Strongly Exponential Lower Bounds for Monotone Computation

Toniann Pitassi and Robert Robere
Department of Computer Science
University of Toronto

STOC 2017
Montréal, Canada
Boolean Circuits

Basic model for computing boolean functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$

Assume fan-in 2, and a basis of AND, OR, NOT gates.

Central Question.

What boolean functions are hard to compute?
Boolean Circuits

Every $f : \{0, 1\}^n \rightarrow \{0, 1\}$ has a circuit of size $O(n2^n)$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$f(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Theorem.** [Lupanov 58] Every boolean function on $n$ bits can be computed by a circuit with $(1 + o(1)) \frac{2^n}{n}$ gates (!)
Boolean Circuits

**Theorem.** [Lupanov 58] Every boolean function on \( n \) bits can be computed by a circuit with \((1 + o(1)) \frac{2^n}{n}\) gates. (!)

**Theorem.** [Shannon 1949] For every \( n \), all but an exponentially small fraction of boolean functions on \( n \) bits require circuits with \( \Omega \left( \frac{2^n}{n} \right) \) gates.

**Proof.** Simple counting argument (non-constructive).
Boolean Circuits (Lower Bounds)

Do we have any explicit examples of hard boolean functions?
Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions?

**NO!**
Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions? **NO!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>Circuits</td>
<td></td>
<td>$2^n/n$ [S. 49]</td>
</tr>
</tbody>
</table>
**Boolean Circuits (Lower Bounds)**

Do we have any **explicit** examples of hard boolean functions?

**NO!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Circuits</td>
<td>$5n - o(n)$</td>
<td>$2^n / n$</td>
</tr>
</tbody>
</table>

[IM. 02] [S. 49]
Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions?

**NO!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>Circuits</td>
<td>$5n - o(n)$</td>
<td>$2^n/n$</td>
</tr>
<tr>
<td>$NC^1$</td>
<td>Formula</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions?

**NO!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>Circuits</td>
<td>$5n - o(n)$</td>
<td>$2^n / n$</td>
</tr>
<tr>
<td>$NC^1$</td>
<td>Formula</td>
<td></td>
<td>$2^n / \log n$</td>
</tr>
</tbody>
</table>
Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions? **NO!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{P}$</td>
<td>Circuits</td>
<td>$5n - o(n)$</td>
<td>$2^n / n$</td>
</tr>
<tr>
<td>$\mathbf{NC}^1$</td>
<td>Formula</td>
<td>$n^{3-o(1)}$</td>
<td>$2^n / \log n$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions?

**NO!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Circuits</td>
<td>$5n - o(n)$</td>
<td>$2^n/n$ [S. 49]</td>
</tr>
<tr>
<td>NC$^1$</td>
<td>Formula</td>
<td>$n^{3-o(1)}$</td>
<td>$2^n/\log n$ [RS. 42]</td>
</tr>
<tr>
<td>L</td>
<td>Switching Networks</td>
<td>$n^2/\log n$</td>
<td>$2^n/n$ [S. 49]</td>
</tr>
<tr>
<td>Mod$_p$ L</td>
<td>Span Programs</td>
<td>$n \log n$</td>
<td>GF(2) $\sqrt{2^{n+1}}$ [N.62]</td>
</tr>
<tr>
<td>CC</td>
<td>Comparator Circuits</td>
<td>$n \log n$</td>
<td>$2^n/n$ [S. 49]</td>
</tr>
</tbody>
</table>
Monotone Circuit Complexity

A circuit is *monotone* if it does not use NOT gates.
Monotone Circuit Complexity

A circuit is *monotone* if it does not use NOT gates.

A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is monotone if

$$x \leq y \implies f(x) \leq f(y)$$
Monotone Circuit Complexity

A circuit is *monotone* if it does not use NOT gates.

A function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) is monotone if

\[
x \leq y \implies f(x) \leq f(y)
\]

Monotone circuits have a number of applications in cryptography, proof complexity, communication theory ....
Monotone Boolean Circuits (Lower Bounds)

Do we have any explicit examples of hard boolean functions?
Monotone Boolean Circuits (Lower Bounds)

Do we have any explicit examples of hard boolean functions?

YES!
Monotone Boolean Circuits (Lower Bounds)

Do we have any explicit examples of hard boolean functions? **YES!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type (MONOTONE)</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mP</td>
<td>Circuits</td>
<td></td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
</tbody>
</table>
Monotone Boolean Circuits (Lower Bounds)

Do we have any explicit examples of hard boolean functions?

**YES!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type (MONOTONE)</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mP</td>
<td>Circuits</td>
<td>$2^{\Omega((n/ \log n)^{1/3})}$ [HR 01]</td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
</tbody>
</table>
Monotone Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions?

**YES!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type (MONOTONE)</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{mP}$</td>
<td>Circuits</td>
<td>$2^{\Omega((n/ \log n)^{1/3})}$ [HR 01]</td>
<td>$2^n / n^{3/2}$ [U, P 76]</td>
</tr>
<tr>
<td>$\text{mNC}^1$</td>
<td>Formula</td>
<td>$2^{\Omega(n/ \log n)}$ [GP 14]</td>
<td>$2^n / \sqrt{n \log n}$</td>
</tr>
</tbody>
</table>
Monotone Boolean Circuits (Lower Bounds)

Do we have any explicit examples of hard boolean functions?

Yes!

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type (MONOTONE)</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bounds (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mP</td>
<td>Circuits</td>
<td>$2^{\Omega((n/ \log n)^{1/3})}$ [HR 01]</td>
<td>$2^n / n^{3/2}$ [U, P 76]</td>
</tr>
<tr>
<td>mNC$^1$</td>
<td>Formula</td>
<td>$2^{\Omega(n/ \log n)}$ [GP 14]</td>
<td>$2^n / \sqrt{n \log n}$</td>
</tr>
<tr>
<td>mL</td>
<td>Switching Networks</td>
<td>$2^{\Omega(\sqrt{n/ \log n})}$ [GP 14]</td>
<td>$2^n / n^{3/2}$ [U, P 76]</td>
</tr>
</tbody>
</table>
## Monotone Boolean Circuits (Lower Bounds)

Do we have any **explicit** examples of hard boolean functions?

**YES!**

<table>
<thead>
<tr>
<th>Complexity Measure</th>
<th>Circuit Type (MONOTONE)</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bound (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mP</strong></td>
<td>Circuits</td>
<td>$2 \Omega((n/ \log n)^{1/3})$ [HR 01]</td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
<tr>
<td><strong>mNC$^1$</strong></td>
<td>Formula</td>
<td>$2 \Omega(n/ \log n)$ [GP 14]</td>
<td>$2^n/\sqrt{n \log n}$</td>
</tr>
<tr>
<td><strong>mL</strong></td>
<td>Switching Networks</td>
<td>$2 \Omega(\sqrt{n/ \log n})$ [GP 14]</td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
<tr>
<td><strong>mSPAN$_\mathbb{R}$</strong></td>
<td>Real Span Programs</td>
<td>$2 \Omega(n^{1/7})$ [RPRC 16]</td>
<td></td>
</tr>
<tr>
<td><strong>mCC</strong></td>
<td>Comparator Circuits</td>
<td>$2 \Omega(n^{1/7})$ [RPRC 16]</td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
</tbody>
</table>
Monotone Boolean Circuits (Lower Bounds)

Do we have any explicit examples of hard boolean functions? **YES!**

<table>
<thead>
<tr>
<th>Complexity Measure (MONOTONE)</th>
<th>Circuit Type</th>
<th>Strongest Lower Bound (Explicit)</th>
<th>Strongest Lower Bound (Non-Explicit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$mP$</td>
<td>Circuits</td>
<td>$2^\Omega((n/ \log n)^{1/3})$ [HR 01]</td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
<tr>
<td>$mNC^1$</td>
<td>Formula</td>
<td>$2^{\alpha n}$ [PR 17]</td>
<td>$2^n/\sqrt{n \log n}$</td>
</tr>
<tr>
<td>$mL$</td>
<td>Switching Networks</td>
<td>$2^{\alpha n}$ [PR 17]</td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
<tr>
<td>$mSPAN_{\mathbb{R}}$</td>
<td>Real Span Programs</td>
<td>$2^{\alpha n}$ [PR 17]</td>
<td></td>
</tr>
<tr>
<td>$mCC$</td>
<td>Comparator Networks</td>
<td>$2^{\alpha n}$ [PR 17]</td>
<td>$2^n/n^{3/2}$ [U, P 76]</td>
</tr>
</tbody>
</table>
Result

**Main Theorem.** There is a monotone boolean function \( f \) computable in \( \text{NP} \) (CSP-SAT) such that every monotone

1. formula,
2. switching network,
3. real span program, or
4. comparator circuit

computing \( f \) requires size \( 2^{\alpha n} \) for some universal constant \( \alpha > 0 \).
The Proof (A Flavor)

Columns labelled with $y \in f^{-1}(0)$

Let $f : \{0, 1\}^N \to \{0, 1\}$ be a monotone boolean function.

$K\text{W-Search}^+(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [N]$

Input: $(x, y) \in f^{-1}(1) \times f^{-1}(0)$

Output: $i \in [N]$  $x_i = 1, y_i = 0$

Rows labelled with $x \in f^{-1}(1)$

$f^{-1}(1) \times f^{-1}(0)$
Columns labelled with $y \in f^{-1}(0)$

Let $f : \{0, 1\}^N \to \{0, 1\}$ be a monotone boolean function.

$KW$-Search$^+(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [N]$

**Input:** $(x, y) \in f^{-1}(1) \times f^{-1}(0)$

**Output:** $i \in [N]$ \quad $x_i = 1, y_i = 0$
Columns labelled with $y \in f^{-1}(0)$

Let $f : \{0, 1\}^N \to \{0, 1\}$ be a monotone boolean function.

$KW$-Search$^+(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [N]$  

Input: $(x, y) \in f^{-1}(1) \times f^{-1}(0)$ 
Output: $i \in [N] \quad x_i = 1, y_i = 0$

**Theme:** Complexity of $KW$-Search$(f) \approx$ Circuit Complexity of $f$
Example: Formulas

Columns labelled with $y \in f^{-1}(0)$

Rows labelled with $x \in f^{-1}(1)$

\[ \land = \text{AND} \quad \lor = \text{OR} \]

Theme: Complexity of $\text{KW-Search}(f) \approx \text{Circuit Complexity of } f$
Example: Formulas

Columns labelled with $y \in f^{-1}(0)$

Rows labelled with $x \in f^{-1}(1)$

Lemma. [Khrapchenko 71] Formula for $f : \{0, 1\}^n \rightarrow \{0, 1\}$ with $s$ leaves yields a partition of $f^{-1}(1) \times f^{-1}(0)$ into $s$ mono. rectangles.
Let $\chi(f)$ denote the minimum number of rectangles in any monochromatic partition of $f^{-1}(1) \times f^{-1}(0)$

Columns labelled with $y \in f^{-1}(0)$

Rows labelled with $x \in f^{-1}(1)$

**Idea [Razb. 90]:** Use rank to lower bound $\chi(f)$!
Let $\chi(f)$ denote the minimum number of rectangles in any monochromatic partition of $f^{-1}(1) \times f^{-1}(0)$.

Columns labelled with $y \in f^{-1}(0)$

Rows labelled with $x \in f^{-1}(1)$

**Idea [Razb. 90]:** Use rank to lower bound $\chi(f)$!

Let $A$ be any $|f^{-1}(1)| \times |f^{-1}(0)|$ matrix over a field $\mathbf{F}$.

$$
A = \sum_{i=1}^{\chi(f)} A_i
$$
Let $\chi(f)$ denote the minimum number of rectangles in any monochromatic partition of $f^{-1}(1) \times f^{-1}(0)$.

Columns labelled with $y \in f^{-1}(0)$

\[
A = \sum_{i=1}^{\chi(f)} A_i
\]

\[
\text{rank}(A) \leq \chi(f) \max_i \text{rank}(A_i)
\]

Rows labelled with $x \in f^{-1}(1)$
Let $\chi(f)$ denote the minimum number of rectangles in any monochromatic partition of $f^{-1}(1) \times f^{-1}(0)$. Columns labelled with $y \in f^{-1}(0)$

$$A = \sum_{i=1}^{\chi(f)} A_i$$

rank($A$) $\leq \chi(f) \max_{i} \text{rank}(A_i)$

$\leq \chi(f) \max_{i \in [n]} \text{rank}(A \upharpoonright X_i)$

Rearranging,

$$\chi(f) \geq \frac{\text{rank}(A)}{\max_{i \in [n]} \text{rank}(A \upharpoonright X_i)}$$
Rank Measure

**Theorem [Razb. 90].** For any monotone boolean function $f$ and any $f^{-1}(1) \times f^{-1}(0)$ matrix $A$ over any field, the quantity

$$\mu_A(f) = \frac{\text{rank}(A)}{\max_{i \in [n]} \text{rank}(A \upharpoonright X_i)}$$

is a lower bound on $\chi(f)$ (and the monotone formula size of $f$).

**Theorem [G. 01, RPRC. 16].** $\mu_A(f)$ is also a lower bound on monotone switching networks, monotone span programs, and monotone comparator circuits computing $f$. 
Rank Measure

Theorem [Razb. 90]. For any monotone boolean function $f$ and any $f^{-1}(1) \times f^{-1}(0)$ matrix $A$ over any field, the quantity

$$\mu_A(f) = \frac{\text{rank}(A)}{\max_{i \in [n]} \text{rank}(A \upharpoonright X_i)}$$

is a lower bound on $\chi(f)$ (and the monotone formula size of $f$).

Main Theorem (Restated). There is an explicit function $f$ computable in NP and a matrix $A$ such that $\mu_A(f) \geq 2^{\alpha n}$. 
Proving Lower Bounds on $\mu_A(f)$

**Theorem [Razb. 90]** There is a monotone boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ in NP and a 0/1 matrix $A$ satisfying

$$\mu_A(f) \geq n^{\Omega(\log n)}$$

**[RPRC 16, PR 17]** “Lifting theorem” to prove lower bounds against $\mu_A(f)$

1. Reduce lower bounds on $\mu_A(f)$ to **query complexity** lower bounds for a search problem $\text{Search}(\mathcal{C})$ related to $\text{KW-Search}^+(f)$
2. Prove strong query complexity lower bounds for $\text{Search}(\mathcal{C})$
Search Problems and Algebraic Gaps

\[ \mathcal{C} = C_1 \land C_2 \land \ldots \land C_m \] is an unsatisfiable \textbf{k}-CNF with variables \( z \).

\text{Search}(\mathcal{C}) := \text{given assignment to } z, \text{ output index of falsified clause.}
Algebraic Gap Complexity

Ex. \[ C = \overline{x}_1 \land \overline{x}_2 \land \cdots \land \overline{x}_n \land \bigg( \bigvee_{i=1}^{n} x_i \bigg) \]
Algebraic Gap Complexity

**Ex.** \[ C = \overline{x}_1 \land \overline{x}_2 \land \cdots \land \overline{x}_n \land \left( \bigvee_{i=1}^{n} x_i \right) \]

**Certificate** = minimal partial restriction falsifying a clause
Algebraic Gap Complexity

Ex. \[ C = \overline{x}_1 \land \overline{x}_2 \land \cdots \land \overline{x}_n \land \left( \bigvee_{i=1}^{n} x_i \right) \]

Cert\((C)\) \quad x_1 = 1 \quad x_2 = 1 \quad x_n = 1 \quad x_1 = 0, x_2 = 0, \ldots, x_n = 0

Certificate = minimal partial restriction falsifying a clause
Algebraic Gap Complexity

**Ex.** \( C = \overline{x}_1 \land \overline{x}_2 \land \cdots \land \overline{x}_n \land \left( \bigvee_{i=1}^{n} x_i \right) \)

\[ \text{Cert}(C) \quad x_1 = 1 \quad x_2 = 1 \quad x_n = 1 \quad x_1 = 0, x_2 = 0, \cdots, x_n = 0 \]

**Algebraic Gap Complexity.** Find a polynomial \( p : \{0, 1\}^n \rightarrow \mathbb{R} \)

so that \( \text{gap}_p(C) = \deg(p) - \max_{\pi \in \text{Cert}(C)} \deg(p \upharpoonright \pi) \) is maximized.
Algebraic Gap Complexity

\[ \text{Ex. } \quad \mathcal{C} = \overline{x}_1 \land \overline{x}_2 \land \cdots \land \overline{x}_n \land \left( \bigvee_{i=1}^{n} x_i \right) \]

\[ \text{Cert}(\mathcal{C}) \quad x_1 = 1 \quad x_2 = 1 \quad x_n = 1 \quad x_1 = 0, x_2 = 0, \cdots , x_n = 0 \]

**Algebraic Gap Complexity.** Find a polynomial \( p : \{0, 1\}^n \rightarrow \mathbb{R} \) so that \( \text{gap}_p(\mathcal{C}) = \deg(p) - \max_{\pi \in \text{Cert}(\mathcal{C})} \deg(p \| \pi) \) is maximized.

\[ p = OR_n \quad \Rightarrow \quad \deg(OR_n) = n \quad \text{and} \quad \max_{\pi \in \text{Cert}(\mathcal{C})} \deg(OR_n) = 0 \]
Algebraic Gap Complexity vs. Rank Measure

**Algebraic Gap Complexity.** Given \( \text{Search}(C) \), find polynomial \( p : \{0,1\}^n \to \mathbb{R} \) so that \( \text{gap}_p(C) = \text{deg}(p) - \max_{\pi \in \text{Cert}(C)} \text{deg}(p | \pi) \) is maximized.

**Rank Measure** \( \mu_A(f) \). Given \( f : \{0,1\}^N \to \{0,1\} \), find matrix \( A \) such that

\[
\mu_A(f) = \frac{\text{rank}(A)}{\max_{i \in [n]} \text{rank}(A | X_i)}
\]

is maximized.
Rank Measure Lifting

Theorem [RPRC 16].
For any unsatisfiable $k$-CNF $C$ with $m$ clauses there is a function $f_C$ computable in NP with $N \leq m^{2k+1}$ variables and a real matrix $A$ such that

$$
\mu_A(f_C) \geq \Omega(m^{\text{gap}(C)}) \geq \Omega(N^{\text{gap}(C)}/2^{k+1})
$$
Rank Measure Lifting

Theorem [RPRC 16].
For any unsatisfiable $k$-CNF $\mathcal{C}$ with $m$ clauses there is a function $f_C$ computable in NP with $N \leq m^{2k+1}$ variables and a real matrix $A$ such that

$$\mu_A(f_C) \geq \Omega(m^{\text{gap}(\mathcal{C})}) \geq \Omega(N^{\text{gap}(\mathcal{C})/2k+1})$$

[RPRC 16]. $\mathcal{C}$ = “pebbling contradiction”, then $\text{gap}(\mathcal{C}) \geq m/\log m$

Yields $2^{\Omega(N^\varepsilon)}$ lower bounds!

Problem is the number of variables!
## Gadget Size Blues

<table>
<thead>
<tr>
<th>Query Complexity</th>
<th>\leq</th>
<th>Circuit Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree Depth</td>
<td>[RM 99]</td>
<td>Monotone Circuit Depth</td>
</tr>
<tr>
<td><strong>Algebraic Gap Complexity</strong></td>
<td>[RPRC 16]</td>
<td><strong>(Logarithm of) Rank Measure</strong></td>
</tr>
</tbody>
</table>

For decision trees vs. depth, current constructions yield $N = \omega(m)$ variables.

For critical block sensitivity, we can take $N = O(m)$ variables, but best query lower bounds are $\Omega(m/\log m)$. 
Rank Measure Lifting

Theorem [RPRC 16].
For any unsatisfiable $k$-CNF $C$ with $m$ clauses there is a function $f_C$ computable in NP with $N \leq m^{2k+1}$ variables and a real matrix $A$ such that

$$\mu_A(f_C) \geq \Omega(m^{\text{gap}(C)}) \geq \Omega(N^{\text{gap}(C)/2k+1})$$
Rank Measure Lifting (Refined)

Theorem [PR 17].
For any unsatisfiable $O(1)$-CNF $\mathcal{C}$ with $m$ clauses satisfying
$\text{gap}(\mathcal{C}) = \Omega(m)$ there is a function $f_C$ computable in NP with
$N = O(m)$ variables and a real matrix $A$ such that

$$\mu_A(f_C) \geq 2^{\Omega(m)} \geq 2^{\Omega(N)}$$
Rank Measure Lifting (Refined)

**Theorem [PR 17].**
For any unsatisfiable \( \mathbf{O}(1) \)-CNF \( \mathcal{C} \) with \( m \) clauses satisfying \( \text{gap}(\mathcal{C}) = \Omega(m) \) there is a function \( f_C \) computable in NP with \( N = O(m) \) variables and a real matrix \( A \) such that

\[
\mu_A(f_C) \geq 2^{\Omega(m)} \geq 2^{\Omega(N)}
\]

**Proof. [RPRC 16]** \( \text{KW-Search}^+(f_C) \equiv \text{Search}(\mathcal{C} \circ g^n(x, y)) \)

Rank of **pattern matrix** \( A = [p(g^n(x, y))]_{x,y \in \mathcal{X}^n \times \mathcal{Y}^n} \approx \exp(\deg(p)) \)
Proving Large Algebraic Gaps

**Algebraic Gap Complexity.** Given $\text{Search}(C)$, find polynomial $p : \{0, 1\}^n \rightarrow \mathbb{R}$ so that $\text{gap}_p(C) = \deg(p) - \max_{\pi \in \text{Cert}(C)} \deg(p \mid \pi)$ is maximized.
Proving Large Algebraic Gaps

**Algebraic Gap Complexity.** Given \( \text{Search}(C) \), find polynomial \( p : \{0, 1\}^n \to \mathbb{R} \) so that \( \text{gap}_p(C) = \deg(p) - \max_{\pi \in \text{Cert}(C)} \deg(p | \pi) \) is maximized.

**Tseitin Principle.** Let \( G \) be a k-regular graph with an odd number of vertices.
Proving Large Algebraic Gaps

**Algebraic Gap Complexity.** Given \( \text{Search}(C) \), find polynomial \( p : \{0, 1\}^n \to \mathbb{R} \) so that \( \text{gap}_p(C) = \deg(p) - \max_{\pi \in \text{Cert}(C)} \deg(p \mid \pi) \) is maximized.

**Tseitin Principle.** Let \( G \) be a \( k \)-regular graph with an odd number of vertices.

\[
\text{Tseitin}_G \quad \begin{array}{c}
\text{Variables} \\
Z_{uv} \quad u, v \in E
\end{array} \quad \begin{array}{c}
\text{Constraints} \\
\bigoplus_{u \sim v} Z_{uv} = 1 \quad v \in V
\end{array}
\]
Proving Large Algebraic Gaps

**Algebraic Gap Complexity.** Given $\text{Search}(C)$, find polynomial $p : \{0,1\}^n \rightarrow \mathbb{R}$ so that $\text{gap}_p(C) = \deg(p) - \max_{\pi \in \text{Cert}(C)} \deg(p \mid \pi)$ is maximized.

**Tseitin Principle.** Let $G$ be a $k$-regular graph with an odd number of vertices.

- **Variables**: $Tseitin_G$
- **Constraints**: $z_{uv} \quad uv \in E$

$$\bigoplus_{u \sim v} z_{uv} = 1 \quad v \in V$$

**Theorem.** $\text{gap}(Tseitin_G) \geq \text{Expansion}(G) \cdot m/3d$

**Proof.** Reduction to resolution width of $Tseitin_G$
Rank Measure Lifting

**Theorem [PR 17].** For any unsatisfiable $O(1)$-CNF $\mathcal{C}$ with $m$ clauses satisfying $\text{gap}(\mathcal{C}) = \Omega(m)$ there is a function $f_{\mathcal{C}}$ computable in NP with $N = O(m)$ variables and a real matrix $A$ such that

$$\mu_A(f_{\mathcal{C}}) \geq 2^{\Omega(m)} \geq 2^{\Omega(N)}$$

**Theorem.** $\text{gap}(\text{Tseitin}_G) \geq \text{Expansion}(G) \cdot m/3d$

Choose $G$ to be a strong constant-degree expander and the main theorem is proved!
Conclusion

Prove the first strongly exponential lower bounds for any explicit function, asymptotically matching non-explicit lower bounds from counting in the monotone setting.

Can we sharpen it further?

Further applications of the framework? (In particular, a deeper understanding of the algebraic gap complexity and other exotic query complexity measures for search problems.)
Thanks!