Using Pseudo-Width to Prove Lower Bounds for Highly Overconstrained Formulas

Jakob Nordström

KTH Royal Institute of Technology
Stockholm, Sweden

Computational Complexity of Discrete Problems
Schloss Dagstuhl – Leibniz Center for Informatics
March 22, 2019

Joint work with Susanna F. de Rezende, Kilian Risse, and Dmitry Sokolov

Using Pseudo-Width to Prove Lower Bounds for Highly Overconstrained Formulas

Jakob Nordström

KTH Royal Institute of Technology
Stockholm, Sweden

Computational Complexity of Discrete Problems
Schloss Dagstuhl – Leibniz Center for Informatics
March 22, 2019

Joint work with Susanna F. de Rezende, Kilian Risse, and Dmitry Sokolov

Thanks for help with the slides!

Proof Complexity

- Study of efficiently verifiable certificates of unsatisfiability
- Example: Is the following CNF formula satisfiable?

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

- Study the power of different methods of reasoning (a.k.a. proof systems) in propositional logic
- This talk: resolution

Motivation for Proof Complexity

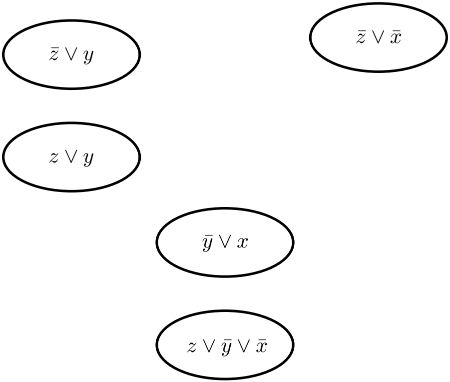
1. Separate NP and coNP
2. Understand how much reasoning power required to prove different mathematical statements
3. Analyse applied satisfiability algorithms (SAT solvers)

Just To Make Sure We're on the Same Page...

- 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
- Empty clause (with no literals) denoted $\perp$ = (contradiction)

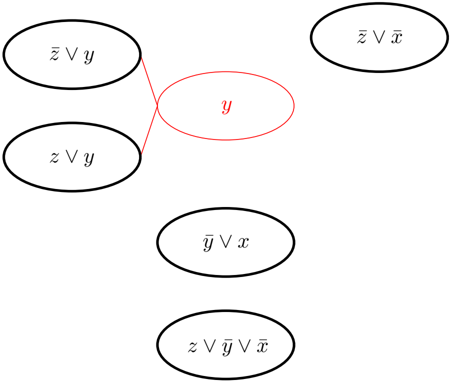
Resolution Proof System

- Derive new clauses using resolution rule: $\frac{C \lor x \quad D \lor \overline{x}}{C \lor D}$
- Certify unsatisfiability by deriving empty clause $\perp$
- Proof of unsatisfiability = refutation



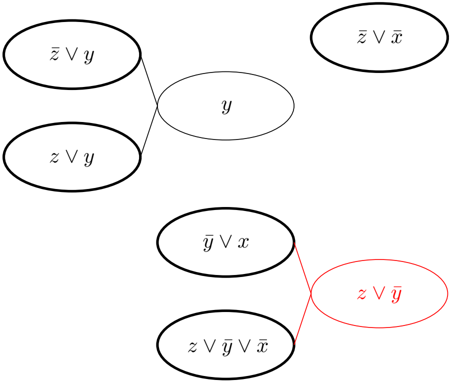
Resolution Proof System

- Derive new clauses using resolution rule: $\frac{C \vee x \quad D \vee \overline{x}}{C \vee D}$
- Certify unsatisfiability by deriving empty clause $\perp$
- Proof of unsatisfiability = refutation



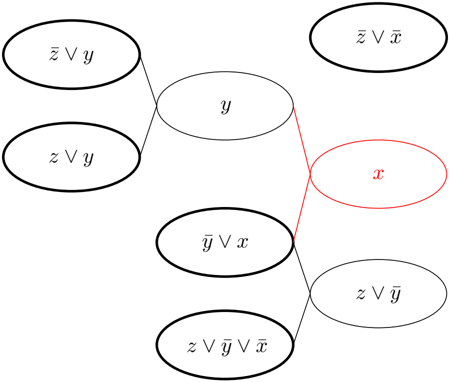
Resolution Proof System

- Derive new clauses using resolution rule: $\frac{C \vee x \quad D \vee \overline{x}}{C \vee D}$
- Certify unsatisfiability by deriving empty clause $\perp$
- Proof of unsatisfiability = refutation



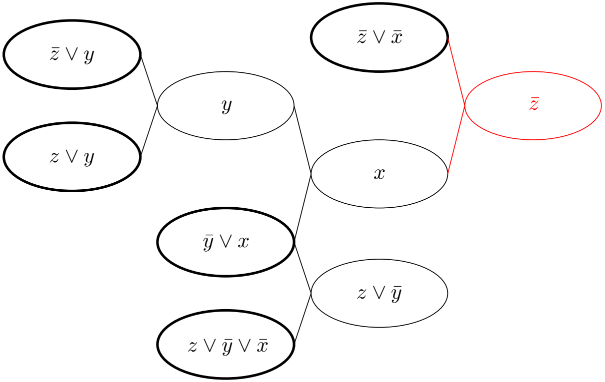
Resolution Proof System

- Derive new clauses using resolution rule: $\frac{C \vee x \quad D \vee \overline{x}}{C \vee D}$
- Certify unsatisfiability by deriving empty clause $\perp$
- Proof of unsatisfiability = refutation



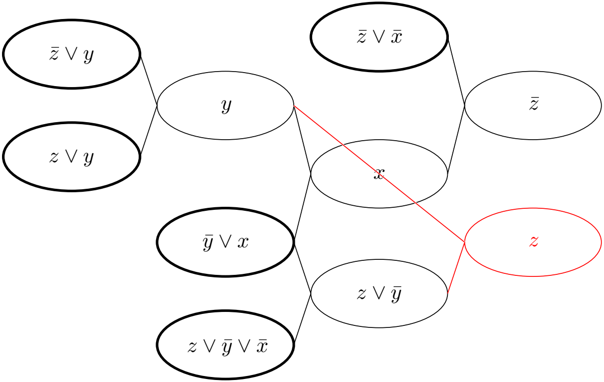
Resolution Proof System

- Derive new clauses using resolution rule: $\frac{C \vee x \quad D \vee \overline{x}}{C \vee D}$
- Certify unsatisfiability by deriving empty clause $\perp$
- Proof of unsatisfiability = refutation



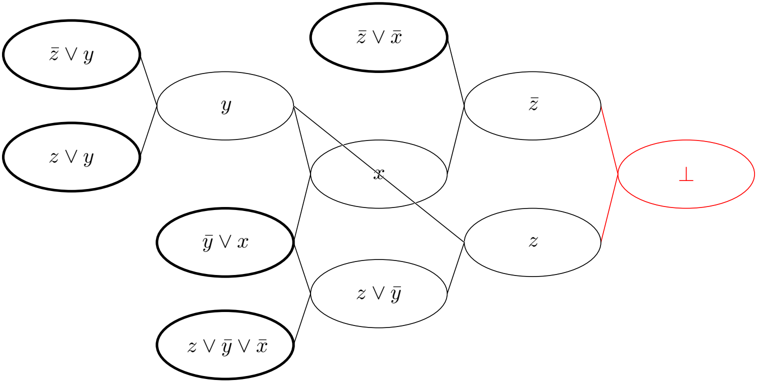
Resolution Proof System

- Derive new clauses using resolution rule: $\frac{C \vee x \quad D \vee \overline{x}}{C \vee D}$
- Certify unsatisfiability by deriving empty clause $\perp$
- Proof of unsatisfiability = refutation

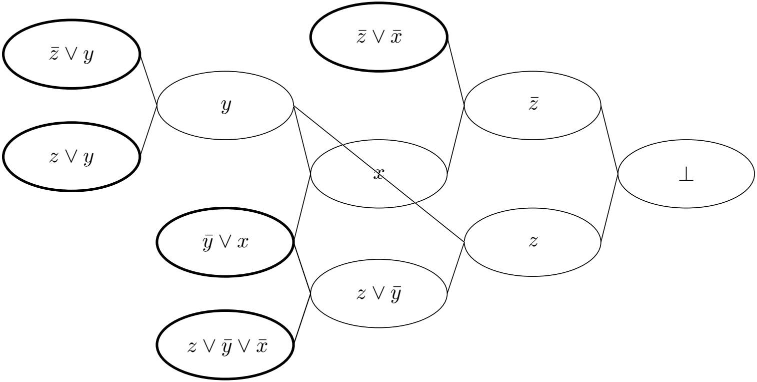


Resolution Proof System

- Derive new clauses using resolution rule: $\frac{C \lor x \quad D \lor \overline{x}}{C \lor D}$
- Certify unsatisfiability by deriving empty clause $\perp$
- Proof of unsatisfiability = refutation

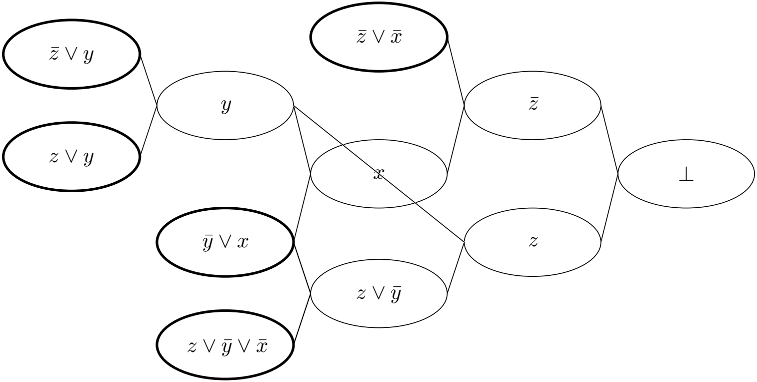


Complexity Measures for Resolution



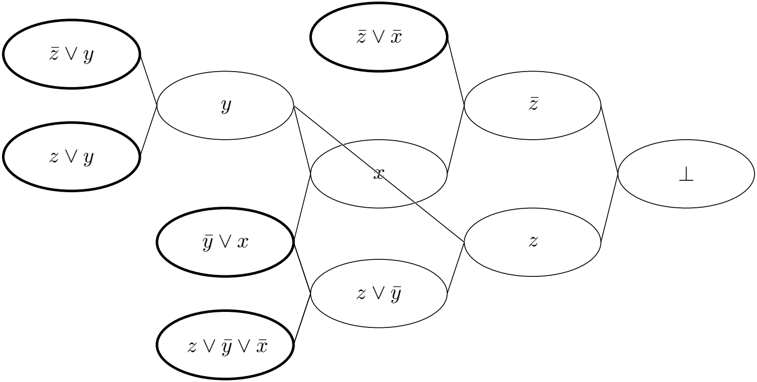
- Length of refutation = #clauses (11 in our example)

Complexity Measures for Resolution



- Length of refutation = #clauses (11 in our example)
- Width of refutation = #literals in largest clause (3 in our example)

Complexity Measures for Resolution



- Length of refutation = #clauses (11 in our example)
- Width of refutation = #literals in largest clause (3 in our example)
- Minimize over all refutations to define length $L(F \vdash \perp)$ and width $W(F \vdash \perp)$ of refuting formula F

Size-Width Lower Bound

Ben-Sasson & Wigderson [BW01]

$$L(F \vdash \bot) = \exp\left(\Omega\left(\frac{W(F \vdash \bot)^2}{\#\text{variables in } F}\right)\right)$$

Size-Width Lower Bound

Ben-Sasson & Wigderson [BW01]

$$L(F \vdash \perp) = \exp\left(\Omega\left(\frac{(W(F \vdash \perp) - W(F))^2}{\#\text{variables in } F}\right)\right)$$

Size-Width Lower Bound

Ben-Sasson & Wigderson [BW01]

$$L(F \vdash \bot) = \exp\left(\Omega\left(\frac{(W(F \vdash \bot) - W(F))^2}{\#\text{variables in } F}\right)\right)$$

- Linear lower bounds on width $\Rightarrow$ exponential lower bounds on length

Size-Width Lower Bound

Ben-Sasson & Wigderson [BW01]

$$L(F \vdash \bot) = \exp\left(\Omega\left(\frac{(W(F \vdash \bot) - W(F))^2}{\#\text{variables in } F}\right)\right)$$

- Linear lower bounds on width $\Rightarrow$ exponential lower bounds on length
- Can be used to prove almost all resolution lower bounds:
  - Pigeonhole principle formulas [Hak85]
  - Tseitin formulas [Urq87]
  - Random k-CNF formulas [CS88, BKPS02]
  - ...

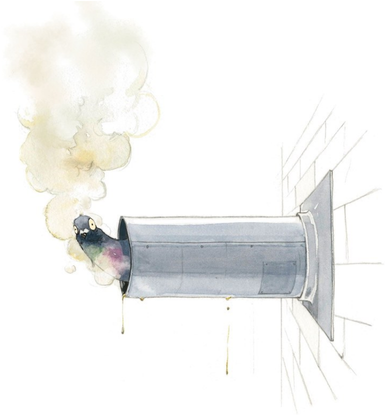
Open Problems

- So are we done with resolution? Not quite...
- Size-width lower bound yields nothing for width $\lessapprox \sqrt{\#\text{variables}}$
- This is essentially tight by [BG01]
- Interesting challenges for resolution lower bounds e.g.:
  - k-clique formulas
  - Pseudo-random generator formulas
  - Weak pigeonhole principle formulas (highly overconstrained)

This talk

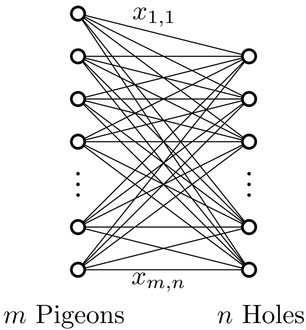
- Strong lower bounds for weak pigeonhole principle formulas
- Using and refining Razborov's pseudo-width method [Raz03, Raz04b]
- Seems like a very powerful tool that could be useful elsewhere

Pigeonhole Principle (PHP) Formulas



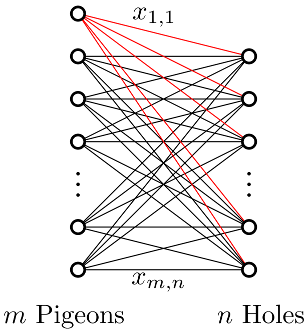
Pigeonhole Principle (PHP) Formulas

- One variable per edge:
  $x_{i,j}$ for $i \in [m]$ and $j \in [n]$
- Pigeon axioms: At least 1 hole
  $\bigvee_{j \in [n]} x_{i,j}$ (for $i \in [m]$)
- Hole axioms: At most 1 pigeon
  $\overline{x}_{i,j} \vee \overline{x}_{i',j}$ (for $i \neq i' \in [m], j \in [n]$)



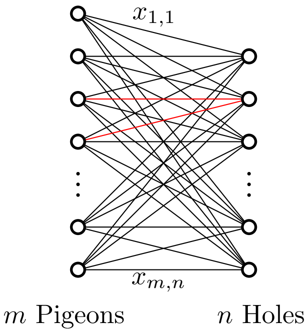
Pigeonhole Principle (PHP) Formulas

- One variable per edge:
  $x_{i,j}$ for $i \in [m]$ and $j \in [n]$
- Pigeon axioms: At least 1 hole
  $\bigvee_{j \in [n]} x_{i,j}$ (for $i \in [m]$)
- Hole axioms: At most 1 pigeon
  $\overline{x}_{i,j} \vee \overline{x}_{i',j}$ (for $i \neq i' \in [m], j \in [n]$)



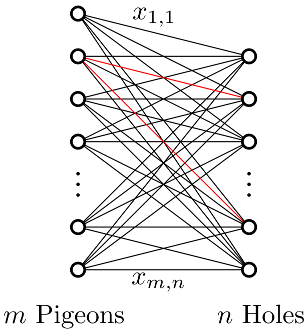
Pigeonhole Principle (PHP) Formulas

- One variable per edge:
  $x_{i,j}$ for $i \in [m]$ and $j \in [n]$
- Pigeon axioms: At least 1 hole
  $\bigvee_{j \in [n]} x_{i,j}$ (for $i \in [m]$)
- Hole axioms: At most 1 pigeon
  $\overline{x}_{i,j} \lor \overline{x}_{i',j}$ (for $i \neq i' \in [m], j \in [n]$)



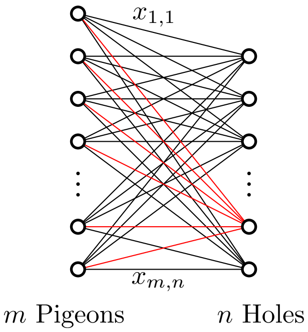
Pigeonhole Principle (PHP) Formulas

- One variable per edge: $x_{i,j}$ for $i \in [m]$ and $j \in [n]$
- Pigeon axioms: At least 1 hole $\bigvee_{j\in[n]} x_{i,j}$ (for $i \in [m]$)
- Hole axioms: At most 1 pigeon $\overline{x}_{i,j} \vee \overline{x}_{i',j}$ (for $i \neq i' \in [m], j \in [n]$)
- Functionality axioms: Only 1 hole $\overline{x}_{i,j} \vee \overline{x}_{i,j'}$ (for $i \in [m], j \neq j' \in [n]$)



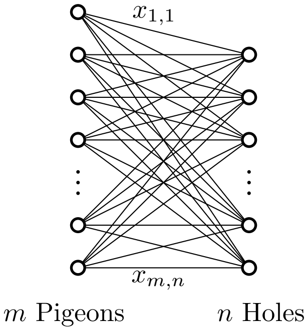
Pigeonhole Principle (PHP) Formulas

- One variable per edge:
  $x_{i,j}$ for $i \in [m]$ and $j \in [n]$
- Pigeon axioms: At least 1 hole
  $\bigvee_{j\in[n]} x_{i,j}$ (for $i \in [m]$)
- Hole axioms: At most 1 pigeon
  $\overline{x}_{i,j} \lor \overline{x}_{i',j}$ (for $i \neq i' \in [m], j \in [n]$)
- Functionality axioms: Only 1 hole
  $\overline{x}_{i,j} \lor \overline{x}_{i,j'}$ (for $i \in [m], j \neq j' \in [n]$)
- Onto axioms: At least 1 pigeon
  $\bigvee_{i\in[m]} x_{i,j}$ (for $j \in [n]$)



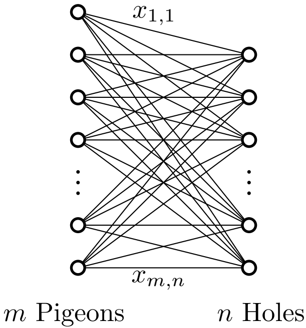
Pigeonhole Principle and Resolution: Some History

- Haken [Hak85]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{n+1} \vdash \perp) = \exp(\Omega(n))$



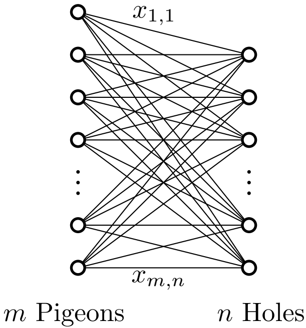
Pigeonhole Principle and Resolution: Some History

- Haken [Hak85]:
  $L(\mathit{Onto\text{-}FPHP}_n^{n+1} \vdash \perp) = \exp(\Omega(n))$
- Buss & Turán [BT88]:
  $L(\mathit{Onto\text{-}FPHP}_n^{m} \vdash \perp) = \exp\left(\Omega\left(\frac{n^2}{m}\right)\right)$



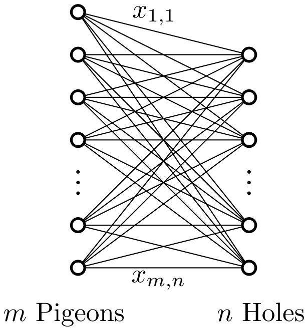
Pigeonhole Principle and Resolution: Some History

- Haken [Hak85]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{n+1} \vdash \bot) = \exp(\Omega(n))$
- Buss & Turán [BT88]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n^2}{m}\right)\right)$
- Raz [Raz04a]:
  $L(\mathit{PHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n}{\log^{10} m}\right)\right)$



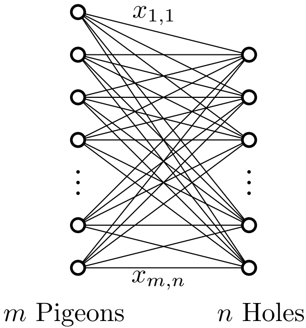
Pigeonhole Principle and Resolution: Some History

- Haken [Hak85]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{n+1} \vdash \bot) = \exp(\Omega(n))$
- Buss & Turán [BT88]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n^2}{m}\right)\right)$
- Raz [Raz04a]:
  $L(\mathit{PHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n}{\log^{10} m}\right)\right)$
- Razborov [Raz03, Raz04b]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n}{\log^{2} m}\right)\right)$



Pigeonhole Principle and Resolution: Some History

- Haken [Hak85]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{n+1} \vdash \bot) = \exp(\Omega(n))$
- Buss & Turán [BT88]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n^2}{m}\right)\right)$
- Raz [Raz04a]:
  $L(\mathit{PHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n}{\log^{10} m}\right)\right)$
- Razborov [Raz03, Raz04b]:
  $L(\mathit{Onto}\text{-}\mathit{FPHP}_n^{m} \vdash \bot) = \exp\left(\Omega\left(\frac{n}{\log^{2} m}\right)\right)$

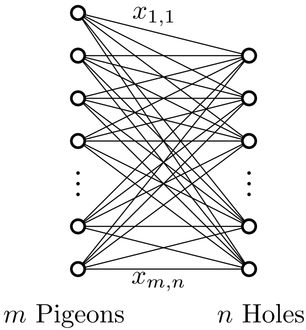


(Much more info in Razborov's survey on PHP in proof complexity [Raz02])

Pigeonhole Principle on Graphs

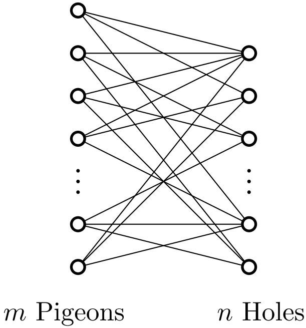
Pigeonhole Principle on Graphs

- Replace complete graph by "good" sparse graph, restricting pigeon choices



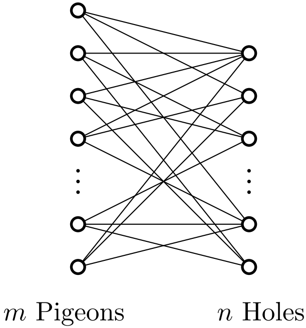
Pigeonhole Principle on Graphs

- Replace complete graph by “good” sparse graph, restricting pigeon choices



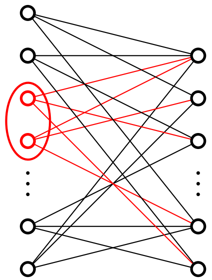
Pigeonhole Principle on Graphs

- Replace complete graph by "good" sparse graph, restricting pigeon choices
- Intuitively, graph is "good" if any small set of pigeons has many partial matchings



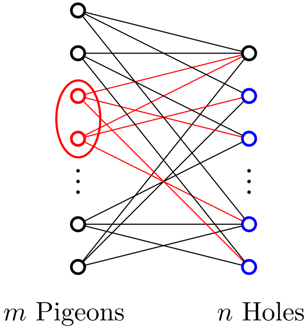
Pigeonhole Principle on Graphs

- Replace complete graph by "good" sparse graph, restricting pigeon choices
- Intuitively, graph is "good" if any small set of pigeons has many partial matchings



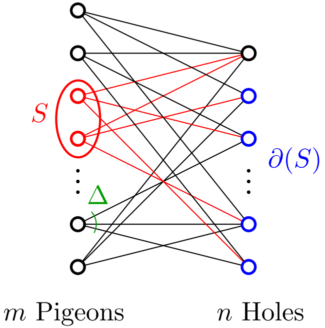
Pigeonhole Principle on Graphs

- Replace complete graph by "good" sparse graph, restricting pigeon choices
- Intuitively, graph is "good" if any small set of pigeons has many partial matchings



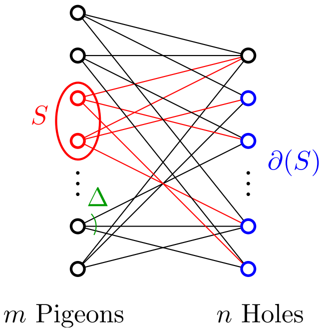
Pigeonhole Principle on Graphs

- Replace complete graph by "good" sparse graph, restricting pigeon choices
- Intuitively, graph is "good" if any small set of pigeons has many partial matchings
- (r, Δ, c)-boundary expander:
  1. every pigeon has degree $\leq \Delta$
  2. all sets $S \subseteq [m]$ of size $\leq r$ have $\geq c \cdot |S|$ unique neighbours



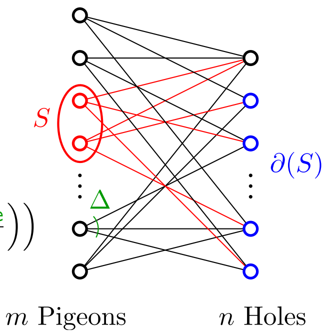
Lower Bounds for Graph PHP Formulas on Expanders

- Ben-Sasson & Wigderson [BW01]:
  For $r = \Omega(n/\log m)$, $\Delta = \log m$ and $c = \frac{3}{4}\log m$:
  $L(FPHP(G) \vdash \perp) = \exp\left(\Omega\left(\frac{n^2}{m \log m}\right)\right)$



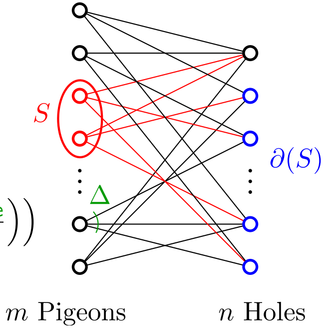
Lower Bounds for Graph PHP Formulas on Expanders

- Ben-Sasson & Wigderson [BW01]:
  For $r = \Omega(n/\log m)$, $\Delta = \log m$ and $c = \frac{3}{4}\log m$:
  $L(FPHP(G) \vdash \perp) = \exp\left(\Omega\left(\frac{n^2}{m \log m}\right)\right)$
- Razborov [Raz03, Raz04b]:
  $L(\mathit{Onto}\text{-}FPHP(G) \vdash \perp) = \exp\left(\Omega\left(\frac{\text{min degree}}{\log^2 m}\right)\right)$



Lower Bounds for Graph PHP Formulas on Expanders

- Ben-Sasson & Wigderson [BW01]:
  For $r = \Omega(n/\log m)$, $\Delta = \log m$ and $c = \frac{3}{4}\log m$:
  $L(FPHP(G) \vdash \perp) = \exp\left(\Omega\left(\frac{n^2}{m \log m}\right)\right)$
- Razborov [Raz03, Raz04b]:
  $L(\mathit{Onto}\text{-}FPHP(G) \vdash \perp) = \exp\left(\Omega\left(\frac{\text{min degree}}{\log^2 m}\right)\right)$
- What about $m \gg n^2$ and $\Delta \approx \log m$?



Our Weak Graph PHP Lower Bounds

- For $m \leq n^{\mathrm{o}(\log n)}$, $\Delta = \log m$ and G sampled from $\mathcal{G}(m, n, \Delta)$:

$$L(FPHP(G) \vdash \perp) \geq \exp\left(n^{1-o(1)}\right)$$

Our Weak Graph PHP Lower Bounds

- For $m \leq n^{\mathrm{o}(\log n)}$, $\Delta = \log m$ and G sampled from $\mathcal{G}(m, n, \Delta)$:

$$L\big(FPHP(G) \vdash \bot\big) \geq \exp\left(n^{1-o(1)}\right)$$

- For $m = n^k$, $\Delta = 32\left(\frac{k}{\varepsilon}\right)^2$ and G sampled from $\mathcal{G}(m, n, \Delta)$:

$$L\big(FPHP(G) \vdash \bot\big) \geq \exp\big(n^{1-\varepsilon}\big)$$

Our Weak Graph PHP Lower Bounds

- For $m \leq n^{\mathrm{o}(\log n)}$, $\Delta = \log m$ and G sampled from $\mathcal{G}(m, n, \Delta)$:

$$L(FPHP(G) \vdash \bot) \geq \exp\left(n^{1-o(1)}\right)$$

- For $m = n^k$, $\Delta = 32\left(\frac{k}{\varepsilon}\right)^2$ and G sampled from $\mathcal{G}(m, n, \Delta)$:

$$L(FPHP(G) \vdash \bot) \geq \exp(n^{1-\varepsilon})$$

- For $m < \exp\left(n^{1/16}\right)$ and $\Delta = \mathcal{O}(\mathrm{polylog}(m))$, $\exists$ graphs G such that:

$$L(FPHP(G) \vdash \bot) \geq \exp\left(n^{1/5}\right)$$

(using expander construction in [GUV09])

A General Theorem

Theorem

Let G be an $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then

$$L(FPHP(G) \vdash \bot) = \exp\left(\Omega\left(\frac{r}{n^{\varepsilon} \log^2 m}\right)\right)$$

A General Theorem

Theorem

Let G be an $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then

$$L(FPHP(G) \vdash \bot) = \exp\left(\Omega\left(\frac{r}{n^{\varepsilon} \log^2 m}\right)\right)$$

Technical note:

- Need expansion $\lim_{n\to\infty} c = \Delta$
- Would be great to show that $c = (1 - \varepsilon)\Delta$ is enough
- Probably room for improvement also in other parameters

Very High-Level Proof Outline

- Define pseudo-width measure on clauses $\approx$ interesting pigeons

Very High-Level Proof Outline

- Define pseudo-width measure on clauses $\approx$ interesting pigeons
- Short refutations can be transformed into low-width refutations

Very High-Level Proof Outline

- Define pseudo-width measure on clauses $\approx$ interesting pigeons
- Short refutations can be transformed into low-width refutations
- But any refutation of $FPHP(G)$ requires large pseudo-width

Very High-Level Proof Outline

- Define pseudo-width measure on clauses $\approx$ interesting pigeons
- Short refutations can be transformed into low-width refutations
- But any refutation of $FPHP(G)$ requires large pseudo-width
- Hence, no short refutations can exist $\blacksquare$

Pseudo-Width

Each clause has 3 different kinds of pigeons:

Pseudo-Width

Each clause has 3 different kinds of pigeons:

obese

Pseudo-Width

Each clause has 3 different kinds of pigeons:

obese

stout

Pseudo-Width

Each clause has 3 different kinds of pigeons:

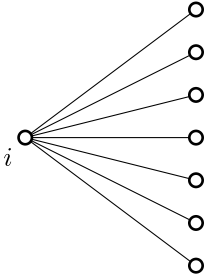
obese

stout

slim

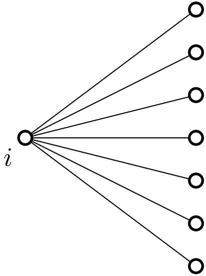
Measuring the Strength of Clauses

- How strong is a clause C?



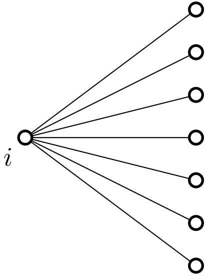
Measuring the Strength of Clauses

- How strong is a clause C?
- Depends on how many pigeon-to-hole matchings C rules out



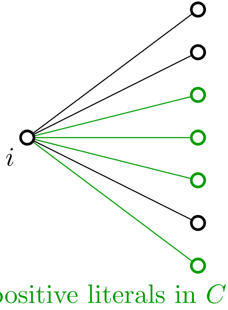
Measuring the Strength of Clauses

- How strong is a clause C?
- Depends on how many pigeon-to-hole matchings C rules out
- I.e., how many matchings falsify C



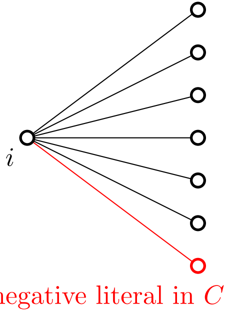
Measuring the Strength of Clauses

- How strong is a clause C?
- Depends on how many pigeon-to-hole matchings C rules out
- I.e., how many matchings falsify C
- Positive literals — don't match edge



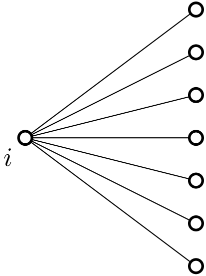
Measuring the Strength of Clauses

- How strong is a clause C?
- Depends on how many pigeon-to-hole matchings C rules out
- I.e., how many matchings falsify C
- Positive literals — don't match edge
- Negative literal — have to match edge



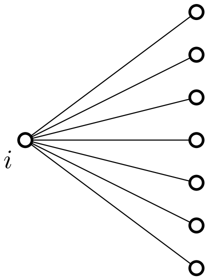
Measuring the Strength of Clauses

- How strong is a clause C?
- Depends on how many pigeon-to-hole matchings C rules out
- I.e., how many matchings falsify C
- Positive literals — don't match edge
- Negative literal — have to match edge
- If several negative literals in C, **no** matching ruled out



Measuring the Strength of Clauses

- How strong is a clause C?
- Depends on how many pigeon-to-hole matchings C rules out
- I.e., how many matchings falsify C
- Positive literals — don't match edge
- Negative literal — have to match edge
- If several negative literals in C, **no** matching ruled out



Key take-away

For each pigeon, consider #matchings that C rules out

Obese Pigeons

- (Pseudo-)width: measure of **weakness** of clauses

Obese Pigeons

- (Pseudo-)width: measure of **weakness** of clauses
- More matchings satisfy $C \Rightarrow$ weaker clause

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$

Obese Pigeons

- (Pseudo-)width: measure of **weakness** of clauses
- More matchings satisfy $C \Rightarrow$ weaker clause

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$

- Choose (somehow) filter vector $\vec{d} = (d_1, \ldots, d_m)$, $d_i < \Delta$

Obese Pigeons

- (Pseudo-)width: measure of **weakness** of clauses
- More matchings satisfy $C \Rightarrow$ weaker clause

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$

- Choose (somehow) filter vector $\vec{d} = (d_1, \ldots, d_m)$, $d_i < \Delta$
- If $d_i(C) \geq d_i$, then pigeon i is obese in clause C

Obese Pigeons

- (Pseudo-)width: measure of **weakness** of clauses
- More matchings satisfy $C \Rightarrow$ weaker clause

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$

- Choose (somehow) filter vector $\vec{d} = (d_1, \ldots, d_m)$, $d_i < \Delta$
- If $d_i(C) \geq d_i$, then pigeon i is obese in clause C
- $P_{\mathrm{obese}}(C) = \{i \in [m] \mid d_i(C) \geq d_i\}$

Stout Pigeons and Pseudo-Width

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$
$$\vec{d} = (d_1, \ldots, d_m)$$

Stout Pigeons and Pseudo-Width

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$
$$\vec{d} = (d_1, \ldots, d_m)$$

- Pigeons δ-close to being obese are also somewhat fat...

Stout Pigeons and Pseudo-Width

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$
$$\vec{d} = (d_1, \ldots, d_m)$$

- Pigeons δ-close to being obese are also somewhat fat...
- If $d_i(C) \geq d_i - \delta$ for $\delta \lessapprox \Delta / \log m$, then pigeon i is stout in C

Stout Pigeons and Pseudo-Width

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$
$$\vec{d} = (d_1, \ldots, d_m)$$

- Pigeons δ-close to being obese are also somewhat fat...
- If $d_i(C) \geq d_i - \delta$ for $\delta \lessapprox \Delta / \log m$, then pigeon i is stout in C
- $P_{\text{stout}}(C) = \{i \in [m] \mid d_i(C) \geq d_i - \delta\}$

Stout Pigeons and Pseudo-Width

$$d_i(C) = \#\text{matchings of pigeon } i \text{ that satisfy } C$$
$$\vec{d} = (d_1, \ldots, d_m)$$

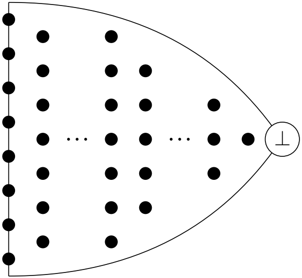
- Pigeons δ-close to being obese are also somewhat fat...
- If $d_i(C) \geq d_i - \delta$ for $\delta \lessapprox \Delta / \log m$, then pigeon i is stout in C
- $P_{\mathrm{stout}}(C) = \{i \in [m] \mid d_i(C) \geq d_i - \delta\}$
- Pseudo-width of clause C is

$$W^*(C) = \left|P_{\mathrm{stout}}(C)\right|$$

i.e., #stout pigeons in C

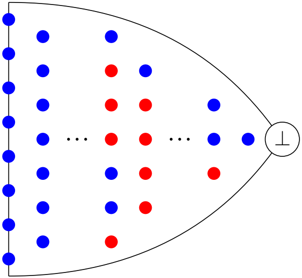
Refined Proof Outline

1. Given refutation, classify clauses as having high or low pseudo-width



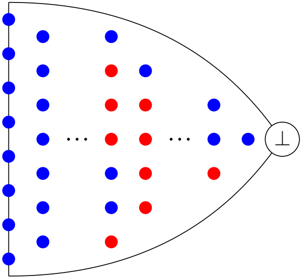
Refined Proof Outline

1. Given refutation, classify clauses as having high or low pseudo-width



Refined Proof Outline

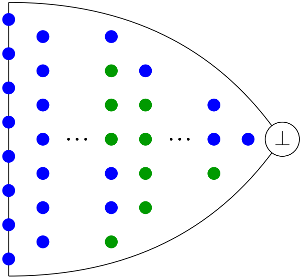
1. Given refutation, classify clauses as having high or low pseudo-width



2. Substitute high-pseudo-width clauses by lower-width fake axioms $\mathcal{A}$

Refined Proof Outline

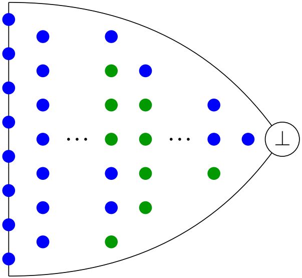
1. Given refutation, classify clauses as having high or low pseudo-width



2. Substitute high-pseudo-width clauses by lower-width fake axioms $\mathcal{A}$

Refined Proof Outline

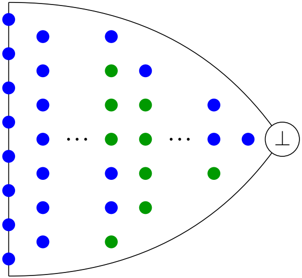
1. Given refutation, classify clauses as having high or low pseudo-width



2. Substitute high-pseudo-width clauses by lower-width fake axioms $\mathcal{A}$
3. By construction
   - $|\mathcal{A}| \leq$ length L of original refutation
   - $\exists$ low-pseudo-width refutation of $FPHP(G) \cup \mathcal{A}$

Refined Proof Outline

1. Given refutation, classify clauses as having high or low pseudo-width



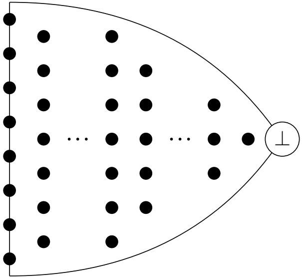
2. Substitute high-pseudo-width clauses by lower-width fake axioms $\mathcal{A}$
3. By construction
   - $|\mathcal{A}| \leq$ length L of original refutation
   - $\exists$ low-pseudo-width refutation of $FPHP(G) \cup \mathcal{A}$
4. Show that since $\mathcal{A}$ not too large, $FPHP(G) \cup \mathcal{A}$ must still require large pseudo-width ↯ ■

Filter Lemma

Lemma (Razborov [Raz03] (with a small twist))

If $\delta \leq \varepsilon \frac{\Delta \log n}{\log m}$ *and length* $L < 2^{w_0}$*, then* $\exists\ \vec{d} = (d_1, \ldots, d_m)$ *such that* $\forall$ *clauses* C *in refutation one of two cases applies:*

1. $|P_{\text{obese}}(C)| \geq w_0$
2. $|P_{\text{stout}}(C)| = W^*(C) \leq \mathcal{O}(w_0 \cdot n^{\varepsilon})$

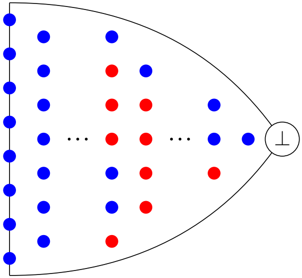


Filter Lemma

Lemma (Razborov [Raz03] (with a small twist))

If $\delta \leq \varepsilon \frac{\Delta \log n}{\log m}$ *and length* $L < 2^{w_0}$*, then* $\exists\ \vec{d} = (d_1, \ldots, d_m)$ *such that* $\forall$ *clauses* C *in refutation one of two cases applies:*

1. $|P_{\text{obese}}(C)| \geq w_0$
2. $|P_{\text{stout}}(C)| = W^*(C) \leq \mathcal{O}(w_0 \cdot n^{\varepsilon})$



Proof of Pseudo-Width Upper Bound

Fake axiom

A fake axiom is any clause with exactly w_0 obese pigeons

Fake intuition: such clauses so weak we can almost give them "for free"

Proof of Pseudo-Width Upper Bound

Fake axiom

A fake axiom is any clause with exactly w_0 obese pigeons

Fake intuition: such clauses so weak we can almost give them "for free"

Corollary (of Filter Lemma)

If $FPHP(G)$ can be refuted in length $L < 2^{w_0}$, then exists

- *filter vector $\vec{d}$*
- *fake axiom set $\mathcal{A}$ with $|\mathcal{A}| \leq L$*

such that $FPHP(G) \cup \mathcal{A}$ can be refuted in pseudo-width $\mathcal{O}(w_0 \cdot n^{\varepsilon})$

Proof of Pseudo-Width Upper Bound

Fake axiom

A fake axiom is any clause with exactly w_0 obese pigeons

Fake intuition: such clauses so weak we can almost give them "for free"

Corollary (of Filter Lemma)

If $FPHP(G)$ can be refuted in length $L < 2^{w_0}$, then exists

- *filter vector $\vec{d}$*
- *fake axiom set $\mathcal{A}$ with $|\mathcal{A}| \leq L$*

such that $FPHP(G) \cup \mathcal{A}$ can be refuted in pseudo-width $\mathcal{O}(w_0 \cdot n^{\varepsilon})$

Proof:

1. Replace type-1 clauses with many obese pigeons by (stronger) fake axioms
2. Now all clauses have low width (type-2 clauses were already OK) — done!

Pseudo-Width Lower Bound: Statement and Intuition

Lemma

Suppose G is $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then refuting $FPHP(G) \cup \mathcal{A}$ requires pseudo-width $\Omega(r \cdot \log n / \log m)$

Pseudo-Width Lower Bound: Statement and Intuition

Lemma

Suppose G is $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then refuting $FPHP(G) \cup \mathcal{A}$ requires pseudo-width $\Omega(r \cdot \log n / \log m)$

Fake proof:

- Measure progress made up to C as fraction of matchings ruled out for $P_{\mathrm{stout}}(C)$

Pseudo-Width Lower Bound: Statement and Intuition

Lemma

Suppose G is $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then refuting $FPHP(G) \cup \mathcal{A}$ requires pseudo-width $\Omega(r \cdot \log n / \log m)$

Fake proof:

- Measure progress made up to C as fraction of matchings ruled out for $P_{\mathrm{stout}}(C)$
- True (original) axioms rule out no matchings

Pseudo-Width Lower Bound: Statement and Intuition

Lemma

Suppose G is $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then refuting $FPHP(G) \cup \mathcal{A}$ requires pseudo-width $\Omega(r \cdot \log n / \log m)$

Fake proof:

- Measure progress made up to C as fraction of matchings ruled out for $P_{\mathrm{stout}}(C)$
- True (original) axioms rule out no matchings
- Fake axioms rule out exponentially small fraction of matchings (hard to match obese pigeons while avoiding to satisfy clause)

Pseudo-Width Lower Bound: Statement and Intuition

Lemma

Suppose G is $(r, \Delta, (1 - \varepsilon \log n/\log m)\Delta)$-boundary expander. Then refuting $FPHP(G) \cup \mathcal{A}$ requires pseudo-width $\Omega(r \cdot \log n/\log m)$

Fake proof:

- Measure progress made up to C as fraction of matchings ruled out for $P_{\mathrm{stout}}(C)$
- True (original) axioms rule out no matchings
- Fake axioms rule out exponentially small fraction of matchings (hard to match obese pigeons while avoiding to satisfy clause)
- Contradiction $\perp$ rules out 100% of partial matchings! (Since $P_{\mathrm{stout}}(\perp) = \emptyset$)

Pseudo-Width Lower Bound: Statement and Intuition

Lemma

Suppose G is $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then refuting $FPHP(G) \cup \mathcal{A}$ requires pseudo-width $\Omega(r \cdot \log n / \log m)$

Fake proof:

- Measure progress made up to C as fraction of matchings ruled out for $P_{\mathrm{stout}}(C)$
- True (original) axioms rule out no matchings
- Fake axioms rule out exponentially small fraction of matchings (hard to match obese pigeons while avoiding to satisfy clause)
- Contradiction $\perp$ rules out 100% of partial matchings! (Since $P_{\mathrm{stout}}(\perp) = \emptyset$)
- Key technical lemma: For small-pseudo-width resolution steps

$$\frac{C \vee x_{i,j} \qquad D \vee \overline{x}_{i,j}}{C \vee D}$$

$C \vee D$ rules out at most same fraction of matchings as $C \vee x_{i,j}$ plus $D \vee \overline{x}_{i,j}$

Pseudo-Width Lower Bound: Statement and Intuition

Lemma

Suppose G is $(r, \Delta, (1 - \varepsilon \log n / \log m)\Delta)$-boundary expander. Then refuting $FPHP(G) \cup \mathcal{A}$ requires pseudo-width $\Omega(r \cdot \log n / \log m)$

Fake proof:

- Measure progress made up to C as fraction of matchings ruled out for $P_{\text{stout}}(C)$
- True (original) axioms rule out no matchings
- Fake axioms rule out exponentially small fraction of matchings (hard to match obese pigeons while avoiding to satisfy clause)
- Contradiction $\perp$ rules out 100% of partial matchings! (Since $P_{\text{stout}}(\perp) = \emptyset$)
- Key technical lemma: For small-pseudo-width resolution steps

$$\frac{C \vee x_{i,j} \qquad D \vee \overline{x}_{i,j}}{C \vee D}$$

  $C \vee D$ rules out at most same fraction of matchings as $C \vee x_{i,j}$ plus $D \vee \overline{x}_{i,j}$
- $\Rightarrow$ Too few fake axioms to add up to 100% $\lightning$

More About the Actual Pseudo-Width Lower Bound

A couple of issues:

1. Not true that $C \vee D$ rules out same fraction as $C \vee x_{i,j}$ plus $D \vee \overline{x}_{i,j}$ — pigeon i can cease to be stout in $C \vee D$

More About the Actual Pseudo-Width Lower Bound

A couple of issues:

1. Not true that $C \vee D$ rules out same fraction as $C \vee x_{i,j}$ plus $D \vee \overline{x}_{i,j}$ — pigeon i can cease to be stout in $C \vee D$
2. Also, assignments to a few other stout pigeons $i'_1, \ldots, i'_\Delta$ might occupy all Δ holes available for pigeon i needing to be matched

More About the Actual Pseudo-Width Lower Bound

A couple of issues:

1. Not true that $C \vee D$ rules out same fraction as $C \vee x_{i,j}$ plus $D \vee \overline{x}_{i,j}$ — pigeon i can cease to be stout in $C \vee D$
2. Also, assignments to a few other stout pigeons $i'_1, \ldots, i'_\Delta$ might occupy all Δ holes available for pigeon i needing to be matched

Solutions:

1. Need "lossy counting"
    - Associate matchings with linear subspaces of suitable space
    - Consider span of all matchings ruled out
    - When "enough" matchings for pigeon i, can stop counting

More About the Actual Pseudo-Width Lower Bound

A couple of issues:

1. Not true that $C \vee D$ rules out same fraction as $C \vee x_{i,j}$ plus $D \vee \overline{x}_{i,j}$ — pigeon i can cease to be stout in $C \vee D$
2. Also, assignments to a few other stout pigeons $i'_1, \ldots, i'_\Delta$ might occupy all Δ holes available for pigeon i needing to be matched

Solutions:

1. Need "lossy counting"
   - Associate matchings with linear subspaces of suitable space
   - Consider span of all matchings ruled out
   - When "enough" matchings for pigeon i, can stop counting
2. Consider $P_{\mathrm{crit}}(C) \supseteq P_{\mathrm{stout}}(C)$ so that residual graph $G \setminus (P_{\mathrm{crit}}(C) \times N(P_{\mathrm{crit}}(C)))$ is expander

More About the Actual Pseudo-Width Lower Bound

A couple of issues:

1. Not true that $C \vee D$ rules out same fraction as $C \vee x_{i,j}$ plus $D \vee \overline{x}_{i,j}$ — pigeon i can cease to be stout in $C \vee D$
2. Also, assignments to a few other stout pigeons $i'_1, \ldots, i'_\Delta$ might occupy all Δ holes available for pigeon i needing to be matched

Solutions:

1. Need "lossy counting"
   - Associate matchings with linear subspaces of suitable space
   - Consider span of all matchings ruled out
   - When "enough" matchings for pigeon i, can stop counting
2. Consider $P_{\text{crit}}(C) \supseteq P_{\text{stout}}(C)$ so that residual graph $G \setminus (P_{\text{crit}}(C) \times N(P_{\text{crit}}(C)))$ is expander
3. Do proof on previous slide, but with linear algebra ☺

Technical Details in Their Full Glory

- Fix linear spaces L_i for $i \in [n]$ of dimension $\ell_i \approx \Delta - d_i + \delta/4$

Technical Details in Their Full Glory

- Fix linear spaces L_i for $i \in [n]$ of dimension $\ell_i \approx \Delta - d_i + \delta/4$
- Associate assignment $i \mapsto j$ with vector $\vec{v}_{i,j} \in L_i$ so that

$$|J| \geq \ell_i \;\Rightarrow\; \operatorname{span}(\{\vec{v}_{i,j} \mid j \in J\}) = L_i$$

Technical Details in Their Full Glory

- Fix linear spaces L_i for $i \in [n]$ of dimension $\ell_i \approx \Delta - d_i + \delta/4$
- Associate assignment $i \mapsto j$ with vector $\vec{v}_{i,j} \in L_i$ so that

$$|J| \geq \ell_i \;\Rightarrow\; \mathrm{span}(\{\vec{v}_{i,j} \mid j \in J\}) = L_i$$

- Associate partial matching φ with subspace

$$L(\varphi) = \bigotimes_{i \in \mathrm{dom}(\varphi)} \vec{v}_{i,\varphi(i)} \otimes \bigotimes_{i \notin \mathrm{dom}(\varphi)} L_i$$

Technical Details in Their Full Glory

- Fix linear spaces L_i for $i \in [n]$ of dimension $\ell_i \approx \Delta - d_i + \delta/4$
- Associate assignment $i \mapsto j$ with vector $\vec{v}_{i,j} \in L_i$ so that

$$|J| \geq \ell_i \;\Rightarrow\; \mathrm{span}(\{\vec{v}_{i,j} \mid j \in J\}) = L_i$$

- Associate partial matching φ with subspace

$$L(\varphi) = \bigotimes_{i \in \mathrm{dom}(\varphi)} \vec{v}_{i,\varphi(i)} \otimes \bigotimes_{i \notin \mathrm{dom}(\varphi)} L_i$$

- Strength of clause C measured by

$$Z(C) = \mathrm{span}(\{L(\varphi) \mid \mathrm{dom}(\varphi) = P_{\mathrm{crit}}(C);\ \varphi \text{ doesn't satisfy } C\})$$

Technical Details in Their Full Glory

- Fix linear spaces L_i for $i \in [n]$ of dimension $\ell_i \approx \Delta - d_i + \delta/4$
- Associate assignment $i \mapsto j$ with vector $\vec{v}_{i,j} \in L_i$ so that

$$|J| \geq \ell_i \;\Rightarrow\; \mathrm{span}\big(\{\vec{v}_{i,j} \mid j \in J\}\big) = L_i$$

- Associate partial matching φ with subspace

$$L(\varphi) = \bigotimes_{i \in \mathrm{dom}(\varphi)} \vec{v}_{i,\varphi(i)} \otimes \bigotimes_{i \notin \mathrm{dom}(\varphi)} L_i$$

- Strength of clause C measured by

$$Z(C) = \mathrm{span}\big(\{L(\varphi) \mid \mathrm{dom}(\varphi) = P_{\mathrm{crit}}(C);\ \varphi \text{ doesn't satisfy } C\}\big)$$

Main Technical Lemma

For derivations in low pseudo-width it holds that

$$Z(C \vee D) \subseteq \mathrm{span}\big(Z(C \vee x), Z(D \vee \overline{x})\big)$$

Take-Home Message

- Resolution very well-studied; large toolbox developed
- But many challenging problems remain beyond current techniques
- Razborov's pseudo-width method seems like a powerful tool that might also work for, e.g.,
  - k-clique formulas
  - Pseudo-random generator formulas
- Would be great to extend to other proof systems
  - resolution over parities
  - polynomial calculus

Take-Home Message

- Resolution very well-studied; large toolbox developed
- But many challenging problems remain beyond current techniques
- Razborov's pseudo-width method seems like a powerful tool that might also work for, e.g.,
  - k-clique formulas
  - Pseudo-random generator formulas
- Would be great to extend to other proof systems
  - resolution over parities
  - polynomial calculus

Thanks! Questions?

References I

[BG01] María Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, December 2001. Preliminary version in *FOCS '99*.

[BKPS02] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. The efficiency of resolution and Davis-Putnam procedures. *SIAM Journal on Computing*, 31(4):1048–1075, 2002. Preliminary versions of these results appeared in *FOCS '96* and *STOC '98*.

[BT88] Samuel R. Buss and György Turán. Resolution proofs of generalized pigeonhole principles. *Theoretical Computer Science*, 62(3):311–317, December 1988.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

[CS88] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM*, 56(4):20:1–20:34, July 2009. Preliminary version in *CCC '07*.

References II

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[Raz02] Alexander A. Razborov. Proof complexity of pigeonhole principles. In *5th International Conference on Developments in Language Theory, (DLT '01), Revised Papers*, volume 2295 of *Lecture Notes in Computer Science*, pages 100–116. Springer, July 2002.

[Raz03] Alexander A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theoretical Computer Science*, 1(303):233–243, June 2003.

[Raz04a] Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *Journal of the ACM*, 51(2):115–138, March 2004. Preliminary version in *STOC '02*.

[Raz04b] Alexander A. Razborov. Resolution lower bounds for perfect matching principles. *Journal of Computer and System Sciences*, 69(1):3–27, August 2004. Preliminary version in *CCC '02*.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.