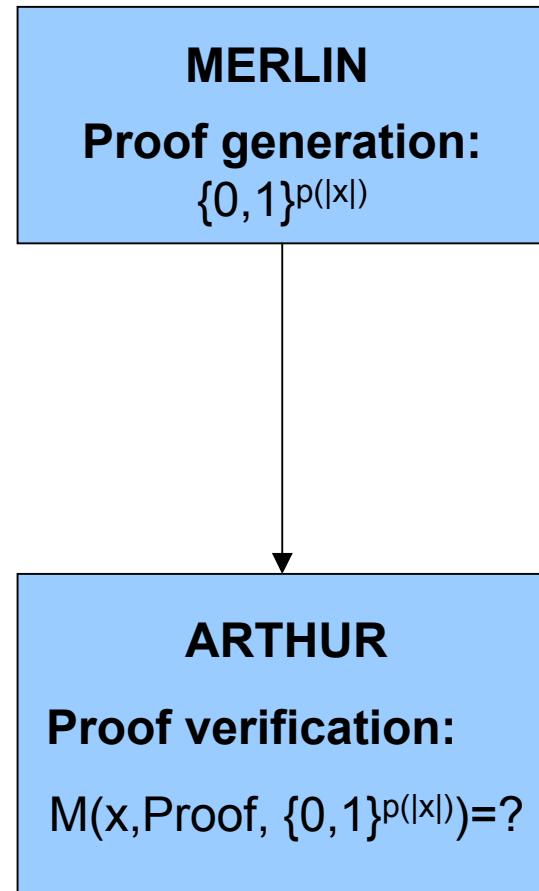
# Proof's bugs

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

□

# MA protocol

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



# REMINDER

- **MA protocols:**  $L \ni \mathbf{MA}$  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:  $\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{PP}}$
- $\mathbf{P}^{\#\mathbf{P}} = \{f: \Sigma^* \rightarrow \mathbf{N} \cup \{0\} \mid \exists \text{ time polynomial NTM } M_f \text{ such that for every } x f(x) = \text{acc}_{M_f}(x)\}$ , where  $\text{acc}_{M_f}(x)$  is the number of ACCEPT paths of machine  $M_f$ .
- $\mathbf{P}^{\mathbf{PP}} = \mathbf{P}^{\#\mathbf{P}}$  : lemma in the proof of Toda's theorem
- $\mathbf{P}^{\#\mathbf{P}}$  has interactive protocol with prover from  $\mathbf{P}^{\#\mathbf{P}}$

# Circuit complexity of PP

- Lemma 1: if  $\mathbf{PP} \subseteq \mathbf{P/poly}$  we have  $\mathbf{P}^{\mathbf{PP}} \subseteq \mathbf{MA}$ .  
Proof: later.
- Lemma 2:  $\mathbf{MA} \subseteq \mathbf{PP}$ .  
Proof: lection 7.
- Theorem:  $\mathbf{PP} \not\subseteq \mathbf{Size}[n^k]$  for every  $k$ .

# Proof: $\text{PP} \subseteq \text{Size}[n^k]$ for every $k$

## REMINDER:

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

## Proof:

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

□

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

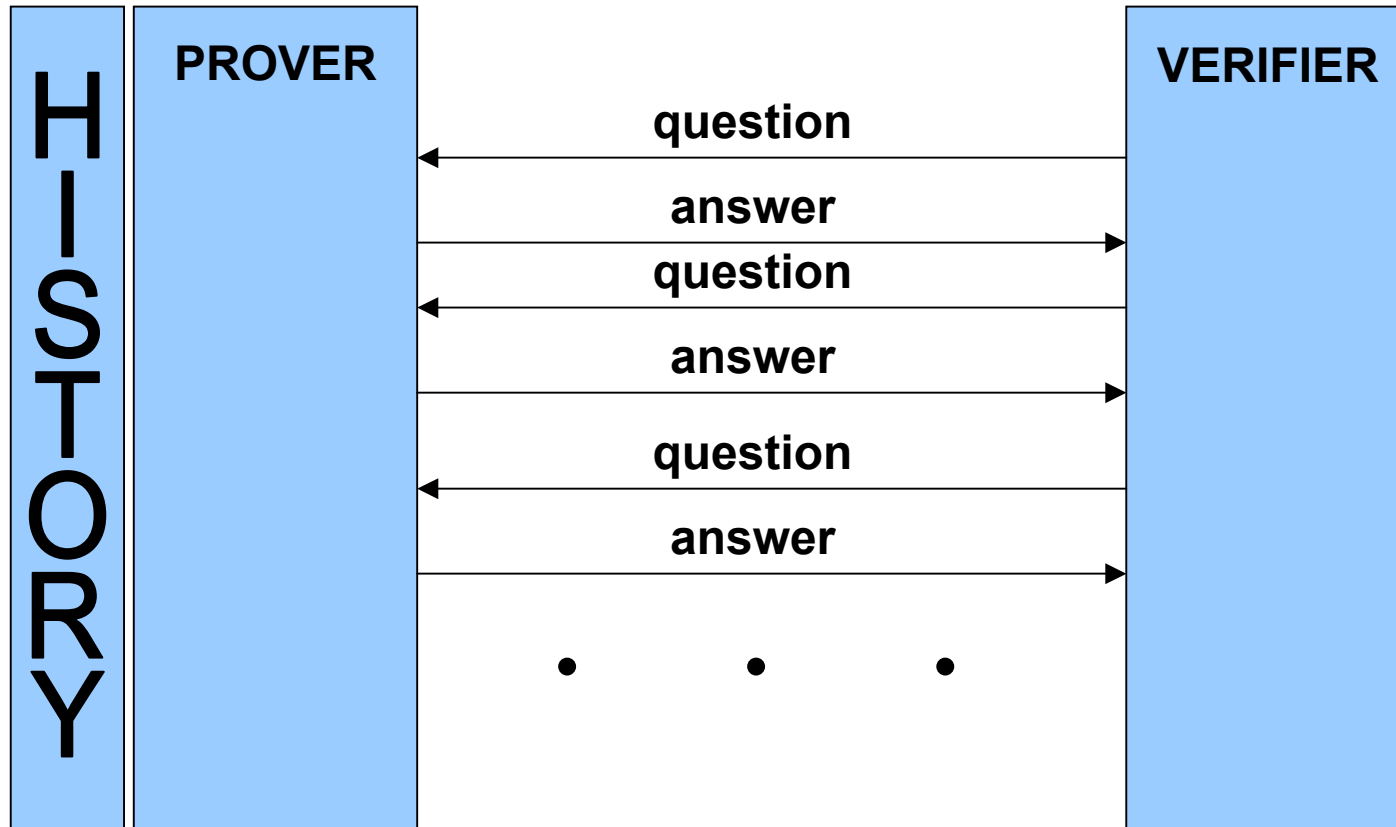
Lemma 1:  $PP \subseteq P/poly \rightarrow P^{PP} \subseteq MA.$

Proof:

- Take  $M$  : polynomial time oracle Turing machine from  $P^{PP}$
- $M$  : asks questions to oracle from  $PP$  of at most polynomial length
- $P^{\#P} = P^{PP} \subseteq P/poly$  :
  - $PP$  has polynomial circuits
  - this circuits can be considered as a **hint** string for machine from  $P/poly$
- $P^{\#P}$  has interactive protocol with prover from  $P^{\#P}$
- 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  $P^{\#P}$  machine

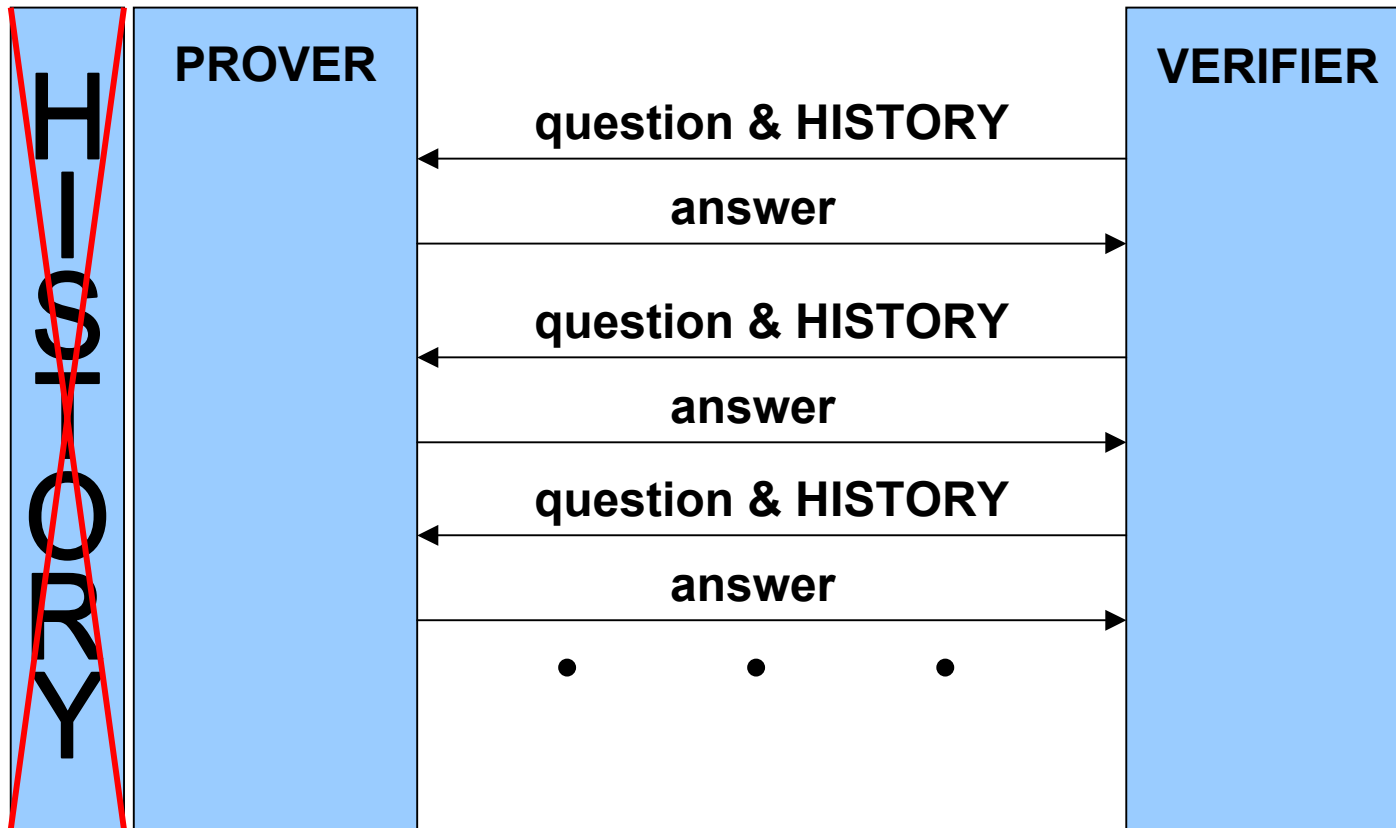
$PP \subseteq P/poly \rightarrow P^{PP} \subseteq MA.$

$P^{\#P}$  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  $\rightarrow$  it acts like a simple  $\mathbf{P}^{\#P}$  machine
- We know:  $\mathbf{P}^{\#P} = \mathbf{P}^{PP} \subseteq \mathbf{P}/\text{poly}$
- **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  $\text{poly}(n) \rightarrow$  circuits are the valid replacement for the prover
- $\mathbf{P}^{\#P} \subseteq \mathbf{P}/\text{poly} \rightarrow \mathbf{P}^{PP} = \mathbf{P}^{\#P} \subseteq \mathbf{MA}$

□

# Conclusion

- P/poly & **Size** $[n^k]$
- P/poly as a computational model
- **SAT** has polynomial circuit  $\rightarrow$  PH collapses on the second level
- **PP**  $\not\subseteq$  **Size** $[n^k]$  for every  $k$ .

Thanks for the Patience

QUESTIONS TIME