NEXP is not contained in $\text{ACC}^0$

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

An outline of the proof by Ryan Williams
May 6, 2011
Outline

- Introduction to complexity classes
- The statement of the main theorem
- History and importance
- Proof
- Future directions
Outline

- Introduction to complexity classes
- The statement of the main theorem
- History and importance
- Proof
- Future directions
Complexity classes and lower bounds

(model of computation, resource bound) complexity class

(Det TM, poly time) $P$

(Non-Det TM, poly time) $NP$

(Non-Det TM, exp time) $NEXP$

(Circuits, poly size) $P/poly$
Circuits as a model of computation

Set of variables
\[ X = \{x_1, x_2, \ldots, x_n\}. \]

Directed acyclic graph (DAG) with labels from \( X \cup \overline{X} \cup \{\land, \lor\} \cup \{0, 1\} \).

Computes a function
\[ f : \{0, 1\}^n \rightarrow \{0, 1\}. \]
Circuits as a model of computation

Let $C = \{C_n\}_{n=0}^\infty$.

Definition (Circuits computing a language)

$C$ is said to compute language $L$ if

$$\forall x : x \in L \cap \{0, 1\}^n \Leftrightarrow C_n(x) = 1$$

- Set of variables
  \[ X = \{x_1, x_2, \ldots, x_n\}. \]
- Directed acyclic graph (DAG) with labels from $X \cup \overline{X} \cup \{\land, \lor\} \cup \{0, 1\}$.
- Computes a function $f : \{0, 1\}^n \to \{0, 1\}$. 

NEXP is not contained in ACC$^0$
Circuits as a model of computation

\[(\overline{x}_1 \lor x_2) \land (x_2 \lor 0)\]

Let \(C = \{C_n\}_{n=0}^{\infty}\).

- Set of variables
  \(X = \{x_1, x_2, \ldots, x_n\}\).
- Directed acyclic graph (DAG) with labels from
  \(X \cup \overline{X} \cup \{\land, \lor\} \cup \{0, 1\}\).
- Computes a function
  \(f : \{0, 1\}^n \rightarrow \{0, 1\}\).

**Definition (Circuits computing a language)**

\(C\) is said to compute language \(L\) if

\[\forall x : x \in L \cap \{0, 1\}^n \Leftrightarrow C_n(x) = 1\]

Allow for a different algorithm per input length.
Can compute even undecidable languages.
**ACC^0 and MOD_m**

**ACC^0 Circuits:**
Constant depth circuits over AND, OR, NOT, MOD_m for any \( m > 1 \).

- Set of variables
  \[ X = \{x_1, x_2, \ldots, x_n\} \].

- Directed acyclic graph (DAG) with labels from \( X \cup \overline{X} \cup \{\land, \lor, \text{MOD}_m\} \cup \{0, 1\} \).

- Computes a function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \).
$\textbf{ACC}^0$ and $\text{MOD}_m$

$\text{MOD}_6$ computing $\text{MOD}_3$

$\text{MOD}_3(x_1, x_2, \ldots, x_n)$
Outline

- Introduction to complexity classes
- The statement of the main theorem
- History and importance
- Proof
- Future directions
NEXP and ACC^0

**Theorem (Williams, 2010)**

There exists a language in NEXP that has no polynomial sized ACC^0 circuits.
Outline

- Introduction to complexity classes
- The statement of the main theorem
- History and importance
- Proof
- Future directions
History: Lower bounds

Circuits allow a different algorithm for every input length.
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP?
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP?
Open
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP?
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP?

No not even for P

Parity does not have constant depth polynomial sized circuits.

[Frust-Saxe-Sipser, 1981], [Ajtai, 1983], [Yao, 1985], [Håstad, 1986]
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? **Open**

Are there polynomial sized constant depth circuits for NP? **No**

Are there poly sized \(O(1)\) depth circuits with \(\text{MOD}_2\) gates for NP?
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_2$ gates for NP? No

not even for P

MOD$_3$ cannot be computed by constant depth polynomial sized circuits with MOD$_2$ gates. [Razborov, 1987], [Smolensky, 1987]
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_2$ gates for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NP?
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP? No

Are there poly sized $O(1)$ depth circuits with $\text{MOD}_2$ gates for NP? No

Are there poly sized $O(1)$ depth circuits with $\text{MOD}_6$ for NP? Open
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_2$ gates for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NP? Open

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for EXP?
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_2$ gates for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NP? Open

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for EXP? Open .. ?
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_2$ gates for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NP? Open

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for EXP? Open

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NEXP?
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open

Are there polynomial sized constant depth circuits for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_2$ gates for NP? No

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NP? Open

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for EXP? Open

Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NEXP? Hmm …
History: Lower bounds

Circuits allow a different algorithm for every input length.

Are there polynomial sized circuits for NP? Open
Are there polynomial sized constant depth circuits for NP? No
Are there poly sized $O(1)$ depth circuits with MOD$_2$ gates for NP? No
Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NP? Open
Are there poly sized $O(1)$ depth circuits with MOD$_6$ for EXP? Open
Are there poly sized $O(1)$ depth circuits with MOD$_6$ for NEXP?
This is what we will study today.
Outline

- Introduction to complexity classes
- The statement of the main theorem
- History and importance
- Proof
- Future directions
Proof

Preliminaries for the proof:

\textbf{Circuit-Sat:}

Given: a circuit $C$ on $n$ variables
Check: does there exist an assignment for the variables
that makes the circuit evaluate to 1.
Proof

Preliminaries for the proof:

**Circuit-Sat:**
- Given: a circuit $C$ on $n$ variables
- Check: does there exist an assignment for the variables that makes the circuit evaluate to 1.

**Succinct-3SAT:**
- Given: a formula $\phi$ on $2^n$ variables and $2^n$ clauses encoded by a circuit of size $\text{poly}(n)$
- Check: is $\phi$ satisfiable?
Proof

Preliminaries for the proof:

**Circuit-Sat:**
- Given: a circuit $C$ on $n$ variables
- Check: does there exist an assignment for the variables that makes the circuit evaluate to 1.

**Succinct-3SAT:**
- Given: a formula $\phi$ on $2^n$ variables and $2^n$ clauses encoded by a circuit of size $\text{poly}(n)$
- Check: is $\phi$ satisfiable?

---

**Theorem (Nondeterministic time heirarchy theorem)**

\[ \forall k > 0 \text{ and } \exists f : \{0, 1\}^* \rightarrow \{0, 1\}, \text{ such that } f \in \text{NTIME}[2^n] \text{ but } f \notin \text{NTIME}[2^n/n^k]. \]
Proof

Proof Outline:

Let $L \in \text{NTIME}[2^n]$.

1. Given $x \in \{0, 1\}^n$. Reduce to an instance of \textsc{Succinct-3SAT} $C_x$

2. From $C_x$ obtain a circuit $D$ such that $D$ is unsatisfiable if and only if $x \in L$.

3. Prove $D$ is ACC$^0$, poly sized.

4. Give a fast algorithm for ACC$^0$-\textsc{Circuit-Sat}, running in time $O(2^n/n^k)$.
Proof

Proof Outline:

Let \( L \in \text{NTIME}[2^n] \).

1. Given \( x \in \{0, 1\}^n \). Reduce to an instance of \textsc{Succinct-3SAT} \( C_x \).

2. From \( C_x \) obtain a circuit \( D \) such that \( D \) is unsatisfiable if and only if \( x \in L \).

3. Prove \( D \) is \( \text{ACC}^0 \), poly sized.

4. Give a fast algorithm for \( \text{ACC}^0\text{-Circuit-Sat} \), running in time \( O(2^n/n^k) \).

\hspace{1cm}

\textbf{Theorem (Nondeterministic time heirarchy theorem)}

\( \forall k > 0 \) and \( \exists f : \{0, 1\}^* \rightarrow \{0, 1\} \), such that \( f \in \text{NTIME}[2^n] \) but \( f \notin \text{NTIME}[2^n/n^k] \).
Proof

Proof Outline:

Let $L \in \text{NTIME}[2^n]$.

1. Given $x \in \{0, 1\}^n$. Reduce to an instance of \textsc{Succinct-3SAT} $C_x$.

2. From $C_x$ obtain a circuit $D$ such that $D$ is unsatisfiable if and only if $x \in L$. *

3. Prove $D$ is $\text{ACC}^0$, poly sized. *

4. Give a fast algorithm for $\text{ACC}^0$-$\text{CIRCUIT-SAT}$, running in time $O(2^n/n^k)$.

* Assuming $\text{NEXP}$ is not contained in $\text{ACC}^0$. 
Step 1: Reduction to Succinct-3SAT

Given: a formula $\phi$ on $2^n$ variables and $2^n$ clauses using a circuit of size $\text{poly}(n)$

Check: is $\phi$ satisfiable?

Deterministic polynomial time reduction from any language $L$ in NTIME[$2^n$] to Succinct-3SAT:

Nutan (IITB)

NEXP is not contained in ACC$^0$

May 2011 12 / 24
Step 1: Reduction to Succinct-3SAT

Succinct-3SAT:

Given: a formula $\phi$ on $2^n$ variables and $2^n$ clauses using a circuit of size $\text{poly}(n)$

Check: is $\phi$ satisfiable?

Nutan (IITB)
**Step 1: Reduction to Succinct-3SAT**

**Succinct-3SAT:**

Given: a formula \( \phi \) on \( 2^n \) variables and \( 2^n \) clauses using a circuit of size \( \text{poly}(n) \)

Check: is \( \phi \) satisfiable?

Deterministic polynomial time reduction from any language \( L \) in NTIME[\( 2^n \)] to Succinct-3SAT:

[Tourlakis, 2001], [Fortnow-Lipton-van Melkebeek-Viglas, 2005]

\[
x \quad \rightarrow \quad C_x
\]

\[
|x| = n \quad \rightarrow \quad \text{size of } C_x \quad O(n^5)
\]

\[
n + 5 \log n \quad \text{inputs}
\]

\[
x \in L \quad \iff \quad C_x \text{ is satisfiable}
\]
Proof Outline

Let $L \in \text{NTIME}[2^n]$.

1. Given $x \in \{0, 1\}^n$. Reduce to an instance of $\text{SUCCINCT-3SAT} \ C_x$

2. From $C_x$ obtain a circuit $D$ such that $D$ is unsatisfiable if and only if $x \in L$. (Assuming NEXP in $\text{ACC}^0$.)

3. Prove $D$ is $\text{ACC}^0$, poly sized. (Assuming NEXP in $\text{ACC}^0$.)

4. Give a fast algorithm for $\text{ACC}^0$-$\text{CIRCUIT-SAT}$, running in $\text{NTIME}[2^n/n^k]$. 

Nutan (IITB)
**Succinct-3SAT to Circuit-Sat**

- **$C_x$**
  - Input: index $i$ of a clause
  - Output: variables $x_{i1}, x_{i2}, x_{i3}$ in clause $i$ with signs

- **$W$**
  - Input: a variable index $i$
  - Output: value of $x_i$ in the lex first satisfying assignment

- **$D$**
  - Input: index $i$ of a clause
  - Output: 1 iff clause $i$ not satisfied by assignment given by $W$

Assume NEXP has polysized circuits.

The satisfiable instance of Succinct-3SAT have polynomial size circuits enclosing the satisfying assignments.

[Impagliazzo-Kabanets-Wigderson, 2002]
**Succinct-3SAT to Circuit-Sat**

**C**
- Input: index $i$ of a clause
- Output: variables $x_{i1}, x_{i2}, x_{i3}$ in clause $i$ with signs

**W**
- Input: a variable index $i$
- Output: value of $x_i$ in the lex first satisfying assignment

**D**
- Input: index $i$ of a clause
- Output: $1$ iff clause $i$ **not** satisfied by assignment given by $W$

If formula encoded by $C_x$ is satisfiable then $D$ is not satisfiable.

If formula encoded by $C_x$ is not satisfiable then $D$ is satisfiable.
**Succinct-3SAT to Circuit-Sat**

\[ C_x \]
Input: index \( i \) of a clause
Output: variables \( x_{i1}, x_{i2}, x_{i3} \) in clause \( i \) with signs

\[ W \]
Input: a variable index \( i \)
Output: value of \( x_i \) in the lex first satisfying assignment

\[ D \]
Input: index \( i \) of a clause
Output: 1 iff clause \( i \) not satisfied by assignment given by \( W \)

Assume \( \text{NEXP} \) has polysized circuits.

The satisfiable instance of \textbf{Succinct-3SAT} have polynomial size circuits encoding the satisfying assignments.

[Impagliazzo-Kabanets-Wigderson, 2002]
Proof Outline

Let $L \in \text{NTIME}[2^n]$.

1. Given $x \in \{0, 1\}^n$. Reduce to an instance of $\text{SUCCINCT}-3\text{SAT} \ C_x$
2. From $C_x$ obtain a circuit $D$ such that $D$ is unsatisfiable if and only if $x \in L$. (Assuming $\text{NEXP} \subseteq \text{ACC}^0$.)
3. Prove $D$ is $\text{ACC}^0$, poly sized. (Assuming $\text{NEXP} \subseteq \text{ACC}^0$.)
4. Give a fast algorithm for $\text{ACC}^0$-$\text{CIRCUIT-SAT}$, running in $\text{TIME}[2^n/n^k]$. 

Making $D$ ACC$^0$?

**Lemma**

For an instance $C_x$ of Succinct-3SAT there is an equivalent ACC$^0$ circuit $C$ such that for all $y \in \{0, 1\}^{n+5\log n}$ $C_x(y) = C(y)$.

Proof: As NEXP is contained in ACC$^0$, P is contained in ACC$^0$.

Let $A$ be an ACC$^0$ circuit for the Circuit Value Problem.

Feed $C_x$ as an input to $A$.

$A(C_x, y)$ evaluates $C_x$ on $y$ and is an ACC$^0$ circuit.
Proof Outline

Let $L \in \text{NTIME}[2^n]$. 

1. Given $x \in \{0, 1\}^n$. Reduce to an instance of $\text{SUCCINCT-3SAT } C_x$
2. From $C_x$ obtain a circuit $D$ such that $D$ is unsatisfiable if and only if $x \in L$. (Assuming NEXP in $\text{ACC}^0$.)
3. Prove $D$ is $\text{ACC}^0$, poly sized. (Assuming NEXP in $\text{ACC}^0$.)
4. Give a fast algorithm for $\text{ACC}^0$-$\text{CIRCUIT-SAT}$, running in $\text{NTIME}[2^n/n^k]$. 

Nutan (IITB) 
NEXP is not contained in $\text{ACC}^0$ 
May 2011
Design fast algorithm for ACC$^0$-Circuit-Sat

Outline:
- Any ACC$^0$ circuit can be converted into a SYM$^+$ circuit.
- There is a fast dynamic programming algorithm for circuit satisfiability of SYM$^+$ circuits.
Design fast algorithm for $\text{ACC}^0$-\textsc{Circuit-Sat}

$\text{SYM}^+$:

Depth 2 circuit which is a symmetric function $*$ of ANDs of the input variables.

$*$ Symmetric function: output depends only on the number of 1s in the input.
Design fast algorithm for $\text{ACC}^0$-$\text{Circuit-Sat}$

$\text{SYM}^+$:

Depth 2 circuit which is a symmetric function of ANDs of the input variables.
$\text{ACC}^0 \text{ to SYM}^+$

**Theorem ([Yao, 1990],[Beigel-Tarui, 1994],[Allender-Gore, 1994])**

Any $\text{ACC}^0$ circuit of size $n^5$ can be converted into a $\text{SYM}^+$ circuit of size $n^{O(\log^d n)}$ in time $n^{O(\log^d n)}$. And the symmetric function can be computed in time $n^{O(\log^d n)}$. ($d$: a constant depending on the depth)
Fast evaluation of $\text{SYM}^+$

Let $S \subseteq [n]$. For a $\text{SYM}^+$ circuit:

Let $h : 2^n \rightarrow \mathbb{N}$ be defined as:

$h(S) = j$ if $j$-many AND gates have $S$ feeding into them.
Fast evaluation of \textbf{SYM}⁺

Let $S \subseteq [n]$. For a SYM⁺ circuit:

Let $h : 2^{[n]} \rightarrow \mathbb{N}$ be defined as:

$h(S) = j$ if $j$-many AND gates have $S$ feeding into them.

Let $g : 2^{[n]} \rightarrow \mathbb{N}$ be defined as: $g(T) = \sum_{S \subseteq T} h(S)$.

$g(T) = \#$ of true AND gates under
$x_i = 1$ for $i \in T$ and $x_i = 0$ for $i \notin T$
Fast evaluation of $\text{SYM}^+$

Let $S \subseteq [n]$. For a $\text{SYM}^+$ circuit:

Let $h : 2^{[n]} \rightarrow \mathbb{N}$ be defined as:

$h(S) = j$ if $j$-many AND gates have $S$ feeding into them.

Let $g : 2^{[n]} \rightarrow \mathbb{N}$ be defined as: $g(T) = \sum_{S \subseteq T} h(S)$.

$g(T) = \#$ of true AND gates under $x_i = 1$ for $i \in T$ and $x_i = 0$ for $i \notin T$

Computing $\text{SYM}^+$ circuit on all its inputs equivalent to

Computing $g$ for all $T \subseteq [n]$
Computing $g$ on all $T \subseteq [n]$

- Computing $h$

  Takes time $O(2^n + s \text{poly}(n))$, where $s$ is the size of the circuit.
Computing $g$ on all $T \subseteq [n]$

- Computing $h$
  
  Takes time $O(2^n + s \text{ poly}(n))$, where $s$ is the size of the circuit.

- Computing $g$: Dynamic Programming
  
  Set $g_0(T) = h(T)$ for all $T \subseteq [n]$

  $$g_{i+1}(T) = \begin{cases} 
  g_i(T) + g_i(T \setminus \{i\}) & \text{if } i \in T \\
  g_i(T) & \text{otherwise}
  \end{cases}$$
Computing $g$ on all $T \subseteq [n]$

- Computing $h$
  
  Takes time $O(2^n + s \text{ poly}(n))$, where $s$ is the size of the circuit.

- Computing $g$: Dynamic Programming
  
  Set $g_0(T) = h(T)$ for all $T \subseteq [n]$

  \[
  g_{i+1}(T) = \begin{cases} 
  g_i(T) + g_i(T \setminus \{i\}) & \text{if } i \in T \\
  g_i(T) & \text{otherwise}
  \end{cases}
  \]

  $g_i(T)$ equals $\sum_{S \subseteq T} h(S)$, for $S$ such that $S \cap [n] \setminus \{i\} = T \cap [n] \setminus \{i\}$

  Note: $g_n = g$.

  The function $g_{i+1}$ can be computed from $g_i$ in time $O(2^n \text{ poly}(n))$

  Therefore the total time $= O(2^n \text{ poly}(n) + s \text{ poly}(n))$. 
Fast ACC$^0$-Circuit-Sat algorithm

**Theorem**

Circuit-Sat for any ACC$^0$ circuit $C$ with $n + 5 \log n$ inputs and $n^5$ size can be determined in time $O(2^{n-\log^2 n} \text{poly}(n))$.

**Proof:** Obtain $C'$ from $C$:

$C'$:
- size: $2^l n^5$,
- inputs: $m - l$, where $m = n + 5 \log n$. 

---

Nutan (IITB)
**Fast ACC\(^0\)-Circuit-Sat algorithm**

**Theorem**

**Circuit-Sat** for any ACC\(^0\) circuit \(C\) with \(n + 5 \log n\) inputs and \(n^5\) size can be determined in time \(O(2^{n - \log^2 n}\text{poly}(n))\).

**Proof:** Obtain \(C'\) from \(C\):

![Diagram of Circuit-Sat algorithm]

\(C'\):
- **size:** \(2^l n^5\),
- **inputs:** \(m - l\), where \(m = n + 5 \log n\).

Recall:

**Theorem ([Yao, 1990],[Beigel-Tarui, 1994],[Allender-Gore, 1994])**

Any ACC\(^0\) circuit of size \(s\) can be converted into a SYM\(^+\) circuit of size \(s^{O(\log^d s)}\) in time \(s^{O(\log^d s)}\). Assuming the symmetric function can be computed in time \(s^{O(\log^d s)}\). (\(d\): a constant depending on the depth)
**Fast ACC$^0$-Circuit-Sat algorithm**

**Theorem**

Circuit-Sat for any ACC$^0$ circuit $C$ with $n + 5 \log n$ inputs and $n^5$ size can be determined in time $O(2^{n - \log^2 n} \text{poly}(n))$.

**Proof:** Obtain $C''$ from $C$:

**SYM$^+$ circuit $C'''$:**

- size: $(2^l n^5)O(l \log d n)$
- inputs: $m - l$

\[ C''': \]

\[ \begin{array}{c}
\text{size:} \\
\text{inputs:}
\end{array} \]

\[ 2^l n^5, \\
m - l, \quad \text{where} \\
m = n + 5 \log n. \]
**Fast ACC⁰-Circuit-Sat algorithm**

**Theorem**

**Circuit-Sat** for any ACC⁰ circuit $C$ with $n + 5 \log n$ inputs and $n^5$ size can be determined in time $O(2^{n - \log^2 n} \text{poly}(n))$.

**Proof:**

**SYM⁺ circuit $C''$:**

- **size:** $(2^l n^5)^{O(l \log^d n)}$
- **inputs:** $m - l$

Fast satisfiability algorithm:

- To evaluate AND gates: $O(2^{m-l} \text{poly}(n))$
- To evaluate symmetric function: $(2^l n^5)^{O(l \log^d n)}$

*Nutan (IITB)*
Fast ACC\(^0\)-Circuit-Sat algorithm

**Theorem**

Circuit-Sat for any ACC\(^0\) circuit \(C\) with \(n + 5 \log n\) inputs and \(n^5\) size can be determined in time \(O(2^{n-\log^2 n} \text{poly}(n))\).

Proof:

**SYM\(^+\) circuit \(C''\):**

- size: \((2^l n^5)O(l \log^d n)\)
- inputs: \(m - l\)

Fast satisfiability algorithm:

For \(l = \log^2 n\)

To evaluate AND gates: \(O(2^{m-l} \text{poly}(n)) = O(2^{n-\log^2 n} \text{poly}(n))\)

To evaluate symmetric function: \((2^l n^5)O(l \log^d n) = nO(\log^{d'} n)\)
**Fast ACC⁰-Circuit-Sat algorithm**

**Theorem**

\textbf{Circuit-Sat} for any ACC⁰ circuit \( C \) with \( n + 5 \log n \) inputs and \( n^5 \) size can be determined in time \( O(2^{n-\log^2 n} \text{poly}(n)) \).

**Proof:**

**SYM⁺ circuit \( C'' \):**

- size: \((2^l n^5)O(l \log^d n)\)
- inputs: \( m - l \)

Fast satisfiability algorithm:

For \( l = \log^2 n \)

To evaluate AND gates: \( O(2^{m-l} \text{poly}(n)) = O(2^{n-\log^2 n} \text{poly}(n)) \)

To evaluate symmetric function: \((2^l n^5)O(l \log^d n) = n^{O(\log^d' n)}\)
Outline

• Introduction to complexity classes
• The statement of the main theorem
• History and importance
• Proof
• Future directions
Thank you!