# A compression algorithm for AC$^0[p]$ circuits using Certifying Polynomials

Srikanth Srinivasan
Department of Mathematics, IIT Bombay.

September 30, 2015

# Meta-algorithmic problems

- Meta-algorithmic problems: Problems that take as input or output the description of a computational model.

# Meta-algorithmic problems

- Meta-algorithmic problems: Problems that take as input or output the description of a computational model.
- Example 1: Circuit SAT. Input: Boolean Circuit $C$, Qn: Is $f_C \equiv 0$?

# Meta-algorithmic problems

- Meta-algorithmic problems: Problems that take as input or output the description of a computational model.
- Example 1: Circuit SAT. Input: Boolean Circuit $C$, Qn: Is $f_C \equiv 0$?
- Example 2: PIT. Input: Algebraic circuit $C$, Qn: Is $P_C \equiv 0$?

# Meta-algorithmic problems

- Meta-algorithmic problems: Problems that take as input or output the description of a computational model.

- Example 1: Circuit SAT. Input: Boolean Circuit $C$, Qn: Is $f_C \equiv 0$?
- Example 2: PIT. Input: Algebraic circuit $C$, Qn: Is $P_C \equiv 0$?
- Example 3: Minimum Circuit Size problem (MCSP).

# Meta-algorithmic problems

- Meta-algorithmic problems: Problems that take as input or output the description of a computational model.
- Example 1: Circuit SAT. Input: Boolean Circuit $C$, Qn: Is $f_C \equiv 0$?
- Example 2: PIT. Input: Algebraic circuit $C$, Qn: Is $P_C \equiv 0$?
- Example 3: Minimum Circuit Size problem (MCSP).
  - Input: $f : \{0,1\}^n \to \{0,1\}$, $k$.
  - Qn: Does $f$ have a circuit of size at most $k$?

# Compression Algorithm for a circuit class $\mathcal{C}$

- $\mathcal{C}$: a class of circuits ($\mathsf{AC}^0$, $\mathsf{AC}^0[p]$, etc.).

# Compression Algorithm for a circuit class $\mathcal{C}$

- $\mathcal{C}$: a class of circuits (AC$^0$, AC$^0[p]$, etc.).
- Compression problem for $\mathcal{C}$ (Chen, Kabanets, Kolokolova, Shaltiel, Zuckerman (2014))

# Compression Algorithm for a circuit class $\mathcal{C}$

- $\mathcal{C}$: a class of circuits (AC$^0$, AC$^0[p]$, etc.).
- Compression problem for $\mathcal{C}$ (Chen, Kabanets, Kolokolova, Shaltiel, Zuckerman (2014))
  - Input: $f : \{0,1\}^n \to \{0,1\}$ with small Boolean circuits from $\mathcal{C}$.

# Compression Algorithm for a circuit class $\mathcal{C}$

- $\mathcal{C}$: a class of circuits ($\mathsf{AC}^0$, $\mathsf{AC}^0[p]$, etc.).
- Compression problem for $\mathcal{C}$ (Chen, Kabanets, Kolokolova, Shaltiel, Zuckerman (2014))
  - Input: $f : \{0,1\}^n \to \{0,1\}$ with small Boolean circuits from $\mathcal{C}$.
  - Qn: Construct a non-trivially small *general* circuit for $f$.

# Compression Algorithm for a circuit class $\mathcal{C}$

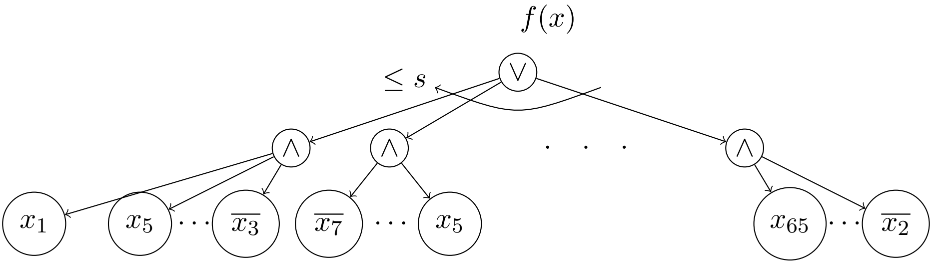
- $\mathcal{C}$: a class of circuits (AC$^0$, AC$^0[p]$, etc.).
- Compression problem for $\mathcal{C}$ (Chen, Kabanets, Kolokolova, Shaltiel, Zuckerman (2014))
  - Input: $f : \{0,1\}^n \to \{0,1\}$ with small Boolean circuits from $\mathcal{C}$.
  - Qn: Construct a non-trivially small *general* circuit for $f$.
- Non-trivially small: size $\ll 2^n/n$.

# Compression Algorithm for a circuit class $\mathcal{C}$

- $\mathcal{C}$: a class of circuits (AC$^0$, AC$^0[p]$, etc.).
- Compression problem for $\mathcal{C}$ (Chen, Kabanets, Kolokolova, Shaltiel, Zuckerman (2014))
  - Input: $f : \{0,1\}^n \to \{0,1\}$ with small Boolean circuits from $\mathcal{C}$.
  - Qn: Construct a non-trivially small *general* circuit for $f$.
- Non-trivially small: size $\ll 2^n/n$.
- (Chen et al.) Compression algorithms imply circuit lower bounds.

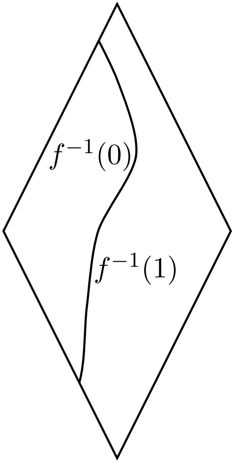
# Compression problem for DNFs

- Input: $f : \{0,1\}^n \to \{0,1\}$ with DNFs of size $s$.
- Qn: Construct a non-trivially small DNF for $f$.



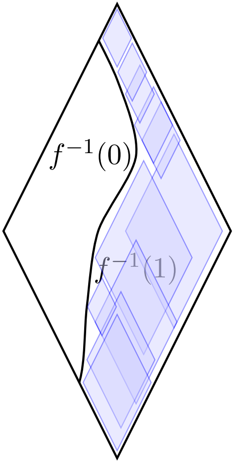
# Compression problem for DNFs

- DNF of size $s$: a union of $s$ subcubes.



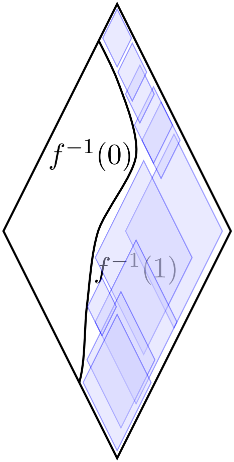
## Compression problem for DNFs

- DNF of size $s$: a union of $s$ subcubes.



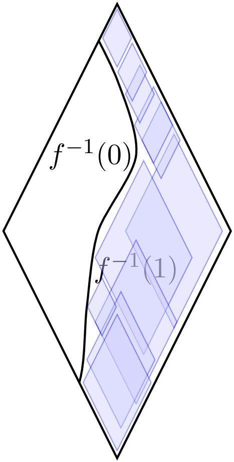
# Compression problem for DNFs

- DNF of size $s$: a union of $s$ subcubes.
- Finding optimal sized DNF for $f$:
  Covering $f$ with subcubes in $f^{-1}(1)$.



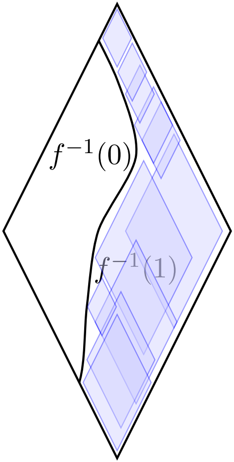
# Compression problem for DNFs

- DNF of size $s$: a union of $s$ subcubes.
- Finding optimal sized DNF for $f$: Covering $f$ with subcubes in $f^{-1}(1)$.
- Number of subcubes: $3^n$.



# Compression problem for DNFs

- DNF of size $s$: a union of $s$ subcubes.
- Finding optimal sized DNF for $f$: Covering $f$ with subcubes in $f^{-1}(1)$.
- Number of subcubes: $3^n$.
- (Lovász 1975) $O(n)$-approximation in time $2^{O(n)}$.



# Compression problem for DNFs

- Input: $f : \{0,1\}^n \to \{0,1\}$ with DNFs of size $s$.
- Qn: Construct a non-trivially small DNF for $f$.
- Say $s = 2^{n-t} = 2^n/\text{superpoly}(n)$.

# Compression problem for DNFs

- Input: $f : \{0,1\}^n \to \{0,1\}$ with DNFs of size $s$.
- Qn: Construct a non-trivially small DNF for $f$.

- Say $s = 2^{n-t} = 2^n/\mathrm{superpoly}(n)$.
- Can obtain a DNF of size $2^{n-\Theta(t)}$.

# Compression algorithms for other classes of circuits (Chen et al. (2014))

- Input: $f : \{0,1\}^n \to \{0,1\}$ with $\mathcal{C}$-circuits of size $s$.
- Qn: Construct a non-trivially small circuit for $f$.
- Say $\mathcal{C}$-circuits of size $s \Rightarrow$ DNFs of size $2^{n-t}$.

# Compression algorithms for other classes of circuits (Chen et al. (2014))

- Input: $f : \{0,1\}^n \to \{0,1\}$ with $\mathcal{C}$-circuits of size $s$.
- Qn: Construct a non-trivially small circuit for $f$.

- Say $\mathcal{C}$-circuits of size $s \Rightarrow$ DNFs of size $2^{n-t}$.
- Can obtain a DNF of size $2^{n-\Theta(t)}$.

## Compression algorithms for other classes of circuits (Chen et al. (2014))

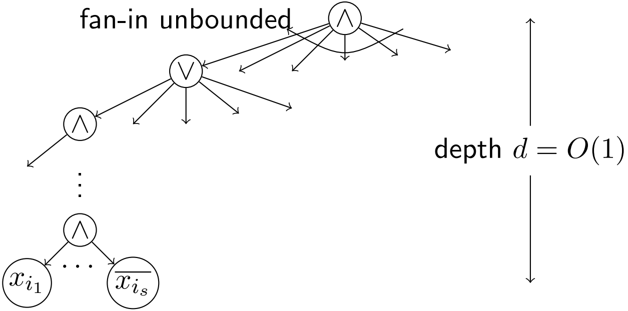
Gives non-trivial compression algorithms for:

- $AC^0$ circuits of size $2^{n^{o(1)}}$ (Impagliazzo-Matthews-Paturi, Håstad).

# Compression algorithms for other classes of circuits (Chen et al. (2014))

Gives non-trivial compression algorithms for:

- $\mathrm{AC}^0$ circuits of size $2^{n^{o(1)}}$ (Impagliazzo-Matthews-Paturi, Håstad).



# Compression algorithms for other classes of circuits (Chen et al. (2014))

Gives non-trivial compression algorithms for:

- $\mathrm{AC}^0$ circuits of size $2^{n^{o(1)}}$ (Impagliazzo-Matthews-Paturi, Håstad).

**Theorem (Chen et al. (2014))**

*Say $f$ has an $AC^0$ circuit of size $s$. Then the algorithm outputs a circuit of size $2^n/M$, where $M = \exp(n/(C\log(s/n))^{d-1})$.*

## Compression algorithms for other classes of circuits (Chen et al. (2014))

Gives non-trivial compression algorithms for:

- $\mathrm{AC}^0$ circuits of size $2^{n^{o(1)}}$ (Impagliazzo-Matthews-Paturi, Håstad).

**Theorem (Chen et al. (2014))**

*Say $f$ has an $AC^0$ circuit of size $s$. Then the algorithm outputs a circuit of size $2^n/M$, where $M = \exp(n/(C\log(s/n))^{d-1})$.*

- DeMorgan formulas of size $\ll n^{1.5}$. (Subbotovskaya, Santhanam)

## Compression algorithms for other classes of circuits (Chen et al. (2014))

Gives non-trivial compression algorithms for:

- $\mathrm{AC}^0$ circuits of size $2^{n^{o(1)}}$ (Impagliazzo-Matthews-Paturi, Håstad).

**Theorem (Chen et al. (2014))**

*Say $f$ has an $AC^0$ circuit of size $s$. Then the algorithm outputs a circuit of size $2^n/M$, where $M = \exp(n/(C\log(s/n))^{d-1})$.*

- DeMorgan formulas of size $\ll n^{1.5}$. (Subbotovskaya, Santhanam)
- Further algorithms using memoization.

# Compression algorithms for other classes of circuits (Chen et al. (2014))

Gives non-trivial compression algorithms for:

- $\mathrm{AC}^0$ circuits of size $2^{n^{o(1)}}$ (Impagliazzo-Matthews-Paturi, Håstad).

**Theorem (Chen et al. (2014))**

*Say $f$ has an $AC^0$ circuit of size $s$. Then the algorithm outputs a circuit of size $2^n/M$, where $M = \exp(n/(C\log(s/n))^{d-1})$.*

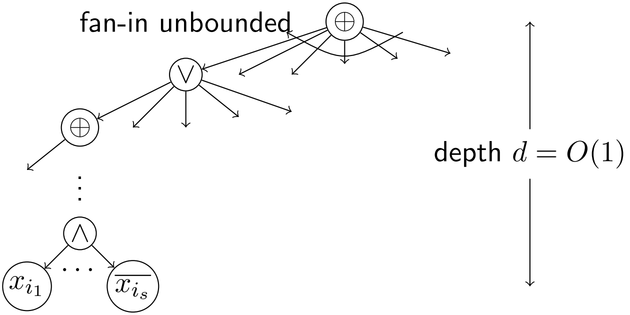
- DeMorgan formulas of size $\ll n^{1.5}$. (Subbotovskaya, Santhanam)
- Further algorithms using memoization.
- General formulas, branching programs.

# More general classes of circuits

- Compression algorithms for more powerful classes of circuits?

# More general classes of circuits

- Compression algorithms for more powerful classes of circuits?
- Natural next question: AC$^0[2]$: AC$^0$ augmented with $\oplus$ gates.



# Compression algorithms for $\mathrm{AC}^0[p]$ circuits

## Theorem (Chen et al. (2014))

*Say $f$ has an $AC^0$ circuit of size $s$. Then we can find in poly$(2^n)$ time a circuit of size $2^n/M$, where $M = \exp(n/(C\log(s/n))^{d-1})$.*

# Compression algorithms for $\mathrm{AC}^0[p]$ circuits

**Theorem (Chen et al. (2014))**

*Say $f$ has an $AC^0$ circuit of size $s$. Then we can find in poly$(2^n)$ time a circuit of size $2^n/M$, where $M = \exp(n/(C\log(s/n))^{d-1})$.*

**Theorem (This work)**

*Say $f$ has an $AC^0[2]$ circuit of size $s$. Then we can find in poly$(2^n)$ time a circuit of size $2^n/M$, where $M = \exp(n/(C\log s)^{2(d-1)})$.*

# Compression algorithms for $\mathsf{AC}^0[p]$ circuits

**Theorem (Chen et al. (2014))**

*Say $f$ has an $AC^0$ circuit of size $s$. Then we can find in poly$(2^n)$ time a circuit of size $2^n/M$, where $M = \exp(n/(C\log(s/n))^{d-1})$.*

**Theorem (This work)**

*Say $f$ has an $AC^0[2]$ circuit of size $s$. Then we can find in poly$(2^n)$ time a circuit of size $2^n/M$, where $M = \exp(n/(C\log s)^{2(d-1)})$.*

Also works for $\mathsf{AC}^0[p]$ ($p$ prime).

# Polynomials and polynomial approximations

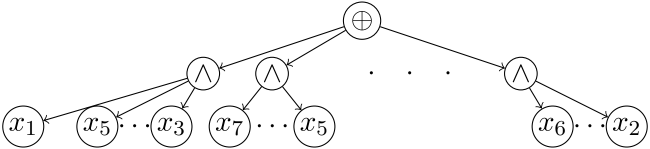
- $P(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$. Multilinear.

# Polynomials and polynomial approximations

- $P(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$. Multilinear.
- E.g.: $x_1x_2 + x_3 + x_1x_5$.

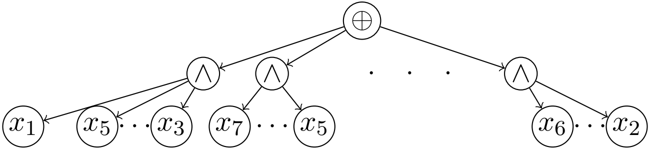
# Polynomials and polynomial approximations

- $P(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$. Multilinear.
- E.g.: $x_1 x_2 + x_3 + x_1 x_5$.
- Special kind of depth-2 $\mathrm{AC}^0[2]$ circuit.



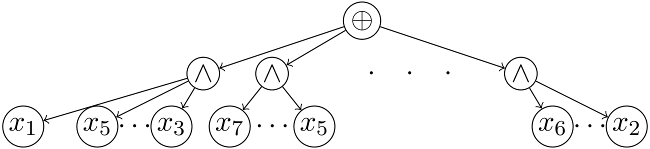
# Polynomials and polynomial approximations

- $P(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$. Multilinear.
- E.g.: $x_1 x_2 + x_3 + x_1 x_5$.
- Special kind of depth-2 $\mathrm{AC}^0[2]$ circuit.
- Degree $D \Rightarrow$ size $\sum_{i \le D} \binom{n}{i} = \binom{n}{\le D}$.



# Polynomials and polynomial approximations

- $P(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$. Multilinear.
- E.g.: $x_1x_2 + x_3 + x_1x_5$.
- Special kind of depth-$2$ $\mathsf{AC}^0[2]$ circuit.
- Degree $D \Rightarrow$ size $\sum_{i \le D} \binom{n}{i} = \binom{n}{\le D}$.
- Say that $P$ $\varepsilon$-approximates $f$ if $\Pr_{x \in \{0,1\}^n}[P(x) \neq f(x)] \le \varepsilon$.

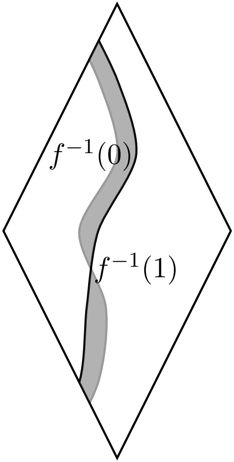


# Razborov approximations to $\mathrm{AC}^0[2]$ circuits

- (Razborov 1987): Can $\varepsilon$-approximate small $\mathrm{AC}^0[2]$ circuits by low-degree polynomials.

# Razborov approximations to $\mathrm{AC}^0[2]$ circuits

- (Razborov 1987): Can $\varepsilon$-approximate small $\mathrm{AC}^0[2]$ circuits by low-degree polynomials.
- Circuit has size $n^{O(1)} \Rightarrow$ degree of polynomial is $O(\log n)^{d-1} \log(1/\varepsilon)$.

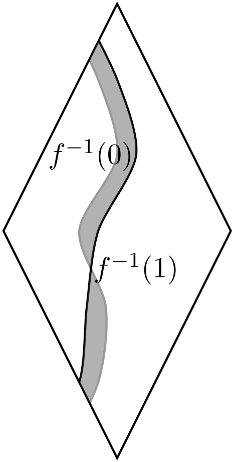


# Natural approach to compression problem

- Given $f : \{0,1\}^n \to \{0,1\}$ with $\mathsf{AC}^0[2]$ circuits of size $s = n^{O(1)}$.

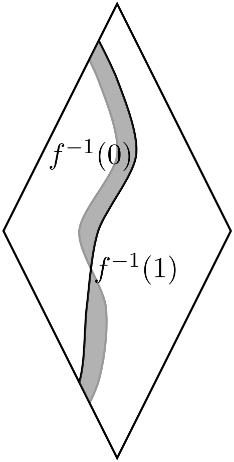
# Natural approach to compression problem

- Given $f : \{0,1\}^n \to \{0,1\}$ with $\mathrm{AC}^0[2]$ circuits of size $s = n^{O(1)}$.
- Find low-degree $\varepsilon$-approximation $P$.



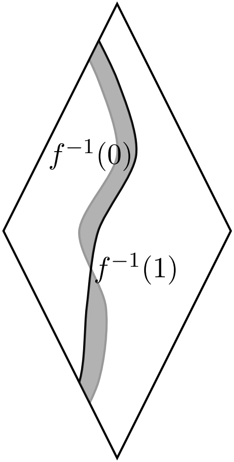
# Natural approach to compression problem

- Given $f : \{0,1\}^n \to \{0,1\}$ with $\mathsf{AC}^0[2]$ circuits of size $s = n^{O(1)}$.
- Find low-degree $\varepsilon$-approximation $P$.
- $\mathrm{Size}(P) = \exp((\log n)^{O(1)} \log(1/\varepsilon))$.



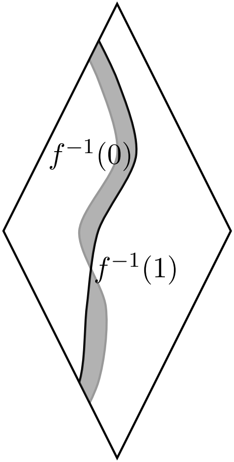
# Natural approach to compression problem

- Given $f : \{0,1\}^n \to \{0,1\}$ with AC$^0[2]$ circuits of size $s = n^{O(1)}$.
- Find low-degree $\varepsilon$-approximation $P$.
- $\mathrm{Size}(P) = \exp((\log n)^{O(1)} \log(1/\varepsilon))$.
- "Fix" $P$ at all the error points with a circuit of size $\varepsilon 2^n$.



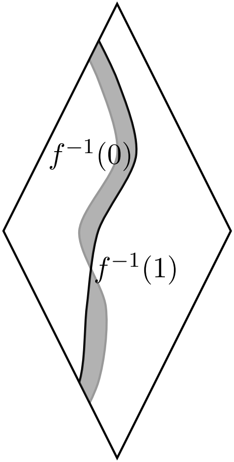
# Natural approach to compression problem

- Given $f : \{0,1\}^n \to \{0,1\}$ with $\mathsf{AC}^0[2]$ circuits of size $s = n^{O(1)}$.
- Find low-degree $\varepsilon$-approximation $P$.
- $\mathrm{Size}(P) = \exp((\log n)^{O(1)} \log(1/\varepsilon))$.
- "Fix" $P$ at all the error points with a circuit of size $\varepsilon 2^n$.
- Overall size: $\mathrm{Size}(P) + \varepsilon 2^n$.



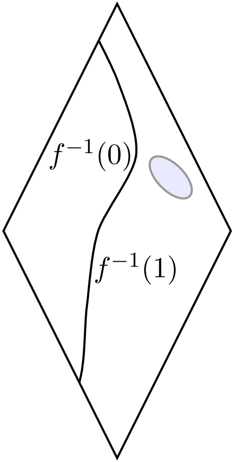
# Natural approach to compression problem

- Given $f : \{0,1\}^n \to \{0,1\}$ with AC$^0[2]$ circuits of size $s = n^{O(1)}$.
- Find low-degree $\varepsilon$-approximation $P$.
- $\mathrm{Size}(P) = \exp((\log n)^{O(1)} \log(1/\varepsilon))$.
- "Fix" $P$ at all the error points with a circuit of size $\varepsilon 2^n$.
- Overall size: $\mathrm{Size}(P) + \varepsilon 2^n$.
- Bottleneck: How to find $P$?



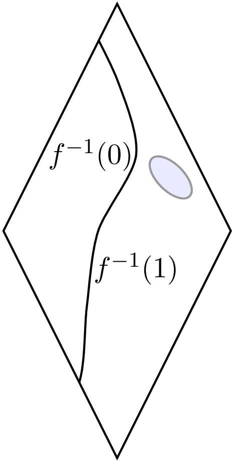
# Certifying polynomials (ABFR '94, Green '95)

- $R \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a Certifying polynomial for $f$ if $R(x) = 1 \Rightarrow f(x) = 1$.



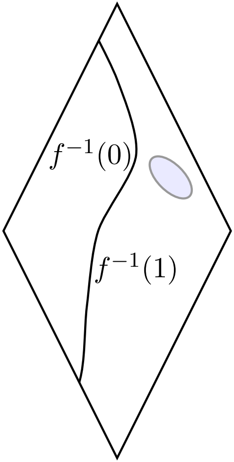
# Certifying polynomials (ABFR '94, Green '95)

- $R \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a Certifying polynomial for $f$ if $R(x) = 1 \Rightarrow f(x) = 1$.
- $R \neq 0$.



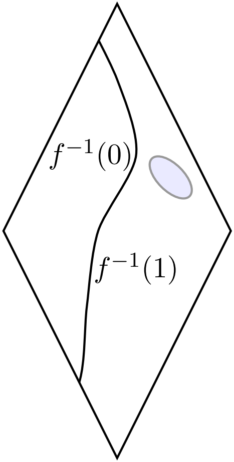
# Certifying polynomials (ABFR '94, Green '95)

- $R \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a Certifying polynomial for $f$ if $R(x) = 1 \Rightarrow f(x) = 1$.
- $R \neq 0$.
- Also studied as Algebraic Immunity, Weak degree.



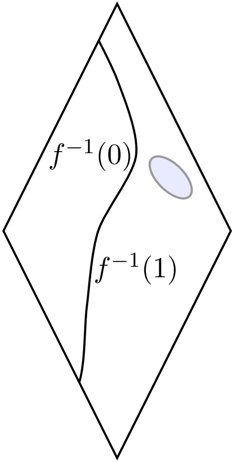
# Certifying polynomials (ABFR '94, Green '95)

- $R \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a Certifying polynomial for $f$ if $R(x) = 1 \Rightarrow f(x) = 1$.
- $R \neq 0$.
- Also studied as Algebraic Immunity, Weak degree.
- Notion of one-sided approximation.



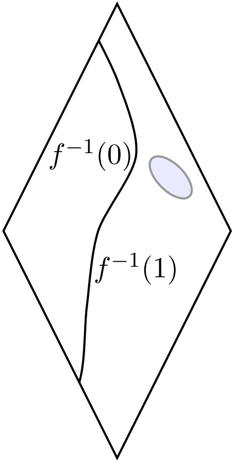
## Certifying polynomials (ABFR '94, Green '95)

- $R \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a Certifying polynomial for $f$ if $R(x) = 1 \Rightarrow f(x) = 1$.
- $R \neq 0$.
- Also studied as Algebraic Immunity, Weak degree.
- Notion of one-sided approximation.
- Any (*) function $f$ has a certifying polynomial of degree at most $n/2$.



# Certifying polynomials (ABFR '94, Green '95)

- $R \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a Certifying polynomial for $f$ if $R(x) = 1 \Rightarrow f(x) = 1$.
- $R \neq 0$.
- Also studied as Algebraic Immunity, Weak degree.
- Notion of one-sided approximation.
- Any (*) function $f$ has a certifying polynomial of degree at most $n/2$.
- Gives a "local" circuit for $f$ of size $\binom{n}{\leq n/2} = 2^{n-1}$.



# Certifying polynomials

- Any function $f$ has a certifying polynomial of degree at most $n/2$.
- Gives a "local" circuit for $f$ of size $\binom{n}{\leq n/2} = 2^{n-1}$.

# Certifying polynomials

- Any function $f$ has a certifying polynomial of degree at most $n/2$.
- Gives a "local" circuit for $f$ of size $\binom{n}{\leq n/2} = 2^{n-1}$.

- Thm (Kopparty-S. '12): $f$ has AC$^0[2]$ circuit of size poly$(n) \Rightarrow$ certifying polynomials of degree $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$.

# Certifying polynomials

- Any function $f$ has a certifying polynomial of degree at most $n/2$.
- Gives a "local" circuit for $f$ of size $\binom{n}{\leq n/2} = 2^{n-1}$.

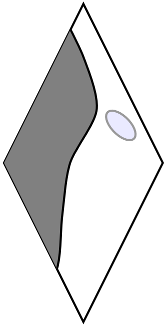
- Thm (Kopparty-S. '12): $f$ has $\mathsf{AC}^0[2]$ circuit of size poly$(n) \Rightarrow$ certifying polynomials of degree $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$.
- Gives a circuit of size $2^n / \exp(n/(\log n)^{O(1)})$.

# Finding Certifying polynomials

- Thm (Kopparty-S. '12): $f$ has AC$^0[2]$ circuit of size poly$(n) \Rightarrow$ certifying polynomials of degree $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$.
- Gives a circuit of size $2^n / \exp(n/(\log n)^{O(1)})$.

# Finding Certifying polynomials

- Thm (Kopparty-S. '12): $f$ has AC$^0[2]$ circuit of size poly$(n) \Rightarrow$ certifying polynomials of degree $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$.
- Gives a circuit of size $2^n / \exp(n/(\log n)^{O(1)})$.

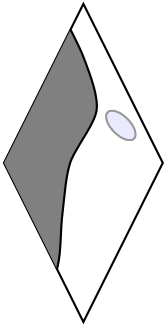


- Finding $R = \sum_{|S| \leq D} \alpha_S \prod_{i \in S} x_i$.

# Finding Certifying polynomials

- Thm (Kopparty-S. '12): $f$ has AC$^0[2]$ circuit of size poly$(n)$ $\Rightarrow$ certifying polynomials of degree $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$.
- Gives a circuit of size $2^n / \exp(n/(\log n)^{O(1)})$.

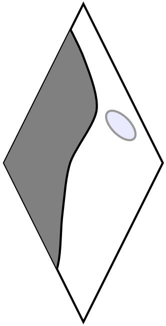
- Finding $R = \sum_{|S| \leq D} \alpha_S \prod_{i \in S} x_i$.
- Need $R(x) = 0$ for all $x \in f^{-1}(0)$.



# Finding Certifying polynomials

- Thm (Kopparty-S. '12): $f$ has AC$^0[2]$ circuit of size poly$(n)$ $\Rightarrow$ certifying polynomials of degree $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$.
- Gives a circuit of size $2^n / \exp(n/(\log n)^{O(1)})$.

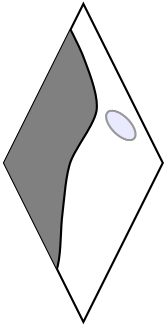
- Finding $R = \sum_{|S| \le D} \alpha_S \prod_{i \in S} x_i$.
- Need $R(x) = 0$ for all $x \in f^{-1}(0)$.
- Has solution set $V_D$.



# Finding Certifying polynomials

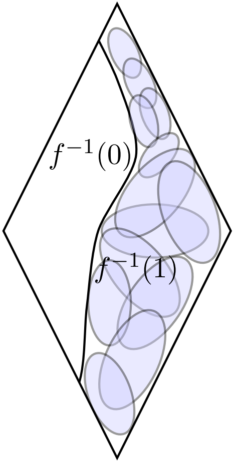
- Thm (Kopparty-S. '12): $f$ has AC$^0[2]$ circuit of size poly$(n) \Rightarrow$ certifying polynomials of degree $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$.
- Gives a circuit of size $2^n / \exp(n/(\log n)^{O(1)})$.

- Finding $R = \sum_{|S| \leq D} \alpha_S \prod_{i \in S} x_i$.
- Need $R(x) = 0$ for all $x \in f^{-1}(0)$.
- Has solution set $V_D$.
- Need non-zero element of $V_D$.



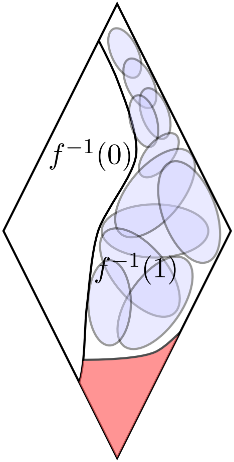
# Using $V_D$ to compress

- Each $R \in V_D$ covers a subset of $f^{-1}(1)$.
- Select a few $R_1, \ldots, R_m \in V_D$ such that $\bigvee_i R_i = f$.



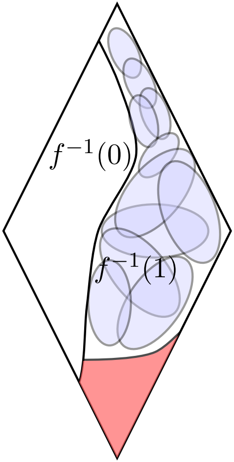
# Problems with the approach

- There are points in $f^{-1}(1)$ that are never covered by $R \in V_D$.
  - $F = \{x \in f^{-1}(1) \mid R \in V_D \Rightarrow R(x) = 0\}$.



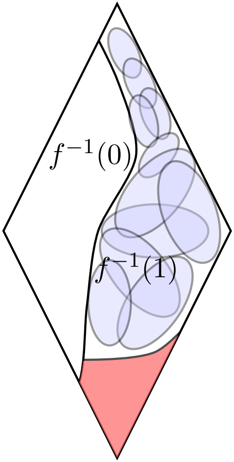
## Problems with the approach

- There are points in $f^{-1}(1)$ that are never covered by $R \in V_D$.
  - $F = \{x \in f^{-1}(1) \mid R \in V_D \Rightarrow R(x) = 0\}$.
  - Need to say that $F$ is small.



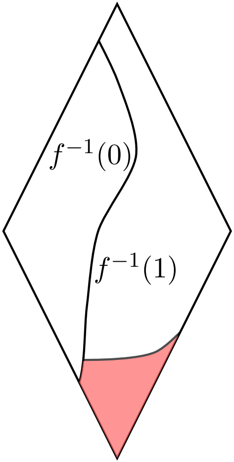
# Problems with the approach

- There are points in $f^{-1}(1)$ that are never covered by $R \in V_D$.
  - $F = \{x \in f^{-1}(1) \mid R \in V_D \Rightarrow R(x) = 0\}$.
  - Need to say that $F$ is small.
- Each $R \in V_D$ might cover only small subset of $f^{-1}(1) \setminus F$.



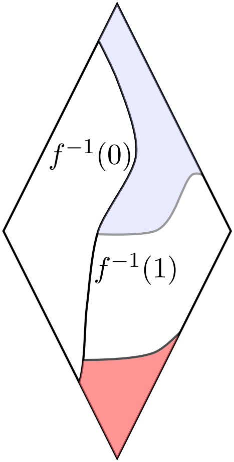
# Handling second issue

- A random $R \in V_D$ covers each $x \notin F$ with probability $\frac{1}{2}$.



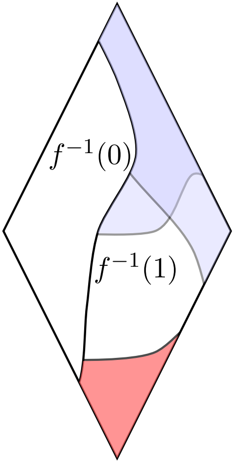
# Handling second issue

- A random $R \in V_D$ covers each $x \notin F$ with probability $\frac{1}{2}$.



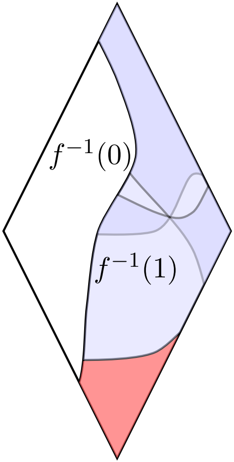
# Handling second issue

- A random $R \in V_D$ covers each $x \notin F$ with probability $\frac{1}{2}$.



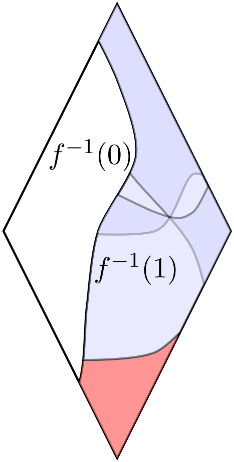
# Handling second issue

- A random $R \in V_D$ covers each $x \notin F$ with probability $\frac{1}{2}$.



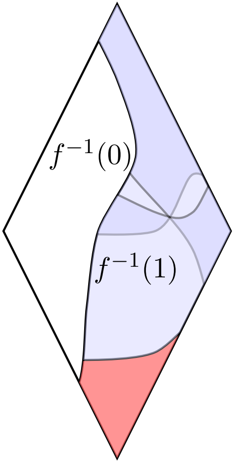
# Handling second issue

- A random $R \in V_D$ covers each $x \notin F$ with probability $\frac{1}{2}$.
- Picking $R_1, \ldots, R_{O(n)} \in_u V_D$ covers $f^{-1}(1)$ with high probability.



# Handling second issue

- A random $R \in V_D$ covers each $x \notin F$ with probability $\frac{1}{2}$.
- Picking $R_1, \ldots, R_{O(n)} \in_u V_D$ covers $f^{-1}(1)$ with high probability.
- Can be easily derandomized using Error-Correcting codes.



# Overall approach summary

- Argue that for $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$, $F$ is small.

# Overall approach summary

- Argue that for $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$, $F$ is small.
- Obtain $m = O(n)$ polynomials $R_1, \ldots, R_m \in V_D$ covering $f^{-1}(1) \setminus F$.

# Overall approach summary

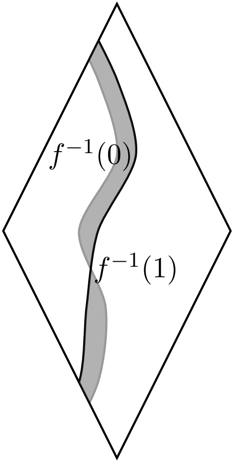
- Argue that for $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$, $F$ is small.
- Obtain $m = O(n)$ polynomials $R_1, \ldots, R_m \in V_D$ covering $f^{-1}(1) \setminus F$.
- Output $C = \bigvee_i R_i \vee \varphi$, where $\varphi$ is a brute-force DNF accepting $F$.

# Overall approach summary

- Argue that for $D = \frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$, $F$ is small.
- Obtain $m = O(n)$ polynomials $R_1, \ldots, R_m \in V_D$ covering $f^{-1}(1) \setminus F$.
- Output $C = \bigvee_i R_i \vee \varphi$, where $\varphi$ is a brute-force DNF accepting $F$.
- $\mathrm{Size}(C) = 2^n / \exp(n/(\log n)^{O(1)}) + |F|$.

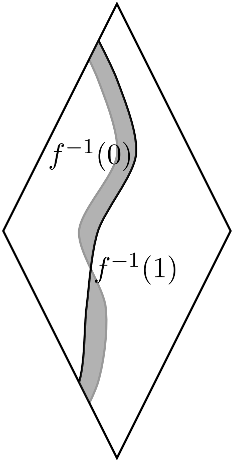
# Constructing certifying polynomials

- $f$ has an $\mathrm{AC}^0[2]$ circuit of size $\mathrm{poly}(n)$.
  - $f$ has $\varepsilon$-approximating polynomial $P$ of degree $D_1 = (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $E$ = error set of $P$. $|E| < \varepsilon 2^n$.



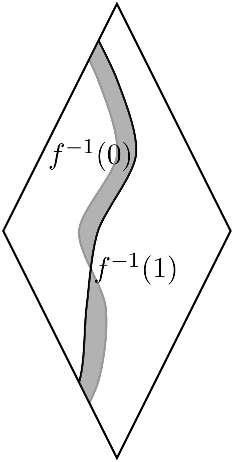
# Constructing certifying polynomials

- $f$ has an $\mathrm{AC}^0[2]$ circuit of size $\mathrm{poly}(n)$.
  - $f$ has $\varepsilon$-approximating polynomial $P$ of degree $D_1 = (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $E$ = error set of $P$. $|E| < \varepsilon 2^n$.
- Find non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$.



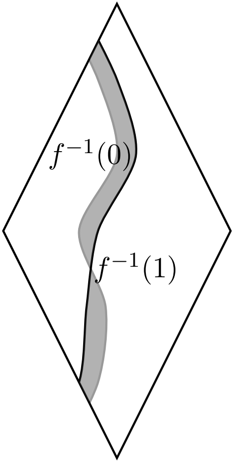
# Constructing certifying polynomials

- $f$ has an $\mathrm{AC}^0[2]$ circuit of size poly$(n)$.
  - $f$ has $\varepsilon$-approximating polynomial $P$ of degree $D_1 = (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $E =$ error set of $P$. $|E| < \varepsilon 2^n$.
- Find non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$.
- Need $\binom{n}{\leq D_2} \geq \varepsilon 2^n$.
- $D_2 = \frac{n}{2} - \Theta(\sqrt{n \log(1/\varepsilon)})$.



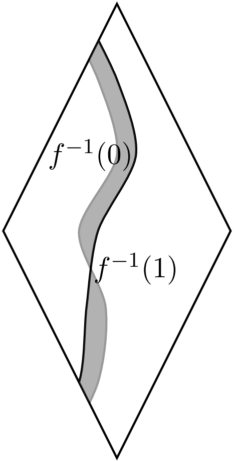
# Constructing certifying polynomials

- $f$ has an $\mathsf{AC}^0[2]$ circuit of size $\mathsf{poly}(n)$.
  - $f$ has $\varepsilon$-approximating polynomial $P$ of degree $D_1 = (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $E$ = error set of $P$. $|E| < \varepsilon 2^n$.
- Find non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$.
- $D_2 = \frac{n}{2} - \Theta(\sqrt{n \log(1/\varepsilon)})$.



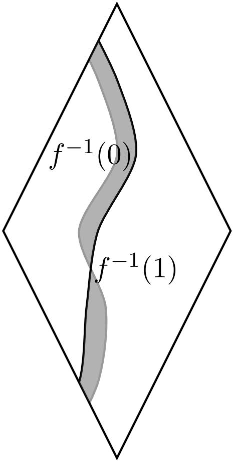
# Constructing certifying polynomials

- $f$ has an $\mathrm{AC}^0[2]$ circuit of size poly$(n)$.
  - $f$ has $\varepsilon$-approximating polynomial $P$ of degree $D_1 = (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $E$ = error set of $P$. $|E| < \varepsilon 2^n$.
- Find non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$.
- $D_2 = \frac{n}{2} - \Theta(\sqrt{n \log(1/\varepsilon)})$.
- $R = Q \cdot P$. $R = 1 \Rightarrow f = 1$.



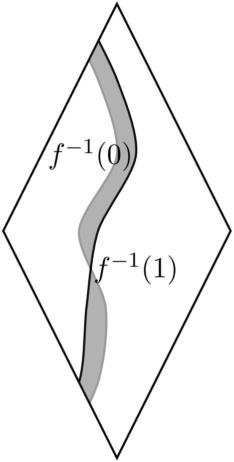
# Constructing certifying polynomials

- $f$ has an $\mathsf{AC}^0[2]$ circuit of size poly$(n)$.
  - $f$ has $\varepsilon$-approximating polynomial $P$ of degree $D_1 = (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $E$ = error set of $P$. $|E| < \varepsilon 2^n$.
- Find non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$.
- $D_2 = \frac{n}{2} - \Theta(\sqrt{n \log(1/\varepsilon)})$.
- $R = Q \cdot P$. $R = 1 \Rightarrow f = 1$.
  - $\deg = \frac{n}{2} - \sqrt{n \log(1/\varepsilon)} + (\log n)^{O(1)} \log(1/\varepsilon)$.



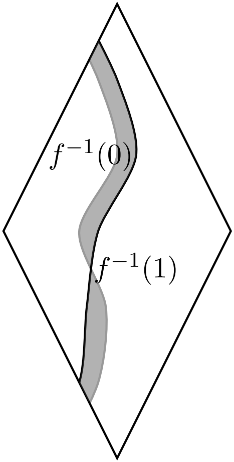
# Constructing certifying polynomials

- $f$ has an $\mathsf{AC}^0[2]$ circuit of size poly$(n)$.
  - $f$ has $\varepsilon$-approximating polynomial $P$ of degree $D_1 = (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $E$ = error set of $P$. $|E| < \varepsilon 2^n$.
- Find non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$.
- $D_2 = \frac{n}{2} - \Theta(\sqrt{n \log(1/\varepsilon)})$.
- $R = Q \cdot P$. $R = 1 \Rightarrow f = 1$.
  - $\deg = \frac{n}{2} - \sqrt{n \log(1/\varepsilon)} + (\log n)^{O(1)} \log(1/\varepsilon)$.
  - $\frac{n}{2} - \frac{n}{(\log n)^{O(1)}}$ if $\varepsilon = \exp(-n/(\log n)^{O(1)})$.



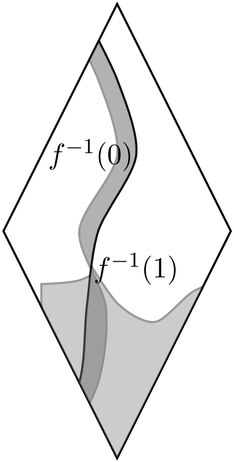
# Bounding $|F|$

- Need non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$. $R = Q \cdot P$.



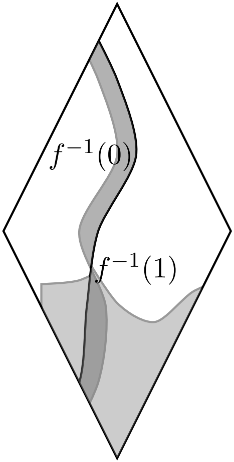
# Bounding $|F|$

- Need non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$. $R = Q \cdot P$.
- Forces $Q$ to be zero on $F'$.



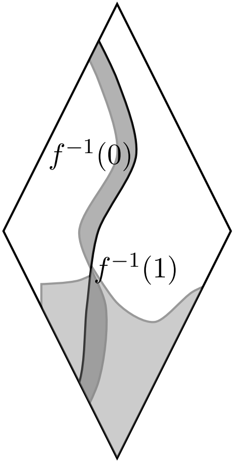
# Bounding $|F|$

- Need non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$. $R = Q \cdot P$.
- Forces $Q$ to be zero on $F'$.
- $x \notin F' \Rightarrow Q(x) = 1$ for some $Q$ s.t. $Q|_E = 0$.



# Bounding $|F|$

- Need non-zero $Q$ of degree $D_2$ s.t. $Q|_E = 0$. $R = Q \cdot P$.
- Forces $Q$ to be zero on $F'$.
- $x \notin F' \Rightarrow Q(x) = 1$ for some $Q$ s.t. $Q|_E = 0$.
- $F \subseteq F'$.



# The problem and its solution

- $E \subseteq \mathbb{F}_2^n$. $|E| \leq \varepsilon 2^n$.

# The problem and its solution

- $E \subseteq \mathbb{F}_2^n$. $|E| \leq \varepsilon 2^n$.
- $F' = \{x \mid \forall Q$ of deg $D_2$, $Q|_E = 0 \Rightarrow Q(x) = 0\}$.

# The problem and its solution

- $E \subseteq \mathbb{F}_2^n$. $|E| \leq \varepsilon 2^n$.
- $F' = \{x \mid \forall Q$ of deg $D_2$, $Q|_E = 0 \Rightarrow Q(x) = 0\}$.
- How large can $|F'|$ be?

**Theorem (Nie-Wang 2014)**

$$\frac{|F'|}{2^n} \leq \frac{|E|}{\binom{n}{\leq D_2}}.$$

Choose $D_2$ so that $\binom{n}{\leq D_2} = \sqrt{\varepsilon} 2^n$.

# Open questions

- Compression algorithms for other/stronger circuit classes?

# Open questions

- Compression algorithms for other/stronger circuit classes?
- For $\mathrm{Maj} \circ \mathsf{AC}^0$?

# Open questions

- Compression algorithms for other/stronger circuit classes?
- For $\mathrm{Maj} \circ \mathsf{AC}^0$?
- Other applications of the Nie-Wang result?

## Open questions

- Compression algorithms for other/stronger circuit classes?
- For $\mathrm{Maj} \circ \mathsf{AC}^0$?
- Other applications of the Nie-Wang result?

# Thank you