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 \{ $\land$, $\lor$, $\neg$ \}
  - “$\land$” and “$\lor$” 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 \supseteq P/poly** if there exists \( \{C_i\}_{i \in \mathbb{N}} \) and polynomial \( p \):
  - \( \forall i \ |C_i| \leq p(i) \)
  - \( x \supseteq L \iff C_{|x|}(x) = 1 \)
- **L \supseteq 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

\[
PH = \sum_2 P \cap \Pi_2 P
\]

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

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

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

We know:

$\sum_1 P = NP$, $\prod_1 P = coNP$
Proof plan

Proof.

- We show: $\sum_3 P = \sum_2 P$
- Take $L : \sum_3 P$-complete language

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

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

- Why $L$ lies in $\sum_2 P$?
- 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:
  
  \[
  L = \{ x | \exists y \forall z (x, y, z) \in R \},
  \]
  
  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
  - \( C = (C_0, \ldots) \) : polynomial circuits that solves SAT
  - \( C_n = (C_0, \ldots, C_n) \) : initial segment of length \( n \)
  - \( C_m \) is a correct initial segment iff \( C_m \) 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 $x$: $G = G[x := \text{true}]$ or $G[x := \text{false}]$

- $w$ – Boolean formula:
  Test($C_n$, $w$):
  - $w$ has variable $x$:
    $C_{|w|}(w) = C_{|w[x:=\text{true}]|}(w[x := \text{true}])$ or $C_{|w[x:=\text{false}]|}(w[x := \text{false}])$
  - $C_{|w|}(\text{true}) = \text{true}$
  - $C_{|w|}(\text{false}) = \text{false}$

- $C_n$ – 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 $\exists C_m \exists y \forall z$ (all of length at most m):

- $C_m(F(x, y, z)) = true$
- $C_m$ is a correct initial segment of length m

What m should we take?

- $x : \exists p: \forall y \forall z(|F(x,y,z)| < p(|x|))$:  
  - F is a reduction from R to SAT
  - R is polynomially balanced
  - F is a polynomial
  - $\rightarrow p$ is a polynomial
- $m = p(|x|)$
Proof ideas: finish

- We prove: $x$ is in $L \iff \exists C_m \exists y : \forall z \forall w$ (all of length at most $p(|x|))$
  - $C_m$ works correct on $w$
  - $C_m (F(x, y, z)) = \text{true}$

$x \in L$

\[ \Rightarrow \exists y \forall z R(x, y, z) \]
\[ \Rightarrow \exists y \forall z F(x, y, z) \in \text{SAT} \]
\[ \Rightarrow \exists \{C_i\}_{i=1}^m \text{– correct initial segment} \]

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

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

\[ \square \]
Second Theorem

- **Theorem (Karp-Lipton):**
  - If \textbf{SAT} has polynomial circuits, then the polynomial hierarchy collapses to the second level.

- **Corollary:**
  - If \textbf{NP} has polynomial circuits, then \( \text{PH} = \Sigma_2 P \cap \Pi_2 P \)
  - Proof: \( \text{PH} \) is closed under complement.
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

\[
\text{PH} = \Sigma_2^P \cap \Pi_2^P
\]

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:** $\sum_4 P$ 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
$\Sigma_2 \text{P} \cap \Pi_2 \text{P } \text{Size}[n^k]$

**Reminder:** PH $\text{Size}[n^k]$:  

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

**Proof:**

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

$\rightarrow$ there exists a polynomial circuit that accepts NP

$\rightarrow$ the polynomial hierarchy collapses on $\Sigma_2 \text{P} \cap \Pi_2 \text{P}$

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

$\square$
Proof of the lemma

**Lemma:** \( \sum_4 \mathbb{P} \text{ Size}[n^k] \) 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 \( 2^{2c^*k^*\log(n)} = 2^{n^{c^*k}} \)
- number of possible circuits of size \( n^k \) is at most \( 2^{n^{k/2} + n} \)
- \( M = \{ f | \forall c \text{ (circuit of size } n^k) \exists x \text{ (input of length } n) : f(x) \neq c(x) \} \)
  - \( (2^{n^{c^*k}} > 2^{n^{k/2} + n} \Rightarrow M \text{ is not empty}) \)
- let \( \leq \) 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

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

\[
y \in L \iff \begin{cases} 
  f(y) = 1 \\
  \forall c \exists x : f(x) \neq c(x) \\
  \forall f' : (\forall c \exists x : (f'(x) \neq c(x)) \implies f \leq f')
\end{cases}
\]

- rewriting:

\[
y \in L \iff \exists f \forall c \forall f' \exists x \exists c' \forall x' : \\
  (f(x) \neq c(x) \land ((f \leq f') \lor f'(x') = c'(x'))) \land f(y) = 1
\]

- \( L \) 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: $\sum_4 \mathbf{P} \quad \text{Size}[n^k]$ 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 $c$ 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
  - $L \quad \text{Size}[n^{k-1}]$
**MA protocol**

- **MA protocols**: \( L \supseteq \text{MA} \) 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

**Proof generation:**
\[
\text{MERLIN} \\
\text{Proof generation:} \\
\{0,1\}^{p(|x|)}
\]

**Proof verification:**
\[
\text{ARTHUR} \\
\text{Proof verification:} \\
M(x, Proof, \{0,1\}^{p(|x|)}) = ?
\]
**MA protocols**: $L \supseteq \text{MA}$ 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**: $\text{PH} \subseteq \text{PPP}$

$\text{P}^\text{#P} = \{f : \Sigma^* \rightarrow \text{N} \cup \{0\} \mid \exists \text{ time polynomial NTM } M_f \text{ such that for every } x f(x) = \text{acc}_{M_f}(x)\}$, where $\text{acc}_{M_f}(x)$ is the number of ACCEPT paths of machine $M_f$.

$\text{PPP} = \text{P}^\text{#P}$: lemma in the proof of Toda’s theorem

$\text{P}^\text{#P}$ has interactive protocol with prover from $\text{P}^\text{#P}$
Lemma 1: if $\text{PP} \subseteq \text{P/poly}$ we have $\text{P}^{\text{PP}} \subseteq \text{MA}$. 
Proof: later.

Lemma 2: $\text{MA} \subseteq \text{PP}$. 
Proof: lection 7.

Theorem: $\text{PP} \quad \text{Size}[n^k]$ for every $k$. 
Proof: \( \text{PP} \subseteq \text{Size}[n^k] \) for every \( k \)

REMEMBER:
- 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: \( \text{PH} \subseteq \text{Size}[n^k] \)
- \( \Rightarrow \text{PP} \subseteq \text{Size}[n^k] \)

\( \square \)
Lemma 1: \( \mathsf{PP} \subseteq \mathsf{P/poly} \rightarrow \mathsf{P}^{\mathsf{PP}} \subseteq \mathsf{MA} \). 

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

$\text{P}^{\#P}$ has interactive protocol with prover from $\text{P}^{\#P}$
PP \subseteq P/poly \rightarrow P^{PP} \subseteq MA.
Lemma’s Proof

- We modified the prover → it acts like a simple $P^{NP}$ machine
- We know: $P^{NP} = P^{PP} \subseteq P/poly$
- $MA$ 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
- $P^{NP} \subseteq P/poly \rightarrow P^{PP} = P^{NP} \subseteq MA$

□
Conclusion

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

- SAT has polynomial circuit $\rightarrow$ PH collapses on the second level

- PP Size$[n^k]$ for every k.
Thanks for the Patience

QUESTIONS TIME