A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds

Mladen Mikša

KTH Royal Institute of Technology
Stockholm, Sweden

30th Annual Computational Complexity Conference

Portland, Oregon, USA
19 June 2015

Joint work with Jakob Nordström

Introduction

Topic: Proof complexity

Focus: Polynomial calculus (Gröbner basis calculations)

Goal: Degree lower bounds ($\Rightarrow$ Size lower bounds)

Lower Bounds via Expansion

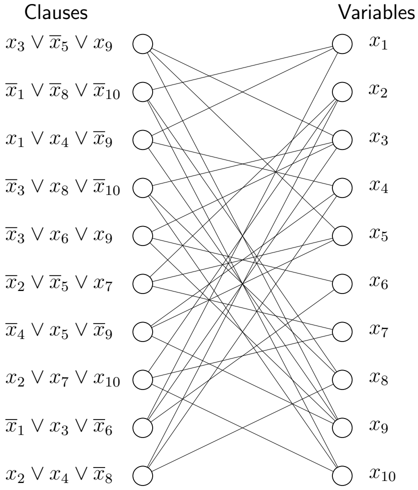
Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

Lower Bounds via Expansion

Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

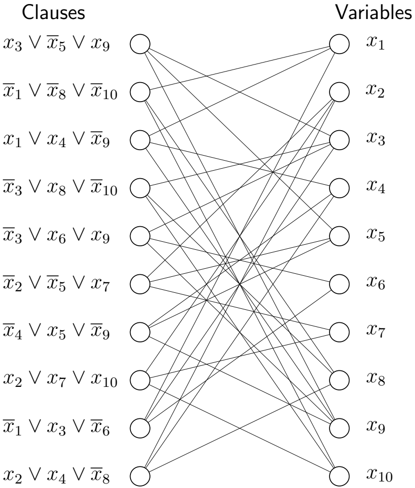


Lower Bounds via Expansion

Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

Boundary expansion:
Subsets of left vertices have many unique neighbors on right.

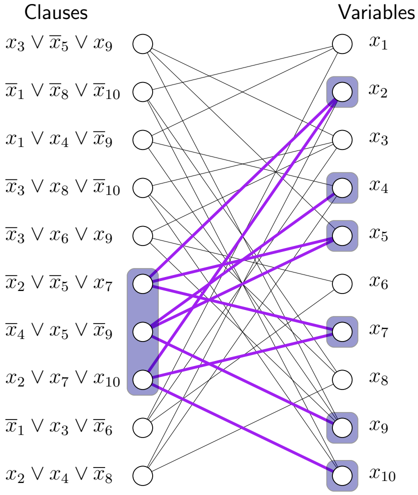


Lower Bounds via Expansion

Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

Boundary expansion:
Subsets of left vertices have many unique neighbors on right.

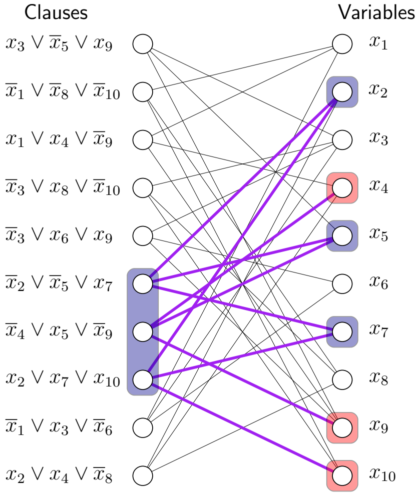


Lower Bounds via Expansion

Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

Boundary expansion:
Subsets of left vertices have many unique neighbors on right.

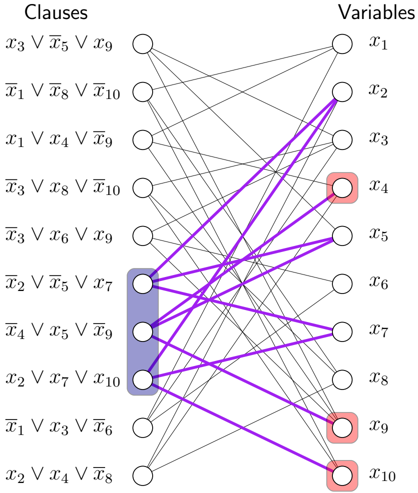


Lower Bounds via Expansion

Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

Boundary expansion:
Subsets of left vertices have many unique neighbors on right.



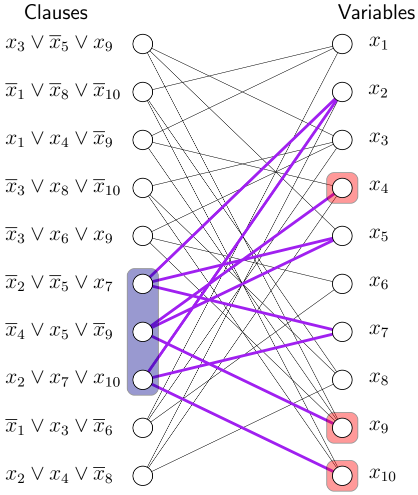
Lower Bounds via Expansion

Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

Boundary expansion:
Subsets of left vertices have many unique neighbors on right.

Problem:
CVIG might lose expansion.



Lower Bounds via Expansion

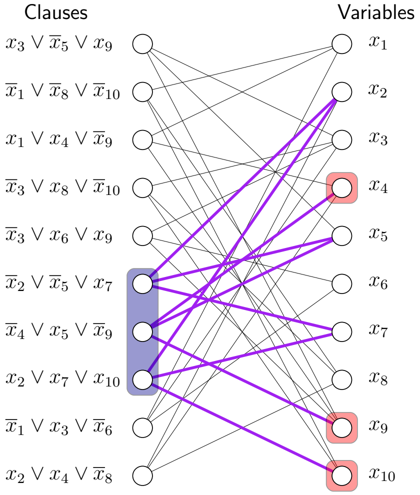
Standard approach: Lower bounds from expansion.

Simplest example: Clause-variable incidence graph (CVIG).

Boundary expansion:
Subsets of left vertices have many unique neighbors on right.

Problem:
CVIG might lose expansion.

Need graph capturing underlying principle!



Our Results

Main Theorem (Informal)

Graph structure on formula such that expansion implies hardness in polynomial calculus.

Extends an approach from [Alekhnovich, Razborov '01].

Unifies (almost) all previous lower bounds.

Corollary

Functional pigeonhole principle is hard for polynomial calculus.

Resolves question in [Razborov '02].

Our Results

Main Theorem (Informal)

Graph structure on formula such that expansion implies hardness in polynomial calculus.

Extends an approach from [Alekhnovich, Razborov '01].

Unifies (almost) all previous lower bounds.

Corollary

Functional pigeonhole principle is hard for polynomial calculus.

Resolves question in [Razborov '02].

Warm-up: Use resolution to present main ideas and challenges.

Resolution

- **Input:** CNF formula $\mathcal{F}$

$$(x \vee \overline{y} \vee z) \wedge (\overline{y} \vee \overline{z}) \wedge (x \vee y) \wedge (\overline{x} \vee \overline{z}) \wedge (\overline{x} \vee z)$$

- **Resolution rule:**

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

- **Goal:** Proof of unsatisfiability (refutation) = Derive empty clause $\perp$

Refer to clauses of formula as axioms.

Complexity Measures for Resolution

Size: number of steps in proof

Width: size of the largest clause

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

Complexity Measures for Resolution

Size: number of steps in proof 9

Width: size of the largest clause

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

Complexity Measures for Resolution

Size: number of steps in proof 9

Width: size of the largest clause

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

Complexity Measures for Resolution

Size: number of steps in proof 9

Width: size of the largest clause 3

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

Complexity Measures for Resolution

Size: number of steps in proof 9

Width: size of the largest clause 3

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

Complexity Measures for Resolution

Size: number of steps in proof 9

Width: size of the largest clause 3

Theorem (Ben-Sasson, Wigderson '99)

$$\mathbf{Size} \gtrsim \exp(\mathbf{Width})$$

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

Complexity Measures for Resolution

Size: number of steps in proof 9

Width: size of the largest clause 3

Theorem (Ben-Sasson, Wigderson '99)

$$\mathbf{Size} \gtrsim \exp(\mathbf{Width})$$

Width lower bounds via expansion argument.

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

Example: Tseitin Formulas

Given set of equations over $\mathbb{F}_2$.

$$
\begin{aligned}
x + w &= 0 \\
x + y &= 0 \\
y + w + z &= 1 \\
z &= 0
\end{aligned}
$$

Example: Tseitin Formulas

Given set of equations over $\mathbb{F}_2$.

$$x + w = 0$$
$$x + y = 0$$
$$y + w + z = 1$$
$$z = 0$$

Encode as clauses.

Clauses

$x \vee \overline{w}$

$\overline{x} \vee w$

$x \vee \overline{y}$

$\overline{x} \vee y$

$y \vee w \vee z$

$\overline{y} \vee \overline{w} \vee z$

$\overline{y} \vee w \vee \overline{z}$

$y \vee \overline{w} \vee \overline{z}$

$\overline{z}$

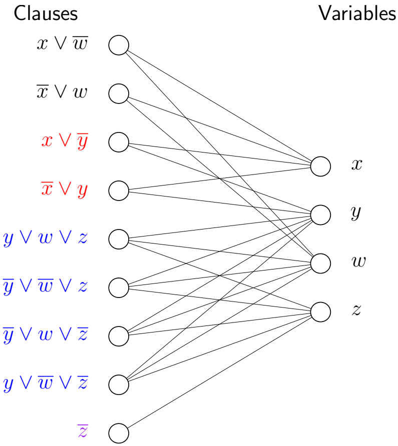
Example: Tseitin Formulas

Given set of equations over $\mathbb{F}_2$.

$$x + w = 0$$
$$x + y = 0$$
$$y + w + z = 1$$
$$z = 0$$

Encode as clauses.

Does CVIG expand?



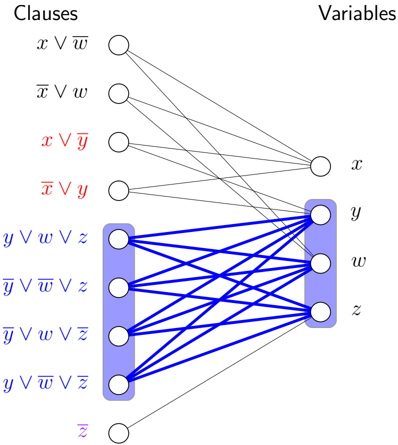
Example: Tseitin Formulas

Given set of equations over $\mathbb{F}_2$.

$$x + w = 0$$
$$x + y = 0$$
$$y + w + z = 1$$
$$z = 0$$

Encode as clauses.

Does CVIG expand? No!



Example: Tseitin Formulas

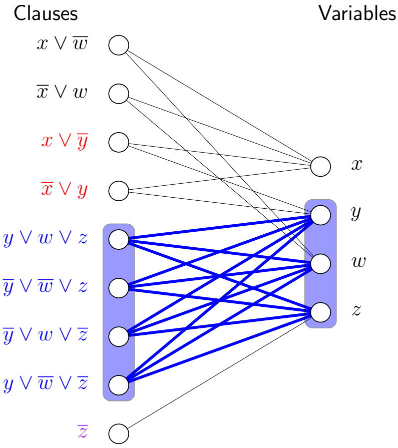
Given set of equations over $\mathbb{F}_2$.

$$x + w = 0$$
$$x + y = 0$$
$$y + w + z = 1$$
$$z = 0$$

Encode as clauses.

Does CVIG expand? No!

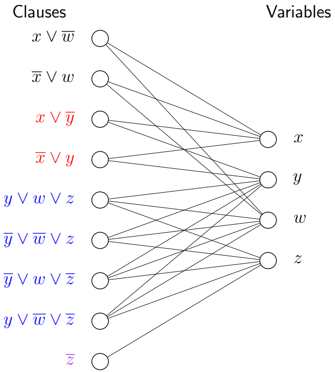
Graph should encode equations not clauses!

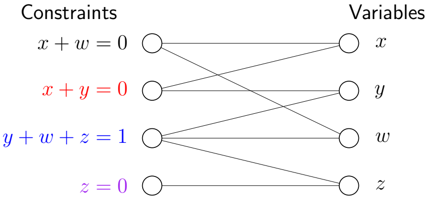


Constraint-Variable Incidence Graph

Have single vertex for each constraint on the left.

Put edge if variable appears in constraint.

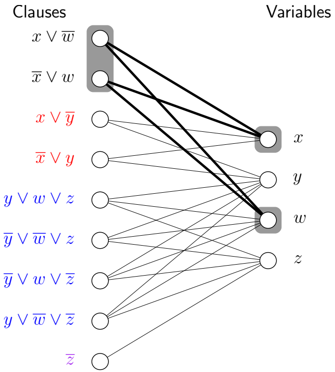


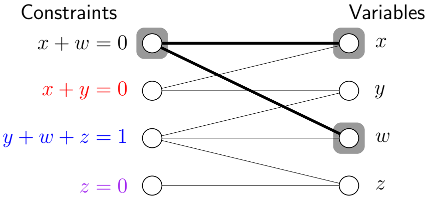


Constraint-Variable Incidence Graph

Have single vertex for each constraint on the left.

Put edge if variable appears in constraint.

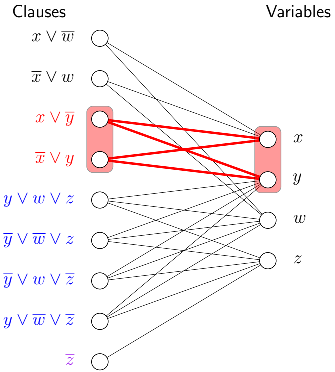


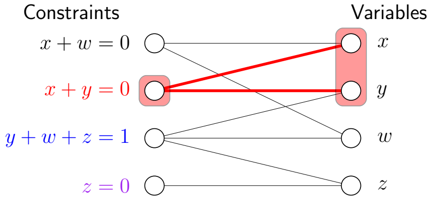


Constraint-Variable Incidence Graph

Have single vertex for each constraint on the left.

Put edge if variable appears in constraint.

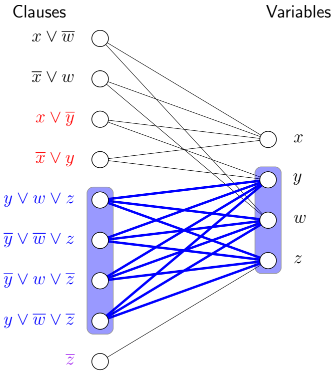


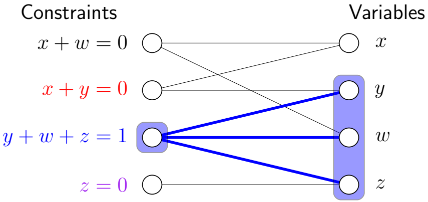


Constraint-Variable Incidence Graph

Have single vertex for each constraint on the left.

Put edge if variable appears in constraint.

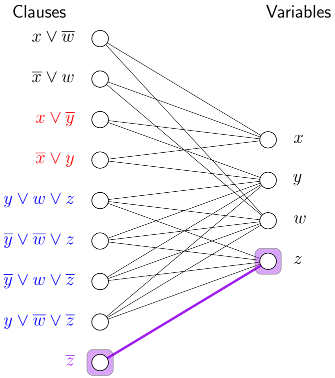


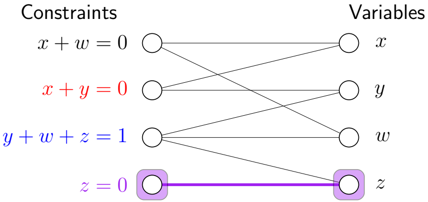


Constraint-Variable Incidence Graph

Have single vertex for each constraint on the left.

Put edge if variable appears in constraint.

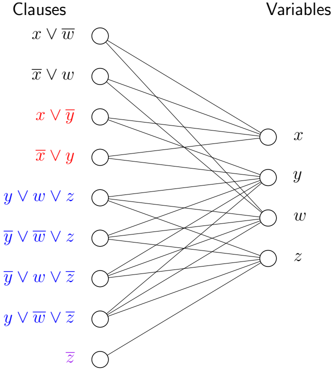


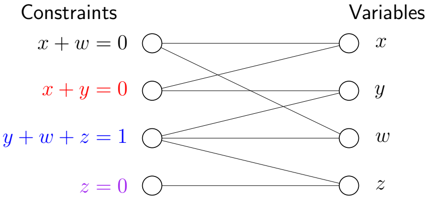


Constraint-Variable Incidence Graph

Have single vertex for each constraint on the left.

Put edge if variable appears in constraint.

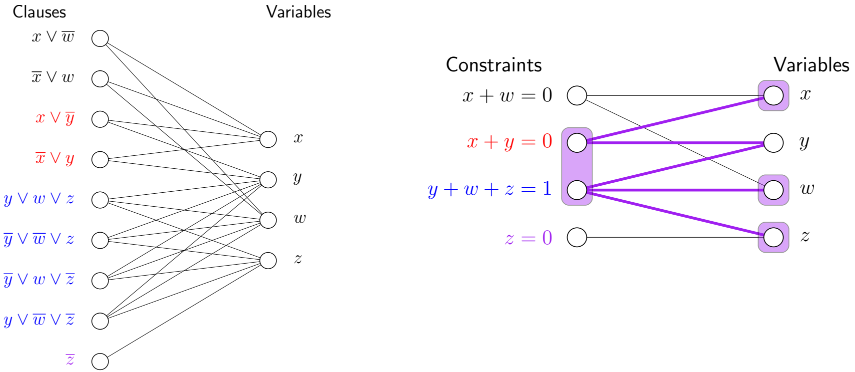




Constraint-Variable Incidence Graph

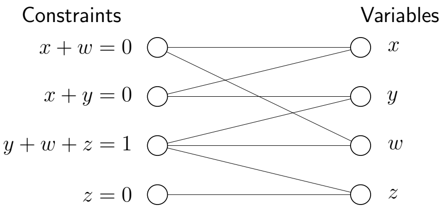
Have single vertex for each constraint on the left.

Put edge if variable appears in constraint.

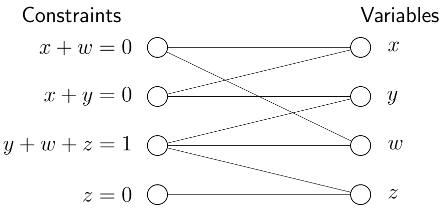


The constraint-variable incidence graph expands!

Proof Sketch of Tseitin Lower Bound

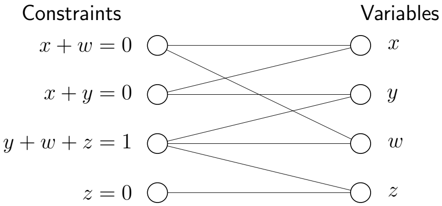


Proof Sketch of Tseitin Lower Bound



1. For each clause, look at constraints needed to derive it.

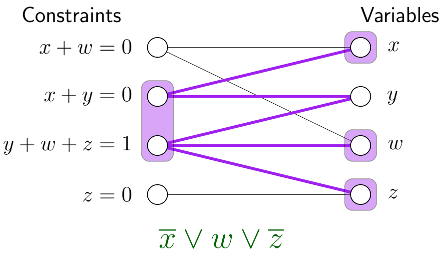
Proof Sketch of Tseitin Lower Bound



1. For each clause, look at constraints needed to derive it.

Axioms:	1 constraint needed
Contradiction ⊥:	All constraints

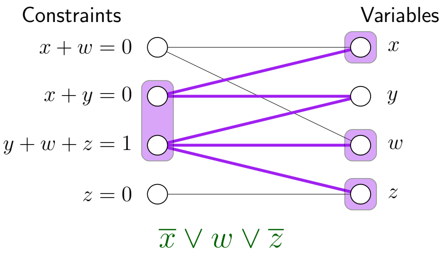
Proof Sketch of Tseitin Lower Bound



1. For each clause, look at constraints needed to derive it.

Axioms:	1 constraint needed
Contradiction $\perp$:	All constraints
Halfway through:	Clause C depending on medium-sized set S

Proof Sketch of Tseitin Lower Bound

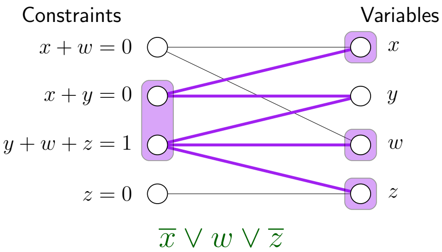


1. For each clause, look at constraints needed to derive it.

Axioms:	1 constraint needed
Contradiction $\perp$:	All constraints
Halfway through:	Clause C depending on medium-sized set S

2. S has large boundary expansion $\Rightarrow$ All boundary variables in C

Proof Sketch of Tseitin Lower Bound



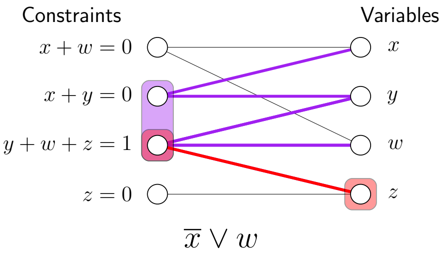
1. For each clause, look at constraints needed to derive it.

Axioms:	1 constraint needed
Contradiction $\perp$:	All constraints
Halfway through:	Clause C depending on medium-sized set S

2. S has large boundary expansion $\Rightarrow$ All boundary variables in C

3. Suppose not $\Rightarrow$ not all of S needed for C

Proof Sketch of Tseitin Lower Bound

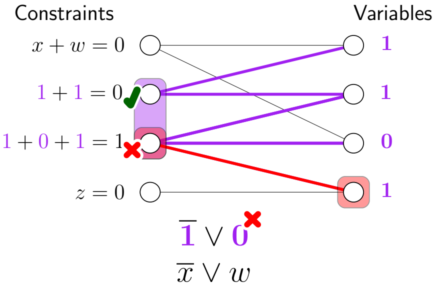


1. For each clause, look at constraints needed to derive it.

Axioms:	1 constraint needed
Contradiction $\bot$:	All constraints
Halfway through:	Clause C depending on medium-sized set S

2. S has large boundary expansion $\Rightarrow$ All boundary variables in C
3. Suppose not $\Rightarrow$ not all of S needed for C; e.g., C doesn't have z

Proof Sketch of Tseitin Lower Bound

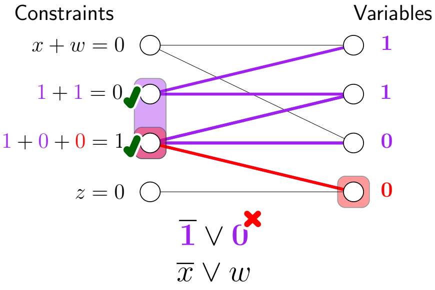


1. For each clause, look at constraints needed to derive it.

Axioms:	1 constraint needed
Contradiction $\perp$:	All constraints
Halfway through:	Clause C depending on medium-sized set S

2. S has large boundary expansion $\Rightarrow$ All boundary variables in C
3. Suppose not $\Rightarrow$ not all of S needed for C; e.g., C doesn't have z

Proof Sketch of Tseitin Lower Bound



1. For each clause, look at constraints needed to derive it.

Axioms:	1 constraint needed
Contradiction $\perp$:	All constraints
Halfway through:	Clause C depending on medium-sized set S

2. S has large boundary expansion $\Rightarrow$ All boundary variables in C
3. Suppose not $\Rightarrow$ not all of S needed for C; e.g., C doesn't have z

Resolution Lower Bounds

Resolution edge game on (P, x)

1. Adversary provides assignment ρ to all variables.
2. Can flip x to some b so that P is satisfied.

Resolution Lower Bounds

Resolution edge game on (P, x)

1. Adversary provides assignment ρ to all variables.
2. Can flip x to some b so that P is satisfied.

Theorem (Ben-Sasson, Wigderson '99)

If from formula $\mathcal{F} = \bigwedge_{P \in \mathcal{F}} P$, we can form graph $\mathcal{G}(\mathcal{F})$ such that

- *$\mathcal{G}(\mathcal{F})$ is expanding, and*
- *for all edges (P, x), P is satisfied by flipping x,*

then refuting $\mathcal{F}$ requires large width.

Polynomial Calculus [CEI '96, ABRW '00]

Lines are polynomial equations over some field $\mathbb{F}$.

- **Input:** Polynomial equations encoding Boolean constraints

  Clause encoded as: $x \vee \overline{y} \vee z \;\rightarrow\; \overline{x} y \overline{z} = 0$

  Additional axioms: $x^2 - x = 0$ and $x + \overline{x} - 1 = 0$

- **Linear combination:**

$$\frac{p = 0 \qquad q = 0}{\alpha p + \beta q = 0}$$

- **Variable multiplication:**

$$\frac{p = 0}{xp = 0}$$

- **Goal:** Derive $1 = 0$ showing that constraints are unsatisfiable

Complexity Measures for Polynomial Calculus

Size: number of monomials in proof

Degree: max degree of monomial

1.	$x\overline{y} = 0$	Axiom
2.	$y = 0$	Axiom
3.	$y + \overline{y} - 1 = 0$	Axiom
4.	$\overline{y} - 1 = 0$	Lin$(2, 3)$
5.	$x\overline{y} - x = 0$	Mul$(4, x)$
6.	$x = 0$	Lin$(1, 5)$
7.	$x + \overline{z} + 1 = 0$	Axiom
8.	$\overline{z} + 1 = 0$	Lin$(6, 7)$
9.	$\overline{z} = 0$	Axiom
10.	$1 = 0$	Lin$(8, 9)$

Complexity Measures for Polynomial Calculus

Size: number of monomials in proof 17

Degree: max degree of monomial

1.	$x\overline{y}^{\mathbf{1}} = 0$	Axiom
2.	$y^{\mathbf{2}} = 0$	Axiom
3.	$y^{\mathbf{3}} + \overline{y}^{\mathbf{4}} - 1^{\mathbf{5}} = 0$	Axiom
4.	$\overline{y}^{\mathbf{6}} - 1^{\mathbf{7}} = 0$	Lin$(2, 3)$
5.	$x\overline{y}^{\mathbf{8}} - x^{\mathbf{9}} = 0$	Mul$(4, x)$
6.	$x^{\mathbf{10}} = 0$	Lin$(1, 5)$
7.	$x^{\mathbf{11}} + \overline{z}^{\mathbf{12}} + 1^{\mathbf{13}} = 0$	Axiom
8.	$\overline{z}^{\mathbf{14}} + 1^{\mathbf{15}} = 0$	Lin$(6, 7)$
9.	$\overline{z}^{\mathbf{16}} = 0$	Axiom
10.	$1^{\mathbf{17}} = 0$	Lin$(8, 9)$

Complexity Measures for Polynomial Calculus

Size: number of monomials in proof 17

Degree: max degree of monomial

1.	$x\overline{y} = 0$	Axiom
2.	$y = 0$	Axiom
3.	$y + \overline{y} - 1 = 0$	Axiom
4.	$\overline{y} - 1 = 0$	Lin$(2, 3)$
5.	$x\overline{y} - x = 0$	Mul$(4, x)$
6.	$x = 0$	Lin$(1, 5)$
7.	$x + \overline{z} + 1 = 0$	Axiom
8.	$\overline{z} + 1 = 0$	Lin$(6, 7)$
9.	$\overline{z} = 0$	Axiom
10.	$1 = 0$	Lin$(8, 9)$

Complexity Measures for Polynomial Calculus

Size: number of monomials in proof 17

Degree: max degree of monomial 2

1.	$x\overline{y} = 0$	Axiom
2.	$y = 0$	Axiom
3.	$y + \overline{y} - 1 = 0$	Axiom
4.	$\overline{y} - 1 = 0$	Lin$(2, 3)$
5.	$x\overline{y} - x = 0$	Mul$(4, x)$
6.	$x = 0$	Lin$(1, 5)$
7.	$x + \overline{z} + 1 = 0$	Axiom
8.	$\overline{z} + 1 = 0$	Lin$(6, 7)$
9.	$\overline{z} = 0$	Axiom
10.	$1 = 0$	Lin$(8, 9)$

Complexity Measures for Polynomial Calculus

Size: number of monomials in proof 17

Degree: max degree of monomial 2

1.	$x\overline{y} = 0$	Axiom
2.	$y = 0$	Axiom
3.	$y + \overline{y} - 1 = 0$	Axiom
4.	$\overline{y} - 1 = 0$	Lin$(2, 3)$
5.	$x\overline{y} - x = 0$	Mul$(4, x)$
6.	$x = 0$	Lin$(1, 5)$
7.	$x + \overline{z} + 1 = 0$	Axiom
8.	$\overline{z} + 1 = 0$	Lin$(6, 7)$
9.	$\overline{z} = 0$	Axiom
10.	$1 = 0$	Lin$(8, 9)$

Complexity Measures for Polynomial Calculus

Size: number of monomials in proof 17

Degree: max degree of monomial 2

Theorem (Impagliazzo, Pudlák, Sgall '99)

$$\mathbf{Size} \gtrsim \exp(\mathbf{Degree})$$

Used in:

- Buss, Grigoriev, Impagliazzo, Pitassi '99
- Ben-Sasson, Impagliazzo '99
- Alekhnovich, Razborov '01
- Galesi, Lauria '10

1.	$x\overline{y} = 0$	Axiom
2.	$y = 0$	Axiom
3.	$y + \overline{y} - 1 = 0$	Axiom
4.	$\overline{y} - 1 = 0$	Lin$(2, 3)$
5.	$x\overline{y} - x = 0$	Mul$(4, x)$
6.	$x = 0$	Lin$(1, 5)$
7.	$x + \overline{z} + 1 = 0$	Axiom
8.	$\overline{z} + 1 = 0$	Lin$(6, 7)$
9.	$\overline{z} = 0$	Axiom
10.	$1 = 0$	Lin$(8, 9)$

Complexity Measures for Polynomial Calculus

Size: number of monomials in proof 17

Degree: max degree of monomial 2

Theorem (Impagliazzo, Pudlák, Sgall '99)

$$\mathbf{Size} \gtrsim \exp\left(\mathbf{Degree}\right)$$

Used in:

- Buss, Grigoriev, Impagliazzo, Pitassi '99
- Ben-Sasson, Impagliazzo '99
- Alekhnovich, Razborov '01
- Galesi, Lauria '10

Polynomial calculus exponentially stronger than resolution.

1.	$x\overline{y} = 0$	Axiom
2.	$y = 0$	Axiom
3.	$y + \overline{y} - 1 = 0$	Axiom
4.	$\overline{y} - 1 = 0$	Lin$(2,3)$
5.	$x\overline{y} - x = 0$	Mul$(4,x)$
6.	$x = 0$	Lin$(1,5)$
7.	$x + \overline{z} + 1 = 0$	Axiom
8.	$\overline{z} + 1 = 0$	Lin$(6,7)$
9.	$\overline{z} = 0$	Axiom
10.	$1 = 0$	Lin$(8,9)$

Polynomial Calculus Edge Game

Tseitin: linear equations $\Rightarrow$ easy over $\mathbb{F}_2$ (Gaussian elimination)

Need stronger guarantee from constraint-variable incidence graph!

Polynomial Calculus Edge Game

Tseitin: linear equations $\Rightarrow$ easy over $\mathbb{F}_2$ (Gaussian elimination)

Need stronger guarantee from constraint-variable incidence graph!

Resolution graph:

- Graph is boundary expander.
- Can play resolution edge game on every edge (P, x).

Polynomial Calculus Edge Game

Tseitin: linear equations $\Rightarrow$ easy over $\mathbb{F}_2$ (Gaussian elimination)

Need stronger guarantee from constraint-variable incidence graph!

Resolution graph:

- Graph is boundary expander.
- Can play resolution edge game on every edge (P, x).

Need to play harder game!

Polynomial Calculus Edge Game

Tseitin: linear equations ⇒ easy over $\mathbb{F}_2$ (Gaussian elimination)

Need stronger guarantee from constraint-variable incidence graph!

Resolution graph:

- Graph is boundary expander.
- Can play resolution edge game on every edge (P, x).

Need to play harder game!

Polynomial calculus edge game on (P, x)

1. Commit to assignment $x = b$ ahead of time.
2. Adversary provides assignment ρ to all variables.
3. Flipping $x = b$ satisfies P.

Can't win this game for Tseitin.

Main Theorem (Tentative Version)

If from formula $\mathcal{F} = \bigwedge_{P \in \mathcal{F}} P$ we can form $\mathcal{G}(\mathcal{F})$:

- $\mathcal{G}(\mathcal{F})$ is expanding, and
- for all edges (P, x), P fixed to true by x,

then refuting $\mathcal{F}$ requires large degree.

Main Theorem (Tentative Version)

If from formula $\mathcal{F} = \bigwedge_{P \in \mathcal{F}} P$ we can form $\mathcal{G}(\mathcal{F})$:

- $\mathcal{G}(\mathcal{F})$ is expanding, and
- for all edges (P, x), P fixed to true by x,

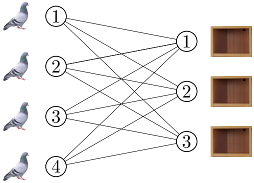
then refuting $\mathcal{F}$ requires large degree.

Not enough to prove functional pigeonhole principle hard!

Pigeonhole Principle (PHP)

Statement: $n+1$ pigeons can fit into n holes

Variable $x_{1,3}$ is true if pigeon 1 sits in hole 3



Pigeonhole Principle (PHP)

Statement: $n + 1$ pigeons can fit into n holes

Variable $x_{1,3}$ is true if pigeon 1 sits in hole 3

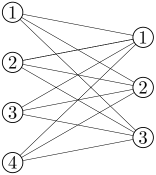
Pigeons

$x_{1,1} \vee x_{1,2} \vee x_{1,3}$

$x_{2,1} \vee x_{2,2} \vee x_{2,3}$

$x_{3,1} \vee x_{3,2} \vee x_{3,3}$

$x_{4,1} \vee x_{4,2} \vee x_{4,3}$



Holes

$\overline{x}_{1,1} \vee \overline{x}_{2,1}$ $\quad$ $\overline{x}_{2,1} \vee \overline{x}_{3,1}$
$\overline{x}_{1,1} \vee \overline{x}_{3,1}$ $\quad$ $\overline{x}_{2,1} \vee \overline{x}_{4,1}$
$\overline{x}_{1,1} \vee \overline{x}_{4,1}$ $\quad$ $\overline{x}_{3,1} \vee \overline{x}_{4,1}$

$\overline{x}_{1,2} \vee \overline{x}_{2,2}$ $\quad$ $\overline{x}_{2,2} \vee \overline{x}_{3,2}$
$\overline{x}_{1,2} \vee \overline{x}_{3,2}$ $\quad$ $\overline{x}_{2,2} \vee \overline{x}_{4,2}$
$\overline{x}_{1,2} \vee \overline{x}_{4,2}$ $\quad$ $\overline{x}_{3,2} \vee \overline{x}_{4,2}$

$\overline{x}_{1,3} \vee \overline{x}_{2,3}$ $\quad$ $\overline{x}_{2,3} \vee \overline{x}_{3,3}$
$\overline{x}_{1,3} \vee \overline{x}_{3,3}$ $\quad$ $\overline{x}_{2,3} \vee \overline{x}_{4,3}$
$\overline{x}_{1,3} \vee \overline{x}_{4,3}$ $\quad$ $\overline{x}_{3,3} \vee \overline{x}_{4,3}$

Pigeonhole Principle (PHP)

Statement: $n+1$ pigeons can fit into n holes

Variable $x_{1,3}$ is true if pigeon 1 sits in hole 3

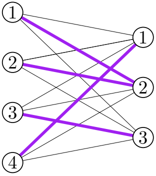
Pigeons

$x_{1,1} \vee x_{1,2} \vee x_{1,3}$

$x_{2,1} \vee x_{2,2} \vee x_{2,3}$

$x_{3,1} \vee x_{3,2} \vee x_{3,3}$

$x_{4,1} \vee x_{4,2} \vee x_{4,3}$



Holes

$\overline{x}_{1,1} \vee \overline{x}_{2,1}$ $\quad \overline{x}_{2,1} \vee \overline{x}_{3,1}$

$\overline{x}_{1,1} \vee \overline{x}_{3,1}$ $\quad \overline{x}_{2,1} \vee \overline{x}_{4,1}$

$\overline{x}_{1,1} \vee \overline{x}_{4,1}$ $\quad \overline{x}_{3,1} \vee \overline{x}_{4,1}$

$\overline{x}_{1,2} \vee \overline{x}_{2,2}$ $\quad \overline{x}_{2,2} \vee \overline{x}_{3,2}$

$\overline{x}_{1,2} \vee \overline{x}_{3,2}$ $\quad \overline{x}_{2,2} \vee \overline{x}_{4,2}$

$\overline{x}_{1,2} \vee \overline{x}_{4,2}$ $\quad \overline{x}_{3,2} \vee \overline{x}_{4,2}$

$\overline{x}_{1,3} \vee \overline{x}_{2,3}$ $\quad \overline{x}_{2,3} \vee \overline{x}_{3,3}$

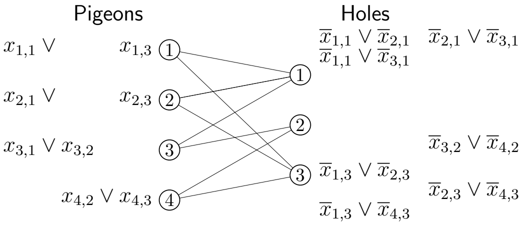
$\overline{x}_{1,3} \vee \overline{x}_{3,3}$ $\quad \overline{x}_{2,3} \vee \overline{x}_{4,3}$

$\overline{x}_{1,3} \vee \overline{x}_{4,3}$ $\quad \overline{x}_{3,3} \vee \overline{x}_{4,3}$

Pigeonhole Principle (PHP)

Statement: $n+1$ pigeons can fit into n holes

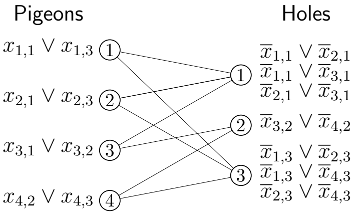
Variable $x_{1,3}$ is true if pigeon 1 sits in hole 3



Pigeonhole Principle (PHP)

Statement: $n+1$ pigeons can fit into n holes

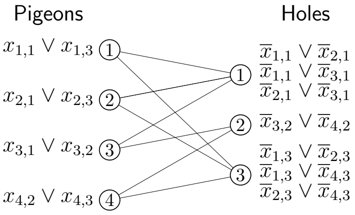
Variable $x_{1,3}$ is true if pigeon 1 sits in hole 3

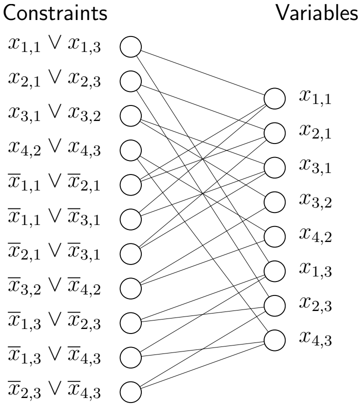


Pigeonhole Principle (PHP)

Statement: $n+1$ pigeons can fit into n holes

Variable $x_{1,3}$ is true if pigeon 1 sits in hole 3

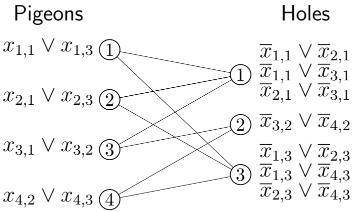


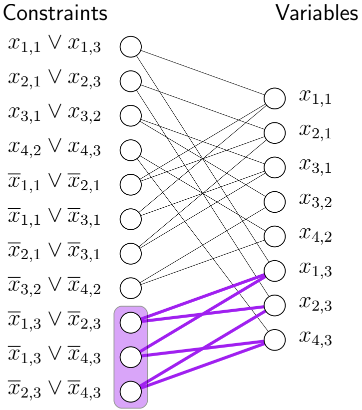


Pigeonhole Principle (PHP)

Statement: $n+1$ pigeons can fit into n holes

Variable $x_{1,3}$ is true if pigeon 1 sits in hole 3





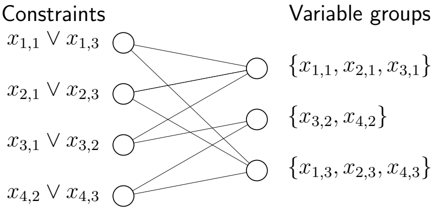
Again CVIG not expanding!

Proving PHP Lower Bound

Isolate hole axioms from graph and group hole variables together!

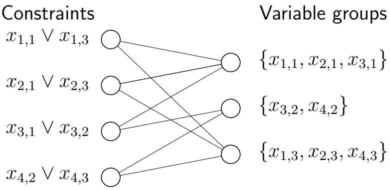
Proving PHP Lower Bound

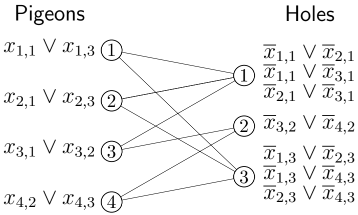
Isolate hole axioms from graph and group hole variables together!



Proving PHP Lower Bound

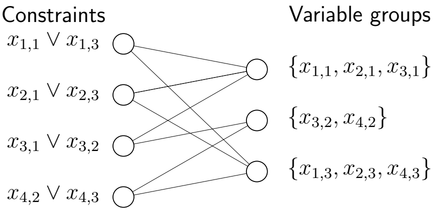
Isolate hole axioms from graph and group hole variables together!





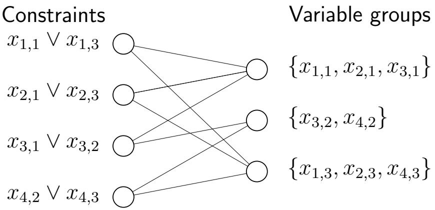
Proving PHP Lower Bound

Isolate hole axioms from graph and group hole variables together!



Proving PHP Lower Bound

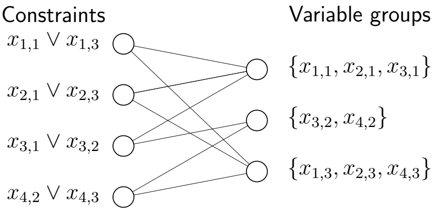
Isolate hole axioms from graph and group hole variables together!



Change the game: Assign group so that hole axioms (E) aren't violated!

Proving PHP Lower Bound

Isolate hole axioms from graph and group hole variables together!



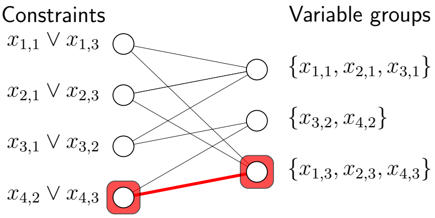
Change the game: Assign group so that hole axioms (E) aren't violated!

Polynomial calculus edge game on (P, V) with E

1. Commit to assignment ρ_V to variables in V ahead of time.
2. Adversary provides assignment ρ to all variables that satisfies E.
3. Flipping V to ρ_V satisfies $P \land E$.

Proving PHP Lower Bound

Isolate hole axioms from graph and group hole variables together!



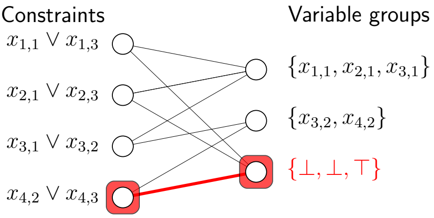
Change the game: Assign group so that hole axioms (E) aren't violated!

Polynomial calculus edge game on (P, V) with E

1. Commit to assignment ρ_V to variables in V ahead of time.
2. Adversary provides assignment ρ to all variables that satisfies E.
3. Flipping V to ρ_V satisfies $P \wedge E$.

Proving PHP Lower Bound

Isolate hole axioms from graph and group hole variables together!



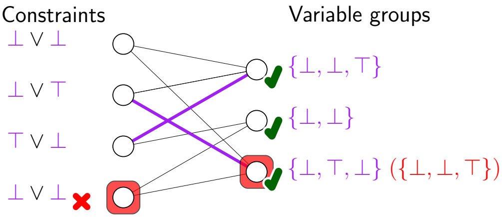
Change the game: Assign group so that hole axioms (E) aren't violated!

Polynomial calculus edge game on (P, V) with E

1. Commit to assignment ρ_V to variables in V ahead of time.
2. Adversary provides assignment ρ to all variables that satisfies E.
3. Flipping V to ρ_V satisfies $P \wedge E$.

Proving PHP Lower Bound

Isolate hole axioms from graph and group hole variables together!



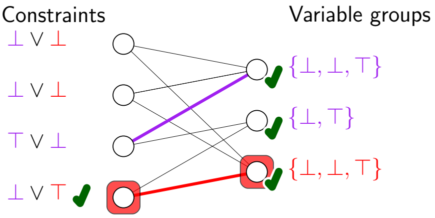
Change the game: Assign group so that hole axioms (E) aren't violated!

Polynomial calculus edge game on (P, V) with E

1. Commit to assignment ρ_V to variables in V ahead of time.
2. Adversary provides assignment ρ to all variables that satisfies E.
3. Flipping V to ρ_V satisfies $P \land E$.

Proving PHP Lower Bound

Isolate hole axioms from graph and group hole variables together!



Change the game: Assign group so that hole axioms (E) aren't violated!

Polynomial calculus edge game on (P, V) with E

1. Commit to assignment ρ_V to variables in V ahead of time.
2. Adversary provides assignment ρ to all variables that satisfies E.
3. Flipping V to ρ_V satisfies $P \wedge E$.

Main Theorem

Main Theorem

If from formula $\mathcal{F} = \mathcal{F}' \wedge E$, where $\mathcal{F}' = \bigwedge_{P \in \mathcal{F}'} P$, we can form $\mathcal{G}(\mathcal{F}')$:

- $\mathcal{G}(\mathcal{F}')$ is expanding, and
- for all edges (P, V), P is fixed to true by V without violating E,

then refuting $\mathcal{F}$ requires large degree.

Gives common framework for previous lower bounds.

- Expanding CNF [Alekhnovich, Razborov '01]
- Pigeonhole principle [Alekhnovich, Razborov '01]
- Graph ordering principle [Galesi, Lauria '10]

Main Theorem

Main Theorem

If from formula $\mathcal{F} = \mathcal{F}' \wedge E$, where $\mathcal{F}' = \bigwedge_{P \in \mathcal{F}'} P$, we can form $\mathcal{G}(\mathcal{F}')$:

- $\mathcal{G}(\mathcal{F}')$ is expanding, and
- for all edges (P, V), P is fixed to true by V without violating E,

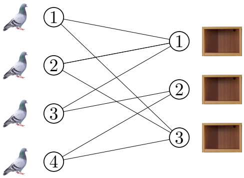
then refuting $\mathcal{F}$ requires large degree.

Gives common framework for previous lower bounds.

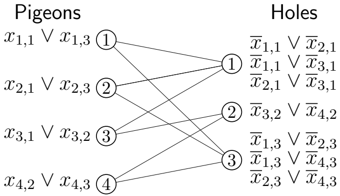
- Expanding CNF [Alekhnovich, Razborov '01]
- Pigeonhole principle [Alekhnovich, Razborov '01]
- Graph ordering principle [Galesi, Lauria '10]

Proves functional PHP hard.

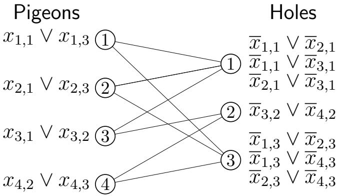
PHP Variants



PHP Variants

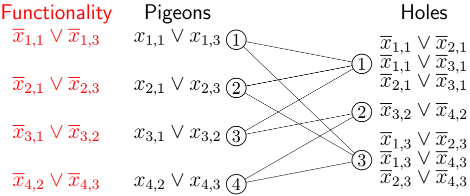


PHP Variants



- Can have fat pigeons which are assigned to multiple holes.

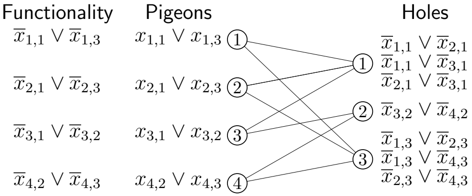
PHP Variants



- Can have fat pigeons which are assigned to multiple holes.

  ⇒ Add functionality axioms (makes mapping 1-to-1).

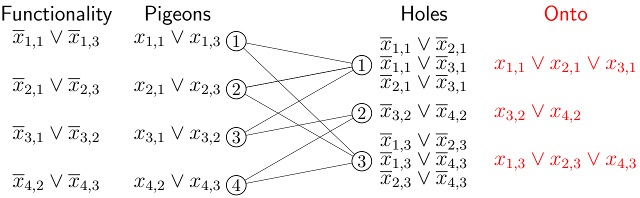
PHP Variants



- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).
- Can have hole with no pigeons.

PHP Variants



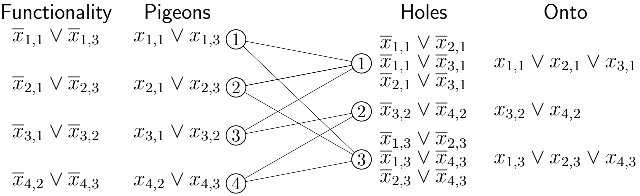
- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).

- Can have hole with no pigeons.

  $\Rightarrow$ Add onto axioms (makes mapping onto).

PHP Variants

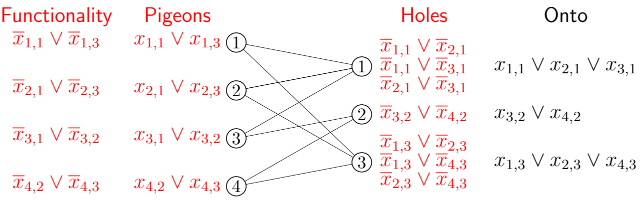


- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).
- Can have hole with no pigeons.

  $\Rightarrow$ Add onto axioms (makes mapping onto).

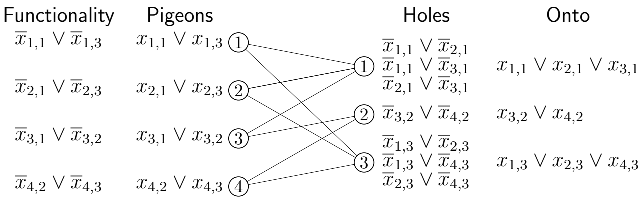
PHP Variants



- Can have fat pigeons which are assigned to multiple holes.
  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).
- Can have hole with no pigeons.
  $\Rightarrow$ Add onto axioms (makes mapping onto).

Functional PHP = PHP + Functionality

PHP Variants



- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).

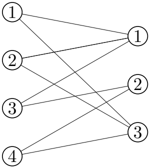
- Can have hole with no pigeons.

  $\Rightarrow$ Add onto axioms (makes mapping onto).

Functional PHP = PHP + Functionality

PHP Variants

Functionality	Pigeons	Holes	Onto
$\overline{x}_{1,1} \lor \overline{x}_{1,3}$	$x_{1,1} \lor x_{1,3}$ (1)	(1) $\overline{x}_{1,1} \lor \overline{x}_{2,1}$, $\overline{x}_{1,1} \lor \overline{x}_{3,1}$, $\overline{x}_{2,1} \lor \overline{x}_{3,1}$	$x_{1,1} \lor x_{2,1} \lor x_{3,1}$
$\overline{x}_{2,1} \lor \overline{x}_{2,3}$	$x_{2,1} \lor x_{2,3}$ (2)	(2) $\overline{x}_{3,2} \lor \overline{x}_{4,2}$	$x_{3,2} \lor x_{4,2}$
$\overline{x}_{3,1} \lor \overline{x}_{3,2}$	$x_{3,1} \lor x_{3,2}$ (3)	(3) $\overline{x}_{1,3} \lor \overline{x}_{2,3}$, $\overline{x}_{1,3} \lor \overline{x}_{4,3}$, $\overline{x}_{2,3} \lor \overline{x}_{4,3}$	$x_{1,3} \lor x_{2,3} \lor x_{4,3}$
$\overline{x}_{4,2} \lor \overline{x}_{4,3}$	$x_{4,2} \lor x_{4,3}$ (4)		



- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).

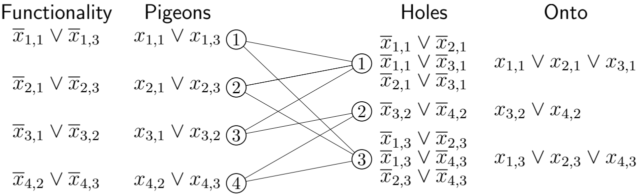
- Can have hole with no pigeons.

  $\Rightarrow$ Add onto axioms (makes mapping onto).

Functional PHP = PHP + Functionality

Onto-PHP = PHP + Onto

PHP Variants



- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).

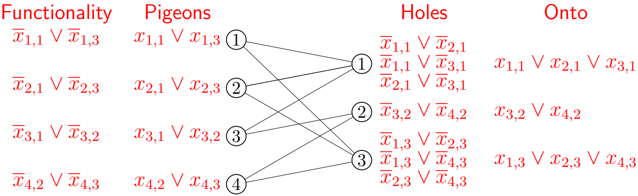
- Can have hole with no pigeons.

  $\Rightarrow$ Add onto axioms (makes mapping onto).

Functional PHP = PHP + Functionality

Onto-PHP = PHP + Onto

PHP Variants



- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).

- Can have hole with no pigeons.

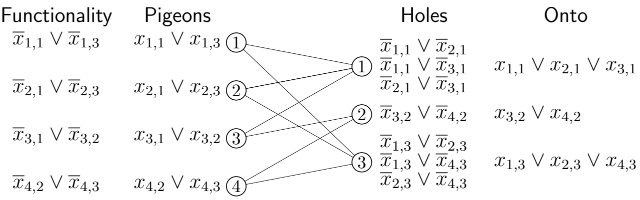
  $\Rightarrow$ Add onto axioms (makes mapping onto).

Functional PHP = PHP + Functionality

Onto-PHP = PHP + Onto

Onto-FPHP = PHP + Functionality + Onto

PHP Variants



- Can have fat pigeons which are assigned to multiple holes.

  $\Rightarrow$ Add functionality axioms (makes mapping 1-to-1).
- Can have hole with no pigeons.

  $\Rightarrow$ Add onto axioms (makes mapping onto).

Functional PHP = PHP + Functionality

Onto-PHP = PHP + Onto

Onto-FPHP = PHP + Functionality + Onto

Hardness of PHP Variants

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

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	
FPHP		
Onto-PHP		
Onto-FPHP		

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	
FPHP		
Onto-PHP		
Onto-FPHP		

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	
FPHP	hard [Haken '85]	
Onto-PHP	hard [Haken '85]	
Onto-FPHP	hard [Haken '85]	

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	
FPHP	hard [Haken '85]	
Onto-PHP	hard [Haken '85]	
Onto-FPHP	hard [Haken '85]	

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	
Onto-PHP	hard [Haken '85]	
Onto-FPHP	hard [Haken '85]	

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	
Onto-PHP	hard [Haken '85]	
Onto-FPHP	hard [Haken '85]	

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	
Onto-PHP	hard [Haken '85]	
Onto-FPHP	hard [Haken '85]	easy [Riis '93]

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	
Onto-PHP	hard [Haken '85]	
Onto-FPHP	hard [Haken '85]	easy [Riis '93]

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	?
Onto-PHP	hard [Haken '85]	?
Onto-FPHP	hard [Haken '85]	easy [Riis '93]

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	?
Onto-PHP	hard [Haken '85]	hard [AR '01]
Onto-FPHP	hard [Haken '85]	easy [Riis '93]

This work

Observe that [AR '01] proves hardness of Onto-PHP.

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	?
Onto-PHP	hard [Haken '85]	hard [AR '01]
Onto-FPHP	hard [Haken '85]	easy [Riis '93]

This work

Observe that [AR '01] proves hardness of Onto-PHP.

Hardness of PHP Variants

Variant	Resolution	Polynomial calculus
PHP	hard [Haken '85]	hard [AR '01]
FPHP	hard [Haken '85]	hard [MN '15]
Onto-PHP	hard [Haken '85]	hard [AR '01]
Onto-FPHP	hard [Haken '85]	easy [Riis '93]

This work

Observe that [AR '01] proves hardness of Onto-PHP.

Prove that FPHP is hard in polynomial calculus.

Open Problems

Open Problems

- Prove polynomial calculus lower bounds for other formulas!

  For instance, graph coloring and independent set formulas.

Open Problems

- Prove polynomial calculus lower bounds for other formulas!

  For instance, graph coloring and independent set formulas.

- Prove size lower bounds via technique that doesn't use degree!

Open Problems

- Prove polynomial calculus lower bounds for other formulas!

  For instance, graph coloring and independent set formulas.

- Prove size lower bounds via technique that doesn't use degree!

- Find truly general method capturing all PC degree lower bounds!

  We generalize part of [AR '01] that doesn't capture [BGIP '99].

Summary

Generalized method for degree lower bounds

- Unified framework for previous lower bounds.
- Hardness of Functional PHP.

Further directions

- Extend techniques to other formulas.
- Devise non-degree-based size lower bound techniques.

Summary

Generalized method for degree lower bounds

- Unified framework for previous lower bounds.
- Hardness of Functional PHP.

Further directions

- Extend techniques to other formulas.
- Devise non-degree-based size lower bound techniques.

Thank you for your attention!