A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds

Jakob Nordström

KTH Royal Institute of Technology
Stockholm, Sweden

Workshop on Proof Complexity
Special Semester Program on Complexity Theory
St Petersburg, Russia
May 19, 2016

Joint work with Mladen Mikša

Proof Complexity and Expansion

- **General goal:** Prove that concrete proof systems cannot efficiently certify unsatisfiability of concrete CNF formulas

- **General theme:**

CNF formula F "expanding"

$$\Downarrow$$

Large proofs needed to refute F

- Well-developed machinery for resolution

- Very much less so for polynomial calculus

Main Message

Theorem (to be made precise later)

CNF formulas that are "expanding" according to (nice and clean) combinatorial condition are hard for polynomial calculus

Main Message

Theorem (to be made precise later)

CNF formulas that are "expanding" according to (nice and clean) combinatorial condition are hard for polynomial calculus

- Extends an approach from [Alekhnovich & Razborov '01]
- Unifies many previous lower bounds for polynomial calculus
- **Corollary:** Lower bound resolving problem in [Razborov '02]

Main Message

Theorem (to be made precise later)

CNF formulas that are "expanding" according to (nice and clean) combinatorial condition are hard for polynomial calculus

- Extends an approach from [Alekhnovich & Razborov '01]
- Unifies many previous lower bounds for polynomial calculus
- **Corollary:** Lower bound resolving problem in [Razborov '02]

This talk:

- Theorem statement clean, but quite involved proof
- Therefore, present main ideas for resolution (way simpler)
- Bonus: general formulation of combinatorial conditions comparing and contrasting resolution and polynomial calculus

Outline

1. Proof Complexity Overview
   - Preliminaries
   - Resolution
   - Polynomial Calculus
2. Lower Bounds from Expansion
   - Resolution Width
   - Polynomial Calculus Degree
   - Pigeonhole Principle
3. Open Problems

Some Notation and Terminology

- Literal a: variable x or its negation $\overline{x}$
- Clause $C = a_1 \vee \cdots \vee a_k$: disjunction of literals
  (Consider as sets, so no repetitions and order irrelevant)
- CNF formula $F = C_1 \wedge \cdots \wedge C_m$: conjunction of clauses
- k-CNF formula: CNF formula with clauses of size $\leq k = \mathcal{O}(1)$
- M = size of formula = # literals ($\approx$ # clauses for k-CNF)
- N = # variables $\leq M$

The Resolution Proof System

Goal: refute **unsatisfiable** CNF

Start with clauses of formula (axioms)

Derive new clauses by resolution rule

$$\frac{C \lor x \qquad D \lor \overline{x}}{C \lor D}$$

Refutation ends when empty clause $\bot$ derived

1. $x \lor y$
2. $x \lor \overline{y} \lor z$
3. $\overline{x} \lor z$
4. $\overline{y} \lor \overline{z}$
5. $\overline{x} \lor \overline{z}$

The Resolution Proof System

Goal: refute **unsatisfiable** CNF

Start with clauses of formula (axioms)

Derive new clauses by resolution rule

$$\frac{C \lor x \qquad D \lor \overline{x}}{C \lor D}$$

Refutation ends when empty clause $\perp$ derived

Can represent refutation as

- annotated list or
- directed acyclic graph

1.	$x \lor y$	Axiom
2.	$x \lor \overline{y} \lor z$	Axiom
3.	$\overline{x} \lor z$	Axiom
4.	$\overline{y} \lor \overline{z}$	Axiom
5.	$\overline{x} \lor \overline{z}$	Axiom
6.	$x \lor \overline{y}$	Res$(2, 4)$
7.	x	Res$(1, 6)$
8.	$\overline{x}$	Res$(3, 5)$
9.	$\perp$	Res$(7, 8)$

The Resolution Proof System

Goal: refute **unsatisfiable** CNF

Start with clauses of formula (axioms)

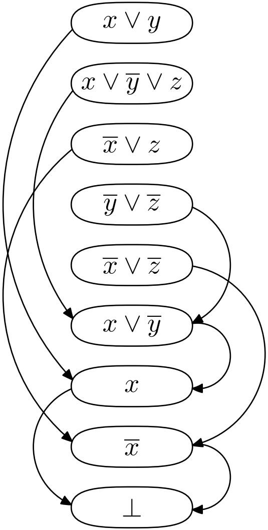
Derive new clauses by resolution rule

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$$

Refutation ends when empty clause $\perp$ derived

Can represent refutation as

- annotated list or
- directed acyclic graph



The Resolution Proof System

Goal: refute **unsatisfiable** CNF

Start with clauses of formula (axioms)

Derive new clauses by resolution rule

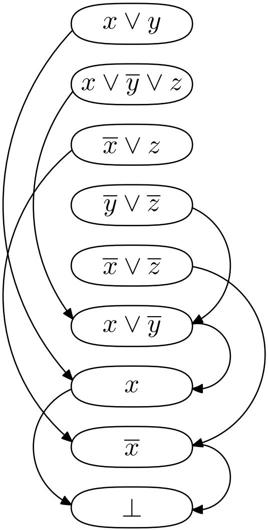
$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$$

Refutation ends when empty clause $\bot$ derived

Can represent refutation as

- annotated list or
- directed acyclic graph

Tree-like resolution if DAG is tree



Resolution Size/Length

Size/length $=$ $\#$ clauses in refutation

Most fundamental measure in proof complexity

Never worse than $\exp(\mathcal{O}(N))$

Matching $\exp(\Omega(M))$ lower bounds known

(Recall $N = \#$ variables $\leq$ formula size $= M$)

Examples of Hard Formulas w.r.t Resolution Size (1/2)

Pigeonhole principle (PHP) [Haken '85]
"$n + 1$ pigeons don't fit into n holes"

Variables $p_{i,j}$ = "pigeon i goes into hole j"

$p_{i,1} \lor p_{i,2} \lor \cdots \lor p_{i,n}$	every pigeon i gets a hole
$\overline{p}_{i,j} \lor \overline{p}_{i',j}$	no hole j gets two pigeons $i \neq i'$

Can also add "functionality" and "onto" axioms

$\overline{p}_{i,j} \lor \overline{p}_{i,j'}$	no pigeon i gets two holes $j \neq j'$
$p_{1,j} \lor p_{2,j} \lor \cdots \lor p_{n+1,j}$	every hole j gets a pigeon

Examples of Hard Formulas w.r.t Resolution Size (1/2)

Pigeonhole principle (PHP) [Haken '85]
"$n+1$ pigeons don't fit into n holes"

Variables $p_{i,j}$ = "pigeon i goes into hole j"

$p_{i,1} \lor p_{i,2} \lor \cdots \lor p_{i,n}$	every pigeon i gets a hole
$\overline{p}_{i,j} \lor \overline{p}_{i',j}$	no hole j gets two pigeons $i \neq i'$

Can also add "functionality" and "onto" axioms

$\overline{p}_{i,j} \lor \overline{p}_{i,j'}$	no pigeon i gets two holes $j \neq j'$
$p_{1,j} \lor p_{2,j} \lor \cdots \lor p_{n+1,j}$	every hole j gets a pigeon

Even onto functional PHP formulas are hard for resolution
"Resolution cannot count"

Examples of Hard Formulas w.r.t Resolution Size (1/2)

Pigeonhole principle (PHP) [Haken '85]
"$n+1$ pigeons don't fit into n holes"

Variables $p_{i,j}$ = "pigeon i goes into hole j"

$p_{i,1} \lor p_{i,2} \lor \cdots \lor p_{i,n}$	every pigeon i gets a hole
$\overline{p}_{i,j} \lor \overline{p}_{i',j}$	no hole j gets two pigeons $i \neq i'$

Can also add "functionality" and "onto" axioms

$\overline{p}_{i,j} \lor \overline{p}_{i,j'}$	no pigeon i gets two holes $j \neq j'$
$p_{1,j} \lor p_{2,j} \lor \cdots \lor p_{n+1,j}$	every hole j gets a pigeon

Even onto functional PHP formulas are hard for resolution
"Resolution cannot count"

But only lower bound $\exp\left(\Omega\left(\sqrt[3]{M}\right)\right)$ in terms of formula size

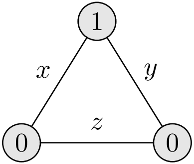
Examples of Hard Formulas w.r.t Resolution Size (2/2)

Tseitin formulas [Urquhart '87]

"Sum of degrees of vertices in graph is even"

Variables = edges (in undirected graph of bounded degree)

- Label every vertex 0/1 so that sum of labels odd
- Write CNF requiring parity of # true incident edges = label



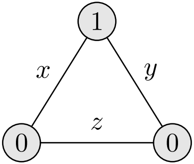
$$
\begin{array}{ll}
(x \lor y) & \land\ (\overline{x} \lor z) \\
\land\ (\overline{x} \lor \overline{y}) & \land\ (y \lor \overline{z}) \\
\land\ (x \lor \overline{z}) & \land\ (\overline{y} \lor z)
\end{array}
$$

Examples of Hard Formulas w.r.t Resolution Size (2/2)

Tseitin formulas [Urquhart '87]
"Sum of degrees of vertices in graph is even"

Variables = edges (in undirected graph of bounded degree)

- Label every vertex 0/1 so that sum of labels odd
- Write CNF requiring parity of # true incident edges = label



$$\begin{array}{llll} & (x \lor y) & \land & (\overline{x} \lor z) \\ \land & (\overline{x} \lor \overline{y}) & \land & (y \lor \overline{z}) \\ \land & (x \lor \overline{z}) & \land & (\overline{y} \lor z) \end{array}$$

Requires size $\exp(\Omega(M))$ on bounded-degree edge expanders
"Resolution cannot count $\bmod 2$**"**

Resolution Width

Width $=$ size of largest clause in refutation (always $\leq N$)

Resolution Width

Width $=$ size of largest clause in refutation (always $\leq N$)

Width upper bound $\Rightarrow$ size upper bound

Proof: at most $(2N)^{\text{width}}$ distinct clauses
(And this counting argument is essentially tight [Atserias et al.'14])

Resolution Width

Width = size of largest clause in refutation (always $\leq N$)

Width upper bound $\Rightarrow$ size upper bound

Proof: at most $(2N)^{\text{width}}$ distinct clauses
(And this counting argument is essentially tight [Atserias et al.'14])

Width lower bound $\Rightarrow$ size lower bound

Much less obvious. . .

Width Lower Bounds Imply Size Lower Bounds

Theorem ([Ben-Sasson & Wigderson '99])

For k-CNF formula over N variables

$$\textit{proof size} \geq \exp\left(\Omega\left(\frac{(\textit{proof width})^2}{N}\right)\right)$$

Width Lower Bounds Imply Size Lower Bounds

Theorem ([Ben-Sasson & Wigderson '99])

For k-CNF formula over N variables

$$\textit{proof size} \geq \exp\left(\Omega\left(\frac{(\textit{proof width})^2}{N}\right)\right)$$

Yields superpolynomial size bounds for width $\omega(\sqrt{N \log N})$
Almost all known lower bounds on size derivable via width

Width Lower Bounds Imply Size Lower Bounds

Theorem ([Ben-Sasson & Wigderson '99])

For k-CNF formula over N variables

$$\textit{proof size} \geq \exp\left(\Omega\left(\frac{(\textit{proof width})^2}{N}\right)\right)$$

Yields superpolynomial size bounds for width $\omega\big(\sqrt{N \log N}\big)$
Almost all known lower bounds on size derivable via width

For tree-like resolution have proof size $\geq 2^{\text{width}}$ [BW99]

General resolution: width up to $\mathcal{O}\big(\sqrt{N \log N}\big)$ implies no size lower bounds — possible to tighten analysis? **No!**

Optimality of the Size-Width Lower Bound

Ordering principles [Stålmarck '96, Bonet & Galesi '99]
"Every (partially) ordered set $\{e_1, \ldots, e_n\}$ has minimal element"

Variables $x_{i,j} =$ "$e_i < e_j$"

$\overline{x}_{i,j} \lor \overline{x}_{j,i}$	anti-symmetry; not both $e_i < e_j$ and $e_j < e_i$
$\overline{x}_{i,j} \lor \overline{x}_{j,k} \lor x_{i,k}$	transitivity; $e_i < e_j$ and $e_j < e_k$ implies $e_i < e_k$
$\bigvee_{1 \leq i \leq n,\, i \neq j} x_{i,j}$	e_j is not a minimal element

Optimality of the Size-Width Lower Bound

Ordering principles [Stålmarck '96, Bonet & Galesi '99]
"Every (partially) ordered set $\{e_1, \ldots, e_n\}$ has minimal element"

Variables $x_{i,j} =$ "$e_i < e_j$"

$\overline{x}_{i,j} \vee \overline{x}_{j,i}$	anti-symmetry; not both $e_i < e_j$ and $e_j < e_i$
$\overline{x}_{i,j} \vee \overline{x}_{j,k} \vee x_{i,k}$	transitivity; $e_i < e_j$ and $e_j < e_k$ implies $e_i < e_k$
$\bigvee_{1 \leq i \leq n,\, i \neq j} x_{i,j}$	e_j is not a minimal element

Refutable in resolution in size $\mathcal{O}\bigl(N^{3/2}\bigr) = \mathcal{O}(M)$
Requires resolution width $\Omega\bigl(\sqrt{N}\bigr)$ (converted to k-CNF)

Conversion to k-CNF "Graph Versions" of Formulas

- Need bounded-width CNFs to use lower bound in [BW99]
- But PHP and ordering principle formulas have wide clauses
- **Solution:** Restrict formulas to bounded-degree graphs

Conversion to k-CNF "Graph Versions" of Formulas

- Need bounded-width CNFs to use lower bound in [BW99]
- But PHP and ordering principle formulas have wide clauses
- **Solution:** Restrict formulas to bounded-degree graphs

For (onto functional) PHP, pigeons can fly only to neighbour holes:

$\bigvee_{j \in \mathcal{N}(i)} p_{i,j}$	pigeon i goes into hole in $\mathcal{N}(i)$
$\bigvee_{i \in \mathcal{N}(j)} p_{i,j}$	hole j gets pigeon from $\mathcal{N}(j)$

For ordering principle, non-minimality only witnessed by neighbours:

$\bigvee_{i \in \mathcal{N}(j)} x_{i,j}$	some e_i for $i \in \mathcal{N}(j)$ shows e_j not minimal

Conversion to k-CNF "Graph Versions" of Formulas

- Need bounded-width CNFs to use lower bound in [BW99]
- But PHP and ordering principle formulas have wide clauses
- **Solution:** Restrict formulas to bounded-degree graphs

For (onto functional) PHP, pigeons can fly only to neighbour holes:

$\bigvee_{j \in \mathcal{N}(i)} p_{i,j}$	pigeon i goes into hole in $\mathcal{N}(i)$
$\bigvee_{i \in \mathcal{N}(j)} p_{i,j}$	hole j gets pigeon from $\mathcal{N}(j)$

For ordering principle, non-minimality only witnessed by neighbours:

$\bigvee_{i \in \mathcal{N}(j)} x_{i,j}$	some e_i for $i \in \mathcal{N}(j)$ shows e_j not minimal

- Now width lower bounds $\Rightarrow$ size lower bounds
- And size lower bounds hold for original, unrestricted formulas

Polynomial Calculus (PC)

From [Clegg et al. '96] with adjustment in [Alekhnovich et al. '02]

Clauses interpreted as polynomial equations over field

Example: $x \vee y \vee \overline{z}$ gets translated to $xy\overline{z} = 0$
(Think of $0 \equiv true$ and $1 \equiv false$)

Polynomial Calculus (PC)

From [Clegg et al. '96] with adjustment in [Alekhnovich et al. '02]

Clauses interpreted as polynomial equations over field

Example: $x \vee y \vee \overline{z}$ gets translated to $xy\overline{z} = 0$
(Think of $0 \equiv \mathit{true}$ and $1 \equiv \mathit{false}$)

Derivation rules

Boolean axioms $\dfrac{}{x^2 - x = 0}$

Negation $\dfrac{}{x + \overline{x} = 1}$

Linear combination $\dfrac{p = 0 \qquad q = 0}{\alpha p + \beta q = 0}$

Multiplication $\dfrac{p = 0}{xp = 0}$

Goal: Derive $1 = 0 \Leftrightarrow$ no common root $\Leftrightarrow$ formula unsatisfiable

Polynomial Calculus Size and Degree

Clauses turn into monomials

Write out all polynomials as sums of monomials

W.l.o.g. all polynomials multilinear (because of Boolean axioms)

Polynomial Calculus Size and Degree

Clauses turn into monomials
Write out all polynomials as sums of monomials
W.l.o.g. all polynomials multilinear (because of Boolean axioms)

Size — analogue of resolution length/size
total # monomials in refutation counted with repetitions

Degree — analogue of resolution width
largest degree of monomial in refutation

Polynomial Calculus Strictly Stronger than Resolution

Polynomial calculus simulates resolution efficiently

- Can mimic resolution refutation step by step
- Essentially no increase in length/size or width/degree
- Hence worst-case upper bounds for resolution carry over

Polynomial Calculus Strictly Stronger than Resolution

Polynomial calculus simulates resolution efficiently

- Can mimic resolution refutation step by step
- Essentially no increase in length/size or width/degree
- Hence worst-case upper bounds for resolution carry over

Polynomial calculus strictly stronger w.r.t. size and degree

- Tseitin formulas (over $\mathrm{GF}(2)$ can do Gaussian elimination)
- Onto functional pigeonhole principle (over any field) [Riis '93]
- Also other examples

Size vs. Degree

- Degree upper bound ⇒ size upper bound [Clegg et al.'96]
  Similar to resolution bound; argument a bit more involved
  Again essentially tight by [Atserias et al.'14]

Size vs. Degree

- Degree upper bound ⇒ size upper bound [Clegg et al.'96]
  Similar to resolution bound; argument a bit more involved
  Again essentially tight by [Atserias et al.'14]

- Degree lower bound ⇒ size lower bound [Impagliazzo et al.'99]
  Precursor of [Ben-Sasson & Wigderson '99] — can do same proof to get exactly same bound

Size vs. Degree

- Degree upper bound ⇒ size upper bound [Clegg et al.'96]
  Similar to resolution bound; argument a bit more involved
  Again essentially tight by [Atserias et al.'14]
- Degree lower bound ⇒ size lower bound [Impagliazzo et al.'99]
  Precursor of [Ben-Sasson & Wigderson '99] — can do same proof to get exactly same bound
- Size-degree bound essentially optimal [Galesi & Lauria '10]
  Example: same ordering principle formulas

Size vs. Degree

- Degree upper bound ⇒ size upper bound [Clegg et al.'96]
  Similar to resolution bound; argument a bit more involved
  Again essentially tight by [Atserias et al.'14]
- Degree lower bound ⇒ size lower bound [Impagliazzo et al.'99]
  Precursor of [Ben-Sasson & Wigderson '99] — can do same proof to get exactly same bound
- Size-degree bound essentially optimal [Galesi & Lauria '10]
  Example: same ordering principle formulas
- Most size lower bounds for polynomial calculus derived via degree lower bounds, but machinery much less developed

Size vs. Degree

- Degree upper bound ⇒ size upper bound [Clegg et al.'96]
  Similar to resolution bound; argument a bit more involved
  Again essentially tight by [Atserias et al.'14]

- Degree lower bound ⇒ size lower bound [Impagliazzo et al.'99]
  Precursor of [Ben-Sasson & Wigderson '99] — can do same proof to get exactly same bound

- Size-degree bound essentially optimal [Galesi & Lauria '10]
  Example: same ordering principle formulas

- Most size lower bounds for polynomial calculus derived via degree lower bounds, but machinery much less developed

- **Open problem:** Are functional PHP and onto PHP formulas hard for polynomial calculus?

Lower Bounds via Graph Expansion

Standard approach:

Lower bounds from expansion

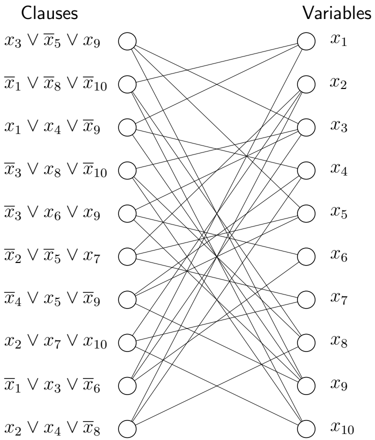
Simplest example is the clause-variable incidence graph (CVIG)

Lower Bounds via Graph Expansion

Standard approach:

Lower bounds from expansion

Simplest example is the clause-variable incidence graph (CVIG)

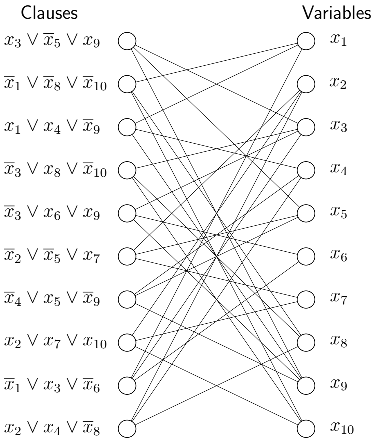


Lower Bounds via Graph Expansion

Standard approach:
Lower bounds from expansion

Simplest example is the clause-variable incidence graph (CVIG)

Boundary expansion:
Subsets of left vertices have many unique right neighbours

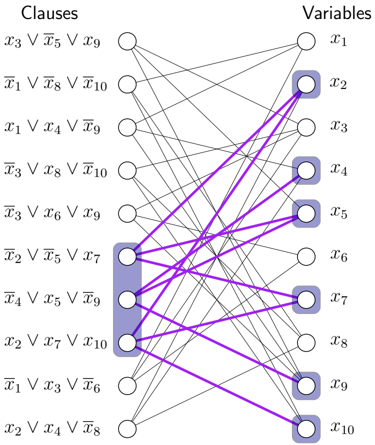


Lower Bounds via Graph Expansion

Standard approach:
Lower bounds from expansion

Simplest example is the clause-variable incidence graph (CVIG)

Boundary expansion:
Subsets of left vertices have many unique right neighbours

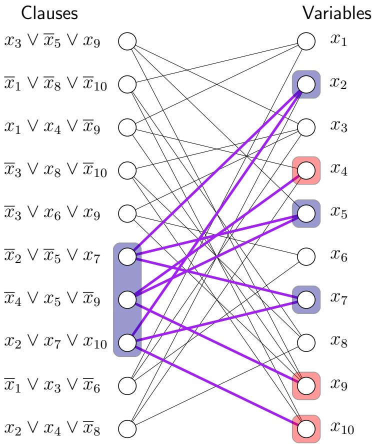


Lower Bounds via Graph Expansion

Standard approach:
Lower bounds from expansion

Simplest example is the clause-variable incidence graph (CVIG)

Boundary expansion:
Subsets of left vertices have many unique right neighbours

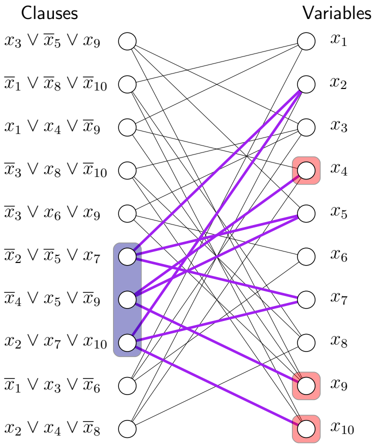


Lower Bounds via Graph Expansion

Standard approach:
Lower bounds from expansion

Simplest example is the clause-variable incidence graph (CVIG)

Boundary expansion:
Subsets of left vertices have many unique right neighbours



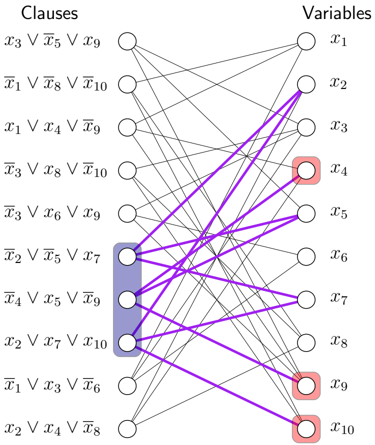
Lower Bounds via Graph Expansion

Standard approach:
Lower bounds from expansion

Simplest example is the clause-variable incidence graph (CVIG)

Boundary expansion:
Subsets of left vertices have many unique right neighbours

Problem:
CVIG often loses expansion of combinatorial problem



Lower Bounds via Graph Expansion

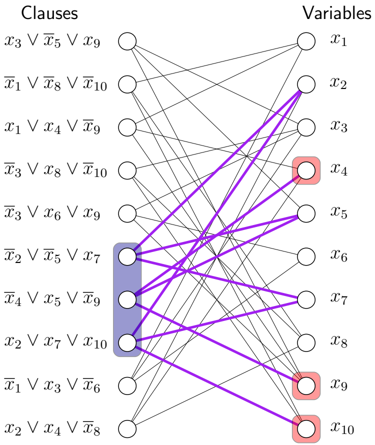
Standard approach:
Lower bounds from expansion

Simplest example is the clause-variable incidence graph (CVIG)

Boundary expansion:
Subsets of left vertices have many unique right neighbours

Problem:
CVIG often loses expansion of combinatorial problem

Need graph capturing combinatorial structure!



Generalized Incidence Graphs for CNF Formulas

Given CNF formula $\mathcal{F}$ over variables $\mathcal{V}$

- Partition clauses into $\mathcal{F} = \bigcup_{i=1}^{m} F_i \cup E$ (for E satisifiable)
- Divide variables into $\mathcal{V} = \bigcup_{j=1}^{n} V_j$ — **not** always partition
- Overlap ℓ: Any x appears in $\leq \ell$ different V_j

Generalized Incidence Graphs for CNF Formulas

Given CNF formula $\mathcal{F}$ over variables $\mathcal{V}$

- Partition clauses into $\mathcal{F} = \bigcup_{i=1}^{m} F_i \cup E$ (for E satisifiable)
- Divide variables into $\mathcal{V} = \bigcup_{j=1}^{n} V_j$ — **not** always partition
- Overlap ℓ: Any x appears in $\leq \ell$ different V_j

Build bipartite $(\mathcal{U}, \mathcal{V})_E$-graph $\mathcal{G}$

- Left vertices $\mathcal{U} = \{F_1, \ldots, F_m\}$
- Right vertices $\mathcal{V} = \{V_1, \ldots, V_n\}$
- Edge (F_i, V_j) if $\mathit{Vars}(F_i) \cap V_j \neq \emptyset$
- Two types of edges depending on how F_i and V_j behave (modulo assignments α satisfying "filtering set" E)

The Importance of Basic Courtesy

$F \in \mathcal{U}$ and $V \in \mathcal{V}$ are E-semirespectful neighbours if

- given any total assignment α such that $\alpha(E) = 1$
- can flip α on V to α' so that $\alpha'(F \wedge E) = 1$

The Importance of Basic Courtesy

$F \in \mathcal{U}$ and $V \in \mathcal{V}$ are E-semirespectful neighbours if

- given any total assignment α such that $\alpha(E) = 1$
- can flip α on V to α' so that $\alpha'(F \wedge E) = 1$

Example

$F_1 = \{x \vee y,\ x \vee \overline{z},\ \overline{x} \vee z\}$, $V = \{x, y\}$, $E = \{\overline{y} \vee z\}$

Not E-semirespectful — consider $\alpha = \{y \mapsto 0, z \mapsto 0\}$

Not allowed to flip $z \notin V$; flipping y falsifies E; but $F_1\restriction_\alpha = \{x, \overline{x}\}$

The Importance of Basic Courtesy

$F \in \mathcal{U}$ and $V \in \mathcal{V}$ are E-semirespectful neighbours if

- given any total assignment α such that $\alpha(E) = 1$
- can flip α on V to α' so that $\alpha'(F \wedge E) = 1$

Example

$F_1 = \{x \vee y,\, x \vee \overline{z},\, \overline{x} \vee z\}$, $V = \{x, y\}$, $E = \{\overline{y} \vee z\}$

Not E-semirespectful — consider $\alpha = \{y \mapsto 0, z \mapsto 0\}$

Not allowed to flip $z \notin V$; flipping y falsifies E; but $F_1{\restriction}_{\alpha} = \{x, \overline{x}\}$

Example

Change to $F_2 = \{x \vee \overline{y},\, x \vee \overline{z},\, \overline{x} \vee y \vee z\}$, $V = \{x, y\}$, $E = \{\overline{y} \vee z\}$

Now F_2 and V E-semirespectful — given any α s.t. $\alpha(\overline{y} \vee z) = 1$

can always flip value assigned to x to $\alpha(y \vee z)$

The Importance of Basic Courtesy

$F \in \mathcal{U}$ and $V \in \mathcal{V}$ are E-semirespectful neighbours if

- given any total assignment α such that $\alpha(E) = 1$
- can flip α on V to α' so that $\alpha'(F \wedge E) = 1$

Example

$F_1 = \{x \vee y,\ x \vee \overline{z},\ \overline{x} \vee z\}$, $V = \{x, y\}$, $E = \{\overline{y} \vee z\}$
Not E-semirespectful — consider $\alpha = \{y \mapsto 0, z \mapsto 0\}$
Not allowed to flip $z \notin V$; flipping y falsifies E; but $F_1{\restriction_\alpha} = \{x, \overline{x}\}$

Example

Change to $F_2 = \{x \vee \overline{y},\ x \vee \overline{z},\ \overline{x} \vee y \vee z\}$, $V = \{x, y\}$, $E = \{\overline{y} \vee z\}$
Now F_2 and V E-semirespectful — given any α s.t. $\alpha(\overline{y} \vee z) = 1$
can always flip value assigned to x to $\alpha(y \vee z)$

(To simplify, think of all edges (F_i, V_j) as being E-semirespectful)

Semirespectful Expanders and Width Lower Bounds

Recall boundary $\partial(\mathcal{U}') = \{V \in \mathcal{N}(\mathcal{U}') \mid \mathcal{N}(V) \cap \mathcal{U}' = \{F\} \text{ unique}\}$

Semirespectful Expanders and Width Lower Bounds

Recall boundary $\partial(\mathcal{U}') = \{V \in \mathcal{N}(\mathcal{U}') \mid \mathcal{N}(V) \cap \mathcal{U}' = \{F\} \text{ unique}\}$

Define semirespectful boundary to be

$\partial^{\mathsf{sr}}_E(\mathcal{U}') := \{V \in \partial(\mathcal{U}') \mid V \text{ and } F = \mathcal{N}(V) \cap \mathcal{U}' \ E\text{-semirespectful}\}$

Semirespectful Expanders and Width Lower Bounds

Recall boundary $\partial(\mathcal{U}') = \{V \in \mathcal{N}(\mathcal{U}') \mid \mathcal{N}(V) \cap \mathcal{U}' = \{F\} \text{ unique}\}$

Define semirespectful boundary to be

$\partial^{\mathsf{sr}}_E(\mathcal{U}') := \{V \in \partial(\mathcal{U}') \mid V \text{ and } F = \mathcal{N}(V) \cap \mathcal{U}' \ E\text{-semirespectful}\}$

Semirespectful expander

An $(\mathcal{U}, \mathcal{V})_E$-graph is an (s, δ, E)-semirespectful expander if for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$ it holds that $|\partial^{\mathsf{sr}}_E(\mathcal{U}')| \geq \delta|\mathcal{U}'|$

Semirespectful Expanders and Width Lower Bounds

Recall boundary $\partial(\mathcal{U}') = \{V \in \mathcal{N}(\mathcal{U}') \mid \mathcal{N}(V) \cap \mathcal{U}' = \{F\} \text{ unique}\}$

Define semirespectful boundary to be

$$\partial_E^{\mathsf{sr}}(\mathcal{U}') := \{V \in \partial(\mathcal{U}') \mid V \text{ and } F = \mathcal{N}(V) \cap \mathcal{U}' \ E\text{-semirespectful}\}$$

Semirespectful expander

An $(\mathcal{U}, \mathcal{V})_E$-graph is an (s, δ, E)-semirespectful expander if for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$ it holds that $|\partial_E^{\mathsf{sr}}(\mathcal{U}')| \geq \delta|\mathcal{U}'|$

Theorem (essentially [BW99])

If $\mathcal{F}$ has (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{resolution proof width} > \frac{\delta s}{2\ell}$$

Progress Measure Approach (1/4)

Theorem (essentially [BW99])

If $\mathcal{F}$ has (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{resolution proof width} > \frac{\delta s}{2\ell}$$

Proof: Define "progress measure" $\mu : \{\text{clauses}\} \rightarrow \mathbb{N}$ such that

1. $\mu(\text{axiom clause}) = \mathcal{O}(1)$
2. $\mu(C \vee D) \leq \mu(C \vee x) + \mu(D \vee \overline{x})$
3. $\mu(\bot) > s$

Progress Measure Approach (1/4)

Theorem (essentially [BW99])

If $\mathcal{F}$ has (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{resolution proof width} > \frac{\delta s}{2\ell}$$

Proof: Define "progress measure" $\mu : \{\text{clauses}\} \to \mathbb{N}$ such that

1. $\mu(\text{axiom clause}) = \mathcal{O}(1)$
2. $\mu(C \vee D) \leq \mu(C \vee x) + \mu(D \vee \overline{x})$
3. $\mu(\bot) > s$

$\Rightarrow$ in any resolution proof $\exists\, C$ with $\mu(C) = \sigma$ for $s/2 < \sigma \leq s$

Progress Measure Approach (1/4)

Theorem (essentially [BW99])

If $\mathcal{F}$ has (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{resolution proof width} > \frac{\delta s}{2\ell}$$

Proof: Define "progress measure" $\mu : \{\text{clauses}\} \to \mathbb{N}$ such that

1. $\mu(\text{axiom clause}) = \mathcal{O}(1)$
2. $\mu(C \lor D) \leq \mu(C \lor x) + \mu(D \lor \overline{x})$
3. $\mu(\perp) > s$

$\Rightarrow$ in any resolution proof $\exists\, C$ with $\mu(C) = \sigma$ for $s/2 < \sigma \leq s$

$\Rightarrow$ such C has width $\geq \delta\sigma/\ell$ □

Progress Measure Approach (2/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ for $\mathcal{F}$, define

$$\mu(C) := \min\{|\mathcal{U}'| \,;\, \bigwedge_{F \in \mathcal{U}'} F \wedge E \vDash C\}$$

Progress Measure Approach (2/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ for $\mathcal{F}$, define

$$\mu(C) := \min\left\{\left|\mathcal{U}'\right| ; \bigwedge_{F \in \mathcal{U}'} F \wedge E \vDash C\right\}$$

1. $\mu(A) = \mathcal{O}(1)$ for axioms $A \in \mathcal{F} = \bigcup_{i=1}^{m} F_i \cup E$
   - $A \in E$: $\mu(A) = 0$ since $E \vDash A$
   - $A \in F_i$: $\mu(A) = 1$ since $F_i \wedge E \vDash A$

Progress Measure Approach (2/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ for $\mathcal{F}$, define

$$\mu(C) := \min\{|\mathcal{U}'| \,;\, \bigwedge_{F \in \mathcal{U}'} F \wedge E \vDash C\}$$

1. $\mu(A) = \mathcal{O}(1)$ for axioms $A \in \mathcal{F} = \bigcup_{i=1}^{m} F_i \cup E$
   - $A \in E$: $\mu(A) = 0$ since $E \vDash A$
   - $A \in F_i$: $\mu(A) = 1$ since $F_i \wedge E \vDash A$
2. $\mu(C \vee D) \leq \mu(C \vee x) + \mu(D \vee \overline{x})$
   - Fix minimal $\mathcal{U}_1$ s.t. $\bigwedge_{F \in \mathcal{U}_1} F \wedge E \vDash C \vee x$
   - Fix minimal $\mathcal{U}_2$ s.t. $\bigwedge_{F \in \mathcal{U}_2} F \wedge E \vDash D \vee \overline{x}$
   - Then it holds that

$$\bigwedge_{F \in \mathcal{U}_1 \cup \mathcal{U}_2} F \wedge E \vDash C \vee D \ ,$$

so $\mu(C \vee D) \leq |\mathcal{U}_1 \cup \mathcal{U}_2| \leq |\mathcal{U}_1| + |\mathcal{U}_2| = \mu(C \vee x) + \mu(D \vee \overline{x})$

Progress Measure Approach (3/4)

3. $\mu(\bot) > s$ for empty clause $\bot$

Progress Measure Approach (3/4)

3. $\mu(\perp) > s$ for empty clause $\perp$
   - Consider any $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| = s$, $\mathcal{U}' = \{F_1, \ldots, F_s\}$

Progress Measure Approach (3/4)

3. $\mu(\bot) > s$ for empty clause $\bot$
   - Consider any $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| = s$, $\mathcal{U}' = \{F_1, \ldots, F_s\}$
   - By expansion $|\partial_E^{\mathsf{sr}}(\mathcal{U}')| \geq \delta|\mathcal{U}'| > 0$

Progress Measure Approach (3/4)

3. $\mu(\perp) > s$ for empty clause $\perp$
   - Consider any $\mathcal{U}' \subseteq \mathcal{U}$, $\left|\mathcal{U}'\right| = s$, $\mathcal{U}' = \{F_1, \ldots, F_s\}$
   - By expansion $\left|\partial^{\mathsf{sr}}_E(\mathcal{U}')\right| \geq \delta\left|\mathcal{U}'\right| > 0$
   - By "peeling argument" $\exists$ matching $F_1 \leftrightarrow V_1, \ldots, F_s \leftrightarrow V_s$ s.t. $V_i \in \mathcal{N}(F_i) \setminus \mathcal{N}\left(\bigcup_{j=1}^{i-1} F_j\right)$ and F_i & V_i E-semirespectful

Progress Measure Approach (3/4)

3. $\mu(\bot) > s$ for empty clause $\bot$
   - Consider any $\mathcal{U}' \subseteq \mathcal{U}$, $\left|\mathcal{U}'\right| = s$, $\mathcal{U}' = \{F_1, \ldots, F_s\}$
   - By expansion $\left|\partial^{\mathsf{sr}}_E(\mathcal{U}')\right| \geq \delta\left|\mathcal{U}'\right| > 0$
   - By "peeling argument" $\exists$ matching $F_1 \leftrightarrow V_1, \ldots, F_s \leftrightarrow V_s$ s.t. $V_i \in \mathcal{N}(F_i) \setminus \mathcal{N}\left(\bigcup_{j=1}^{i-1} F_j\right)$ and F_i & V_i E-semirespectful
   - Given any α s.t. $\alpha(E) = 1$, for $i = 1, 2, \ldots, s$ flip V_i to satisfy F_i without falsifying E

Progress Measure Approach (3/4)

3. $\mu(\bot) > s$ for empty clause $\bot$
   - Consider any $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| = s$, $\mathcal{U}' = \{F_1, \ldots, F_s\}$
   - By expansion $|\partial^{\mathsf{sr}}_E(\mathcal{U}')| \geq \delta|\mathcal{U}'| > 0$
   - By "peeling argument" $\exists$ matching $F_1 \leftrightarrow V_1, \ldots, F_s \leftrightarrow V_s$ s.t. $V_i \in \mathcal{N}(F_i) \setminus \mathcal{N}\left(\bigcup_{j=1}^{i-1} F_j\right)$ and F_i & V_i E-semirespectful
   - Given any α s.t. $\alpha(E) = 1$, for $i = 1, 2, \ldots, s$ flip V_i to satisfy F_i without falsifying E
   - Yields α' s.t. $\alpha'\left(\bigwedge_{F_i \in \mathcal{U}'} F_i \wedge E\right) = 1$

Progress Measure Approach (3/4)

3. $\mu(\bot) > s$ for empty clause $\bot$
   - Consider any $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| = s$, $\mathcal{U}' = \{F_1, \ldots, F_s\}$
   - By expansion $|\partial^{\mathsf{sr}}_E(\mathcal{U}')| \geq \delta|\mathcal{U}'| > 0$
   - By "peeling argument" $\exists$ matching $F_1 \leftrightarrow V_1, \ldots, F_s \leftrightarrow V_s$ s.t. $V_i \in \mathcal{N}(F_i) \setminus \mathcal{N}\left(\bigcup_{j=1}^{i-1} F_j\right)$ and F_i & V_i E-semirespectful
   - Given any α s.t. $\alpha(E) = 1$, for $i = 1, 2, \ldots, s$ flip V_i to satisfy F_i without falsifying E
   - Yields α' s.t. $\alpha'\left(\bigwedge_{F_i \in \mathcal{U}'} F_i \wedge E\right) = 1$
   - So $\bigwedge_{F_i \in \mathcal{U}'} F_i \wedge E \nvDash \bot$ for $|\mathcal{U}'| \leq s$ and hence $\mu(\bot) > s$

Progress Measure Approach (4/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ

Progress Measure Approach (4/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ

Already showed: In any proof $\exists\, C$ with $\mu(C) = \sigma \in (s/2, s]$

Progress Measure Approach (4/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ

Already showed: In any proof $\exists\, C$ with $\mu(C) = \sigma \in (s/2, s]$

Want to show: $\mu(C) = \sigma \leq s$ implies C has width $\geq \delta\sigma/\ell$
Fix minimal $\mathcal{U}_C$ of size $|\mathcal{U}_C| = \sigma$ s.t. $\bigwedge_{F \in \mathcal{U}_C} F \wedge E \vDash C$

Progress Measure Approach (4/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ

Already showed: In any proof $\exists\, C$ with $\mu(C) = \sigma \in (s/2, s]$

Want to show: $\mu(C) = \sigma \leq s$ implies C has width $\geq \delta\sigma/\ell$
Fix minimal $\mathcal{U}_C$ of size $\left|\mathcal{U}_C\right| = \sigma$ s.t. $\bigwedge_{F \in \mathcal{U}_C} F \wedge E \vDash C$

Claim

If $V \in \partial^{\mathsf{sr}}_E(\mathcal{U}_C)$, then $V \cap \mathit{Vars}(C) \neq \emptyset$

Progress Measure Approach (4/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ

Already showed: In any proof $\exists\, C$ with $\mu(C) = \sigma \in (s/2, s]$

Want to show: $\mu(C) = \sigma \leq s$ implies C has width $\geq \delta\sigma/\ell$
Fix minimal $\mathcal{U}_C$ of size $|\mathcal{U}_C| = \sigma$ s.t. $\bigwedge_{F \in \mathcal{U}_C} F \wedge E \vDash C$

Claim

If $V \in \partial^{\mathsf{sr}}_E(\mathcal{U}_C)$, then $V \cap \mathit{Vars}(C) \neq \emptyset$

Since every variable occurs in $\leq \ell$ sets V, the clause C then must have width $\geq \left|\partial^{\mathsf{sr}}_E(\mathcal{U}_C)\right|/\ell \geq \delta\left|\mathcal{U}_C\right|/\ell = \delta\sigma/\ell$ □

Progress Measure Approach (4/4)

Given (s, δ, E)-semirespectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ

Already showed: In any proof $\exists\, C$ with $\mu(C) = \sigma \in (s/2, s]$

Want to show: $\mu(C) = \sigma \leq s$ implies C has width $\geq \delta\sigma/\ell$
Fix minimal $\mathcal{U}_C$ of size $|\mathcal{U}_C| = \sigma$ s.t. $\bigwedge_{F \in \mathcal{U}_C} F \wedge E \vDash C$

Claim

If $V \in \partial^{\mathsf{sr}}_E(\mathcal{U}_C)$, then $V \cap \mathit{Vars}(C) \neq \emptyset$

Since every variable occurs in $\leq \ell$ sets V, the clause C then must have width $\geq |\partial^{\mathsf{sr}}_E(\mathcal{U}_C)|/\ell \geq \delta|\mathcal{U}_C|/\ell = \delta\sigma/\ell$ $\square$

Proof of claim: Another flipping argument using semirespectfulness

- Fix $V \in \partial^{\mathsf{sr}}_E(\mathcal{U}_C)$ and unique neighbour $F_V \in \mathcal{U}_C$ of V
- By minimality, $\exists\, \alpha$ s.t. $\alpha\big(\bigwedge_{F \in \mathcal{U}_C \setminus \{F_V\}} F \wedge E\big) = 1$ but $\alpha(C) = 0$
- If $V \cap \mathit{Vars}(C) = \emptyset$, then E-semirespectfully flip α on V to satisfy F_V ↯

Applications: Tseitin and Onto-FPHP

Tseitin formulas

- F_i = clauses encoding parity constraint for ith vertex
- V_j = singleton set with jth edge (so overlap $\ell = 1$)
- $E = \emptyset$
- If underlying graph edge expander, then $(\mathcal{U}, \mathcal{V})_E$-graph semirespectful boundary expander with same parameters

Applications: Tseitin and Onto-FPHP

Tseitin formulas

- F_i = clauses encoding parity constraint for ith vertex
- V_j = singleton set with jth edge (so overlap $\ell = 1$)
- $E = \emptyset$
- If underlying graph edge expander, then $(\mathcal{U}, \mathcal{V})_E$-graph semirespectful boundary expander with same parameters

Onto functional PHP formulas

- F_i = singleton set with pigeon axiom for pigeon i
- V_j = all variables $p_{i,j}$ mentioning hole j (again overlap $\ell = 1$)
- E = all hole, functional, and onto axioms
- If onto FPHP restricted to bipartite graph, then $(\mathcal{U}, \mathcal{V})_E$-graph semirespectful boundary expander with same parameters

From Resolution to Polynomial Calculus

Obtain resolution width lower bounds from expander graphs where we can win following game on edges

Resolution edge game on (F, V) with side constraints E

1. Adversary provides total assignment α such that $\alpha(E) = 1$
2. Choose $\alpha_V : V \rightarrow \{0, 1\}$ and flip so that $\alpha[\alpha_V/V](F \wedge E) = 1$

From Resolution to Polynomial Calculus

Obtain resolution width lower bounds from expander graphs where we can win following game on edges

Resolution edge game on (F, V) with side constraints E

1. Adversary provides total assignment α such that $\alpha(E) = 1$
2. Choose $\alpha_V : V \to \{0,1\}$ and flip so that $\alpha[\alpha_V/V](F \wedge E) = 1$

But Tseitin and onto FPHP both easy for polynomial calculus!

So semirespectful boundary expanders cannot yield any lower bounds for polynomial calculus

A Harder Edge Game for Polynomial Calculus

Resolution edge game on (F, V) with side constraints E

1. Adversary provides total assignment α such that $\alpha(E) = 1$
2. Choose $\alpha_V : V \to \{0, 1\}$ and flip so that $\alpha[\alpha_V/V](F \wedge E) = 1$

A Harder Edge Game for Polynomial Calculus

Resolution edge game on (F, V) with side constraints E

1. Adversary provides total assignment α such that $\alpha(E) = 1$
2. Choose $\alpha_V : V \to \{0, 1\}$ and flip so that $\alpha[\alpha_V/V](F \wedge E) = 1$

To get polynomial calculus degree lower bounds need winning strategy for harder game on expander graphs

Polynomial calculus edge game on (F, V) with side constraints E

1. Commit to $\alpha_V : V \to \{0, 1\}$
2. Adversary provides total assignment α such that $\alpha(E) = 1$
3. Flipping α on V to α_V should yield $\alpha[\alpha_V/V](F \wedge E) = 1$

Fully Respectful Neighbours

$F \in \mathcal{U}$ and $V \in \mathcal{V}$ are E-respectful neighbours if possible to find $\alpha_V : V \to \{0,1\}$ such that

- $\alpha_V(F) = 1$
- $\alpha_V(C) = 1$ for all clauses $C \in E$ with $V \cap \mathit{Vars}(C) \neq \emptyset$

Fully Respectful Neighbours

$F \in \mathcal{U}$ and $V \in \mathcal{V}$ are E-respectful neighbours if possible to find $\alpha_V : V \to \{0,1\}$ such that

- $\alpha_V(F) = 1$
- $\alpha_V(C) = 1$ for all clauses $C \in E$ with $V \cap Vars(C) \neq \emptyset$

Example

$F_2 = \{x \vee \overline{y},\ x \vee \overline{z},\ \overline{x} \vee y \vee z\}$, $V = \{x, y\}$, $E = \{\overline{y} \vee z\}$
Recall F_2 and V E-semirespectful — can always flip x to $\alpha(y \vee z)$
Not E-respectful — α_V needs $y \mapsto 0$, but $F_2{\restriction}_{y=0} = \{x \vee \overline{z},\ \overline{x} \vee z\}$

Fully Respectful Neighbours

$F \in \mathcal{U}$ and $V \in \mathcal{V}$ are E-respectful neighbours if possible to find $\alpha_V : V \to \{0,1\}$ such that

- $\alpha_V(F) = 1$
- $\alpha_V(C) = 1$ for all clauses $C \in E$ with $V \cap \mathit{Vars}(C) \neq \emptyset$

Example

$F_2 = \{x \vee \overline{y},\ x \vee \overline{z},\ \overline{x} \vee y \vee z\}$, $V = \{x, y\}$, $E = \{\overline{y} \vee z\}$
Recall F_2 and V E-semirespectful — can always flip x to $\alpha(y \vee z)$
Not E-respectful — α_V needs $y \mapsto 0$, but $F_2\restriction_{y=0} = \{x \vee \overline{z},\ \overline{x} \vee z\}$

Example

Change to $F_2 = \{x \vee \overline{y},\ x \vee \overline{z},\ \overline{x} \vee y \vee z\}$, $V = \{x, y\}$, $E' = \{y \vee \overline{z}\}$
Now F_2 and V E'-respectful — for $\alpha_V = \{x \mapsto 1, y \mapsto 1\}$ we have $\alpha_V\left(F_2 \wedge E'\right) = 1$

Respectful Expanders and Degree Lower Bounds

Define respectful boundary to be

$$\partial^{\mathsf{r}}_{E}(\mathcal{U}') := \left\{ V \in \partial(\mathcal{U}') \,\middle|\, V \text{ and } F = \mathcal{N}(V) \cap \mathcal{U}' \; E\text{-respectful} \right\}$$

Respectful Expanders and Degree Lower Bounds

Define respectful boundary to be

$$\partial^{\mathsf{r}}_E(\mathcal{U}') := \{V \in \partial(\mathcal{U}') \mid V \text{ and } F = \mathcal{N}(V) \cap \mathcal{U}' \ E\text{-respectful}\}$$

Respectful expander

An $(\mathcal{U}, \mathcal{V})_E$-graph is an (s, δ, E)-respectful expander if for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$ it holds that $|\partial^{\mathsf{r}}_E(\mathcal{U}')| \geq \delta|\mathcal{U}'|$

Respectful Expanders and Degree Lower Bounds

Define respectful boundary to be

$$\partial^{\mathsf{r}}_E(\mathcal{U}') := \left\{ V \in \partial(\mathcal{U}') \,\middle|\, V \text{ and } F = \mathcal{N}(V) \cap \mathcal{U}' \ E\text{-respectful} \right\}$$

Respectful expander

An $(\mathcal{U}, \mathcal{V})_E$-graph is an (s, δ, E)-respectful expander if for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$ it holds that $\left|\partial^{\mathsf{r}}_E(\mathcal{U}')\right| \geq \delta |\mathcal{U}'|$

Theorem ([Mikša & Nordström '15] building on [AR01])

If $\mathcal{F}$ has (s, δ, E)-respectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{PC proof degree} > \frac{\delta s}{2\ell}$$

Respectful Expanders and Degree Lower Bounds

Define respectful boundary to be

$$\partial^{\mathsf{r}}_{E}(\mathcal{U}') := \{V \in \partial(\mathcal{U}') \mid V \text{ and } F = \mathcal{N}(V) \cap \mathcal{U}' \ E\text{-respectful}\}$$

Respectful expander

An $(\mathcal{U}, \mathcal{V})_E$-graph is an (s, δ, E)-respectful expander if for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$ it holds that $|\partial^{\mathsf{r}}_{E}(\mathcal{U}')| \geq \delta|\mathcal{U}'|$

Theorem ([Mikša & Nordström '15] building on [AR01])

If $\mathcal{F}$ has (s, δ, E)-respectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{PC proof degree} > \frac{\delta s}{2\ell}$$

(Also holds for sets of polynomials not obtained from CNFs)

Generalized Method for Degree Lower Bounds

Theorem ([Mikša & Nordström '15] building on [AR01])

If $\mathcal{F}$ has (s, δ, E)-respectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{PC proof degree} > \frac{\delta s}{2\ell}$$

Generalized Method for Degree Lower Bounds

Theorem ([Mikša & Nordström '15] building on [AR01])

If $\mathcal{F}$ has (s, δ, E)-respectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{PC proof degree} > \frac{\delta s}{2\ell}$$

Proof by careful adaptation of [Alekhnovich & Razborov '01]
(but fairly involved — can't say anything much)

Generalized Method for Degree Lower Bounds

Theorem ([Mikša & Nordström '15] building on [AR01])

If $\mathcal{F}$ has (s, δ, E)-respectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{PC proof degree} > \frac{\delta s}{2\ell}$$

Proof by careful adaptation of [Alekhnovich & Razborov '01]
(but fairly involved — can't say anything much)

Provides common framework for previous lower bounds:

- CNFs with expanding CVIGs [Alekhnovich & Razborov '01]
- "Vanilla" PHP formulas [Alekhnovich & Razborov '01]
- Ordering principle [Galesi & Lauria '10]
- Subset cardinality formulas [Mikša & Nordström '14]

Generalized Method for Degree Lower Bounds

Theorem ([Mikša & Nordström '15] building on [AR01])

If $\mathcal{F}$ has (s, δ, E)-respectful expander $(\mathcal{U}, \mathcal{V})_E$ with overlap ℓ, then

$$\textit{PC proof degree} > \frac{\delta s}{2\ell}$$

Proof by careful adaptation of [Alekhnovich & Razborov '01] (but fairly involved — can't say anything much)

Provides common framework for previous lower bounds:

- CNFs with expanding CVIGs [Alekhnovich & Razborov '01]
- "Vanilla" PHP formulas [Alekhnovich & Razborov '01]
- Ordering principle [Galesi & Lauria '10]
- Subset cardinality formulas [Mikša & Nordström '14]

New contribution: Functional PHP is hard

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP		
FPHP		
Onto-PHP		
Onto-FPHP		

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP	hard [Hak85]	
FPHP		
Onto-PHP		
Onto-FPHP		

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP	hard [Hak85]	
FPHP	hard [Hak85]	
Onto-PHP	hard [Hak85]	
Onto-FPHP	hard [Hak85]	

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP	hard [Hak85]	hard [AR01]
FPHP	hard [Hak85]	
Onto-PHP	hard [Hak85]	
Onto-FPHP	hard [Hak85]	

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP	hard [Hak85]	hard [AR01]
FPHP	hard [Hak85]	
Onto-PHP	hard [Hak85]	
Onto-FPHP	hard [Hak85]	easy! [Rii93]

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP	hard [Hak85]	hard [AR01]
FPHP	hard [Hak85]	?
Onto-PHP	hard [Hak85]	?
Onto-FPHP	hard [Hak85]	easy! [Rii93]

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP	hard [Hak85]	hard [AR01]
FPHP	hard [Hak85]	?
Onto-PHP	hard [Hak85]	hard [AR01]
Onto-FPHP	hard [Hak85]	easy! [Rii93]

This work

- Observe that [AR01] proves hardness of Onto-PHP

Hardness of Different Flavours of PHP

Variant	Resolution	Polynomial calculus
PHP	hard [Hak85]	hard [AR01]
FPHP	hard [Hak85]	hard [MN15]
Onto-PHP	hard [Hak85]	hard [AR01]
Onto-FPHP	hard [Hak85]	easy! [Rii93]

This work

- Observe that [AR01] proves hardness of Onto-PHP
- Prove that FPHP is hard in polynomial calculus

Degree Lower Bound for Functional PHP

Theorem ([MN15])

If G is a (standard) bipartite (s, δ)-boundary expander with left degree $\leq d$, then $FPHP_G$ requires PC degree $> \delta s/(2d)$.

Degree Lower Bound for Functional PHP

Theorem ([MN15])

If G is a (standard) bipartite (s, δ)-boundary expander with left degree $\leq d$, then $FPHP_G$ requires PC degree $> \delta s/(2d)$.

Proof: Just need to build expanding $(\mathcal{U}, \mathcal{V})_E$-graph

Degree Lower Bound for Functional PHP

Theorem ([MN15])

If G is a (standard) bipartite (s, δ)-boundary expander with left degree $\leq d$, then $FPHP_G$ requires PC degree $> \delta s/(2d)$.

Proof: Just need to build expanding $(\mathcal{U}, \mathcal{V})_E$-graph

- $F_i =$ pigeon axiom for pigeon i

Degree Lower Bound for Functional PHP

Theorem ([MN15])

If G is a (standard) bipartite (s, δ)-boundary expander with left degree $\leq d$, then $FPHP_G$ requires PC degree $> \delta s/(2d)$.

Proof: Just need to build expanding $(\mathcal{U}, \mathcal{V})_E$-graph

- F_i = pigeon axiom for pigeon i
- E = all hole and functional axioms

Degree Lower Bound for Functional PHP

Theorem ([MN15])

If G is a (standard) bipartite (s, δ)-boundary expander with left degree $\leq d$, then $FPHP_G$ requires PC degree $> \delta s/(2d)$.

Proof: Just need to build expanding $(\mathcal{U}, \mathcal{V})_E$-graph

- F_i = pigeon axiom for pigeon i
- E = all hole and functional axioms
- $V_j = \{p_{i',j'} \,|\, i' \in \mathcal{N}(j)$ and $j' \in \mathcal{N}(i')\}$

Degree Lower Bound for Functional PHP

Theorem ([MN15])

If G is a (standard) bipartite (s, δ)-boundary expander with left degree $\leq d$, then $FPHP_G$ requires PC degree $> \delta s/(2d)$.

Proof: Just need to build expanding $(\mathcal{U}, \mathcal{V})_E$-graph

- F_i = pigeon axiom for pigeon i
- E = all hole and functional axioms
- $V_j = \left\{ p_{i',j'} \middle| i' \in \mathcal{N}(j) \text{ and } j' \in \mathcal{N}(i') \right\}$
- Can prove (straightforward exercise):
  - Overlap ℓ satisfies $1 < \ell \leq d$
  - All V_j and F_i for $i \in \mathcal{N}(j)$ E-respectful neighbours
  - Original graph G and $(\mathcal{U}, \mathcal{V})_E$ isomorphic

Degree Lower Bound for Functional PHP

Theorem ([MN15])

If G is a (standard) bipartite (s, δ)-boundary expander with left degree $\leq d$, then $FPHP_G$ requires PC degree $> \delta s/(2d)$.

Proof: Just need to build expanding $(\mathcal{U}, \mathcal{V})_E$-graph

- F_i = pigeon axiom for pigeon i
- E = all hole and functional axioms
- $V_j = \left\{p_{i',j'} \middle| i' \in \mathcal{N}(j) \text{ and } j' \in \mathcal{N}(i')\right\}$
- Can prove (straightforward exercise):
  - Overlap ℓ satisfies $1 < \ell \leq d$
  - All V_j and F_i for $i \in \mathcal{N}(j)$ E-respectful neighbours
  - Original graph G and $(\mathcal{U}, \mathcal{V})_E$ isomorphic
- So get same expansion parameters, and theorem follows ☐

Open Problems

- Prove polynomial calculus lower bounds for other formulas

Open Problems

- Prove polynomial calculus lower bounds for other formulas
  - graph colouring formulas
  - independent set formulas

Open Problems

- Prove polynomial calculus lower bounds for other formulas
  - graph colouring formulas
  - independent set formulas
- Prove size lower bounds via technique that doesn't use degree

Open Problems

- Prove polynomial calculus lower bounds for other formulas
  - graph colouring formulas
  - independent set formulas
- Prove size lower bounds via technique that doesn't use degree
  - k-clique formulas
  - weak pigeonhole principle formulas ($\geq n^2$ pigeons)

Open Problems

- Prove polynomial calculus lower bounds for other formulas
  - graph colouring formulas
  - independent set formulas
- Prove size lower bounds via technique that doesn't use degree
  - k-clique formulas
  - weak pigeonhole principle formulas ($\geq n^2$ pigeons)
- Find truly general framework capturing all degree bounds

Open Problems

- Prove polynomial calculus lower bounds for other formulas
  - graph colouring formulas
  - independent set formulas
- Prove size lower bounds via technique that doesn't use degree
  - k-clique formulas
  - weak pigeonhole principle formulas ($\geq n^2$ pigeons)
- Find truly general framework capturing all degree bounds
  - We generalize only part of [Alekhnovich & Razborov '01]
  - Cannot deal with lower bounds à la [Buss et al. '99]

Open Problems

- Prove polynomial calculus lower bounds for other formulas
  - graph colouring formulas
  - independent set formulas
- Prove size lower bounds via technique that doesn't use degree
  - k-clique formulas
  - weak pigeonhole principle formulas ($\geq n^2$ pigeons)
- Find truly general framework capturing all degree bounds
  - We generalize only part of [Alekhnovich & Razborov '01]
  - Cannot deal with lower bounds à la [Buss et al. '99]
- Go beyond polynomial calculus (e.g. to Positivstellensatz)

Take-away Message

Generalized method for PC degree lower bounds

- Unified framework for most previous lower bounds
- Exponential size lower bound for functional PHP

Future directions

- Extend techniques further to other tricky formulas
- Develop non-degree-based size lower bound techniques

Take-away Message

Generalized method for PC degree lower bounds

- Unified framework for most previous lower bounds
- Exponential size lower bound for functional PHP

Future directions

- Extend techniques further to other tricky formulas
- Develop non-degree-based size lower bound techniques

Thank you for your attention!