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
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
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)
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** $C = a_1 \lor \cdots \lor a_k$: disjunction of literals
  (Consider as sets, so no repetitions and order irrelevant)

- **CNF formula** $F = C_1 \land \cdots \land C_m$: 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

Refutation ends when empty clause $\bot$ 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. $\bot$ Res(7, 8)
The Resolution Proof System

Goal: refute unsatisfiable CNF

Start with clauses of formula (axioms)

Derive new clauses by resolution rule

Refutation ends when empty clause $\bot$ 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

\[
\frac{C \lor x}{C \lor D} \quad \frac{D \lor \overline{x}}{C \lor \overline{x}}
\]

Refutation ends when empty clause $\bot$ 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)

**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. $\bot$ Res$(7, 8)$
**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 ...
**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 $\exp(O(N))$

- Space always at most $N + O(1)$

- 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 \((C_0, \ldots, C_\tau)\)
where \(C_t\) obtained from \(C_{t-1}\) by:

- **Download** \(C_t = C_{t-1} \cup \{C\}\) for axiom clause \(C \in F\)
- **Inference** \(C_t = C_{t-1} \cup \{D\}\) inferred by resolution on clauses in \(C_{t-1}\)
- **Erasure** \(C_t = C_{t-1} \setminus \{D\}\) for some \(D \in C_{t-1}\)
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 \((\mathcal{C}_0, \ldots, \mathcal{C}_\tau)\) where \(\mathcal{C}_t\) obtained from \(\mathcal{C}_{t-1}\) by:

\[
\begin{align*}
\text{Download} & \quad \mathcal{C}_t = \mathcal{C}_{t-1} \cup \{C\} \quad \text{for axiom clause } C \in F \\
\text{Inference} & \quad \mathcal{C}_t = \mathcal{C}_{t-1} \cup \{D\} \quad \text{inferred by resolution on clauses in } \mathcal{C}_{t-1} \\
\text{Erasure} & \quad \mathcal{C}_t = \mathcal{C}_{t-1} \setminus \{D\} \quad \text{for some } D \in \mathcal{C}_{t-1}
\end{align*}
\]

Size = \# download & inference steps
Space = \(\max_{0 \leq t \leq \tau} \{|\mathcal{C}_t|\}\)
Space Lower Bound as Two-Person Game

\( F \) requires space \( s \) \iff all \( C_t \) derived from \( F \) in space \( < s \) satisfiable
Space Lower Bound as Two-Person Game

\[ F \text{ requires space } s \iff \text{ all } C_t \text{ derived from } F \text{ in space } < s \text{ satisfiable} \]

Given derivation \((C_0, \ldots, C_\tau)\), construct \(\alpha_t\) satisfying \(C_t\)
Space Lower Bound as Two-Person Game

$F$ requires space $s \iff$ all $C_t$ derived from $F$ in space $< s$ satisfiable

Given derivation $(C_0, \ldots, C_T)$, construct $\alpha_t$ satisfying $C_t$

**Space game**

**Download** Pick $\alpha_t$ of size $\leq |C_t|$

**Inference** Do nothing

**Erasure** Pick $\alpha_t$ of size $\leq |C_t|$
Space Lower Bound as Two-Person Game

\( F \) requires space \( s \) \iff all \( C_t \) derived from \( F \) in space \( < s \) satisfiable

Given derivation \((C_0, \ldots, C_\tau)\), construct \( \alpha_t \) satisfying \( C_t \)

<table>
<thead>
<tr>
<th>Action</th>
<th>Space game</th>
<th>Lower bound game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download</td>
<td>Pick ( \alpha_t ) of size ( \leq</td>
<td>C_t</td>
</tr>
<tr>
<td>Inference</td>
<td>Do nothing</td>
<td>Do nothing</td>
</tr>
<tr>
<td>Erasure</td>
<td>Pick ( \alpha_t ) of size ( \leq</td>
<td>C_t</td>
</tr>
</tbody>
</table>
Space Complexity
Size-Space Trade-offs
Open Problems

Space Lower Bound as Two-Person Game

\( F \) requires space \( s \iff \) all \( C_t \) derived from \( F \) in space \(< s \) satisfiable

Given derivation \((C_0, \ldots, C_\tau)\), construct \( \alpha_t \) satisfying \( C_t \)

<table>
<thead>
<tr>
<th>Space game</th>
<th>Lower bound game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download</td>
<td>Pick ( \alpha_t ) of size ( \leq</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Inference</td>
<td>Do nothing</td>
</tr>
<tr>
<td>Erasure</td>
<td>Pick ( \alpha_t ) of size ( \leq</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Space game exactly characterizes space (but hard to play)

Restricted lower bound game: can construct \( \alpha_t \) inductively (but no guarantee this will work)
Hard to get a handle on structure of derived configuration $C_t$.

Construct auxiliary configuration $D_t$ (view $\alpha_t$ as 1-CNF) that is easier to understand but still gives information about $C_t$:

1. $D_t \implies C_t$ (i.e., $D_t$ is "stronger" than $C_t$).
2. $D_t$ is satisfiable (so, in particular, $C_t$ also satisfiable).
3. $|D_t| \leq |C_t|$ (all we know about space of $C_t$).
4. At derivation step $C_t 
\rightarrow C_t$, can do a local update $D_t 
\rightarrow D_t$ if $|D_t| \text{ small enough (i.e., less than } s)$.

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

Two observations:

"On the safe side" of adversary ($D_t$ stronger than $C_t$).

History-dependent (can get different $D_t$ for same $C_t$).
General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $C_t$

Construct auxiliary configuration $D_t$ (view $\alpha_t$ as 1-CNF) that is easier to understand but still gives information about $C_t$:

1. $D_t$ implies $C_t$ (i.e., $D_t$ “stronger” than $C_t$)
General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $C_t$

Construct auxiliary configuration $D_t$ (view $\alpha_t$ as 1-CNF) that is easier to understand but still gives information about $C_t$:

1. $D_t$ implies $C_t$ (i.e., $D_t$ “stronger” than $C_t$)
2. $D_t$ is satisfiable (so, in particular, $C_t$ also satisfiable)
General Proof Strategy for Space Lower Bound

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

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

1. $\mathcal{D}_t$ implies $\mathcal{C}_t$ (i.e., $\mathcal{D}_t$ “stronger” than $\mathcal{C}_t$)
2. $\mathcal{D}_t$ is satisfiable (so, in particular, $\mathcal{C}_t$ also satisfiable)
3. $|\mathcal{D}_t| \leq |\mathcal{C}_t|$ (all we know about space of $\mathcal{C}_t$)
General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $C_t$

Construct auxiliary configuration $D_t$ (view $\alpha_t$ as 1-CNF) that is easier to understand but still gives information about $C_t$:

1. $D_t$ implies $C_t$ (i.e., $D_t$ “stronger” than $C_t$)
2. $D_t$ is satisfiable (so, in particular, $C_t$ also satisfiable)
3. $|D_t| \leq |C_t|$ (all we know about space of $C_t$)
4. At derivation step $C_{t-1} \rightsquigarrow C_t$, can do a local update $D_{t-1} \rightsquigarrow D_t$ if $|D_{t-1}|$ small enough (i.e., less than $s$)
General Proof Strategy for Space Lower Bound

Hard to get a handle on structure of derived configuration $C_t$

Construct auxiliary configuration $D_t$ (view $\alpha_t$ as 1-CNF) that is easier to understand but still gives information about $C_t$:

1. $D_t$ implies $C_t$ (i.e., $D_t$ “stronger” than $C_t$)
2. $D_t$ is satisfiable (so, in particular, $C_t$ also satisfiable)
3. $|D_t| \leq |C_t|$ (all we know about space of $C_t$)
4. At derivation step $C_{t-1} \leadsto C_t$, can do a local update $D_{t-1} \leadsto D_t$ if $|D_{t-1}|$ 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 ($D_t$ stronger than $C_t$)
- History-dependent (can get different $D_t$ for same $C_t$)
Random $k$-CNF formulas

$\Delta n$ 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
**Random \( k \)-CNF formulas**

\( \Delta n \) 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 \( G(U \cup V, E) \)
- \( U \) = set of clauses; \( V \) = set of variables
- Edge \((C, x)\) if variable \( x \) occurs in \( C \) [ignore sign of literal]
Random $k$-CNF formulas

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

$(d, \delta, s)$-bipartite expander

- Bipartite graph $G(U \cup V, E)$ with left degree $d$
- Every $A \subseteq U$ s.t. $|A| \leq s$ has neighbourhood $|N_G(A)| \geq \delta|A|$
Theorem ([BG03])

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

Proof sketch.

Given small-space derivation $(C_0, C_1, C_2, \ldots)$ from $F$, inductively construct 1-CNF $D_t$ implying $C_t$ and satisfying $|D_t| \leq |C_t|$: 

1. **Download of $C \in F$:** Since $G(F)$ has expansion $1 + \epsilon$, can find variable in $C$ not in $D_{t-1}$ [needs an argument, of course]
2. **Inference:** Set $D_t = D_{t-1}$
3. **Erasure:** Pick $D_t \subseteq D_{t-1}$ of size $|D_t| \leq |C_t|$ implying $C_t$
Taking Care of Erasures by Locality Lemma

Lemma (Locality lemma for resolution)

Suppose $D$ 1-CNF; $C$ clause configuration; $D$ implies $C$

Then $\exists$ 1-CNF $D'$ of size $|D'| \leq |C|$ s.t. $D'$ implies $C$
Taking Care of Erasures by Locality Lemma

**Lemma (Locality lemma for resolution)**

Suppose $D$ 1-CNF; $C$ clause configuration; $D$ implies $C$

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

**Proof.**

Consider bipartite graph with

- clauses $C \in C$ on left; unit clauses $\in D$ on right
- edge between $C$ and $D$ if $D \models C$ (share a literal)
Lemma (Locality lemma for resolution)

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

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

Proof.

Consider bipartite graph with

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

For every $C \in \mathcal{C}$, pick one neighbour $D \in \mathcal{D}$ (must exist) to form 1-CNF $\mathcal{D}'$
Lemma (Locality lemma for resolution)

Suppose $D$ 1-CNF; $C$ clause configuration; $D$ implies $C$

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

Proof.

Consider bipartite graph with

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

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

Then by construction:

- $|D'| \leq |C|
- $D' \models C$
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 $\text{space} \geq \text{width} + O(1)$
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 $\text{space} \geq \text{width} + O(1)$*

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 $x, \bar{x}, y, \bar{y}, z, \bar{z}, \ldots$ (where $x$ and $\bar{x}$ distinct variables)

**Example:** $x \lor y \lor \bar{z}$ gets translated to $xy\bar{z} = 0$

Think of $0 \equiv true$ and $1 \equiv false$
Polynomial Calculus (or Actually PCR)

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

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

**Example:** \(x \lor y \lor \bar{z}\) gets translated to \(xy\bar{z} = 0\)

Think of \(0 \equiv \text{true}\) and \(1 \equiv \text{false}\)

### Derivation rules

<table>
<thead>
<tr>
<th><strong>Boolean axioms</strong></th>
<th>(x^2 - x = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Negation</strong></td>
<td>(x + \bar{x} = 1)</td>
</tr>
<tr>
<td><strong>Linear combination</strong></td>
<td>(p = 0) (q = 0)</td>
</tr>
<tr>
<td>(\alpha p + \beta q = 0)</td>
<td></td>
</tr>
<tr>
<td><strong>Multiplication</strong></td>
<td>(p = 0)</td>
</tr>
<tr>
<td>(xp = 0)</td>
<td></td>
</tr>
</tbody>
</table>

**Goal:** Derive \(1 = 0 \iff\) no common root \(\iff\) 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)
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 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 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 Bound 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 \((P_0, P_1, P_2, \ldots)\) in small space

Want to construct “auxiliary configurations” \(D_0, D_1, D_2, \ldots\) s.t.

- \(D_t\) highly structured, so easier to understand than \(P_t\)
- but still gives information about \(P_t\)
What We Want (Recap of Lower Bound Proof Strategy)

Given PCR derivation \((P_0, P_1, P_2, \ldots)\) in small space

Want to construct “auxiliary configurations” \(D_0, D_1, D_2, \ldots\) s.t.

- \(D_t\) highly structured, so easier to understand than \(P_t\)
- but still gives information about \(P_t\)

Maintain invariants for \(D_t\):

1. \(D_t\) implies \(P_t\) (i.e., \(D_t\) “stronger” than \(P_t\))
2. \(D_t\) is satisfiable (so, in particular, \(P_t\) also satisfiable)
3. space of \(D_t\) \(\leq\) space of \(P_t\) (all we know about space of \(P_t\))
4. For \(P_{t-1} \leadsto P_t\), can do update \(D_{t-1} \leadsto D_t\) if \(D_{t-1}\) small
What We Want (Recap of Lower Bound Proof Strategy)

Given PCR derivation \((P_0, P_1, P_2, \ldots)\) in small space

Want to construct “auxiliary configurations" \(D_0, D_1, D_2, \ldots\) s.t.

- \(D_t\) highly structured, so easier to understand than \(P_t\)
- but still gives information about \(P_t\)

Maintain invariants for \(D_t\):

1. \(D_t\) implies \(P_t\) (i.e., \(D_t\) “stronger” than \(P_t\))
2. \(D_t\) is satisfiable (so, in particular, \(P_t\) also satisfiable)
3. space of \(D_t\) \(\leq\) space of \(P_t\) (all we know about space of \(P_t\))
4. For \(P_{t-1} \rightsquigarrow P_t\), can do update \(D_{t-1} \rightsquigarrow D_t\) if \(D_{t-1}\) 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]
Theorem ([BG13])

If $F$ is random $k$-CNF for $k \geq 4$ over $n$ variables with $\Delta n$ clauses then $F$ requires PCR space $\Omega(n)$ almost surely.
Theorem ([BG13])

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

Proof approach:

- Structured auxiliary configurations: $2$-CNFs $\mathbb{D}_t = A_t \land B_t$
Theorem ([BG13])

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

Proof approach:

- Structured auxiliary configurations: $2$-CNFs $\mathcal{D}_t = \mathcal{A}_t \land \mathcal{B}_t$
- Each $\mathcal{A} \in \mathcal{A}_t$ is subclause of axiom $C \in F$
Theorem ([BG13])

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

Proof approach:

- Structured auxiliary configurations: 2-CNFs $D_t = A_t \land B_t$
- Each $A \in A_t$ is subclause of axiom $C \in F$
- No distinct $A, A' \in A_t$ 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 \( \Delta n \) clauses then \( F \) requires PCR space \( \Omega(n) \) almost surely

Proof approach:

- Structured auxiliary configurations: 2-CNFs \( \mathbb{D}_t = \mathbb{A}_t \land \mathbb{B}_t \)
- Each \( A \in \mathbb{A}_t \) is subclause of axiom \( C \in F \)
- No distinct \( A, A' \in \mathbb{A}_t \) share any variables
- Every \( B \in \mathbb{B}_t \) associated to two unique \( A_B, A'_B \in \mathbb{A}_t \)
Theorem ([BG13])

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

Proof approach:

- Structured auxiliary configurations: 2-CNFs $D_t = A_t \land B_t$
- Each $A \in A_t$ is subclause of axiom $C \in F$
- No distinct $A, A' \in A_t$ share any variables
- Every $B \in B_t$ associated to two unique $A_B, A'_B \in A_t$
- $B$ contains one variable from $A_B$ and one variable from $A'_B$
Theorem ([BG13])

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

Proof approach:

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

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

- $D_t = A_t \land B_t$ structured auxiliary configuration
- $D_t$ implies $P_t$
- $|D_t| \leq 6 \cdot (\# \text{[distinct] monomials in } P_t)$

Proof is by case analysis over derivation step
Inductive Proof: Invariants and Inference

**Proof invariants:**

- $D_t = A_t \land B_t$ structured auxiliary configuration
- $D_t$ implies $P_t$
- $|D_t| \leq 6 \cdot (\# \text{[distinct] monomials in } P_t)$

Proof is by case analysis over derivation step

1. **Inference** $P_t = P_t \cup \{Q\}$ for polynomial $Q$ derived from $P_{t-1}$
   - Set $D_t := D_{t-1}$
   - $D_t = D_{t-1}$ implies $Q$ by soundness
   - Space of $D_t$ stays the same
   - Space of $P_t$ goes up
2. **Download** $P_t = P_t \cup \{C\}$ for $C \in F$

- For simplicity, assume extra download of $C' \in F$
- Without loss of generality: can then immediately erase $C'$
2. **Download** $P_t = P_t \cup \{C\}$ 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 $B$
2. **Download** $P_t = P_t \cup \{C\}$ 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 $B$
- $A_t := A_{t-1} \cup \{A, A'\}$
- $B_t := B_{t-1} \cup \{B\}$
- Space of $D_t = A_t \wedge B_t$ up by 3
- Space of $P_t$ up by 1
3. **Erasure** \( P_t = P_{t-1} \setminus \{Q\} \) for \( Q \in P_{t-1} \)

- Know \( D_{t-1} \) implies \( P_t \subseteq P_{t-1} \)
- But \(|D_{t-1}|\) might be far too large
- Need to find smaller auxiliary configuration that implies \( P_t \)
  (Was very easy for resolution; now not clear at all what to do)
Inductive Proof: Erasure

3. **Erasure** $P_t = P_{t-1} \setminus \{Q\}$ for $Q \in P_{t-1}$

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

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

Suppose

- $D = A \land B$ structured auxiliary configuration
- $P$ PCR-configuration
- $D$ implies $P$

Then

$\exists D^* = A^* \land B^*$ with $|D^*| \leq 6 \cdot (\# \text{ monomials in } P)$ s.t. $D^*$ implies $P$
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 $G = (U \cup V, E)$
- $U = \text{distinct monomials } M \text{ in } \mathbb{P}$

$G = (U \cup V, E)$

$m_1 \bigcirc$

$m_2 \bigcirc$

$m_3 \bigcirc$

$m_4 \bigcirc$

$m_5 \bigcirc$
Proof sketch for Locality Lemma for PCR (1/4)

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

Let $\Gamma \subseteq M$ set of maximal size such that $|N(\Gamma)| \leq 2 \cdot |\Gamma|$

Assume $\Gamma \neq M$ (else done)

For all $S \subseteq M \setminus \Gamma$ by maximality $|N(S) \setminus N(\Gamma)| > 2 \cdot |S|$.

⇒ There exists a matching of each $m \in M \setminus \Gamma$ to 2 distinct $B', B'' \in N(\Gamma)$.

(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 $P$
- $V =$ clauses in $B$
- Edge between $m \in M$ and $B \in B$
  - if $\exists$ common variable

Diagram:

- $m_1$ connected to $B_1$, $B_2$, $B_3$
- $m_2$ connected to $B_4$, $B_5$, $B_6$
- $m_3$ connected to $B_7$, $B_8$, $B_9$
- $m_4$ connected to $B_{10}$, $B_{11}$, $B_{12}$
- $m_5$ connected to $B_{13}$
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 $P$
- $V =$ clauses in $B$
- Edge between $m \in M$ and $B \in B$ 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 $P$
- $V =$ clauses in $B$
- Edge between $m \in M$ and $B \in B$
  
  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 $B', B'' \in B \setminus N(\Gamma)$
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 $G = (U \cup V, E)$
- $U = \text{distinct monomials } M \text{ in } \mathbb{P}$
- $V = \text{clauses in } \mathcal{B}$
- 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 $P$
- $V =$ clauses in $B$
- Edge between $m \in M$ and $B \in B$ 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 $B', B'' \in B \setminus N(\Gamma)$
- (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 $B' = \overline{x} \lor \overline{y}$ and $B'' = \overline{z} \lor \overline{w}$.
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)
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

1. $B' \leftrightarrow A'_1 = x \lor x'$ and $A'_2 = y \lor y'$
2. $B'' \leftrightarrow A''_1 = z \lor z'$ and $A''_2 = w \lor w'$

\[m \quad m_1 \quad m_2 \quad m_3 \quad m_4\]

\[B_1 \quad B_2 \quad B_3 \quad B_4 \quad B_5 \quad B_6 \quad B_7 \quad B_8 \quad B_{11} \quad B_{12} \quad B_{13}\]
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 $A''_2 = w \lor w'$

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

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

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

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

1. $D^*$ structured auxiliary configuration
   - Straightforward to verify

2. $D^*$ has the right size
   - OK, since $|D^*| \leq 6 \cdot |M| \leq 6 \cdot (\# \text{ monomials in } P)$

3. $D^*$ implies $P$
   - Perhaps a priori not so clear. . .

Prove implication in slightly roundabout way:

Given any $\beta$ satisfying $D^*$, find $\alpha$ such that $P(\alpha) = P(\beta)$

$\alpha$ satisfies $D$
Proof sketch for Locality Lemma for PCR (3/4)

Need to prove three things:

1. $D^*$ structured auxiliary configuration
   Straightforward to verify
Need to prove three things:

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

2. $\mathcal{D}^*$ has the right size
   OK, since $|\mathcal{D}^*| \leq 6 \cdot |M| \leq 6 \cdot (\# \text{ monomials in } P)$
Need to prove three things:

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

2. $\mathcal{D}^*$ has the right size
   OK, since $|\mathcal{D}^*| \leq 6 \cdot |M| \leq 6 \cdot (\# \text{ monomials in } \mathbb{P})$

3. $\mathcal{D}^*$ implies $\mathbb{P}$
   Perhaps a priori not so clear...
Proof sketch for Locality Lemma for PCR (3/4)

Need to prove three things:

1. \( \mathcal{D}^* \) structured auxiliary configuration
   Straightforward to verify

2. \( \mathcal{D}^* \) has the right size
   OK, since \( |\mathcal{D}^*| \leq 6 \cdot |M| \leq 6 \cdot (\# \text{ monomials in } \mathcal{P}) \)

3. \( \mathcal{D}^* \) implies \( \mathcal{P} \)
   Perhaps a priori not so clear . . .

Prove implication in slightly roundabout way:
Given any \( \beta \) satisfying \( \mathcal{D}^* \), find \( \alpha \) such that

- \( \mathcal{P}(\alpha) = \mathcal{P}(\beta) \)
- \( \alpha \) satisfies \( \mathcal{D} \)
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 $D^*$ [satisfied by $\beta$]
- $B^* = x \lor z$, $A^*_1 = x \lor x'$, $A^*_2 = z \lor z'$
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 \( D^* \) [satisfied by \( \beta \)]

- \( B^* = x \lor z, A_1^* = x \lor x', A_2^* = z \lor z' \)

Old clauses in \( D \) [to be satisfied by \( \alpha \)] are:

- \( B' = \overline{x} \lor \overline{y} \leftrightarrow A_1' = x \lor x', A_2' = y \lor y' \)
- \( B'' = \overline{z} \lor \overline{w} \leftrightarrow A_1'' = z \lor z', A_2'' = w \lor w' \)
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 \( D^* \) [satisfied by \( \beta \)]

- \( B^* = x \lor z, A_1^* = x \lor x', A_2^* = z \lor z' \)

Old clauses in \( D \) [to be satisfied by \( \alpha \)] are:

- \( B' = \overline{x} \lor \overline{y} \leftrightarrow A_1' = x \lor x', A_2' = y \lor y' \)

- \( B'' = \overline{z} \lor \overline{w} \leftrightarrow A_1'' = z \lor z', A_2'' = w \lor w' \)

Let \( \alpha = \beta \) except that for \( m \in M \setminus \Gamma \) we set

\( y = w = \text{false} \) and \( x' = y' = z' = w' = \text{true} \)
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 $\beta$]
- $B^* = x \lor z$, $A_1^* = x \lor x'$, $A_2^* = z \lor z'$

Old clauses in $\mathbb{D}$ [to be satisfied by $\alpha$] are:
- $B' = \overline{x} \lor \overline{y} \leftrightarrow A_1' = x \lor x'$, $A_2' = y \lor y'$
- $B'' = \overline{z} \lor \overline{w} \leftrightarrow A_1'' = z \lor z'$, $A_2'' = w \lor w'$

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

- $\alpha(m) = \beta(m)$ 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 $\mathbb{D}^*$]
- $\alpha$ satisfies $\mathbb{D}$ and hence $P$
- But then $\beta$ must also satisfy $P$, Q.E.D.
Another Intriguing Problem: Space vs. Degree

Open Problem (analogue of [AD08])

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

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 $\text{space} \geq \text{degree} + O(1)$?

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 $O(1)$
- require space $\Omega(N)$
- but separating formulas depend on characteristic of field
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
∃ constant space refutation ⇒ ∃ polynomial size refutation [AD03]
∃ constant space refutation ⇒ ∃ 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
∃ constant space refutation ⇒ ∃ 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 \( \mathcal{O}(N/\log N) \)
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 $N$
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

And an open problem:

Seems likely that \( 3\sqrt[3]{N} \) above should be possible to improve to \( \sqrt{N} \), but don’t know how to prove this...
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 \( \approx 3\sqrt[3]{N} \) such that
- any refutation in space \( \ll 3\sqrt[3]{N} \) 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 \( g(N) \)
- refutable in size linear in \( N \) and space \( \approx \sqrt[3]{N} \)
- 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. \( \overline{u} \lor \overline{v} \lor x \)
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. \( \overline{u} \lor \overline{v} \lor x \)
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. $\overline{z}$

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):

\[
\overline{x} \lor y
\]

\[
\downarrow
\]

\[
\neg (x_1 \oplus x_2) \lor (y_1 \oplus y_2)
\]

\[
\downarrow
\]

\[
(x_1 \lor \overline{x}_2 \lor y_1 \lor y_2)
\]

\[
\land (x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2)
\]

\[
\land (\overline{x}_1 \lor x_2 \lor y_1 \lor y_2)
\]

\[
\land (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2)
\]
Let $F[⊕]$ denote formula with XOR $x_1 ⊕ x_2$ substituted for $x$

Obvious approach for refuting $F[⊕]$: 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 $F$
Key Technical Result: Substitution Theorem

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

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

\[
\begin{array}{c}
  x \\
  \overline{x} \lor y \\
  y
\end{array}
\]
Let $F[⊕]$ denote formula with XOR $x_1 ⊕ x_2$ substituted for $x$.

Obvious approach for refuting $F[⊕]$: mimic refutation of $F$.

\[
\begin{array}{c}
  x \\
  \overline{x} \lor y \\
  y \\
\end{array}
\]

\[
\begin{array}{c}
  x_1 \lor x_2 \\
  \overline{x}_1 \lor \overline{x}_2 \\
\end{array}
\]
Let $F[\oplus]$ denote formula with $\text{XOR } x_1 \oplus x_2$ substituted for $x$

Obvious approach for refuting $F[\oplus]$: mimic refutation of $F$
Key Technical Result: Substitution Theorem

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

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

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

$$
\begin{align*}
x_1 \lor x_2 \\
\overline{x}_1 \lor \overline{x}_2 \\
x_1 \lor \overline{x}_2 \lor y_1 \lor y_2 \\
x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2 \\
\overline{x}_1 \lor x_2 \lor y_1 \lor y_2 \\
\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2 \\
y_1 \lor y_2 \\
\overline{y}_1 \lor \overline{y}_2
\end{align*}
$$
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 $F$.

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 $F$

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 $F[⊕]$, extract “shadow refutation” of $F$

<table>
<thead>
<tr>
<th>XOR formula $F[⊕]$</th>
<th>Original formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Sketch of Proof of Substitution Theorem

Given refutation of \( F[⊕] \), extract “shadow refutation” of \( F \)

<table>
<thead>
<tr>
<th>XOR formula ( F[⊕] )</th>
<th>Original formula ( F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. ( \neg(x_1 ⊕ x_2) \lor (y_1 ⊕ y_2) \ldots )</td>
<td></td>
</tr>
</tbody>
</table>
Sketch of Proof of Substitution Theorem

Given refutation of $F[⊕]$, extract “shadow refutation” of $F$

<table>
<thead>
<tr>
<th>XOR formula $F[⊕]$</th>
<th>Original formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. $\neg(x_1 ⊕ x_2) ∨ (y_1 ⊕ y_2)...$</td>
<td>write $\overline{x} ∨ y$ on shadow blackboard</td>
</tr>
</tbody>
</table>
Given refutation of $F[\oplus]$, extract “shadow refutation” of $F$.

<table>
<thead>
<tr>
<th>XOR formula $F[\oplus]$</th>
<th>Original formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2) \ldots$</td>
<td>write $\overline{x} \lor y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations...</td>
<td></td>
</tr>
</tbody>
</table>
Given refutation of $F[\oplus]$, extract “shadow refutation” of $F$

<table>
<thead>
<tr>
<th>XOR formula $F[\oplus]$</th>
<th>Original formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)$...</td>
<td>write $\overline{x} \lor y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations...</td>
<td>can get between corresponding shadow blackboards by legal resolution derivation steps</td>
</tr>
</tbody>
</table>
Sketch of Proof of Substitution Theorem

Given refutation of $F[⊕]$, extract “shadow refutation” of $F$

<table>
<thead>
<tr>
<th>XOR formula $F[⊕]$</th>
<th>Original formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)$...</td>
<td>write $\bar{x} \lor y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations...</td>
<td>can get between corresponding shadow blackboards by legal resolution derivation steps</td>
</tr>
<tr>
<td></td>
<td>Size of shadow blackboard derivation ...</td>
</tr>
</tbody>
</table>
Given refutation of $F[\oplus]$, extract "shadow refutation" of $F$

<table>
<thead>
<tr>
<th>XOR formula $F[\oplus]$</th>
<th>Original formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. $\neg (x_1 \oplus x_2) \lor (y_1 \oplus y_2)$...</td>
<td>write $\overline{x} \lor y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations...</td>
<td>can get between corresponding shadow blackboards by legal resolution derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation size</td>
<td>Size of shadow blackboard derivation ...</td>
</tr>
</tbody>
</table>
Given refutation of $F[⊕]$, extract “shadow refutation” of $F$

<table>
<thead>
<tr>
<th>XOR formula $F[⊕]$</th>
<th>Original formula $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)$...</td>
<td>write $\overline{x} \lor y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations...</td>
<td>can get between corresponding shadow blackboards by legal resolution derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation size</td>
<td>Size of shadow blackboard derivation ...</td>
</tr>
<tr>
<td></td>
<td># variables mentioned on shadow blackboard...</td>
</tr>
</tbody>
</table>
Given refutation of $F[⊕]$, extract “shadow refutation” of $F$

<table>
<thead>
<tr>
<th><strong>XOR formula $F[⊕]$</strong></th>
<th><strong>Original formula $F$</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>If XOR blackboard implies e.g. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)\ldots$</td>
<td>write $\overline{x} \lor y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations...</td>
<td>can get between corresponding shadow blackboards by legal resolution derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation size</td>
<td>Size of shadow blackboard derivation ...</td>
</tr>
<tr>
<td>... is at most $#$ clauses on XOR blackboard</td>
<td>$#$ variables mentioned on shadow blackboard...</td>
</tr>
</tbody>
</table>
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 in [BN11] fail for polynomial calculus and PCR (see [Nor13] for examples)
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 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 \textbf{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 \( \text{space} \geq \text{degree} + O(1) \)?
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 \( x + y + (1 - z) \geq 1 \)

<table>
<thead>
<tr>
<th>Derivation rules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable axioms</strong></td>
</tr>
<tr>
<td><strong>Addition</strong></td>
</tr>
<tr>
<td><strong>Multiplication</strong></td>
</tr>
<tr>
<td><strong>Division</strong></td>
</tr>
</tbody>
</table>

**Goal:** Derive \( 0 \geq 1 \iff \text{formula unsatisfiable} \)
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 II


References V


<table>
<thead>
<tr>
<th>Reference</th>
<th>Author</th>
<th>Title</th>
<th>Journal</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Preliminary version appeared in CCC ’10.</td>
<td></td>
</tr>
</tbody>
</table>