A (Biased) Survey of Space Complexity and Time-Space Trade-offs in Proof Complexity

Jakob Nordström

KTH Royal Institute of Technology
Stockholm, Sweden

FLoC Workshop on Proof Complexity
Federated Logic Conference
Vienna, Austria
July 12–13, 2014

Topic of This Survey

Study of space in proof complexity initiated in late 1990s
Motivated by considerations of SAT solver memory usage
But also (and mainly?) intrinsically interesting for proof complexity

Topic of This Survey

Study of space in proof complexity initiated in late 1990s
Motivated by considerations of SAT solver memory usage
But also (and mainly?) intrinsically interesting for proof complexity

This talk intended to give overview of

- space complexity
- size-space trade-offs (a.k.a. time-space trade-offs)

Topic of This Survey

Study of space in proof complexity initiated in late 1990s
Motivated by considerations of SAT solver memory usage
But also (and mainly?) intrinsically interesting for proof complexity

This talk intended to give overview of

- space complexity
- size-space trade-offs (a.k.a. time-space trade-offs)

Make most sense for relatively weak proof systems — focus on:

- resolution
- polynomial calculus
- cutting planes (only mention very briefly)

By necessity, selective coverage — apologies for omissions

Outline

1. Space Complexity
   - Preliminaries
   - Space Lower Bounds for Resolution
   - Space Lower Bounds for Polynomial Calculus
2. Size-Space Trade-offs
   - Trade-offs for Resolution
   - Trade-offs for Polynomial Calculus
   - Trade-offs for Superlinear Space
3. Open Problems
   - Open Problems for Resolution
   - Open Problems for Polynomial Calculus
   - Open Problems for Cutting Planes

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 $\leq k$
  (where k is some constant)
- Mostly assume formulas k-CNFs (for simplicity of exposition)
  Conversion to 3-CNF most often doesn't change much
  [except sometimes the difference is huge. . .]
- N denotes size of formula ($\#$ literals, which is $\approx \#$ clauses)

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
- DAG

Tree-like resolution if DAG is tree

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

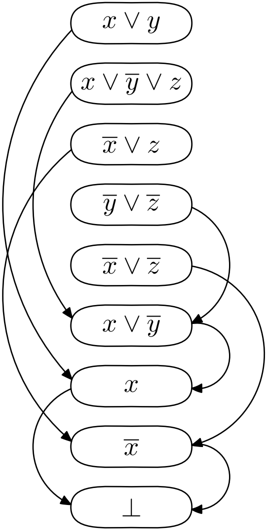
$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$$

Refutation ends when empty clause $\perp$ derived

Can represent refutation as

- annotated list or
- DAG

Tree-like resolution if DAG is tree



The Resolution Proof System

Goal: refute unsatisfiable CNF

Start with clauses of formula (axioms)

Derive new clauses by resolution rule

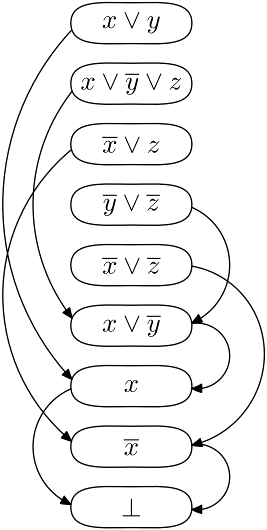
$$\frac{C \lor x \qquad D \lor \overline{x}}{C \lor D}$$

Refutation ends when empty clause $\perp$ derived

Can represent refutation as

- annotated list or
- DAG

Tree-like resolution if DAG is tree



Resolution Size and Space

Size/length = total # clauses in refutation

Space = max # clauses in memory when performing refutation

(Exist other space measures also — focus here on most well-studied one)

Space at step t: # clauses at steps $\leq t$ used at steps $\geq t$

Example: Space at step 7 . . .

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$(7, 8)$

Resolution Size and Space

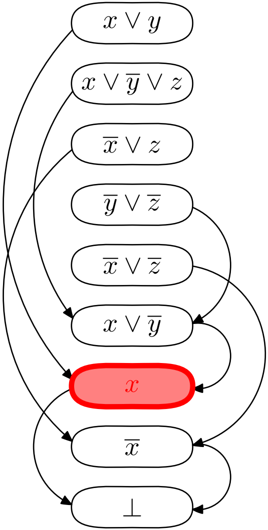
Size/length = total # clauses in refutation

Space = max # clauses in memory when performing refutation

(Exist other space measures also — focus here on most well-studied one)

Space at step t: # clauses at steps $\leq t$ used at steps $\geq t$

Example: Space at step $7 \ldots$



Resolution Size and Space

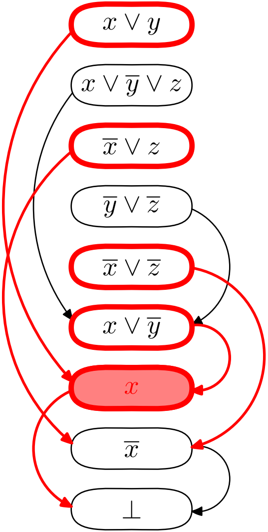
Size/length = total # clauses in refutation

Space = max # clauses in memory when performing refutation

(Exist other space measures also — focus here on most well-studied one)

Space at step t: # clauses at steps $\leq t$ used at steps $\geq t$

Example: Space at step 7 is 5



Upper Bounds on Resolution Size and Space

- **Size / space of refuting formula** defined by taking minimum over all resolution refutations
- Size always at most $\exp(\mathcal{O}(N))$
- Space always at most $N + \mathcal{O}(1)$
- Can be achieved simultaneously (even in tree-like resolution) [ET01]

Blackboard Definition of Resolution

Think of resolution refutation as being presented on blackboard:

- Write down axiom clauses from formula
- Apply resolution rule (only to clauses currently on board)
- Erase clauses (when no longer needed)

Blackboard Definition of Resolution

Think of resolution refutation as being presented on blackboard:

- Write down axiom clauses from formula
- Apply resolution rule (only to clauses currently on board)
- Erase clauses (when no longer needed)

Define derivation as sequence of clause configurations $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$ where $\mathbb{C}_t$ obtained from $\mathbb{C}_{t-1}$ by:

Download $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{C\}$ for axiom clause $C \in F$

Inference $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{D\}$ inferred by resolution on clauses in $\mathbb{C}_{t-1}$

Erasure $\mathbb{C}_t = \mathbb{C}_{t-1} \setminus \{D\}$ for some $D \in \mathbb{C}_{t-1}$

Blackboard Definition of Resolution

Think of resolution refutation as being presented on blackboard:

- Write down axiom clauses from formula
- Apply resolution rule (only to clauses currently on board)
- Erase clauses (when no longer needed)

Define derivation as sequence of clause configurations $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$ where $\mathbb{C}_t$ obtained from $\mathbb{C}_{t-1}$ by:

Download $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{C\}$ for axiom clause $C \in F$

Inference $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{D\}$ inferred by resolution on clauses in $\mathbb{C}_{t-1}$

Erasure $\mathbb{C}_t = \mathbb{C}_{t-1} \setminus \{D\}$ for some $D \in \mathbb{C}_{t-1}$

Size $=$ $\#$ download & inference steps
Space $= \max_{0 \leq t \leq \tau} \{|\mathbb{C}_t|\}$

Space Lower Bound as Two-Person Game

F requires space $s \Leftrightarrow$ all $\mathbb{C}_t$ derived from F in space $< s$ satisfiable

Space Lower Bound as Two-Person Game

F requires space $s \Leftrightarrow$ all $\mathbb{C}_t$ derived from F in space $< s$ satisfiable

Given derivation $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$, construct α_t satisfying $\mathbb{C}_t$

Space Lower Bound as Two-Person Game

F requires space $s \Leftrightarrow$ all $\mathbb{C}_t$ derived from F in space $< s$ satisfiable

Given derivation $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$, construct α_t satisfying $\mathbb{C}_t$

	Space game
Download	Pick α_t of size $\leq \lvert\mathbb{C}_t\rvert$
Inference	Do nothing
Erasure	Pick α_t of size $\leq \lvert\mathbb{C}_t\rvert$

Space Lower Bound as Two-Person Game

F requires space $s \Leftrightarrow$ all $\mathbb{C}_t$ derived from F in space $< s$ satisfiable

Given derivation $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$, construct α_t satisfying $\mathbb{C}_t$

	Space game	**Lower bound game**
Download	Pick α_t of size $\leq \lvert\mathbb{C}_t\rvert$	Enlarge to $\alpha_t \supseteq \alpha_{t-1}$ of size $\leq \lvert\mathbb{C}_t\rvert$
Inference	Do nothing	Do nothing
Erasure	Pick α_t of size $\leq \lvert\mathbb{C}_t\rvert$	Shrink to $\alpha_t \subseteq \alpha_{t-1}$ of size $\leq \lvert\mathbb{C}_t\rvert$

Space Lower Bound as Two-Person Game

F requires space $s \Leftrightarrow$ all $\mathbb{C}_t$ derived from F in space $< s$ satisfiable

Given derivation $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$, construct α_t satisfying $\mathbb{C}_t$

	Space game	**Lower bound game**
Download	Pick α_t of size $\leq |\mathbb{C}_t|$	Enlarge to $\alpha_t \supseteq \alpha_{t-1}$ of size $\leq |\mathbb{C}_t|$
Inference	Do nothing	Do nothing
Erasure	Pick α_t of size $\leq |\mathbb{C}_t|$	Shrink to $\alpha_t \subseteq \alpha_{t-1}$ of size $\leq |\mathbb{C}_t|$

Space game exactly characterizes space (but hard to play)

Restricted lower bound game: can construct α_t inductively (but no guarantee this will work)

General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $\mathbb{C}_t$

Construct auxiliary configuration $\mathbb{D}_t$ (view α_t as 1-CNF) that is easier to understand but still gives information about $\mathbb{C}_t$:

General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $\mathbb{C}_t$

Construct auxiliary configuration $\mathbb{D}_t$ (view α_t as 1-CNF) that is easier to understand but still gives information about $\mathbb{C}_t$:

1. $\mathbb{D}_t$ implies $\mathbb{C}_t$ (i.e., $\mathbb{D}_t$ "stronger" than $\mathbb{C}_t$)

General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $\mathbb{C}_t$

Construct auxiliary configuration $\mathbb{D}_t$ (view α_t as 1-CNF) that is easier to understand but still gives information about $\mathbb{C}_t$:

1. $\mathbb{D}_t$ implies $\mathbb{C}_t$ (i.e., $\mathbb{D}_t$ "stronger" than $\mathbb{C}_t$)
2. $\mathbb{D}_t$ is satisfiable (so, in particular, $\mathbb{C}_t$ also satisfiable)

General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $\mathbb{C}_t$

Construct auxiliary configuration $\mathbb{D}_t$ (view α_t as 1-CNF) that is easier to understand but still gives information about $\mathbb{C}_t$:

1. $\mathbb{D}_t$ implies $\mathbb{C}_t$ (i.e., $\mathbb{D}_t$ "stronger" than $\mathbb{C}_t$)
2. $\mathbb{D}_t$ is satisfiable (so, in particular, $\mathbb{C}_t$ also satisfiable)
3. $|\mathbb{D}_t| \leq |\mathbb{C}_t|$ (all we know about space of $\mathbb{C}_t$)

General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $\mathbb{C}_t$

Construct auxiliary configuration $\mathbb{D}_t$ (view α_t as 1-CNF) that is easier to understand but still gives information about $\mathbb{C}_t$:

1. $\mathbb{D}_t$ implies $\mathbb{C}_t$ (i.e., $\mathbb{D}_t$ "stronger" than $\mathbb{C}_t$)
2. $\mathbb{D}_t$ is satisfiable (so, in particular, $\mathbb{C}_t$ also satisfiable)
3. $|\mathbb{D}_t| \leq |\mathbb{C}_t|$ (all we know about space of $\mathbb{C}_t$)
4. At derivation step $\mathbb{C}_{t-1} \rightsquigarrow \mathbb{C}_t$, can do a local update $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$ if $|\mathbb{D}_{t-1}|$ small enough (i.e., less than s)

General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $\mathbb{C}_t$

Construct auxiliary configuration $\mathbb{D}_t$ (view α_t as 1-CNF) that is easier to understand but still gives information about $\mathbb{C}_t$:

1. $\mathbb{D}_t$ implies $\mathbb{C}_t$ (i.e., $\mathbb{D}_t$ "stronger" than $\mathbb{C}_t$)
2. $\mathbb{D}_t$ is satisfiable (so, in particular, $\mathbb{C}_t$ also satisfiable)
3. $|\mathbb{D}_t| \leq |\mathbb{C}_t|$ (all we know about space of $\mathbb{C}_t$)
4. At derivation step $\mathbb{C}_{t-1} \rightsquigarrow \mathbb{C}_t$, can do a local update $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$ if $|\mathbb{D}_{t-1}|$ small enough (i.e., less than s)

If we can do this, clearly we get lower bound on space

Two observations:

- "On the safe side" of adversary ($\mathbb{D}_t$ stronger than $\mathbb{C}_t$)
- History-dependent (can get different $\mathbb{D}_t$ for same $\mathbb{C}_t$)

Resolution Space Lower Bound for Random k-CNFs (1/2)

Random k-CNF formulas

Δn randomly sampled k-clauses over n variables

Resolution space lower bound $\Omega(n)$ [BG03]

In fact, holds for any CNF whose graph is good enough expander

Resolution Space Lower Bound for Random k-CNFs (1/2)

Random k-CNF formulas

Δn randomly sampled k-clauses over n variables

Resolution space lower bound $\Omega(n)$ [BG03]

In fact, holds for any CNF whose graph is good enough expander

Graph $G(F)$ of CNF F

- Bipartite graph $G(U \dot\cup V, E)$
- U = set of clauses; V = set of variables
- Edge (C, x) if variable x occurs in C [ignore sign of literal]

Resolution Space Lower Bound for Random k-CNFs (1/2)

Random k-CNF formulas

Δn randomly sampled k-clauses over n variables

Resolution space lower bound $\Omega(n)$ [BG03]

In fact, holds for any CNF whose graph is good enough expander

Graph $G(F)$ of CNF F

- Bipartite graph $G(U \dot\cup V, E)$
- U = set of clauses; V = set of variables
- Edge (C, x) if variable x occurs in C [ignore sign of literal]

(d, δ, s)-bipartite expander

- Bipartite graph $G(U \dot\cup V, E)$ with left degree d
- Every $A \subseteq U$ s.t. $|A| \leq s$ has neighbourhood $|N_G(A)| \geq \delta |A|$

Resolution Space Lower Bound for Random k-CNFs (2/2)

Theorem ([BG03])

If F is random k-CNF for $k \geq 3$ over n variables with Δn clauses then F requires space $\Omega(n)$ almost surely

Proof sketch.

Given small-space derivation $(\mathbb{C}_0, \mathbb{C}_1, \mathbb{C}_2, \ldots)$ from F, inductively construct 1-CNF $\mathbb{D}_t$ implying $\mathbb{C}_t$ and satisfying $|\mathbb{D}_t| \leq |\mathbb{C}_t|$:

1. Download of $C \in F$: Since $G(F)$ has expansion $1 + \epsilon$, can find variable in C not in $\mathbb{D}_{t-1}$ [needs an argument, of course]
2. Inference: Set $\mathbb{D}_t = \mathbb{D}_{t-1}$
3. Erasure: Pick $\mathbb{D}_t \subseteq \mathbb{D}_{t-1}$ of size $|\mathbb{D}_t| \leq |\mathbb{C}_t|$ implying $\mathbb{C}_t$ □

Taking Care of Erasures by Locality Lemma

Lemma (Locality lemma for resolution)

Suppose $\mathbb{D}$ *1-CNF;* $\mathbb{C}$ *clause configuration;* $\mathbb{D}$ *implies* $\mathbb{C}$

Then $\exists$ *1-CNF* $\mathbb{D}'$ *of size* $|\mathbb{D}'| \leq |\mathbb{C}|$ *s.t.* $\mathbb{D}'$ *implies* $\mathbb{C}$

Taking Care of Erasures by Locality Lemma

Lemma (Locality lemma for resolution)

Suppose $\mathbb{D}$ *1-CNF;* $\mathbb{C}$ *clause configuration;* $\mathbb{D}$ *implies* $\mathbb{C}$
Then $\exists$ *1-CNF* $\mathbb{D}'$ *of size* $|\mathbb{D}'| \leq |\mathbb{C}|$ *s.t.* $\mathbb{D}'$ *implies* $\mathbb{C}$

Proof.

Consider bipartite graph with

- clauses $C \in \mathbb{C}$ on left; unit clauses $\in \mathbb{D}$ on right
- edge between C and D if $D \vDash C$ (share a literal)

Taking Care of Erasures by Locality Lemma

Lemma (Locality lemma for resolution)

Suppose $\mathbb{D}$ *1-CNF;* $\mathbb{C}$ *clause configuration;* $\mathbb{D}$ *implies* $\mathbb{C}$
Then $\exists$ *1-CNF* $\mathbb{D}'$ *of size* $|\mathbb{D}'| \leq |\mathbb{C}|$ *s.t.* $\mathbb{D}'$ *implies* $\mathbb{C}$

Proof.

Consider bipartite graph with

- clauses $C \in \mathbb{C}$ on left; unit clauses $\in \mathbb{D}$ on right
- edge between C and D if $D \vDash C$ (share a literal)

For every $C \in \mathbb{C}$, pick one neighbour $D \in \mathbb{D}$ (must exist) to form 1-CNF $\mathbb{D}'$

Taking Care of Erasures by Locality Lemma

Lemma (Locality lemma for resolution)

Suppose $\mathbb{D}$ *1-CNF;* $\mathbb{C}$ *clause configuration;* $\mathbb{D}$ *implies* $\mathbb{C}$
Then $\exists$ *1-CNF* $\mathbb{D}'$ *of size* $|\mathbb{D}'| \leq |\mathbb{C}|$ *s.t.* $\mathbb{D}'$ *implies* $\mathbb{C}$

Proof.

Consider bipartite graph with

- clauses $C \in \mathbb{C}$ on left; unit clauses $\in \mathbb{D}$ on right
- edge between C and D if $D \vDash C$ (share a literal)

For every $C \in \mathbb{C}$, pick one neighbour $D \in \mathbb{D}$ (must exist) to form 1-CNF $\mathbb{D}'$

Then by construction:

- $|\mathbb{D}'| \leq |\mathbb{C}|$
- $\mathbb{D}' \vDash \mathbb{C}$ $\square$

Space Lower Bounds from Width Lower Bounds

Tight space lower bound obtained in this way also for

- Pigeonhole principle [ABRW02, ET01]
- Tseitin formulas [ABRW02, ET01]

Matching width lower bounds (min size of largest clause in proof)
Under the hood proofs of very similar flavour... What is going on?

Space Lower Bounds from Width Lower Bounds

Tight space lower bound obtained in this way also for

- Pigeonhole principle [ABRW02, ET01]
- Tseitin formulas [ABRW02, ET01]

Matching width lower bounds (min size of largest clause in proof)
Under the hood proofs of very similar flavour... What is going on?

Theorem ([AD03])

For k-CNF formulas it holds that space $\geq$ width $+ \mathcal{O}(1)$

Space Lower Bounds from Width Lower Bounds

Tight space lower bound obtained in this way also for

- Pigeonhole principle [ABRW02, ET01]
- Tseitin formulas [ABRW02, ET01]

Matching width lower bounds (min size of largest clause in proof)
Under the hood proofs of very similar flavour... What is going on?

Theorem ([AD03])

For k-CNF formulas it holds that space $\geq$ width $+ \mathcal{O}(1)$

With hindsight, almost all space lower bounds obtainable this way

But not quite — get back to this later

Polynomial Calculus (or Actually PCR)

Introduced in [CEI96]; below modified version from [ABRW02]

Clauses interpreted as polynomial equations over (fixed) field in variables $x, \overline{x}, y, \overline{y}, z, \overline{z}, \ldots$ (where x and $\overline{x}$ distinct variables)

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

Polynomial Calculus (or Actually PCR)

Introduced in [CEI96]; below modified version from [ABRW02]

Clauses interpreted as polynomial equations over (fixed) field in variables $x, \overline{x}, y, \overline{y}, z, \overline{z}, \ldots$ (where x and $\overline{x}$ distinct variables)

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 $\dfrac{p = 0 \quad q = 0}{\alpha p + \beta q = 0}$

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

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

Size, Degree and Space

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

Size, Degree and Space

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

Size — analogue of resolution size
total # monomials in refutation (counted with repetitions)
[Can also define length measure — might be much smaller]

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

(Monomial) space — analogue of resolution (clause) space
max # monomials in memory during refutation (with repetitions)

PCR Strictly Stronger than Resolution

Polynomial calculus simulates resolution efficiently with respect to length/size, width/degree, and space simultaneously

- Can mimic resolution refutation step by step
- Hence worst-case upper bounds for resolution carry over

PCR Strictly Stronger than Resolution

Polynomial calculus simulates resolution efficiently with respect to length/size, width/degree, and space simultaneously

- Can mimic resolution refutation step by step
- Hence worst-case upper bounds for resolution carry over

PCR strictly stronger w.r.t. size and degree

- Tseitin formulas on expanders over $\mathrm{GF}(2)$ (just do Gaussian elimination)
- Onto functional pigeonhole principle [Rii93]

PCR Strictly Stronger than Resolution

Polynomial calculus simulates resolution efficiently with respect to length/size, width/degree, and space simultaneously

- Can mimic resolution refutation step by step
- Hence worst-case upper bounds for resolution carry over

PCR strictly stronger w.r.t. size and degree

- Tseitin formulas on expanders over $\mathrm{GF}(2)$ (just do Gaussian elimination)
- Onto functional pigeonhole principle [Rii93]

Open Problem

Show that PCR is strictly stronger than resolution w.r.t. space

Lower Bounds on PCR Space

Lower bound for PHP with wide clauses [ABRW02]

k-CNFs much trickier — sequence of lower bounds for

- Obfuscated 4-CNF versions of PHP [FLN$^+$12]
- Random 4-CNFs + general technique [BG13]
- Tseitin formulas on (some) expanders [FLM$^+$13]

Lower Bounds on PCR Space

Lower bound for PHP with wide clauses [ABRW02]

k-CNFs much trickier — sequence of lower bounds for

- Obfuscated 4-CNF versions of PHP [FLN^{+}12]
- Random 4-CNFs + general technique [BG13]
- Tseitin formulas on (some) expanders [FLM^{+}13]

Open Problem

- *Prove tight space lower bounds for Tseitin on any expander*
- *Prove tight space lower bounds for ordering principle formulas*
- *Prove any space lower bound on random 3-CNFs*
- *Prove any space lower bound for any 3-CNF!?*

What We Want (Recap of Lower Bound Proof Strategy)

Given PCR derivation $(\mathbb{P}_0, \mathbb{P}_1, \mathbb{P}_2, \ldots)$ in small space

Want to construct "auxiliary configurations" $\mathbb{D}_0, \mathbb{D}_1, \mathbb{D}_2, \ldots$ s.t.

- $\mathbb{D}_t$ highly structured, so easier to understand than $\mathbb{P}_t$
- but still gives information about $\mathbb{P}_t$

What We Want (Recap of Lower Bound Proof Strategy)

Given PCR derivation $(\mathbb{P}_0, \mathbb{P}_1, \mathbb{P}_2, \ldots)$ in small space

Want to construct "auxiliary configurations" $\mathbb{D}_0, \mathbb{D}_1, \mathbb{D}_2, \ldots$ s.t.

- $\mathbb{D}_t$ highly structured, so easier to understand than $\mathbb{P}_t$
- but still gives information about $\mathbb{P}_t$

Maintain invariants for $\mathbb{D}_t$:

1. $\mathbb{D}_t$ implies $\mathbb{P}_t$ (i.e., $\mathbb{D}_t$ "stronger" than $\mathbb{P}_t$)
2. $\mathbb{D}_t$ is satisfiable (so, in particular, $\mathbb{P}_t$ also satisfiable)
3. space of $\mathbb{D}_t$ $\leq$ space of $\mathbb{P}_t$ (all we know about space of $\mathbb{P}_t$)
4. For $\mathbb{P}_{t-1} \rightsquigarrow \mathbb{P}_t$, can do update $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$ if $\mathbb{D}_{t-1}$ small

What We Want (Recap of Lower Bound Proof Strategy)

Given PCR derivation $(\mathbb{P}_0, \mathbb{P}_1, \mathbb{P}_2, \ldots)$ in small space

Want to construct "auxiliary configurations" $\mathbb{D}_0, \mathbb{D}_1, \mathbb{D}_2, \ldots$ s.t.

- $\mathbb{D}_t$ highly structured, so easier to understand than $\mathbb{P}_t$
- but still gives information about $\mathbb{P}_t$

Maintain invariants for $\mathbb{D}_t$:

1. $\mathbb{D}_t$ implies $\mathbb{P}_t$ (i.e., $\mathbb{D}_t$ "stronger" than $\mathbb{P}_t$)
2. $\mathbb{D}_t$ is satisfiable (so, in particular, $\mathbb{P}_t$ also satisfiable)
3. space of $\mathbb{D}_t \leq$ space of $\mathbb{P}_t$ (all we know about space of $\mathbb{P}_t$)
4. For $\mathbb{P}_{t-1} \rightsquigarrow \mathbb{P}_t$, can do update $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$ if $\mathbb{D}_{t-1}$ small

If so, small-space derivation cannot derive contradiction

So What's the Problem?

Resolution (clause) space $s \Rightarrow \exists$ satisfying assignment of size $\leq s$

So What's the Problem?

Resolution (clause) space $s \Rightarrow \exists$ satisfying assignment of size $\leq s$

Not true for polynomials!

Example

- Consider polynomial equation $-1 + xyzuvw = 0$
- Monomial space 2
- But have to set 6 variables to satisfy
- Obviously generalizes to arbitrary number of variables

Cannot use 1-CNFs / assignments as auxiliary configurations!

So What's the Problem?

Resolution (clause) space $s \Rightarrow \exists$ satisfying assignment of size $\leq s$

Not true for polynomials!

Example

- Consider polynomial equation $-1 + xyzuvw = 0$
- Monomial space 2
- But have to set 6 variables to satisfy
- Obviously generalizes to arbitrary number of variables

Cannot use 1-CNFs / assignments as auxiliary configurations!

But miraculously, 2-**CNFs sometimes work!** [ABRW02]

PCR Space Lower Bound for Random k-CNFs

Theorem ([BG13])

If F is random k-CNF for $k \geq 4$ over n variables with Δn clauses then F requires PCR space $\Omega(n)$ almost surely

PCR Space Lower Bound for Random k-CNFs

Theorem ([BG13])

If F is random k-CNF for $k \geq 4$ over n variables with Δn clauses then F requires PCR space $\Omega(n)$ almost surely

Proof approach:

- Structured auxiliary configurations: 2-CNFs $\mathbb{D}_t = \mathcal{A}_t \land \mathcal{B}_t$

PCR Space Lower Bound for Random k-CNFs

Theorem ([BG13])

If F is random k-CNF for $k \geq 4$ over n variables with Δn clauses then F requires PCR space $\Omega(n)$ almost surely

Proof approach:

- Structured auxiliary configurations: 2-CNFs $\mathbb{D}_t = \mathcal{A}_t \wedge \mathcal{B}_t$
- Each $A \in \mathcal{A}_t$ is subclause of axiom $C \in F$

PCR Space Lower Bound for Random k-CNFs

Theorem ([BG13])

If F is random k-CNF for $k \geq 4$ over n variables with Δn clauses then F requires PCR space $\Omega(n)$ almost surely

Proof approach:

- Structured auxiliary configurations: 2-CNFs $\mathbb{D}_t = \mathcal{A}_t \land \mathcal{B}_t$
- Each $A \in \mathcal{A}_t$ is subclause of axiom $C \in F$
- No distinct $A, A' \in \mathcal{A}_t$ share any variables

PCR Space Lower Bound for Random k-CNFs

Theorem ([BG13])

If F is random k-CNF for $k \geq 4$ over n variables with Δn clauses then F requires PCR space $\Omega(n)$ almost surely

Proof approach:

- Structured auxiliary configurations: 2-CNFs $\mathbb{D}_t = \mathcal{A}_t \wedge \mathcal{B}_t$
- Each $A \in \mathcal{A}_t$ is subclause of axiom $C \in F$
- No distinct $A, A' \in \mathcal{A}_t$ share any variables
- Every $B \in \mathcal{B}_t$ associated to two unique $A_B, A'_B \in \mathcal{A}_t$

PCR Space Lower Bound for Random k-CNFs

Theorem ([BG13])

If F is random k-CNF for $k \geq 4$ over n variables with Δn clauses then F requires PCR space $\Omega(n)$ almost surely

Proof approach:

- Structured auxiliary configurations: 2-CNFs $\mathbb{D}_t = \mathcal{A}_t \wedge \mathcal{B}_t$
- Each $A \in \mathcal{A}_t$ is subclause of axiom $C \in F$
- No distinct $A, A' \in \mathcal{A}_t$ share any variables
- Every $B \in \mathcal{B}_t$ associated to two unique $A_B, A'_B \in \mathcal{A}_t$
- B contains one variable from A_B and one variable from A'_B

PCR Space Lower Bound for Random k-CNFs

Theorem ([BG13])

If F is random k-CNF for $k \geq 4$ over n variables with Δn clauses then F requires PCR space $\Omega(n)$ almost surely

Proof approach:

- Structured auxiliary configurations: 2-CNFs $\mathbb{D}_t = \mathcal{A}_t \wedge \mathcal{B}_t$
- Each $A \in \mathcal{A}_t$ is subclause of axiom $C \in F$
- No distinct $A, A' \in \mathcal{A}_t$ share any variables
- Every $B \in \mathcal{B}_t$ associated to two unique $A_B, A'_B \in \mathcal{A}_t$
- B contains one variable from A_B and one variable from A'_B

(Straightforward to verify that any such $\mathbb{D}_t$ is satisfiable)

Inductive Proof: Invariants and Inference

Proof invariants:

- $\mathbb{D}_t = \mathcal{A}_t \wedge \mathcal{B}_t$ structured auxiliary configuration
- $\mathbb{D}_t$ implies $\mathbb{P}_t$
- $|\mathbb{D}_t| \leq 6 \cdot (\#$ [distinct] monomials in $\mathbb{P}_t)$

Proof is by case analysis over derivation step

Inductive Proof: Invariants and Inference

Proof invariants:

- $\mathbb{D}_t = \mathcal{A}_t \wedge \mathcal{B}_t$ structured auxiliary configuration
- $\mathbb{D}_t$ implies $\mathbb{P}_t$
- $|\mathbb{D}_t| \leq 6 \cdot (\#$ [distinct] monomials in $\mathbb{P}_t)$

Proof is by case analysis over derivation step

1. Inference $\mathbb{P}_t = \mathbb{P}_t \cup \{Q\}$ for polynomial Q derived from $\mathbb{P}_{t-1}$

- Set $\mathbb{D}_t := \mathbb{D}_{t-1}$
- $\mathbb{D}_t = \mathbb{D}_{t-1}$ implies Q by soundness
- Space of $\mathbb{D}_t$ stays the same
- Space of $\mathbb{P}_t$ goes up

Inductive Proof: Axiom Download

2. **Download** $\mathbb{P}_t = \mathbb{P}_t \cup \{C\}$ for $C \in F$
   - For simplicity, assume extra download of $C' \in F$
   - Without loss of generality: can then immediately erase C'

Inductive Proof: Axiom Download

2. **Download** $\mathbb{P}_t = \mathbb{P}_t \cup \{C\}$ for $C \in F$
   - For simplicity, assume extra download of $C' \in F$
   - Without loss of generality: can then immediately erase C'
   - Since $G(F)$ has expansion $2 + \epsilon$, can find 2-clauses $A \subseteq C$ and $A' \subseteq C'$ on disjoint sets of variables [argument analogous to [BG03] but expansion requires 4-CNF]
   - Pick one arbitrary literal each from A and A' to form B

Inductive Proof: Axiom Download

2. **Download** $\mathbb{P}_t = \mathbb{P}_t \cup \{C\}$ for $C \in F$

- For simplicity, assume extra download of $C' \in F$
- Without loss of generality: can then immediately erase C'
- Since $G(F)$ has expansion $2 + \epsilon$, can find 2-clauses $A \subseteq C$ and $A' \subseteq C'$ on disjoint sets of variables [argument analogous to [BG03] but expansion requires 4-CNF]
- Pick one arbitrary literal each from A and A' to form B
- $\mathcal{A}_t := \mathcal{A}_{t-1} \cup \{A, A'\}$
- $\mathcal{B}_t := \mathcal{B}_{t-1} \cup \{B\}$
- Space of $\mathbb{D}_t = \mathcal{A}_t \wedge \mathcal{B}_t$ up by 3
- Space of $\mathbb{P}_t$ up by 1

Inductive Proof: Erasure

3. **Erasure** $\mathbb{P}_t = \mathbb{P}_{t-1} \setminus \{Q\}$ for $Q \in \mathbb{P}_{t-1}$
   - Know $\mathbb{D}_{t-1}$ implies $\mathbb{P}_t \subseteq \mathbb{P}_{t-1}$
   - But $|\mathbb{D}_{t-1}|$ might be far too large
   - Need to find smaller auxiliary configuration that implies $\mathbb{P}_t$
     (Was very easy for resolution; now not clear at all what to do)

Inductive Proof: Erasure

3. Erasure $\mathbb{P}_t = \mathbb{P}_{t-1} \setminus \{Q\}$ for $Q \in \mathbb{P}_{t-1}$

- Know $\mathbb{D}_{t-1}$ implies $\mathbb{P}_t \subseteq \mathbb{P}_{t-1}$
- But $|\mathbb{D}_{t-1}|$ might be far too large
- Need to find smaller auxiliary configuration that implies $\mathbb{P}_t$
  (Was very easy for resolution; now not clear at all what to do)

Lemma (Locality lemma for PCR [ABRW02, BG13])

Suppose

- $\mathbb{D} = \mathcal{A} \wedge \mathcal{B}$ *structured auxiliary configuration*
- $\mathbb{P}$ *PCR-configuration*
- $\mathbb{D}$ *implies* $\mathbb{P}$

Then

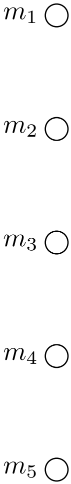
$\exists\, \mathbb{D}^* = \mathcal{A}^* \wedge \mathcal{B}^*$ *with* $|\mathbb{D}^*| \leq 6 \cdot (\#$ *monomials in* $\mathbb{P})$ *s.t.* $\mathbb{D}^*$ *implies* $\mathbb{P}$

Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$

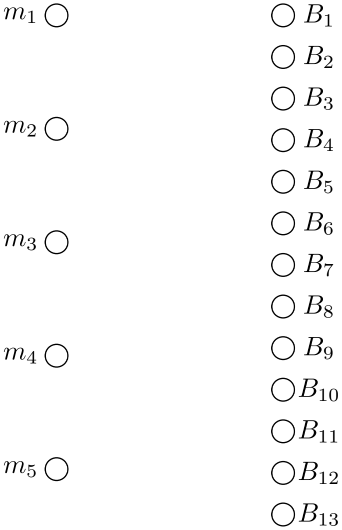
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$



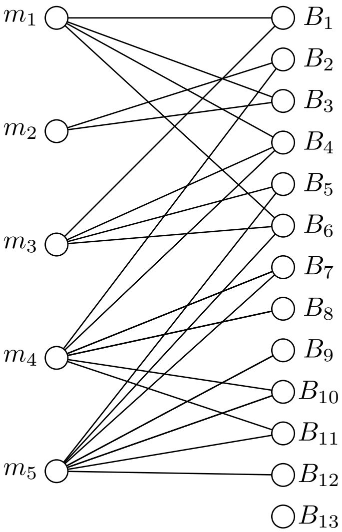
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$



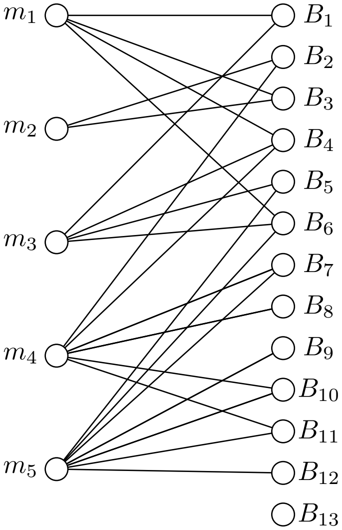
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable



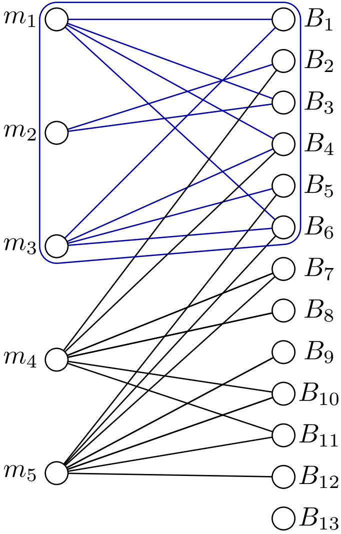
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$



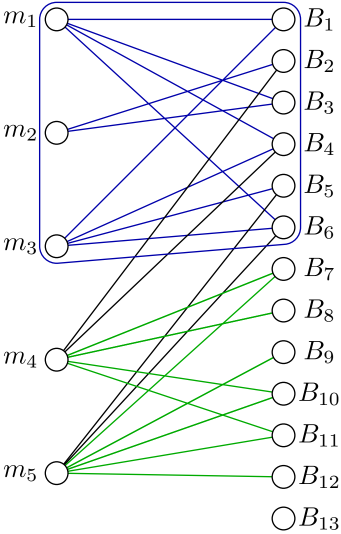
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$
- Assume $\Gamma \neq M$ (else done)



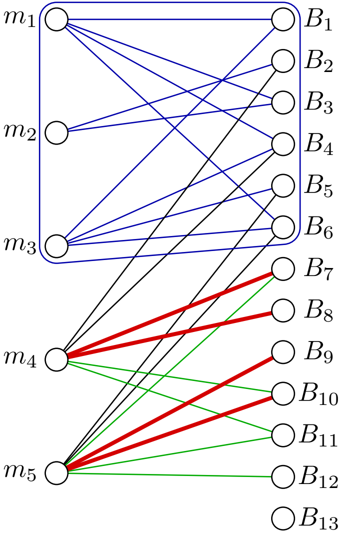
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$
- Assume $\Gamma \neq M$ (else done)
- $\forall\ S \subseteq M \setminus \Gamma$ by maximality $|N(S) \setminus N(\Gamma)| > 2 \cdot |S|$



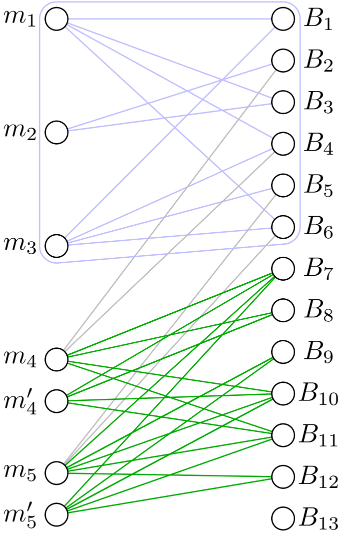
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$
- Assume $\Gamma \neq M$ (else done)
- $\forall\ S \subseteq M \setminus \Gamma$ by maximality $|N(S) \setminus N(\Gamma)| > 2 \cdot |S|$
- $\Rightarrow$ $\exists$ matching of each $m \in M \setminus \Gamma$ to 2 distinct $B', B'' \in \mathcal{B} \setminus N(\Gamma)$



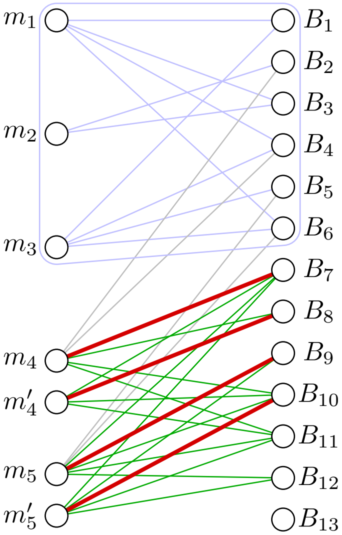
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$
- Assume $\Gamma \neq M$ (else done)
- $\forall\ S \subseteq M \setminus \Gamma$ by maximality $|N(S) \setminus N(\Gamma)| > 2 \cdot |S|$
- $\Rightarrow$ $\exists$ matching of each $m \in M \setminus \Gamma$ to 2 distinct $B', B'' \in \mathcal{B} \setminus N(\Gamma)$
- (Make 2 copies of each $m \in M \setminus \Gamma$ and apply Hall's theorem)



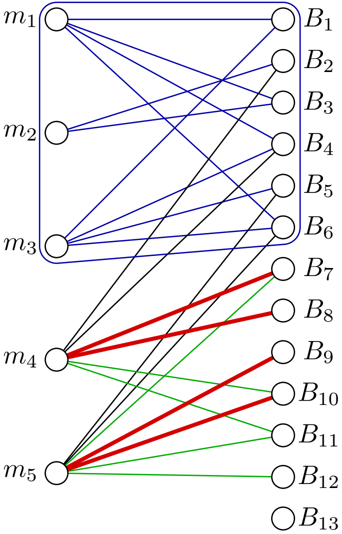
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G=(U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$
- Assume $\Gamma \neq M$ (else done)
- $\forall\ S \subseteq M \setminus \Gamma$ by maximality $|N(S) \setminus N(\Gamma)| > 2 \cdot |S|$
- $\Rightarrow$ $\exists$ matching of each $m \in M \setminus \Gamma$ to 2 distinct $B', B'' \in \mathcal{B} \setminus N(\Gamma)$
- (Make 2 copies of each $m \in M \setminus \Gamma$ and apply Hall's theorem)



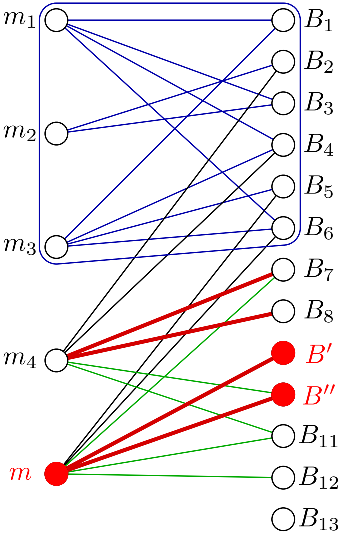
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$
- Assume $\Gamma \neq M$ (else done)
- $\forall\ S \subseteq M \setminus \Gamma$ by maximality $|N(S) \setminus N(\Gamma)| > 2 \cdot |S|$
- $\Rightarrow$ $\exists$ matching of each $m \in M \setminus \Gamma$ to 2 distinct $B', B'' \in \mathcal{B} \setminus N(\Gamma)$
- (Make 2 copies of each $m \in M \setminus \Gamma$ and apply Hall's theorem)



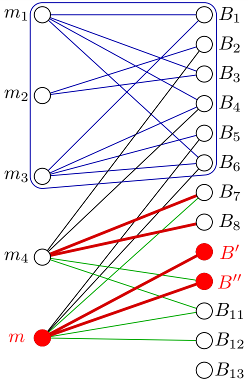
Proof sketch for Locality Lemma for PCR (1/4)

- Build graph $G = (U \cup V, E)$
- U = distinct monomials M in $\mathbb{P}$
- V = clauses in $\mathcal{B}$
- Edge between $m \in M$ and $B \in \mathcal{B}$ if $\exists$ common variable
- Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$
- Assume $\Gamma \neq M$ (else done)
- $\forall\ S \subseteq M \setminus \Gamma$ by maximality $|N(S) \setminus N(\Gamma)| > 2 \cdot |S|$
- $\Rightarrow$ $\exists$ matching of each $m \in M \setminus \Gamma$ to 2 distinct $B', B'' \in \mathcal{B} \setminus N(\Gamma)$
- (Make 2 copies of each $m \in M \setminus \Gamma$ and apply Hall's theorem)



Proof sketch for Locality Lemma for PCR (2/4)

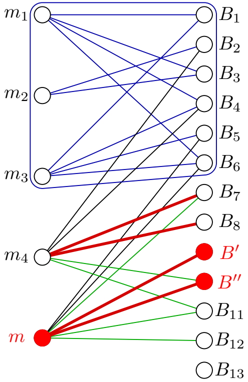
Look at $m \in M \setminus \Gamma$ — suppose matched to $B' = \overline{x} \vee \overline{y}$ and $B'' = \overline{z} \vee \overline{w}$



Proof sketch for Locality Lemma for PCR (2/4)

Look at $m \in M \setminus \Gamma$ — suppose matched to $B' = \overline{x} \vee \overline{y}$ and $B'' = \overline{z} \vee \overline{w}$

Say x, z common variables and $m = xz \cdot m'$ (maybe y and/or w in m' — don't care)



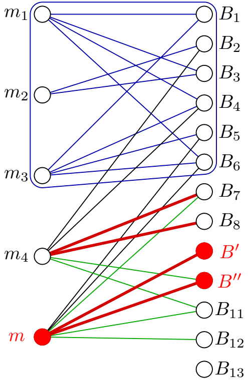
Proof sketch for Locality Lemma for PCR (2/4)

Look at $m \in M \setminus \Gamma$ — suppose matched to $B' = \overline{x} \lor \overline{y}$ and $B'' = \overline{z} \lor \overline{w}$

Say x, z common variables and $m = xz \cdot m'$ (maybe y and/or w in m' — don't care)

Suppose further

- $B' \leftrightarrow A'_1 = x \lor x'$ and $A'_2 = y \lor y'$
- $B'' \leftrightarrow A''_1 = z \lor z'$ and $A''_2 = w \lor w'$



Proof sketch for Locality Lemma for PCR (2/4)

Look at $m \in M \setminus \Gamma$ — suppose matched to $B' = \overline{x} \vee \overline{y}$ and $B'' = \overline{z} \vee \overline{w}$

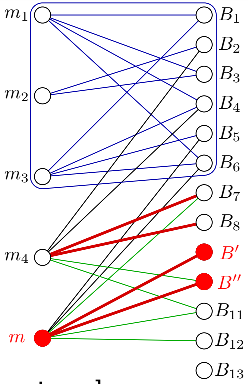
Say x, z common variables and $m = xz \cdot m'$ (maybe y and/or w in m' — don't care)

Suppose further

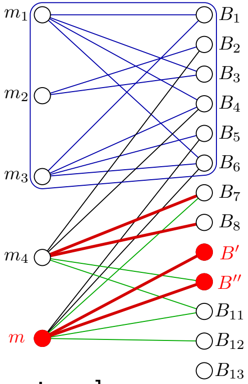
- $B' \leftrightarrow A'_1 = x \vee x'$ and $A'_2 = y \vee y'$
- $B'' \leftrightarrow A''_1 = z \vee z'$ and $A''_2 = w \vee w'$

New clauses for m in $\mathbb{D}^*$ will be

- $B^* = x \vee z$ [common variables with signs as in m]
- $A^*_1 = x \vee x'$ [A-clause associated to x]
- $A^*_2 = z \vee z'$ [A-clause associated to z]



Proof sketch for Locality Lemma for PCR (2/4)



Look at $m \in M \setminus \Gamma$ — suppose matched to $B' = \overline{x} \vee \overline{y}$ and $B'' = \overline{z} \vee \overline{w}$

Say x, z common variables and $m = xz \cdot m'$ (maybe y and/or w in m' — don't care)

Suppose further

- $B' \leftrightarrow A'_1 = x \vee x'$ and $A'_2 = y \vee y'$
- $B'' \leftrightarrow A''_1 = z \vee z'$ and $A''_2 = w \vee w'$

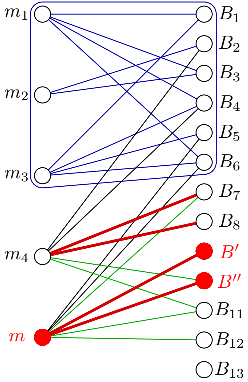
New clauses for m in $\mathbb{D}^*$ will be

- $B^* = x \vee z$ [common variables with signs as in m]
- $A^*_1 = x \vee x'$ [A-clause associated to x]
- $A^*_2 = z \vee z'$ [A-clause associated to z]

Plus keep all B-clauses in $N(\Gamma)$ and their A-clauses — **Done!**

Proof sketch for Locality Lemma for PCR (3/4)

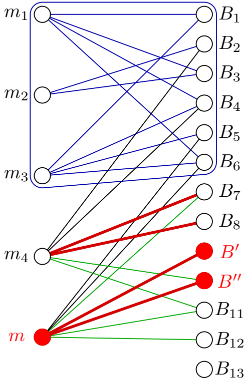
Need to prove three things:



Proof sketch for Locality Lemma for PCR (3/4)

Need to prove three things:

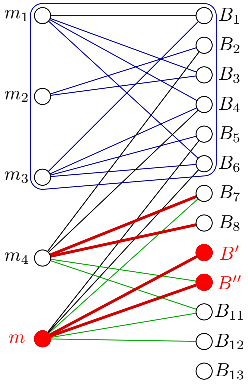
1. $\mathbb{D}^*$ structured auxiliary configuration
   Straightforward to verify



Proof sketch for Locality Lemma for PCR (3/4)

Need to prove three things:

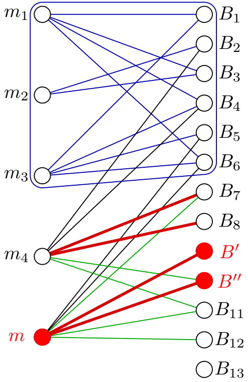
1. $\mathbb{D}^*$ structured auxiliary configuration
   Straightforward to verify
2. $\mathbb{D}^*$ has the right size
   OK, since $|\mathbb{D}^*| \leq 6 \cdot |M| \leq$
   $\leq 6 \cdot (\#$ monomials in $\mathbb{P})$



Proof sketch for Locality Lemma for PCR (3/4)

Need to prove three things:

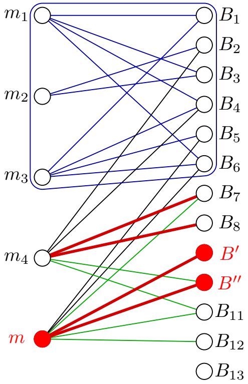
1. $\mathbb{D}^*$ structured auxiliary configuration
   Straightforward to verify
2. $\mathbb{D}^*$ has the right size
   OK, since $|\mathbb{D}^*| \leq 6 \cdot |M| \leq$
   $\leq 6 \cdot (\#$ monomials in $\mathbb{P})$
3. $\mathbb{D}^*$ implies $\mathbb{P}$
   Perhaps a priori not so clear...



Proof sketch for Locality Lemma for PCR (3/4)

Need to prove three things:

1. $\mathbb{D}^*$ structured auxiliary configuration
   Straightforward to verify
2. $\mathbb{D}^*$ has the right size
   OK, since $|\mathbb{D}^*| \leq 6 \cdot |M| \leq$
   $\leq 6 \cdot (\#$ monomials in $\mathbb{P})$
3. $\mathbb{D}^*$ implies $\mathbb{P}$
   Perhaps a priori not so clear...



Prove implication in slightly roundabout way:
Given any β satisfying $\mathbb{D}^*$, find α such that

- $\mathbb{P}(\alpha) = \mathbb{P}(\beta)$
- α satisfies $\mathbb{D}$

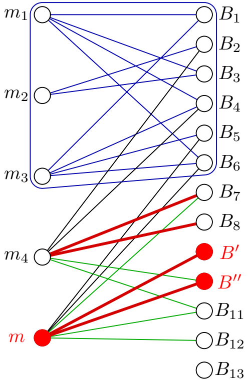
Proof sketch for Locality Lemma for PCR (4/4)

Look at our example monomial

- $m = xz \cdot m' \in M \setminus \Gamma$

with new clauses in $\mathbb{D}^*$ [satisfied by β]

- $B^* = x \vee z$, $A_1^* = x \vee x'$, $A_2^* = z \vee z'$



Proof sketch for Locality Lemma for PCR (4/4)

Look at our example monomial

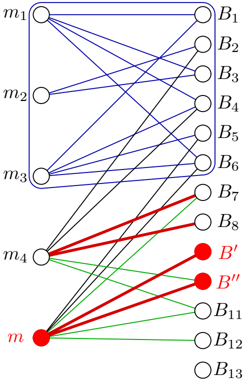
- $m = xz \cdot m' \in M \setminus \Gamma$

with new clauses in $\mathbb{D}^*$ [satisfied by β]

- $B^* = x \vee z,\ A_1^* = x \vee x',\ A_2^* = z \vee z'$

Old clauses in $\mathbb{D}$ [to be satisfied by α] are:

- $B' = \overline{x} \vee \overline{y} \leftrightarrow A_1' = x \vee x',\ A_2' = y \vee y'$
- $B'' = \overline{z} \vee \overline{w} \leftrightarrow A_1'' = z \vee z',\ A_2'' = w \vee w'$



Proof sketch for Locality Lemma for PCR (4/4)

Look at our example monomial

- $m = xz \cdot m' \in M \setminus \Gamma$

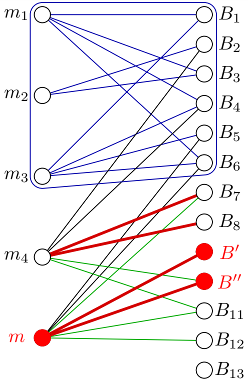
with new clauses in $\mathbb{D}^*$ [satisfied by β]

- $B^* = x \vee z,\ A_1^* = x \vee x',\ A_2^* = z \vee z'$

Old clauses in $\mathbb{D}$ [to be satisfied by α] are:

- $B' = \overline{x} \vee \overline{y} \leftrightarrow A_1' = x \vee x',\ A_2' = y \vee y'$
- $B'' = \overline{z} \vee \overline{w} \leftrightarrow A_1'' = z \vee z',\ A_2'' = w \vee w'$

Let $\alpha = \beta$ except that for $m \in M \setminus \Gamma$ we set $y = w = \mathit{false}$ and $x' = y' = z' = w' = \mathit{true}$



Proof sketch for Locality Lemma for PCR (4/4)

Look at our example monomial

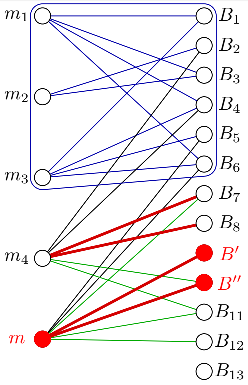
- $m = xz \cdot m' \in M \setminus \Gamma$

with new clauses in $\mathbb{D}^*$ [satisfied by β]

- $B^* = x \vee z,\ A_1^* = x \vee x',\ A_2^* = z \vee z'$

Old clauses in $\mathbb{D}$ [to be satisfied by α] are:

- $B' = \overline{x} \vee \overline{y} \leftrightarrow A_1' = x \vee x',\ A_2' = y \vee y'$
- $B'' = \overline{z} \vee \overline{w} \leftrightarrow A_1'' = z \vee z',\ A_2'' = w \vee w'$



Let $\alpha = \beta$ except that for $m \in M \setminus \Gamma$ we set $y = w = \mathit{false}$ and $x' = y' = z' = w' = \mathit{true}$

- $\alpha(m) = \beta(m)$ for all $m \in \Gamma$ [didn't touch those variables]
- $\alpha(m) = \beta(m) = 0$ for all $m \in M \setminus \Gamma$ [by construction of $\mathbb{D}^*$]
- α satisfies $\mathbb{D}$ and hence $\mathbb{P}$
- But then β must also satisfy $\mathbb{P}$, **Q.E.D.**

Another Intriguing Problem: Space vs. Degree

Open Problem (analogue of [AD08])

Is it true that space $\geq$ degree $+ \mathcal{O}(1)$?

Partial progress: if formula requires large resolution width, then XOR-substituted version requires large space [FLM^{+}13]

Another Intriguing Problem: Space vs. Degree

Open Problem (analogue of [AD08])

Is it true that space $\geq$ degree $+ \mathcal{O}(1)$?

Partial progress: if formula requires large resolution width, then XOR-substituted version requires large space [FLM+13]

Optimal separation of space and degree in [FLM+13] by flavour of Tseitin formulas which

- can be refuted in degree $\mathcal{O}(1)$
- require space $\Omega(N)$
- but separating formulas depend on characteristic of field

Comparing Size and Space

Some "rescaling" needed to get meaningful comparisons of size/length and space

- Size exponential in formula size in worst case
- Space at most linear in worst case
- So natural to compare space to logarithm of size

Size-Space Correlations and/or Trade-offs?

$\exists$ constant space refutation $\Rightarrow$ $\exists$ polynomial size refutation [AD03]

Size-Space Correlations and/or Trade-offs?

$\exists$ constant space refutation $\Rightarrow$ $\exists$ polynomial size refutation [AD03]

For tree-like resolution: any polynomial size refutation can be carried out in logarithmic space [ET01]

So essentially no trade-offs for tree-like resolution

Size-Space Correlations and/or Trade-offs?

$\exists$ constant space refutation $\Rightarrow$ $\exists$ polynomial size refutation [AD03]

For tree-like resolution: any polynomial size refutation can be carried out in logarithmic space [ET01]

So essentially no trade-offs for tree-like resolution

Does short size imply small space for general resolution?

Are there size-space trade-offs for general resolution?
(Some trade-off results in restricted settings in [Ben02, Nor09])

An Optimal Size-Space Separation

Size and space in resolution are "completely uncorrelated"

Theorem ([BN08])

There are k-CNF formula families of size N with

- *refutation size* $\mathcal{O}(N)$
- *refutation space* $\Omega(N/\log N)$

Optimal separation of size and space — given size $\mathcal{O}(N)$, always possible to get clause space $\mathcal{O}(N/\log N)$

Size-Space Trade-offs

There is a rich collection of size-space trade-offs

Results hold for

- resolution
- even k-DNF resolution (which we won't go into here)

Different trade-offs covering (almost) whole range of space from constant to linear

Simple, explicit formulas

One Example: Robust Trade-offs for Small Space

Theorem ([BN11] (informal))

For any arbitrarily slowly growing function g there exist explicit k-CNF formulas of size N

One Example: Robust Trade-offs for Small Space

Theorem ([BN11] (informal))

For any arbitrarily slowly growing function g there exist explicit k-CNF formulas of size N

- *refutable in resolution in space $g(N)$ and*

One Example: Robust Trade-offs for Small Space

Theorem ([BN11] (informal))

For any arbitrarily slowly growing function g there exist explicit k-CNF formulas of size N

- *refutable in resolution in space $g(N)$ and*
- *refutable in size linear in N and space $\approx \sqrt[3]{N}$ such that*

One Example: Robust Trade-offs for Small Space

Theorem ([BN11] (informal))

For any arbitrarily slowly growing function g there exist explicit k-CNF formulas of size N

- *refutable in resolution in space $g(N)$ and*
- *refutable in size linear in N and space $\approx \sqrt[3]{N}$ such that*
- *any refutation in space $\ll \sqrt[3]{N}$ requires superpolynomial size*

One Example: Robust Trade-offs for Small Space

Theorem ([BN11] (informal))

For any arbitrarily slowly growing function g there exist explicit k-CNF formulas of size N

- *refutable in resolution in space $g(N)$ and*
- *refutable in size linear in N and space $\approx \sqrt[3]{N}$ such that*
- *any refutation in space $\ll \sqrt[3]{N}$ requires superpolynomial size*

And an open problem:

Open Problem

Seems likely that $\sqrt[3]{N}$ above should be possible to improve to $\sqrt{N}$, but don't know how to prove this...

Proof Strategy for Size-Space Separations and Trade-offs

- Both of these theorems proved in the same way
- Want to sketch intuition and main ideas in proofs
- For details, see survey [Nor13]
- To prove the theorems, need to go back to the early days of computer science. . .

A Detour into Combinatorial Games

Want to find formulas that

- can be quickly refuted but require large space
- have space-efficient refutations requiring much time

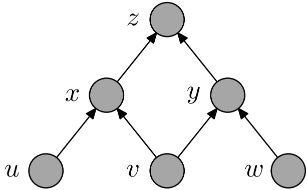
Such time-space trade-off questions well-studied for pebble games modelling calculations described by DAGs ([CS76] and many others)

- Time needed for calculation: # pebbling moves
- Space needed for calculation: max # pebbles required

Pebbling Formulas: Vanilla Version

CNF formulas encoding pebble games on DAGs

1. u
2. v
3. w
4. $\overline{u} \vee \overline{v} \vee x$
5. $\overline{v} \vee \overline{w} \vee y$
6. $\overline{x} \vee \overline{y} \vee z$
7. $\overline{z}$

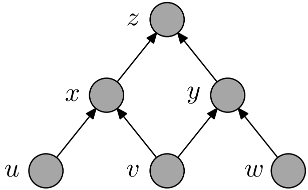


- sources are true
- truth propagates upwards
- but sink is false

Pebbling Formulas: Vanilla Version

CNF formulas encoding pebble games on DAGs

1. u
2. v
3. w
4. $\overline{u} \lor \overline{v} \lor x$
5. $\overline{v} \lor \overline{w} \lor y$
6. $\overline{x} \lor \overline{y} \lor z$
7. $\overline{z}$

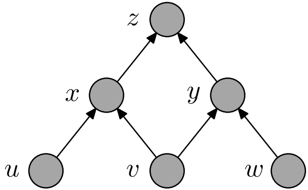


- sources are true
- truth propagates upwards
- but sink is false

Pebbling Formulas: Vanilla Version

CNF formulas encoding pebble games on DAGs

1. u
2. v
3. w
4. $\overline{u} \lor \overline{v} \lor x$
5. $\overline{v} \lor \overline{w} \lor y$
6. $\overline{x} \lor \overline{y} \lor z$
7. $\overline{z}$

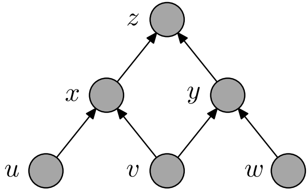


- sources are true
- truth propagates upwards
- but sink is false

Pebbling Formulas: Vanilla Version

CNF formulas encoding pebble games on DAGs

1. u
2. v
3. w
4. $\overline{u} \lor \overline{v} \lor x$
5. $\overline{v} \lor \overline{w} \lor y$
6. $\overline{x} \lor \overline{y} \lor z$
7. $\overline{z}$

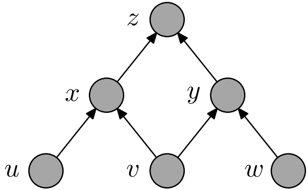


- sources are true
- truth propagates upwards
- but sink is false

Pebbling Formulas: Vanilla Version

CNF formulas encoding pebble games on DAGs

1. u
2. v
3. w
4. $\overline{u} \vee \overline{v} \vee x$
5. $\overline{v} \vee \overline{w} \vee y$
6. $\overline{x} \vee \overline{y} \vee z$
7. $\overline{z}$



- sources are true
- truth propagates upwards
- but sink is false

Extensive literature on pebbling space and time-space trade-offs from 1970s and 80s

Have been useful in proof complexity before in various contexts

Hope that pebbling properties of DAG somehow carry over to resolution refutations of pebbling formulas

Pebbling Formula Trade-offs

- Reduction from resolution to pebbling [Ben02]
- Pebbling time-space trade-offs $\Rightarrow$ size-variable space trade-offs in resolution [BN11]
- In fact, size-variable space trade-offs for any "semantic" proof system [BNT13]
- But we want trade-offs for stronger space measures!
- And pebbling formulas supereasy — can do constant (clause) space and linear size simultaneously

Key New (Old?) Idea: Variable Substitution

Make formula harder by substituting exclusive or $x_1 \oplus x_2$ of two new variables x_1 and x_2 for every variable x (also works for other Boolean functions with "right" properties):

$$\overline{x} \lor y$$

$$\Downarrow$$

$$\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)$$

$$\Downarrow$$

$$\begin{aligned} & (x_1 \lor \overline{x}_2 \lor y_1 \lor y_2) \\ \land\ & (x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2) \\ \land\ & (\overline{x}_1 \lor x_2 \lor y_1 \lor y_2) \\ \land\ & (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2) \end{aligned}$$

Key Technical Result: Substitution Theorem

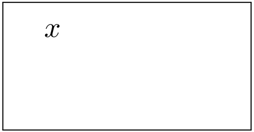
Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

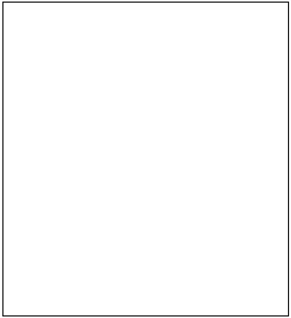
Obvious approach for refuting $F[\oplus]$: mimic refutation of F

Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F





Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F

x
$\overline{x} \lor y$

Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F

x

$\overline{x} \vee y$

y

Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F

$$x$$
$$\overline{x} \lor y$$
$$y$$

$$x_1 \lor x_2$$
$$\overline{x}_1 \lor \overline{x}_2$$

Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F

x
$\overline{x} \lor y$
y

$x_1 \lor x_2$
$\overline{x}_1 \lor \overline{x}_2$
$x_1 \lor \overline{x}_2 \lor y_1 \lor y_2$
$x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2$
$\overline{x}_1 \lor x_2 \lor y_1 \lor y_2$
$\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2$

Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F

x
$\overline{x} \lor y$
y

$x_1 \lor x_2$
$\overline{x}_1 \lor \overline{x}_2$
$x_1 \lor \overline{x}_2 \lor y_1 \lor y_2$
$x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2$
$\overline{x}_1 \lor x_2 \lor y_1 \lor y_2$
$\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2$
$y_1 \lor y_2$
$\overline{y}_1 \lor \overline{y}_2$

Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F

$$\boxed{\begin{array}{l} x \\ \overline{x} \lor y \\ y \end{array}}$$

$$\boxed{\begin{array}{l} x_1 \lor x_2 \\ \overline{x}_1 \lor \overline{x}_2 \\ x_1 \lor \overline{x}_2 \lor y_1 \lor y_2 \\ x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2 \\ \overline{x}_1 \lor x_2 \lor y_1 \lor y_2 \\ \overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2 \\ y_1 \lor y_2 \\ \overline{y}_1 \lor \overline{y}_2 \end{array}}$$

For such refutation of $F[\oplus]$:

- size $\geq$ size for F
- clause space $\geq$ # variables on board in proof for F

Key Technical Result: Substitution Theorem

Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

Obvious approach for refuting $F[\oplus]$: mimic refutation of F

x
$\overline{x} \lor y$
y

$x_1 \lor x_2$
$\overline{x}_1 \lor \overline{x}_2$
$x_1 \lor \overline{x}_2 \lor y_1 \lor y_2$
$x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2$
$\overline{x}_1 \lor x_2 \lor y_1 \lor y_2$
$\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2$
$y_1 \lor y_2$
$\overline{y}_1 \lor \overline{y}_2$

For such refutation of $F[\oplus]$:

- size $\geq$ size for F
- clause space $\geq$ # variables on board in proof for F

Prove that this is (sort of) best one can do for $F[\oplus]$!

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \vee (y_1 \oplus y_2) \ldots$	

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \vee (y_1 \oplus y_2) \ldots$	write $\overline{x} \vee y$ on shadow blackboard

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \vee (y_1 \oplus y_2) \ldots$	write $\overline{x} \vee y$ on shadow blackboard
For consecutive XOR blackboard configurations...	

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \vee (y_1 \oplus y_2)$...	write $\overline{x} \vee y$ on shadow blackboard
For consecutive XOR blackboard configurations...	can get between corresponding shadow blackboards by legal resolution derivation steps

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \vee (y_1 \oplus y_2) \ldots$	write $\overline{x} \vee y$ on shadow blackboard
For consecutive XOR blackboard configurations...	can get between corresponding shadow blackboards by legal resolution derivation steps
	Size of shadow blackboard derivation ...

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)$...	write $\overline{x} \lor y$ on shadow blackboard
For consecutive XOR blackboard configurations...	can get between corresponding shadow blackboards by legal resolution derivation steps
... (sort of) upper-bounded by XOR derivation size	Size of shadow blackboard derivation ...

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \vee (y_1 \oplus y_2)$...	write $\overline{x} \vee y$ on shadow blackboard
For consecutive XOR blackboard configurations...	can get between corresponding shadow blackboards by legal resolution derivation steps
... (sort of) upper-bounded by XOR derivation size	Size of shadow blackboard derivation ...
	# variables mentioned on shadow blackboard...

Sketch of Proof of Substitution Theorem

Given refutation of $F[\oplus]$, extract "shadow refutation" of F

XOR formula $F[\oplus]$	**Original formula** F
If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \vee (y_1 \oplus y_2)$...	write $\overline{x} \vee y$ on shadow blackboard
For consecutive XOR blackboard configurations...	can get between corresponding shadow blackboards by legal resolution derivation steps
... (sort of) upper-bounded by XOR derivation size	Size of shadow blackboard derivation ...
... is at most # clauses on XOR blackboard	# variables mentioned on shadow blackboard...

Putting the Pieces Together

Making variable substitutions in pebbling formulas

- lifts lower bound from number of variables to (clause) space
- maintains upper bound in terms of space and size

Putting the Pieces Together

Making variable substitutions in pebbling formulas

- lifts lower bound from number of variables to (clause) space
- maintains upper bound in terms of space and size

Get our results by

- using known pebbling results from literature of 70s and 80s
- proving a couple of new pebbling results [Nor12]

Some Philosophical Notes

- Projections "on the wrong side" of adversary (we throw away info and get weaker configuration)
- Independent of history (always same projection from same configuration)
- Only technique for proving space lower bounds without dependence on width lower bounds (pebbling formulas refutable in constant width)
- Is there a "safe side of adversary," history-dependent space lower bound proof for pebbling formulas?

Projections v.s. Restrictions for Polynomial Calculus

Projections in [BN11] fail for polynomial calculus and PCR (see [Nor13] for examples)

Projections v.s. Restrictions for Polynomial Calculus

Projections in [BN11] fail for polynomial calculus and PCR (see [Nor13] for examples)

Use XOR-substitution + random restrictions

- If refutation short $\Rightarrow$ restriction kills all high-degree monomials
- If also monomial space small $\Rightarrow$ get small variable space
- But then size-variable space trade-off kicks in!

Projections v.s. Restrictions for Polynomial Calculus

Projections in [BN11] fail for polynomial calculus and PCR (see [Nor13] for examples)

Use XOR-substitution + random restrictions

- If refutation short $\Rightarrow$ restriction kills all high-degree monomials
- If also monomial space small $\Rightarrow$ get small variable space
- But then size-variable space trade-off kicks in!

Obtain similar trade-offs as for resolution but with some loss in parameters [BNT13]

No unconditional space lower bounds — inherent limitation due to random restriction argument

Going Beyond Linear Space...

- All formulas in [BN11] refutable in linear size (and hence simultaneously also in linear space)
- Could it be that optimal proof size sometimes requires larger than linear space? (Which is worst-case space upper bound)

Going Beyond Linear Space...

- All formulas in [BN11] refutable in linear size (and hence simultaneously also in linear space)
- Could it be that optimal proof size sometimes requires larger than linear space? (Which is worst-case space upper bound)
- **Yes!** For Tseitin formulas over "long, skinny grids" [BBI12, BNT13]
- Holds even for PCR [BNT13]

Going Beyond Linear Space...

- All formulas in [BN11] refutable in linear size (and hence simultaneously also in linear space)
- Could it be that optimal proof size sometimes requires larger than linear space? (Which is worst-case space upper bound)
- **Yes!** For Tseitin formulas over "long, skinny grids" [BBI12, BNT13]
- Holds even for PCR [BNT13]
- Superlinear space regime more challenging than sublinear
- Trade-offs not as dramatic as in [BN11] so in that sense results are incomparable

Going Beyond Linear Space...

- All formulas in [BN11] refutable in linear size (and hence simultaneously also in linear space)
- Could it be that optimal proof size sometimes requires larger than linear space? (Which is worst-case space upper bound)
- **Yes!** For Tseitin formulas over "long, skinny grids" [BBI12, BNT13]
- Holds even for PCR [BNT13]
- Superlinear space regime more challenging than sublinear
- Trade-offs not as dramatic as in [BN11] so in that sense results are incomparable
- Don't have time to go into any details — topic for a separate talk, probably...

Some Open Problems for Resolution

Resolution arguably fairly well-understood by now, but several good open questions remain

For instance:

- Can we get (much) sharper trade-offs for superlinear space than in [BBI12, BNT13]?
- Are there trade-offs between proof size and proof width? Or can both measures be minimized simultaneously?

Some Open Problems for Polynomial Calculus/PCR

Long list of open problems — mentioned in this talk:

- Show that PCR is strictly stronger than resolution w.r.t. space
- Prove PCR space lower bounds for
  - Tseitin on any expander
  - ordering principle formulas
  - random 3-CNFs
  - Or any 3-CNF, really...
- Is it true for PCR that space $\geq$ degree $+ \mathcal{O}(1)$?

Definition of Cutting Planes [CCT87]

Clauses interpreted as linear inequalities over the reals with integer coefficients

Example: $x \lor y \lor \overline{z}$ gets translated to $x + y + (1 - z) \geq 1$

Derivation rules

Variable axioms $\dfrac{}{0 \leq x \leq 1}$

Multiplication $\dfrac{\sum a_i x_i \geq A}{\sum c a_i x_i \geq cA}$

Addition $\dfrac{\sum a_i x_i \geq A \quad \sum b_i x_i \geq B}{\sum (a_i + b_i) x_i \geq A + B}$

Division $\dfrac{\sum c a_i x_i \geq A}{\sum a_i x_i \geq \lceil A/c \rceil}$

Goal: Derive $0 \geq 1 \Leftrightarrow$ formula unsatisfiable

Size, Length and Space

Length = total # lines/inequalities in refutation

Size = sum also size of coefficients

Space = max # lines in memory during refutation

No (useful) analogue of width/degree

Size, Length and Space

Length = total # lines/inequalities in refutation

Size = sum also size of coefficients

Space = max # lines in memory during refutation

No (useful) analogue of width/degree

Cutting planes

- simulates resolution efficiently w.r.t. length/size and space simultaneously
- is strictly stronger w.r.t. length/size — can refute PHP efficiently [CCT87]

Open Problem

Show cutting planes strictly stronger than resolution w.r.t. space

Hard Formulas w.r.t Cutting Planes Space?

No space lower bounds known except conditional ones

All short cutting planes refutations of

- Tseitin formulas on expanders require large space [GP14]
  (But such short refutations probably don't exist anyway)
- (some) pebbling formulas require large space [HN12, GP14]
  (and such short refutations do exist; hard to see how exponential length could help bring down space)

Above results obtained via communication complexity

No (true) length-space trade-off results known
Although results above can also be phrased as trade-offs

Summing up

- Survey of space complexity and size-space trade-offs
- Focus on resolution and polynomial calculus/PCR
- Resolution fairly well understood
- Polynomial calculus less so — several nice open problems
- And cutting planes almost not at all understood!

Summing up

- Survey of space complexity and size-space trade-offs
- Focus on resolution and polynomial calculus/PCR
- Resolution fairly well understood
- Polynomial calculus less so — several nice open problems
- And cutting planes almost not at all understood!

Thank you for your attention!

References I

[ABRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version appeared in *STOC '00*.

[AD03] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. In *Proceedings of the 18th IEEE Annual Conference on Computational Complexity (CCC '03)*, pages 239–247, July 2003. Journal version in [AD08].

[AD08] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, May 2008. Preliminary version appeared in *CCC '03*.

[BBI12] Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 213–232, May 2012.

References II

[Ben02] Eli Ben-Sasson. Size space tradeoffs for resolution. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 457–464, May 2002. Journal version in [Ben09].

[Ben09] Eli Ben-Sasson. Size-space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, May 2009. Preliminary version appeared in *STOC '02*.

[BG03] Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003. Preliminary version appeared in *CCC '01*.

[BG13] Ilario Bonacina and Nicola Galesi. Pseudo-partitions, transversality and locality: A combinatorial characterization for the space measure in algebraic proof systems. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS '13)*, pages 455–472, January 2013.

References III

[BN08] Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 709–718, October 2008.

[BN11] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, January 2011.

[BNT13] Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.

[CCT87] William Cook, Collette Rene Coullard, and Gyorgy Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, November 1987.

References IV

[CEI96] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 174–183, May 1996.

[CS76] Stephen A. Cook and Ravi Sethi. Storage requirements for deterministic polynomial time recognizable languages. *Journal of Computer and System Sciences*, 13(1):25–37, 1976. Preliminary version appeared in *STOC '74*.

[ET01] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Preliminary versions of these results appeared in *STACS '99* and *CSL '99*.

[FLM+13] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (extended abstract). In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 437–448. Springer, July 2013.

References V

[FLN+12] Yuval Filmus, Massimo Lauria, Jakob Nordström, Neil Thapen, and Noga Ron-Zewi. Space complexity in polynomial calculus (extended abstract). In *Proceedings of the 27th Annual IEEE Conference on Computational Complexity (CCC '12)*, pages 334–344, June 2012.

[GP14] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC '14)*, pages 847–856, May 2014.

[HN12] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity (extended abstract). In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 233–248, May 2012.

[Nor09] Jakob Nordström. A simplified way of proving trade-off results for resolution. *Information Processing Letters*, 109(18):1030–1035, August 2009. Preliminary version appeared in ECCC report TR07-114, 2007.

References VI

[Nor12] Jakob Nordström. On the relative strength of pebbling and resolution. *ACM Transactions on Computational Logic*, 13(2):16:1–16:43, April 2012. Preliminary version appeared in *CCC '10*.

[Nor13] Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

[Rii93] Søren Riis. *Independence in Bounded Arithmetic*. PhD thesis, University of Oxford, 1993.