Exponential Lower Bounds for Monotone Span Programs

Stephen A. Cook
Toniann Pitassi
Robert Robere
Benjamin Rossman

University of Toronto

FOCS 2016

Familiar Picture

$$NC^1 \subseteq L \subseteq NL \subseteq NC \subseteq P$$

Familiar Picture

$$\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{NC} \subseteq \mathsf{P}$$

Formulas

Familiar Picture

Switching Networks
(Branching Programs)

$$NC^1 \subseteq L \subseteq NL \subseteq NC \subseteq P$$

Formulas

Familiar Picture

Switching Networks
(Branching Programs)

$$\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{NC} \subseteq \mathsf{P}$$

Formulas

Directed Switching Networks
(Non-det. Branching Programs)

Familiar Picture

Switching Networks
(Branching Programs)

Polylog-depth
Circuits

$$NC^1 \subseteq L \subseteq NL \subseteq NC \subseteq P$$

Formulas

Directed Switching Networks
(Non-det. Branching Programs)

Familiar Picture

Switching Networks
(Branching Programs)

Polylog-depth
Circuits

$$NC^1 \subseteq L \subseteq NL \subseteq NC \subseteq P$$

Circuits

Formulas

Directed Switching Networks
(Non-det. Branching Programs)

Familiar Picture

$$\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{NC} \subseteq \mathsf{P}$$

(Less) Familiar Picture

$$
\begin{array}{ccccccccc}
 & & & & \mathsf{CC} & & & & \\
 & & & & \rotatebox{90}{$\subseteq$} & & & & \\
\mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\
 & & \rotatebox{-90}{$\subseteq$} & & & & & & \\
 & & \mathsf{SPAN}_{\mathbf{F}} & & & & & &
\end{array}
$$

(Less) Familiar Picture

$$
\begin{array}{ccccccccc}
 & & & & \mathsf{CC} & & & & \\
 & & & & \rotatebox{90}{$\subseteq$} & & & & \\
\mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\
 & & \rotatebox{-90}{$\subseteq$} & & & & & & \\
 & & \mathsf{SPAN}_{\mathbf{F}} & & & & & &
\end{array}
$$

Span Programs over field **F** [KW '90]

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

A

Matrix over **F**

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

1	**0**	**0**	**1**
0	**0**	**1**	**0**
0	**1**	**0**	**0**
0	**1**	**1**	**0**

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

1	**0**	**0**	**1**
0	**0**	**1**	**0**
0	**1**	**0**	**0**
0	**1**	**1**	**0**

Rows labelled with input literals.

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

Accept assignment if the consistent rows span all-1s vector

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

$$x_1 = \text{True}$$
$$x_2 = \text{True}$$
$$x_3 = \text{True}$$

Accept assignment if the consistent rows span all-1s vector

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

$x_1 = \text{True}$
$x_2 = \text{True}$
$x_3 = \text{True}$

Accept assignment if the consistent rows span all-1s vector

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

$x_1 = \text{True}$

$x_2 = \text{True}$

$x_3 = \text{True}$

ACCEPT!

Accept assignment if the consistent rows span all-1s vector

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

$x_1 = \text{True}$
$x_2 = \text{False}$
$x_3 = \text{False}$

Accept assignment if the consistent rows span all-1s vector

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

ACCEPT!

$$x_1 = \text{True}$$
$$x_2 = \text{False}$$
$$x_3 = \text{False}$$

Accept assignment if the consistent rows span all-1s vector

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

$$x_1 = \text{False}$$
$$x_2 = \text{False}$$
$$x_3 = \text{False}$$

Accept assignment if the consistent rows span all-1s vector

Span Programs [KW '90]

What is a **Span Program** over a field **F**?

REJECT!

x_1	**1**	**0**	**0**	**1**
x_1	**0**	**0**	**1**	**0**
x_2	**0**	**1**	**0**	**0**
$\overline{x_3}$	**0**	**1**	**1**	**0**

$$x_1 = \text{False}$$
$$x_2 = \text{False}$$
$$x_3 = \text{False}$$

Accept assignment if the consistent rows span all-1s vector

(Less) Familiar Picture

$$
\begin{array}{ccccccccc}
 & & & & \mathsf{CC} & & & & \\
 & & & & \rotatebox{90}{$\subseteq$} & & & & \\
\mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\
 & & \rotatebox{90}{$\subseteq$} & & & & & & \\
 & & \mathsf{SPAN}_{\mathbf{F}} & & & & & &
\end{array}
$$

Span Programs over field **F** [KW '90]
Capture logspace counting classes.

(Less) Familiar Picture

Comparator Circuits

$$\begin{array}{ccccccccc} & & & & \mathsf{CC} & & & & \\ & & & & \cup\!\!| & & & & \\ \mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\ & & |\!\!\cap & & & & & & \\ & & \mathsf{SPAN}_{\mathbf{F}} & & & & & & \end{array}$$

Span Programs over field **F** [KW '90]
Capture logspace counting classes.

(Less) Familiar Picture

Comparator Circuits
~ Sorting networks.

$$\begin{array}{ccccccccc} & & & & \mathsf{CC} & & & & \\ & & & & \cup\!\!| & & & & \\ \mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\ & & |\!\!\cap & & & & & & \\ & & \mathsf{SPAN}_{\mathbf{F}} & & & & & & \end{array}$$

Span Programs over field **F** [KW '90]
Capture logspace counting classes.

(Less) Familiar Picture

$$
\begin{array}{ccccccccc}
 & & & & \mathsf{CC} & & & & \\
 & & & & \rotatebox{90}{$\subseteq$} & & & & \\
\mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\
 & & \rotatebox{-90}{$\subseteq$} & & & & & & \\
 & & \mathsf{SPAN}_{\mathbf{F}} & & & & & &
\end{array}
$$

Familiar Picture

$$
\begin{array}{ccccccccc}
 & & & & \mathsf{CC} & & & & \\
 & & & & \rotatebox{90}{$\subseteq$} & & & & \\
\mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\
 & & \rotatebox{-90}{$\subseteq$} & & & & & & \\
 & & \mathsf{SPAN}_{\mathbf{F}} & & & & & &
\end{array}
$$

Familiar Picture

How many separations do we have?

$$
\begin{array}{ccccccccc}
 & & & & \mathsf{CC} & & & & \\
 & & & & \rotatebox{90}{$\subseteq$} & & & & \\
\mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\
 & & \rotatebox{90}{$\supseteq$} & & & & & & \\
 & & \mathsf{SPAN}_{\mathbf{F}} & & & & & &
\end{array}
$$

Familiar Picture

How many separations do we have?

$$\begin{array}{ccccccccc} & & & & \mathsf{CC} & & & & \\ & & & & \rotatebox{90}{$\subseteq$} & & & & \\ \mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\ & & \rotatebox{-90}{$\subseteq$} & & & & & & \\ & & \mathsf{SPAN}_{\mathbf{F}} & & & & & & \end{array}$$

Familiar Picture

How many separations do we have?

$$\begin{array}{ccccccccc}
 & & & & \mathsf{CC} & & & & \\
 & & & & \rotatebox{90}{$\subseteq$} & & & & \\
\mathsf{NC}^1 & \subseteq & \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{NC} & \subseteq & \mathsf{P} \\
 & & \rotatebox{-90}{$\subseteq$} & & & & & & \\
 & & \mathsf{SPAN}_{\mathbf{F}} & & & & & &
\end{array}$$

Fortunately, this is easy to fix.

Familiar Picture

How many separations do we have?

$$\begin{array}{c} \mathsf{mCC} \\ \cup\!| \\ \mathsf{mNC}^1 \subsetneq \mathsf{mL} \subsetneq \mathsf{mNL} \subsetneq \mathsf{mNC} \subsetneq \mathsf{mP} \end{array}$$

$$\mathsf{mL} \supseteq \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP}$$

Fortunately, this is easy to fix.

Monotone = No Negations in Circuit Models

Familiar Picture

How many separations do we have?

$$\begin{array}{l} \mathsf{mCC} \\ \cup\!\vert \\ \mathsf{mNC}^1 \subsetneq \mathsf{mL} \subsetneq \mathsf{mNL} \subsetneq \mathsf{mNC} \subsetneq \mathsf{mP} \\ \qquad\quad \vert\!\cap \\ \qquad\quad \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP} \end{array}$$

Familiar Picture

[Potechin '10]
(Directed st-connectivity)

$$\mathsf{mNC}^1 \subseteq \mathsf{mCC}$$

$$\mathsf{mNC}^1 \subsetneq \mathsf{mL} \subsetneq \mathsf{mNL} \subsetneq \mathsf{mNC} \subsetneq \mathsf{mP}$$

$$\mathsf{mL} \subseteq \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP}$$

[Karchmer-Wigderson '88]
(Undirected st-connectivity)

[Raz-Mckenzie '97]
(GEN)

[Babai, Gal, Wigderson '99]
(Odd Factor)

Familiar Picture

$$\begin{array}{l} \mathsf{mCC} \\ \;\;\cup\!\!\mid \\ \mathsf{mNC}^1 \subsetneq \mathsf{mL} \subsetneq \mathsf{mNL} \subsetneq \mathsf{mNC} \subsetneq \mathsf{mP} \\ \qquad\quad\;\; \mid\!\!\cap \\ \qquad\quad \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP} \end{array}$$

Familiar Picture

$$\begin{array}{ccccccccc}
\mathsf{mCC} & & & & & & & & \\
\rotatebox{90}{$\subseteq$} & & & & & & & & \\
\mathsf{mNC}^1 & \subsetneq & \mathsf{mL} & \subsetneq & \mathsf{mNL} & \subsetneq & \mathsf{mNC} & \subsetneq & \mathsf{mP} \\
 & & \rotatebox{90}{$\supseteq$} & & & & & & \\
 & & \multicolumn{3}{l}{\mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP}} & & & &
\end{array}$$

Familiar Picture

$$\mathrm{mSPAN}_{\mathbf{F}}$$

[Babai et al '96] Quasipolynomial lower bounds against mNP.

Familiar Picture

$\mathrm{mSPAN}_{\mathbf{F}}$

[Babai et al '96] Quasipolynomial lower bounds against mNP.

[Gal '98] Improved lower bounds using rank measure (still quasipolynomial).

Familiar Picture

$\mathrm{mSPAN}_{\mathbf{F}}$

[Babai et al '96] Quasipolynomial lower bounds against mNP.

[Gal '98] Improved lower bounds using rank measure (still quasipolynomial).

[BW '05] Quasipolynomial against nonmonotone NC

Familiar Picture

$\mathrm{mSPAN}_{\mathbf{F}}$

[Babai et al '96] Quasipolynomial lower bounds against mNP.

[Gal '98] Improved lower bounds using rank measure (still quasipolynomial).

[BW '05] Quasipolynomial against nonmonotone NC

Extra Motivation:

Familiar Picture

$\mathrm{mSPAN}_{\mathbf{F}}$

[Babai et al '96] Quasipolynomial lower bounds against mNP.

[Gal '98] Improved lower bounds using rank measure (still quasipolynomial).

[BW '05] Quasipolynomial against nonmonotone NC

Extra Motivation:
Equivalent to Linear Secret Sharing Schemes (!) [KW '90]

Familiar Picture

$$\begin{array}{l} \mathsf{mCC} \\ \cup\!\!| \\ \mathsf{mNC}^1 \subsetneq \mathsf{mL} \subsetneq \mathsf{mNL} \subsetneq \mathsf{mNC} \subsetneq \mathsf{mP} \\ \qquad\quad\ \ |\!\!\cap \\ \qquad\quad \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP} \end{array}$$

Familiar Picture

$$\begin{array}{ccccccccc}
\mathsf{mCC} & & & & & & & & \\
\rotatebox{90}{$\subseteq$} & & & & & & & & \\
\mathsf{mNC}^1 & \subsetneq & \mathsf{mL} & \subsetneq & \mathsf{mNL} & \subsetneq & \mathsf{mNC} & \subsetneq & \mathsf{mP} \\
 & & \rotatebox{-90}{$\subseteq$} & & & & & & \\
 & & \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP} & & & & & &
\end{array}$$

Familiar Picture

Essentially nothing known!

Exponential bounds for Clique

Cannot even prove it contains mNL or mL

$$\mathsf{mCC}$$

$$\cup\!\!\mid$$

$$\mathsf{mNC}^1 \subsetneq \mathsf{mL} \subsetneq \mathsf{mNL} \subsetneq \mathsf{mNC} \subsetneq \mathsf{mP}$$

$$\mid\!\!\cap$$

$$\mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP}$$

Familiar Picture

$$\begin{array}{lllllllll}
\mathsf{mCC} & & & & & & & & \\
\ \ \cup\!\!\mid & & & & & & & & \\
\mathsf{mNC}^1 & \subsetneq & \mathsf{mL} & \subsetneq & \mathsf{mNL} & \subsetneq & \mathsf{mNC} & \subsetneq & \mathsf{mP} \\
 & & \mid\!\!\cap & & & & & & \\
 & & \mathsf{mSPAN}_{\mathbf{F}} & \not\subseteq & \mathsf{mP} & & & &
\end{array}$$

Familiar Picture

$$
\begin{array}{ccccccccc}
\mathsf{mCC} & & & & & & & & \\
\cup\!| & & & & & & & & \\
\mathsf{mNC}^1 & \subsetneq & \mathsf{mL} & \subsetneq & \mathsf{mNL} & \subsetneq & \mathsf{mNC} & \subsetneq & \mathsf{mP} \\
 & & |\!\cap & & & & & & \\
 & & \mathsf{mSPAN}_{\mathbf{F}} & \not\subseteq & \mathsf{mP} & & & &
\end{array}
$$

Natural Questions:

Familiar Picture

$$\begin{array}{ccccccccc} \mathsf{mCC} & & & & & & & & \\ \cup\!\mathsf{I} & & & & & & & & \\ \mathsf{mNC}^1 & \subsetneq & \mathsf{mL} & \subsetneq & \mathsf{mNL} & \subsetneq & \mathsf{mNC} & \subsetneq & \mathsf{mP} \\ & & \mathsf{I}\!\cap & & & & & & \\ & & \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP} & & & & & & \end{array}$$

Natural Questions:

Can we separate mSPAN from mP? mNL?

Familiar Picture

$$\begin{array}{lllllllll} \mathsf{mCC} & & & & & & & & \\ \cup\!| & & & & & & & & \\ \mathsf{mNC}^1 & \subsetneq & \mathsf{mL} & \subsetneq & \mathsf{mNL} & \subsetneq & \mathsf{mNC} & \subsetneq & \mathsf{mP} \\ & & |\!\cap & & & & & & \\ & & \multicolumn{3}{l}{\mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP}} & & & & \end{array}$$

Natural Questions:

Can we separate mSPAN from mP? mNL?

Can we separate mCC from mP? mNL?

Familiar Picture

$$\begin{array}{l} \mathsf{mCC} \\ \ \ \cup\!| \\ \mathsf{mNC}^1 \subsetneq \mathsf{mL} \subsetneq \mathsf{mNL} \subsetneq \mathsf{mNC} \subsetneq \mathsf{mP} \\ \qquad\qquad\ \ |\!\cap \\ \qquad\qquad \mathsf{mSPAN}_{\mathbf{F}} \not\subseteq \mathsf{mP} \end{array}$$

Natural Questions:

Can we separate mSPAN from mP? mNL?

Can we separate mCC from mP? mNL?

Yes --- also unify nearly all lower bounds in mP.

Rank Measure

Rank Measure

$f : \{0,1\}^n \rightarrow \{0,1\}$

Rank Measure

$f : \{0,1\}^n \to \{0,1\}$

monotone

Rank Measure

$f : \{0,1\}^n \rightarrow \{0,1\}$

monotone

$f^{-1}(0)$

$f^{-1}(1)$

Rank Measure

$f : \{0,1\}^n \to \{0,1\}$

monotone

$f^{-1}(0)$

$f^{-1}(1)$

A

Matrix

Not the
0-1 Communication
Matrix

Rank Measure

$f : \{0,1\}^n \to \{0,1\}$

monotone

For any input index i,
take submatrix of

$f^{-1}(0)$

$f^{-1}(1)$

A

Matrix

Not the
0-1 Communication
Matrix

Rank Measure

$$f : \{0,1\}^n \to \{0,1\}$$

monotone

For any input index i,
take submatrix of

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

$$x_i = 1$$

$$y_i = 0$$

$f^{-1}(0)$

$f^{-1}(1)$

A

Matrix

Not the
0-1 Communication
Matrix

Rank Measure

$$f : \{0,1\}^n \to \{0,1\}$$

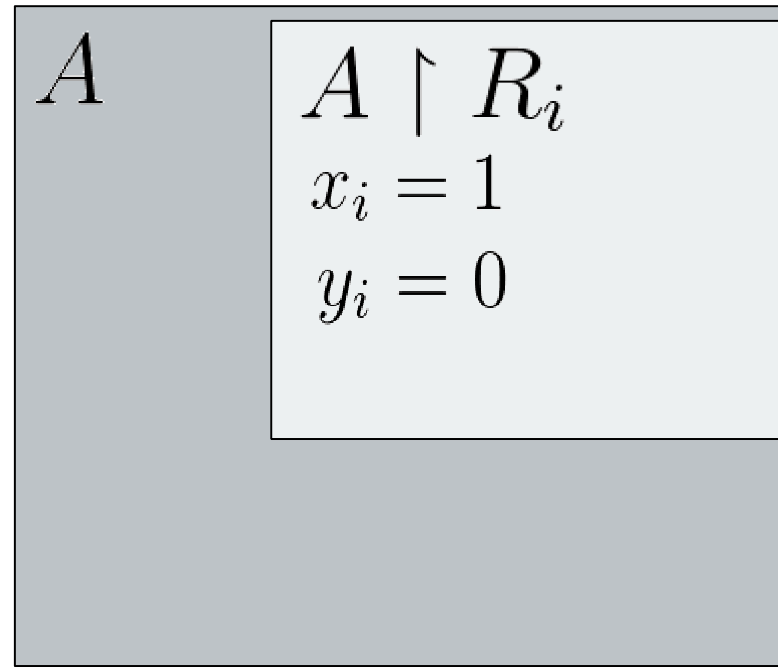
monotone

For any input index i,
take submatrix of

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

$$x_i = 1$$

$$y_i = 0$$

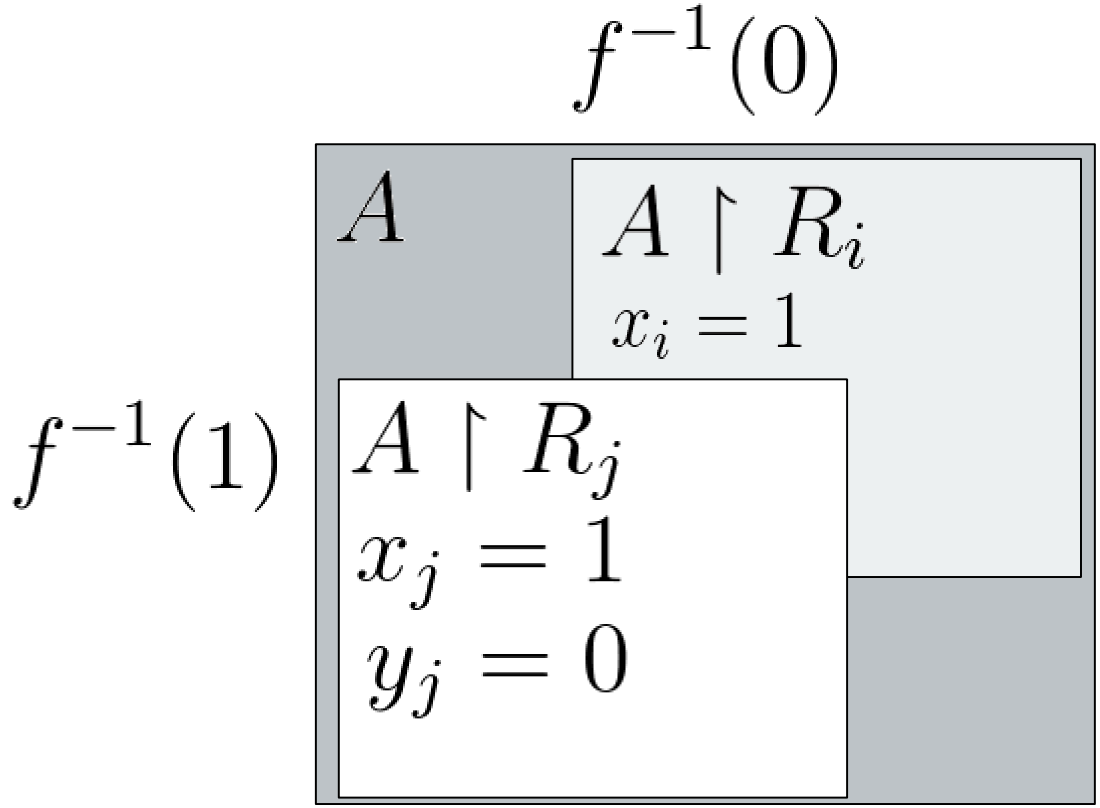


Rank Measure

$f : \{0,1\}^n \to \{0,1\}$

monotone

Ranging over inputs...

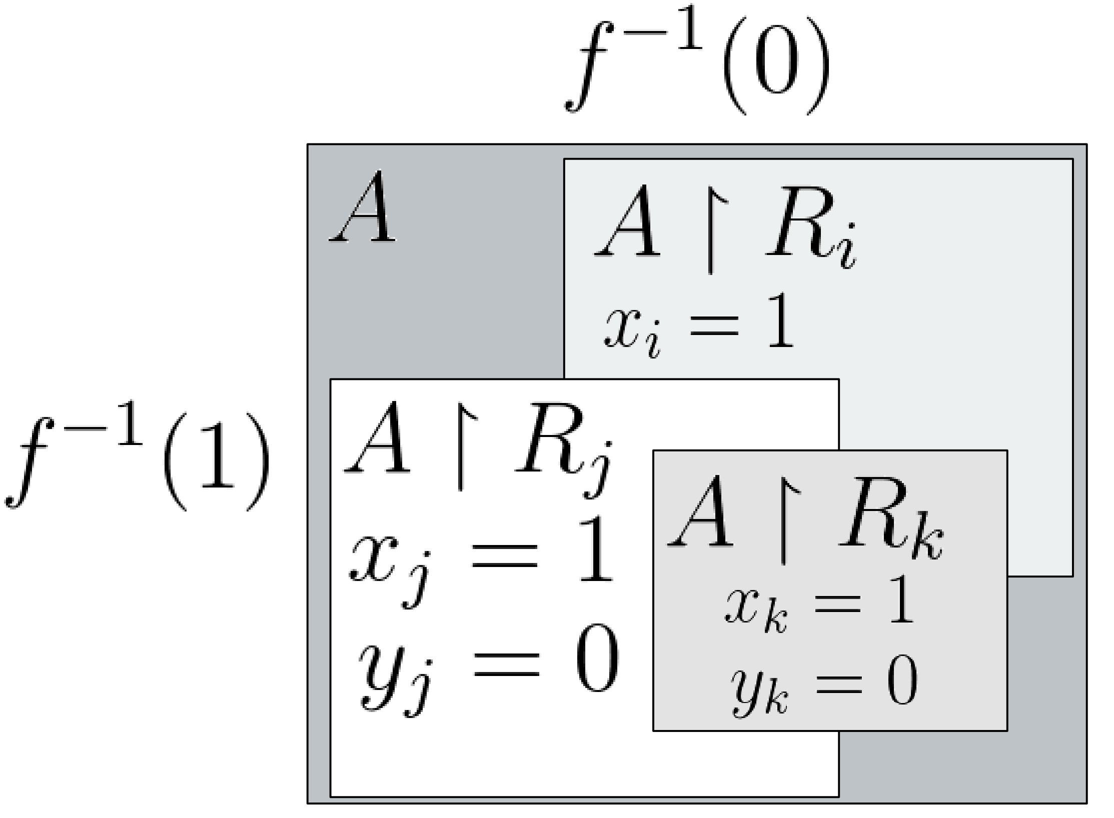


Rank Measure

$$f : \{0,1\}^n \to \{0,1\}$$

monotone

Ranging over inputs...



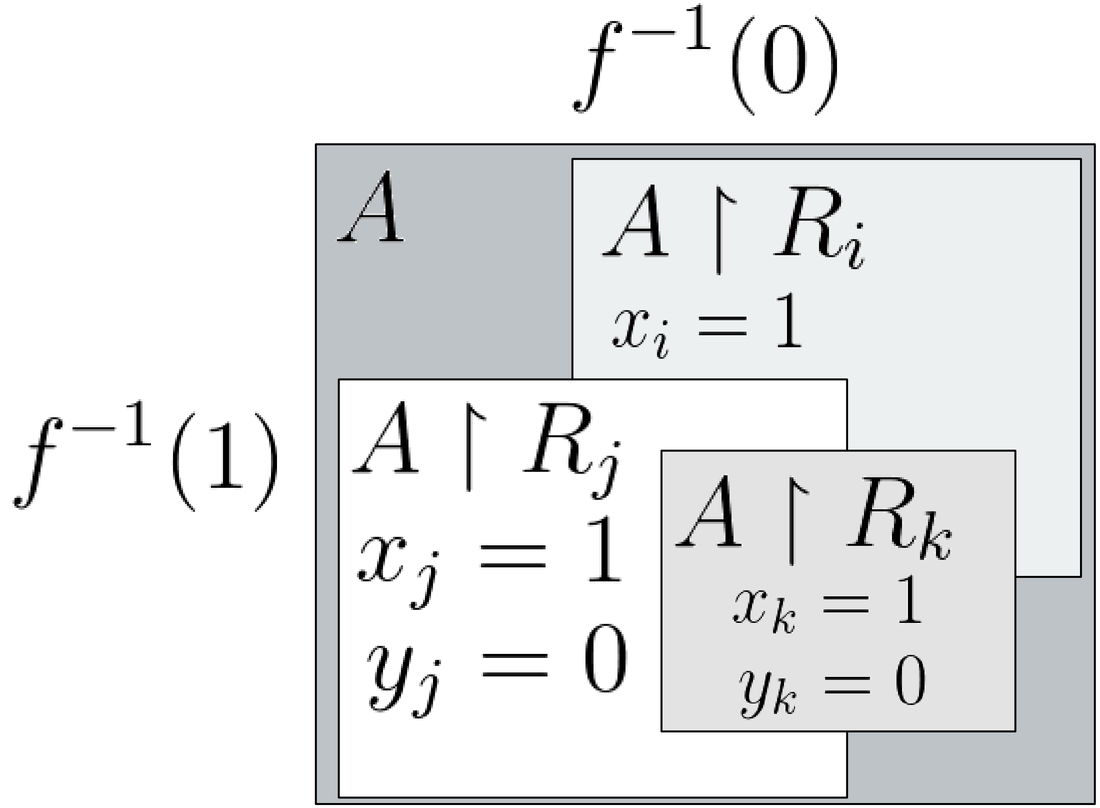
Rank Measure

$f : \{0,1\}^n \to \{0,1\}$

monotone

Ranging over inputs...

All rectangles cover A!



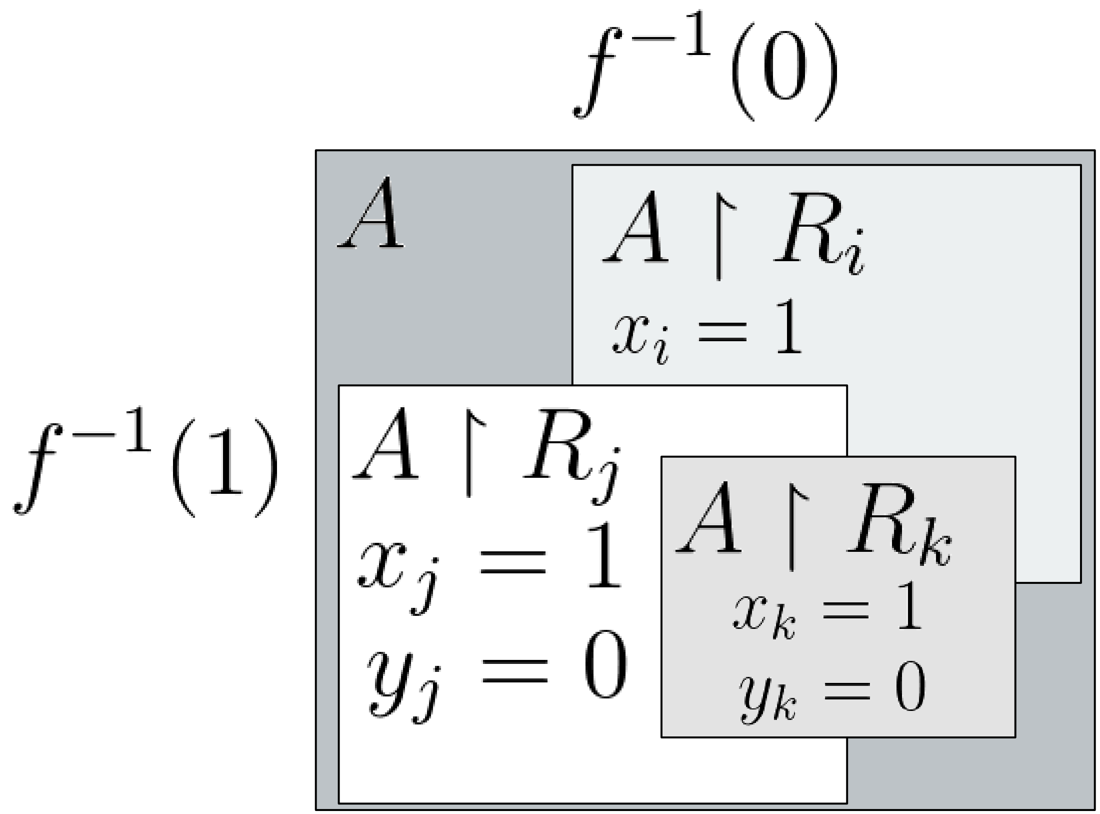
Rank Measure

$f : \{0,1\}^n \to \{0,1\}$

monotone

Ranging over inputs...

All rectangles cover A!



<u>Rank Measure</u> [Razborov '90]**:**

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

Rank Measure

Rank Measure [Razborov '90]**:**

$$\mu_A(f) = \frac{\operatorname{rank}(A)}{\max_{i \in [n]} \operatorname{rank}(A \restriction R_i)}$$

Rank Measure

Rank Measure [Razborov '90]**:**

$$\mu_A(f) = \frac{\mathrm{rank}(A)}{\max_{i \in [n]} \mathrm{rank}(A \restriction R_i)}$$

Theorem [R '90, KW '90, G '98, **CPRR '16**]**:**

For any field **F**, any boolean function f, and any matrix A over **F**,

$$\mu_A(f) \leq \mathsf{mSPAN}_{\mathbf{F}}(f) \leq \mathsf{mL}(f) \leq \mathsf{mNC}^1(f)$$

$$\mu_A(f) \leq \mathsf{mCC}(f)$$

Rank Measure

Rank Measure [Razborov '90]:

$$\mathrm{rank}(A)$$

Best prior lower bounds:

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

f **in NP!**

$$\mu_A(f) \leq \mathrm{mSPAN}_{\mathbb{F}}(f) \leq \mathrm{mL}(f) \leq \mathrm{mNC}^1(f)$$

$$\mu_A(f) \leq \mathsf{mCC}(f)$$

Main Theorem

Theorem: There is a function f (GEN) in **mP** and a **real** matrix A such that $\mu_A(f) \geq 2^{\Omega(N^\varepsilon)}$

There is a function g (STCONN) in **mNL** and a **real** matrix B such that $\mu_B(g) \geq N^{\Omega(\log N)}$

Main Theorem

<u>Theorem</u>: There is a function f (GEN) in **mP** and a **real** matrix A such that $\mu_A(f) \geq 2^{\Omega(N^{\varepsilon})}$

There is a function g (STCONN) in **mNL** and a **real** matrix B such that $\mu_B(g) \geq N^{\Omega(\log N)}$

<u>Prior Work:</u>
Unified proof of many previous monotone separations between classes within P.

Simplification of $\mathsf{mL} \not\subseteq \mathsf{mNL}$ [Potechin '10]

Main Theorem

Theorem: There is a function f (GEN) in **mP** and a **real** matrix A such that $\mu_A(f) \geq 2^{\Omega(N^\varepsilon)}$

There is a function g (STCONN) in **mNL** and a **real** matrix B such that $\mu_B(g) \geq N^{\Omega(\log N)}$

Span Programs:

First exponential lower bounds for monotone **span programs** and linear secret sharing schemes.

First separations between monotone **span programs** and monotone P, monotone NL

Example of a function computable by non-monotone **span programs over GF(2)**, not computable by **monotone span programs over reals**

Main Theorem

Theorem: There is a function f (GEN) in **mP** and a **real** matrix A such that $\mu_A(f) \geq 2^{\Omega(N^{\varepsilon})}$

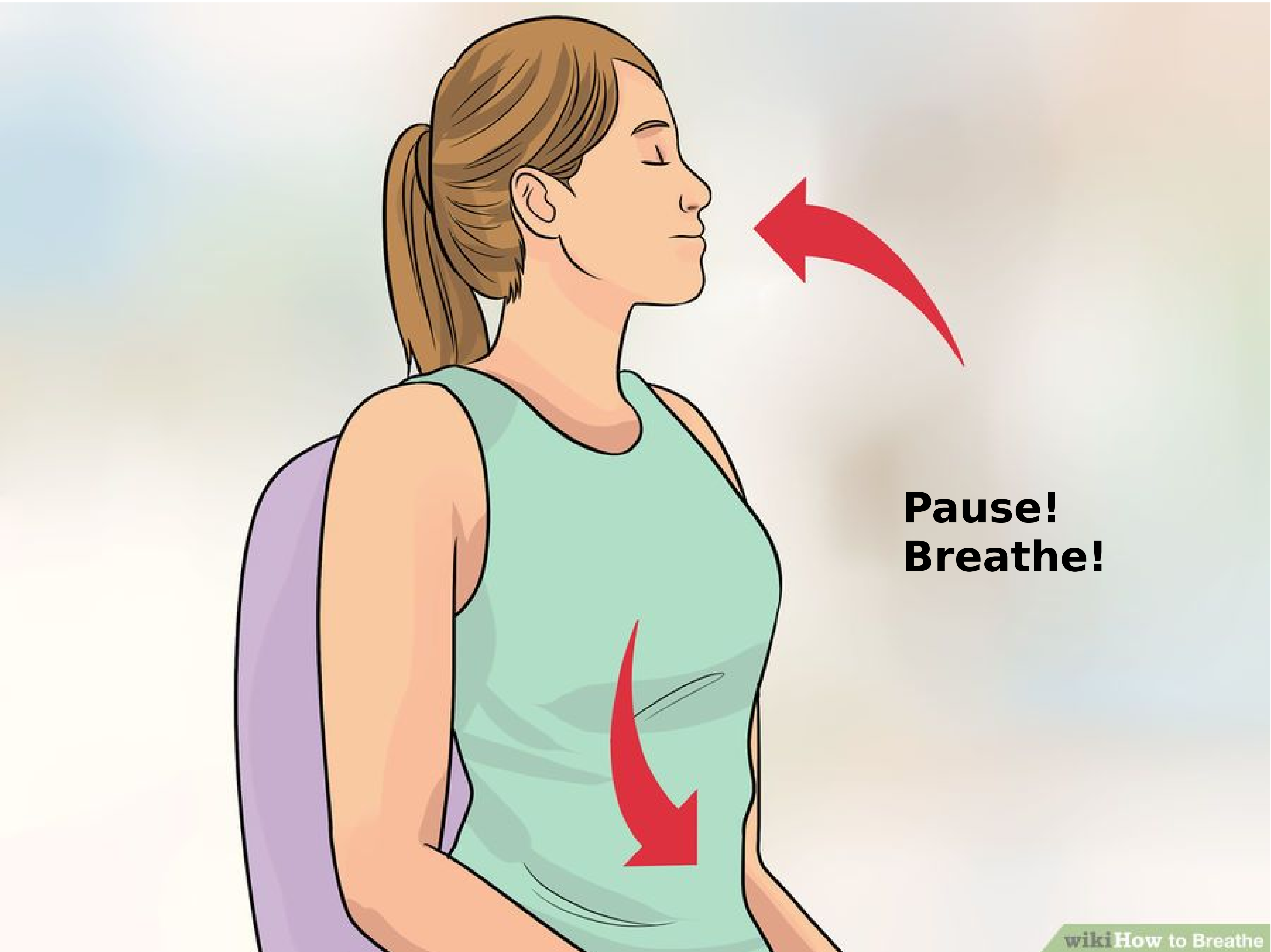
There is a function g (STCONN) in **mNL** and a **real** matrix B such that $\mu_B(g) \geq N^{\Omega(\log N)}$

Comparator Circuits:

First exponential lower bounds for **comparator circuits** computing a function in monotone P.

First separations between monotone **comparator circuits** and monotone P, monotone NL

Example of a function computable by non-monotone **comparator circuits**, not efficiently computable by monotone **comparator circuits**


Pause!
Breathe!
wikiHow to Breathe

The Proof

The Proof

Previous Proofs:

The Proof

Previous Proofs:

Direct combinatorial constructions

The Proof

Previous Proofs:

Direct combinatorial constructions

Resulting matrices have {0,1} entries, for which we have quasipolynomial **upper** bounds [Razborov '90].

The Proof

Previous Proofs:

Direct combinatorial constructions

Resulting matrices have {0,1} entries, for which we have quasipolynomial **upper** bounds [Razborov '90].

Our Proof:

The Proof

Previous Proofs:

Direct combinatorial constructions

Resulting matrices have {0,1} entries, for which we have quasipolynomial **upper** bounds [Razborov '90].

Our Proof:

Prove a new **lifting theorem** to reduce the lower bound to bounding a new **algebraic query measure** on search problems.

The Proof

Previous Proofs:

Direct combinatorial constructions

Resulting matrices have {0,1} entries, for which we have quasipolynomial **upper** bounds [Razborov '90].

Our Proof:

Prove a new **lifting theorem** to reduce the lower bound to bounding a new **algebraic query measure** on search problems.

Our matrices have entries in $\mathbb{R}$, and so we can avoid the above obstacle.

The Proof

Overview

Rank Measure [Razborov '90]**:**

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

The Proof

Overview

Rank Measure [Razborov '90]**:**

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

1 Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)

The Proof

Overview

<u>Rank Measure</u> [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f)

The Proof

Overview

Rank Measure [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f)
3. Actually **prove** the query lower bounds against Search(f)

The Proof

Overview

<u>Rank Measure</u> [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f)
3. Actually **prove** the query lower bounds against Search(f)

The Proof

Overview

Rank Measure [Razborov '90]:

$$\frac{\ldots k(A)}{\ldots k(A \upharpoonright R_i)}$$

Follows from
[Raz-Mckenzie '97]
[Goos-Pitassi '15]

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f)
3. Actually **prove** the query lower bounds against Search(f)

The Proof

Overview

Rank Measure [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f)
3. Actually **prove** the query lower bounds against Search(f)

The Proof

Lifting Theorem

The Proof

Lifting Theorem

(Communication Setting)

The Proof

Lifting Theorem

(Communication Setting)

Search Problem

S = Search(f)

$$S \subseteq \{0,1\}^n \times Q$$

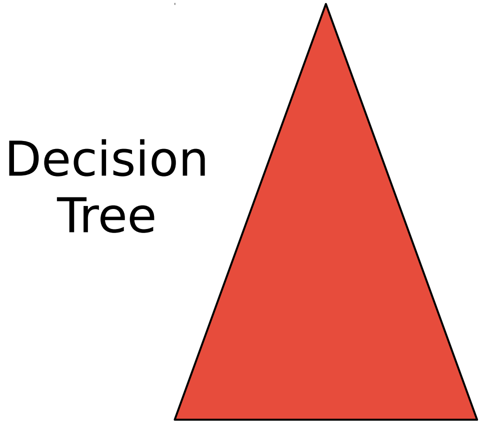
The Proof

Lifting Theorem

(Communication Setting)

Search Problem
S = Search(f)

$$S \subseteq \{0,1\}^n \times Q$$



Hard for
Weak Complexity
Measure

The Proof

Lifting Theorem

(Communication Setting)

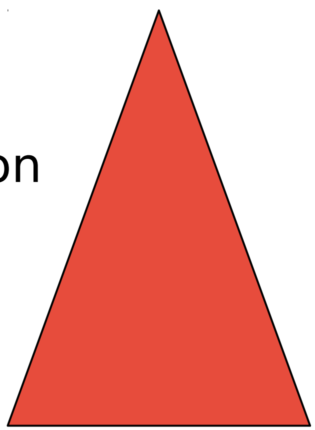
Search Problem

S = Search(f)

$S \subseteq \{0,1\}^n \times Q$

$x \in \mathcal{A}^n, y \in \mathcal{B}^n$

Decision Tree



Hard for
Weak Complexity
Measure

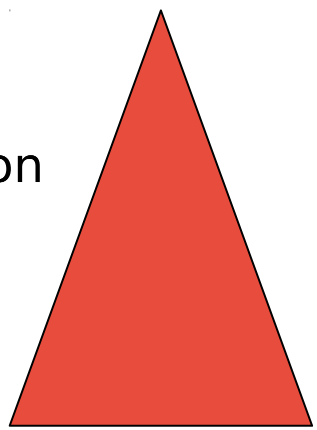
The Proof

Lifting Theorem
(Communication Setting)

Search Problem
S = Search(f)

$$S \subseteq \{0,1\}^n \times Q$$

Decision
Tree



Hard for
Weak Complexity
Measure

$$x \in \mathcal{A}^n, y \in \mathcal{B}^n$$

$$S(g(x_1, y_1), \ldots g(x_n, y_n))$$

Compose S with some
two input function g

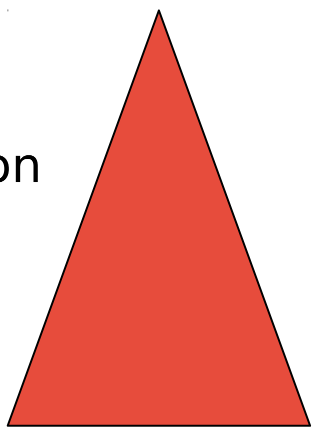
Alice gets x inputs
Bob gets y inputs

The Proof

Lifting Theorem
(Communication Setting)

Search Problem
S = Search(f)
$S \subseteq \{0,1\}^n \times Q$

Decision
Tree



$x \in \mathcal{A}^n, y \in \mathcal{B}^n$
$S(g(x_1, y_1), \ldots g(x_n, y_n))$

$\longrightarrow$

Compose S with some two input function g

Alice gets x inputs
Bob gets y inputs

$\mathcal{B}^n$

$\mathcal{A}^n$

Hard for
Weak Complexity
Measure

Hard for
Strong Complexity
Measure

The Proof

Lifting Theorem

(Our Setting)

The Proof

Lifting Theorem

(Our Setting)

Search Problem

S = Search(f)

$S \subseteq \{0,1\}^n \times Q$

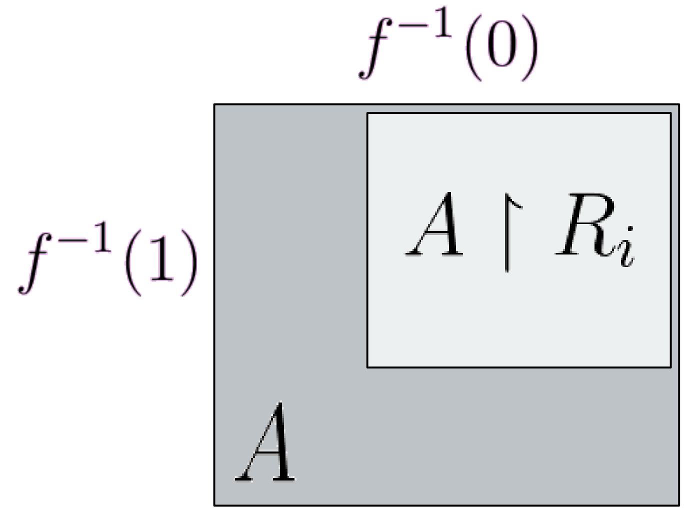
The Proof

Lifting Theorem

(Our Setting)

Search Problem
S = Search(f)

$$S \subseteq \{0,1\}^n \times Q$$



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

Hard for
Strong Complexity
Measure

The Proof

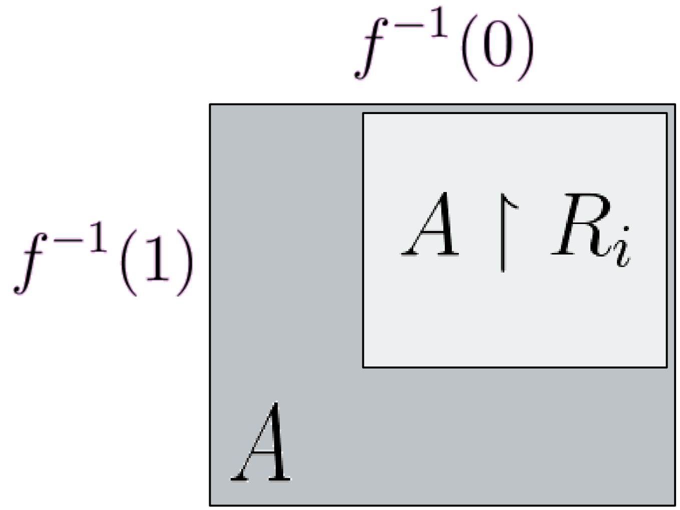
Lifting Theorem
(Our Setting)

Search Problem
S = Search(f)

$S \subseteq \{0,1\}^n \times Q$

?

Hard for
Weak Complexity
Measure



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

Hard for
Strong Complexity
Measure

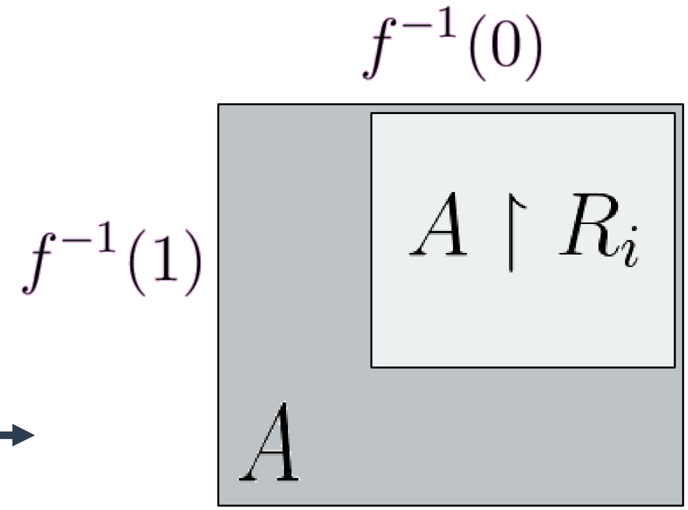
The Proof

Lifting Theorem
(Our Setting)

Search Problem
S = Search(f)
$$S \subseteq \{0,1\}^n \times Q$$

?

?



$$\mu_A(f) = \frac{\mathrm{rank}(A)}{\max_{i \in [n]} \mathrm{rank}(A \restriction R_i)}$$

Hard for
Weak Complexity
Measure

Hard for
Strong Complexity
Measure

The Proof

Lifting Theorem

(Our Setting)

Search Problem

S = Search(f)

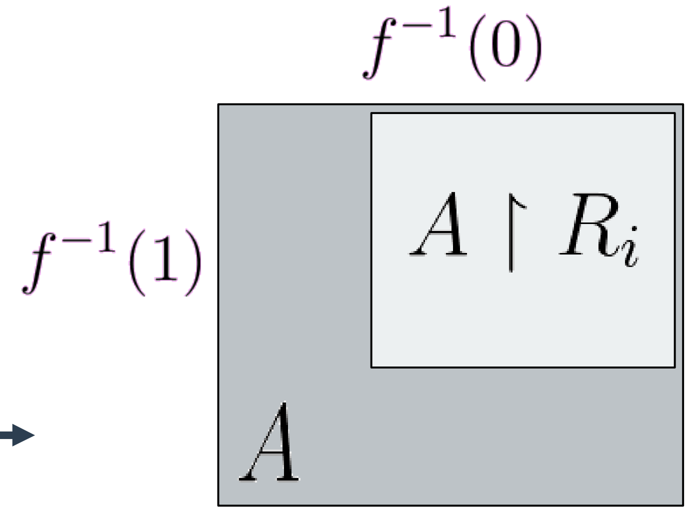
$S \subseteq \{0,1\}^n \times Q$

Polynomial

$p : \{0,1\}^n \to \mathbf{R}$

certifying a large algebraic gap for S

$$\xrightarrow{\quad ? \quad}$$



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

Hard for Weak Complexity Measure

Hard for Strong Complexity Measure

The Proof

Lifting Theorem

(Our Setting)

Search Problem

S = Search(f)

$$S \subseteq \{0,1\}^n \times Q$$

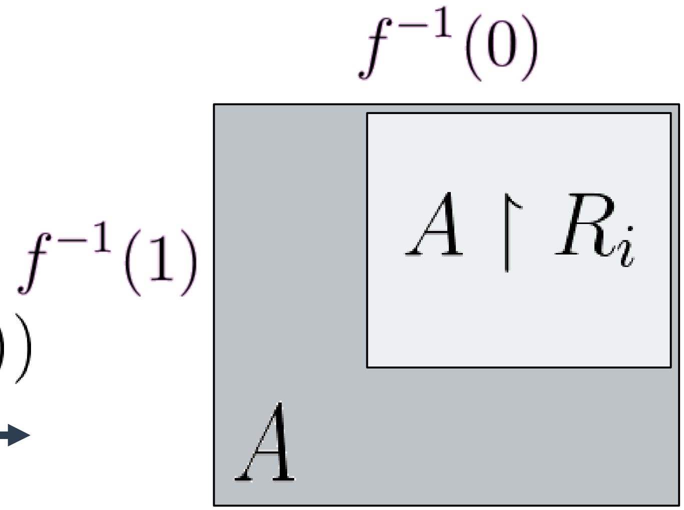
Polynomial

$$p : \{0,1\}^n \to \mathbf{R}$$

certifying a large algebraic gap for S

$$p(g(x_1, y_1), \ldots, g(x_n, y_n))$$

Compose p with two-input function g instead!



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

Hard for Weak Complexity Measure

Hard for Strong Complexity Measure

Lifting Theorem (ST-CONN)

<u>Theorem:</u> (**Lifting Theorem for Rank Measure**)

Consider layered ST-CONN on the $2m^2 \times m$ grid, and let k be the **algebraic gap complexity** of the ST-CONN search problem. There is a real matrix A such that

$$\mu_A(\text{ST-CONN}) \geq \frac{m^k}{6}$$

Lifting Theorem (ST-CONN)

<u>Theorem:</u> (**Lifting Theorem for Rank Measure**)

Consider layered ST-CONN on the $2m^2 \times m$ grid, and let k be the **algebraic gap complexity** of the ST-CONN search problem. There is a real matrix A such that

$$\mu_A(\text{ST-CONN}) \geq \frac{m^k}{6}$$

Proof: Intuition on previous slide, extension of the Pattern Matrix Method [Sherstov '08].

The Proof

Overview

<u>Rank Measure</u> [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f)
3. Actually **prove** the query lower bounds against Search(f)

The Proof

Overview

<u>Rank Measure</u> [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f) $\mu_A(f) \geq n^{\mathsf{gap}(f)}$
3. Actually **prove** the query lower bounds against Search(f)

The Proof

Lifting Theorem

Algebraic Gaps

The Proof

Lifting Theorem

Algebraic Gaps

Def: Let $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be an unsatisfiable CNF. Then Search(F) is the following problem: Given an assignment x to the variables of F, output the name of a clause falsified by x.

The Proof

Lifting Theorem

Algebraic Gaps

Def: Let $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be an unsatisfiable CNF. Then Search(F) is the following problem: Given an assignment x to the variables of F, output the name of a clause falsified by x.

Def: Let $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a total search problem. The **algebraic gap complexity** of Search(F) is the maximum k for which there is a polynomial $p : \{0,1\}^n \to \mathbf{R}$ such that

$$\deg(p) = n, \quad \deg(p\restriction_C) \leq n - k$$

for each certificate C of Search(F).

The Proof

Lifting Theorem

Algebraic Gaps

Def: Let $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a total search problem. The **algebraic gap complexity** of Search(F) is the maximum k for which there is a polynomial $p : \{0,1\}^n \to \mathbf{R}$ such that

$$\deg(p) = n, \quad \deg(p\restriction_C) \leq n - k$$

for each certificate C of Search(F).

The Proof

Lifting Theorem

Algebraic Gaps

<u>Def</u>: Let $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a total search problem. The **algebraic gap complexity** of Search(F) is the maximum k for which there is a polynomial $p : \{0,1\}^n \to \mathbf{R}$ such that

$$\deg(p) = n, \quad \deg(p\restriction_C) \leq n - k$$

for each certificate C of Search(F).

We give lower bounds on the algebraic gap complexity for the search problems corresponding to GEN and ST-CONN by reducing to **reversible pebbling**.

The Proof

Overview

Rank Measure [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f) $\mu_A(f) \geq n^{\mathsf{gap}(f)}$
3. Actually **prove** the query lower bounds against Search(f)

The Proof

Overview

<u>Rank Measure</u> [Razborov '90]**:**

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

1. Associate with certain special functions f (like GEN and ST-CONN) a search problem Search(f)
2. (**Lift**) Reduce constructing a good matrix A for f to lower bounding a complexity measure on Search(f) $\mu_A(f) \geq n^{\mathsf{gap}(f)}$
3. Actually **prove** the query lower bounds against Search(f) $\mathsf{gap}(\text{ST-CONN}) = \log n$

Conclusion

Conclusion

Unified lower bounds against monotone models by “lifting”.

Conclusion

Unified lower bounds against monotone models by “lifting”.

Algebraic gaps → other applications?

Conclusion

Unified lower bounds against monotone models by "lifting".

Algebraic gaps → other applications?

Average case lower bounds?

Conclusion

Unified lower bounds against monotone models by "lifting".

Algebraic gaps → other applications?

Average case lower bounds?

Sharpen lifting theorems further?

Conclusion

Unified lower bounds against monotone models by “lifting”.

Algebraic gaps → other applications?

Average case lower bounds?

Sharpen lifting theorems further?

Other algebraic query complexity measures for search problems?

Conclusion

Unified lower bounds against monotone models by “lifting”.

Algebraic gaps → other applications?

Average case lower bounds?

Sharpen lifting theorems further?

Other algebraic query complexity measures for search problems?

Thanks for listening!

References

Babai, Gal, Kollar, Ronyai, Szabo, Wigderson. *Extremal bipartite graphs and superpolynomial lower bounds for monotone span programs.* STOC '96.

Gal. *A characterization of span program size and improved lower bounds for monotone span programs.* STOC '98.

Potechin. *Bounds on monotone switching networks for directed connectivity.* FOCS '10.

Chan, Potechin. *Tight bounds for monotone switching networks via Fourier analysis.* STOC '12.

Karchmer, Wigderson. *Monotone circuits for connectivity require super-logarithmic depth.* STOC '88.

Karchmer, Wigderson. *On span programs.* Structure in Complexity Theory '93.

Raz, Mckenzie. *Separation of the monotone NC hierarchy.* FOCS '97.

Razborov. *Applications of matrix methods to the theory of lower bounds in computational complexity.* Combinatorica '90.

Sherstov. *The pattern matrix method for lower bounds on quantum communication.* STOC '08.