Understanding Space in Proof Complexity: Separations and Trade-offs via Substitutions

Jakob Nordström
jakobn@mit.edu
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Barriers in Computational Complexity
Center for Computational Intractability, Princeton
August 25–29, 2009

Joint work with Eli Ben-Sasson
Resolution: proof system for refuting CNF formulas

Perhaps the most studied system in proof complexity

Basis of current state-of-the-art SAT-solvers (e.g. winners in SAT 2008 competition)

Key resources: time and space

What are the connections between these resources? Time-space correlations? Trade-offs?

Study these questions for more general $k$-DNF resolution proof systems introduced by [Krajíček ’01]
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
- **Term** $T = a_1 \land \cdots \land a_k$: conjunction of literals
- **CNF formula** $F = C_1 \land \cdots \land C_m$: conjunction of clauses
  - $k$-CNF formula: CNF formula with clauses of size $\leq k$
- **DNF formula** $D = T_1 \lor \cdots \lor T_m$: disjunction of terms
  - $k$-DNF formula: DNF formula with terms of size $\leq k$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us
Example \( k \)-DNF Resolution Refutation \((k = 2)\)

Can write down axioms, infer new formulas, and erase used formulas

1. \(x\)
2. \(\overline{x} \lor y\)
3. \(\overline{y} \lor z\)
4. \(\overline{z}\)

Rules:
- Infer new formulas only from formulas currently on board
- Only \(k\)-DNF formulas can appear on board (for \(k\) fixed)
- Details about derivation rules won’t matter for us
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Write down axiom 1: $x$

Rules:

- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Write down axiom 1: $x$
Write down axiom 3: $\overline{y} \lor z$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

1. $x$
2. $\neg x \lor y$
3. $\neg y \lor z$
4. $\neg z$

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Write down axiom 1: $x$
Write down axiom 3: $\neg y \lor z$
Combine $x$ and $\neg y \lor z$ to get $(x \land \neg y) \lor z$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Write down axiom 1: $x$
Write down axiom 3: $\overline{y} \lor z$
Combine $x$ and $\overline{y} \lor z$ to get $(x \land \overline{y}) \lor z$
Example \textit{k-DNF Resolution Refutation (}k = 2\textit{)}

Can \textit{write down axioms, infer new formulas, and erase used formulas}

1. \( x \)
2. \( \overline{x} \lor y \)
3. \( y \lor z \)
4. \( z \)

Rules:
- Infer new formulas only from formulas \textit{currently on board}
- \textit{Only k-DNF formulas} can appear on board (for \( k \) fixed)
- Details about derivation rules won’t matter for us

Write down axiom 1: \( x \)
Write down axiom 3: \( \overline{y} \lor z \)
Combine \( x \) and \( \overline{y} \lor z \) to get \( (x \land \overline{y}) \lor z \)
Erase the line \( x \)
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Write down axiom 1: $x$
Write down axiom 3: $\overline{y} \lor z$
Combine $x$ and $\overline{y} \lor z$ to get $(x \land \overline{y}) \lor z$
Erase the line $x$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Write down axiom 3: $\overline{y} \lor z$
Combine $x$ and $\overline{y} \lor z$ to get $(x \land \overline{y}) \lor z$
Erase the line $x$
Erase the line $\overline{y} \lor z$

$\overline{y} \lor z$
$(x \land \overline{y}) \lor z$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

1. $x$
2. $\neg x \lor y$
3. $\neg y \lor z$
4. $\neg z$

Rules:

- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Write down axiom 3: $\neg y \lor z$

Combine $x$ and $\neg y \lor z$

to get $(x \land \neg y) \lor z$

Erase the line $x$

Erase the line $\neg y \lor z$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Combine $x$ and $\overline{y} \lor z$ to get $(x \land \overline{y}) \lor z$
Erase the line $x$
Erase the line $\overline{y} \lor z$
Write down axiom 2: $\overline{x} \lor y$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Erase the line $x$
Erase the line $\overline{y} \lor z$
Write down axiom 2: $\overline{x} \lor y$
Infer $z$ from $\overline{x} \lor y$ and $(x \land \overline{y}) \lor z$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Erase the line $x$
Erase the line $\overline{y} \lor z$
Write down axiom 2: $\overline{x} \lor y$
Infer $z$ from $\overline{x} \lor y$ and $(x \land \overline{y}) \lor z$
Example \( k \)-DNF Resolution Refutation (\( k = 2 \))

Can write down axioms, infer new formulas, and erase used formulas

1. \( x \)
2. \( \overline{x} \lor y \)
3. \( \overline{y} \lor z \)
4. \( \overline{z} \)

Rules:
- Infer new formulas only from formulas currently on board
- Only \( k \)-DNF formulas can appear on board (for \( k \) fixed)
- Details about derivation rules won’t matter for us

\[
(x \land \overline{y}) \lor z \\
\overline{x} \lor y \\
z
\]

Erase the line \( \overline{y} \lor z \)
Write down axiom 2: \( \overline{x} \lor y \)
Infer \( z \) from
\[
\overline{x} \lor y \text{ and } (x \land \overline{y}) \lor z \\
\]
Erase the line \( (x \land \overline{y}) \lor z \)
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Erase the line $\overline{y} \lor z$

Write down axiom 2: $\overline{x} \lor y$

Infer $z$ from $\overline{x} \lor y$ and $(x \land \overline{y}) \lor z$

Erase the line $(x \land \overline{y}) \lor z$
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:

- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Write down axiom 2: $\overline{x} \lor y$
Infer $z$ from
$\overline{x} \lor y$ and $(x \land \overline{y}) \lor z$
Erase the line $(x \land \overline{y}) \lor z$
Erase the line $\overline{x} \lor y$
Example \(k\)-DNF Resolution Refutation (\(k = 2\))

Can write down axioms, infer new formulas, and erase used formulas:

1. \(x\)
2. \(\overline{x} \lor y\)
3. \(\overline{y} \lor z\)
4. \(\overline{z}\)

Rules:
- Infer new formulas only from formulas currently on board.
- Only \(k\)-DNF formulas can appear on board (for \(k\) fixed).
- Details about derivation rules won’t matter for us.

Write down axiom 2: \(\overline{x} \lor y\)

Infer \(z\) from
\(\overline{x} \lor y\) and \((x \land \overline{y}) \lor z\)

Erase the line \((x \land \overline{y}) \lor z\)

Erase the line \(\overline{x} \lor y\)
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Infer $z$ from $\overline{x} \lor y$ and $(x \land \overline{y}) \lor z$
Erase the line $(x \land \overline{y}) \lor z$
Erase the line $\overline{x} \lor y$
Write down axiom 4: $\overline{z}$
Example \( k \)-DNF Resolution Refutation (\( k = 2 \))

Can write down axioms, infer new formulas, and erase used formulas

1. \( x \)
2. \( \overline{x} \lor y \)
3. \( \overline{y} \lor z \)
4. \( \overline{z} \)

Rules:
- Infer new formulas only from formulas currently on board
- Only \( k \)-DNF formulas can appear on board (for \( k \) fixed)
- Details about derivation rules won’t matter for us

Erase the line \((x \land \overline{y}) \lor z\)
Erase the line \(\overline{x} \lor y\)
Write down axiom 4: \(\overline{z}\)
Infer 0 from \(\overline{z}\) and \(z\)
Example $k$-DNF Resolution Refutation ($k = 2$)

Can write down axioms, infer new formulas, and erase used formulas

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

Rules:
- Infer new formulas only from formulas currently on board
- Only $k$-DNF formulas can appear on board (for $k$ fixed)
- Details about derivation rules won’t matter for us

Erase the line $(x \land \overline{y}) \lor z$
Erase the line $\overline{x} \lor y$
Write down axiom 4: $\overline{z}$
Infer 0 from $\overline{z}$ and $z$
Complexity Measures of Interest: Length and Space

- **Length**: Lower bound on time for proof search algorithm
- **Space**: Lower bound on memory for proof search algorithm

**Length**

# formulas written on blackboard counted with repetitions
(Or total # derivation steps)

**Space**

Somewhat less straightforward—several ways of measuring

\[
\begin{align*}
  x & \quad \text{Formula space: 3} \\
  \bar{y} \lor z & \quad \text{Total space: 6} \\
  (x \land \bar{y}) \lor z & \quad \text{Variable space: 3}
\end{align*}
\]
Complexity Measures of Interest: Length and Space

- **Length**: Lower bound on *time* for proof search algorithm
- **Space**: Lower bound on *memory* for proof search algorithm

**Length**

# formulas written on blackboard counted with repetitions
(Or total # derivation steps)

**Space**

Somewhat less straightforward—several ways of measuring

\[
x \quad \bar{y} \lor z \quad (x \land \bar{y}) \lor z
\]

- Formula space: 3
- Total space: 6
- Variable space: 3
Complexity Measures of Interest: Length and Space

- **Length:** Lower bound on *time* for proof search algorithm
- **Space:** Lower bound on *memory* for proof search algorithm

**Length**

# formulas written on blackboard counted with repetitions  
(Or total # derivation steps)

**Space**

Somewhat less straightforward—several ways of measuring

\[ x \]
\[ \overline{y} \lor z \]
\[ (x \land \overline{y}) \lor z \]

- **Formula space:** 3
- **Total space:** 6
- **Variable space:** 3
Let $n =$ size of formula

**Length:** at most $2^n$
Lower bound $\exp(\Omega(n))$ [Urquhart '87, Chvátal & Szemerédi '88]

**Formula space (a.k.a. clause space):** at most $n$
Lower bound $\Omega(n)$ [Torán '99, Alekhnovich et al. '00]

**Total space:** at most $n^2$
No better lower bound than $\Omega(n)$!?

**Variable space:** at most $n$
Lower bound $\Omega(n)$ [Ben-Sasson & Wigderson '99]
For restricted system of so-called **tree-like resolution**: length and space strongly correlated [Esteban & Torán ’99]

So essentially no trade-offs for tree-like resolution

No (nontrivial) length-space correlation for general resolution [Ben-Sasson & Nordström ’08]

Nothing known about time-space trade-offs for

- resolution refutations of
- explicit formulas in
- general, unrestricted resolution

(Results in restricted settings in [Ben-Sasson ’02, Hertel & Pitassi ’07, Nordström ’07])
Length-Space Trade-offs for Resolution?

For restricted system of so-called tree-like resolution: length and space strongly correlated [Esteban & Torán ’99]

So essentially no trade-offs for tree-like resolution

No (nontrivial) length-space correlation for general resolution [Ben-Sasson & Nordström ’08]

Nothing known about time-space trade-offs for

- resolution refutations of
- explicit formulas in
- general, unrestricted resolution

(Results in restricted settings in [Ben-Sasson ’02, Hertel & Pitassi ’07, Nordström ’07])
Length-Space Trade-offs for Resolution?

For restricted system of so-called tree-like resolution: length and space strongly correlated [Esteban & Torán ’99]

So essentially no trade-offs for tree-like resolution

No (nontrivial) length-space correlation for general resolution [Ben-Sasson & Nordström ’08]

Nothing known about time-space trade-offs for
- resolution refutations of
- explicit formulas in
- general, unrestricted resolution

(Results in restricted settings in [Ben-Sasson ’02, Hertel & Pitassi ’07, Nordström ’07])
Previous Work on $k$-DNF Resolution ($k \geq 2$)

**Length:** lower bound $\exp(\Omega(n^{1-o(1)}))$ [Alekhnovich ’05]

**Formula space:** lower bound $\Omega(n)$ [Esteban et al. ’02]

(Suppressing dependencies on $k$)

$(k+1)$-DNF resolution exponentially stronger than $k$-DNF resolution w.r.t. length [Segerlind et al. ’04]

No hierarchy known w.r.t. space

Except for tree-like $k$-DNF resolution [Esteban et al. ’02]

(But tree-like $k$-DNF weaker than standard resolution)

No trade-off results known
Previous Work on $k$-DNF Resolution ($k \geq 2$)

**Length:** lower bound $\exp(\Omega(n^{1-o(1)}))$ [Alekhnovich ’05]

**Formula space:** lower bound $\Omega(n)$ [Esteban et al. ’02]

(Suppressing dependencies on $k$)

$(k+1)$-DNF resolution exponentially stronger than $k$-DNF resolution w.r.t. length [Segerlind et al. ’04]

No hierarchy known w.r.t. space

Except for tree-like $k$-DNF resolution [Esteban et al. ’02]

(But tree-like $k$-DNF weaker than standard resolution)

No trade-off results known
Previous Work on $k$-DNF Resolution ($k \geq 2$)

**Length:** lower bound $\exp(\Omega(n^{1-o(1)}))$ [Alekhnovich ’05]

**Formula space:** lower bound $\Omega(n)$ [Esteban et al. ’02]

(Suppressing dependencies on $k$)

$(k+1)$-DNF resolution exponentially stronger than $k$-DNF resolution w.r.t. length [Segerlind et al. ’04]

No hierarchy known w.r.t. space

Except for tree-like $k$-DNF resolution [Esteban et al. ’02]

(But tree-like $k$-DNF weaker than standard resolution)

No trade-off results known
We prove a collection of time-space trade-offs

Results hold for

- resolution (essentially tight analysis)
- $k$-DNF resolution, $k \geq 2$ (with slightly worse parameters)

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

For any $\omega(1)$ function and any fixed $k$ there exist explicit CNF formulas of size $O(n)$

- refutable in resolution in total space $\omega(1)$
- refutable in resolution in length $O(n)$ and total space $\approx 3\sqrt{n}$
- any resolution refutation in formula space $\leq 3\sqrt{n}$ requires superpolynomial length
- any $k$-DNF resolution refutation in formula space $\leq n^{1/3(k+1)}$ requires superpolynomial length
One Example: Robust Trade-offs for Small Space

Theorem

For any $\omega(1)$ function and any fixed $k$ there exist explicit CNF formulas of size $\mathcal{O}(n)$

- refutable in resolution in total space $\omega(1)$
- refutable in resolution in length $\mathcal{O}(n)$ and total space $\approx 3\sqrt[3]{n}$
- any resolution refutation in formula space $\leq 3\sqrt[3]{n}$ requires superpolynomial length
- any $k$-DNF resolution refutation in formula space $\leq n^{1/3(k+1)}$ requires superpolynomial length
One Example: Robust Trade-offs for Small Space

**Theorem**

For any $\omega(1)$ function and any fixed $k$ there exist explicit CNF formulas of size $\mathcal{O}(n)$

- refutable in resolution in total space $\omega(1)$
- refutable in resolution in length $\mathcal{O}(n)$ and total space $\approx 3\sqrt{n}$
- any resolution refutation in formula space $\leq 3\sqrt{n}$ requires superpolynomial length
- any $k$-DNF resolution refutation in formula space $\leq n^{1/3(k+1)}$ requires superpolynomial length
Theorem

For any $\omega(1)$ function and any fixed $k$ there exist explicit CNF formulas of size $O(n)$

- refutable in resolution in total space $\omega(1)$
- refutable in resolution in length $O(n)$ and total space $\approx 3\sqrt{n}$
- any resolution refutation in formula space $\leq 3\sqrt{n}$ requires superpolynomial length
- any $k$-DNF resolution refutation in formula space $\leq n^{1/(3(k+1))}$ requires superpolynomial length

One Example: Robust Trade-offs for Small Space
One Example: Robust Trade-offs for Small Space

**Theorem**

For any $\omega(1)$ function and any fixed $k$ there exist explicit CNF formulas of size $O(n)$

- refutable in resolution in total space $\omega(1)$
- refutable in resolution in length $O(n)$ and total space $\approx 3\sqrt{n}$
- any resolution refutation in formula space $\leq 3\sqrt{n}$ requires superpolynomial length
- any $k$-DNF resolution refutation in formula space $\leq n^{1/3(k+1)}$ requires superpolynomial length
Some Quick Technical Remarks

Upper bounds hold for
- total space (# literals)
- standard syntactic derivation rules

Lower bounds hold for
- formula space (# lines)
- semantic derivation rules (exponentially stronger)

Space definition reminder

\[
\begin{align*}
x & \quad \text{Formula space: 3} \\
\overline{y} \lor z & \quad \text{Total space: 6} \\
(x \land \overline{y}) \lor z & \quad \text{Variable space: 3}
\end{align*}
\]
New Results 2: Space Hierarchy for $k$-DNF Resolution

We also separate $k$-DNF resolution from $(k+1)$-DNF resolution w.r.t. formula space

**Theorem**

For any constant $k$ there are explicit CNF formulas of size $O(n)$

- refutable in $(k+1)$-DNF resolution in formula space $O(1)$ but such that
- any $k$-DNF resolution refutation requires formula space $\Omega\left(\frac{k+1}{\log n}\sqrt{n} \right)$
Rest of This Talk

- Study old combinatorial game from the 1970s
- Prove new theorem about variable substitution and proof space
- Combine the two
How to Get a Handle on Time-Space Relations?

Time-space trade-off questions well-studied for pebble games modelling calculations described by DAGs ([Cook & Sethi ’76] and many others)

- **Time** needed for calculation: \# pebbling moves
- **Space** needed for calculation: max \# pebbles required
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them

<table>
<thead>
<tr>
<th># moves</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current # pebbles</td>
<td>2</td>
</tr>
<tr>
<td>Max # pebbles so far</td>
<td>3</td>
</tr>
</tbody>
</table>
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them
The Black-White Pebble Game

**Goal:** get single black pebble on sink vertex of $G$

1. Can **place black pebble** on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always **remove black pebble** from vertex
3. Can always **place white pebble** on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them

Graph:

- $u$, $v$, $w$, $x$, $y$, $z$
- $u$ connects to $x$
- $x$ connects to $v$, $y$
- $v$ connects to $w$, $x$
- $y$ connects to $w$, $z$
- $z$ connects to $w$

| # moves | 6 |
| Current # pebbles | 2 |
| Max # pebbles so far | 3 |
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can **place black pebble** on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always **remove black pebble** from vertex
3. Can always **place white pebble** on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them

<table>
<thead>
<tr>
<th># moves</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current # pebbles</td>
<td>3</td>
</tr>
<tr>
<td>Max # pebbles so far</td>
<td>3</td>
</tr>
</tbody>
</table>
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them
The Black-White Pebble Game

**Goal**: get single black pebble on sink vertex of $G$

1. Can **place black pebble** on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always **remove black pebble** from vertex
3. Can always **place white pebble** on (empty) vertex
4. Can **remove white pebble** from $v$ if all immediate predecessors have pebbles on them

<table>
<thead>
<tr>
<th># moves</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current # pebbles</td>
<td>2</td>
</tr>
<tr>
<td>Max # pebbles so far</td>
<td>3</td>
</tr>
</tbody>
</table>
### The Black-White Pebble Game

**Goal:** get single black pebble on sink vertex of $G$

1. Can **place black pebble** on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always **remove black pebble** from vertex
3. Can always **place white pebble** on (empty) vertex
4. Can **remove white pebble** from $v$ if all immediate predecessors have pebbles on them

**Example:**

- **# moves:** 9
- **Current # pebbles:** 3
- **Max # pebbles so far:** 3
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

- Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
- Can always remove black pebble from vertex
- Can always place white pebble on (empty) vertex
- Can remove white pebble from $v$ if all immediate predecessors have pebbles on them

Here is a table showing the number of moves, current number of pebbles, and maximum number of pebbles so far:

<table>
<thead>
<tr>
<th># moves</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current # pebbles</td>
<td>4</td>
</tr>
<tr>
<td>Max # pebbles so far</td>
<td>4</td>
</tr>
</tbody>
</table>
### The Black-White Pebble Game

**Goal:** get single black pebble on sink vertex of $G$

1. Can **place black pebble** on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always **remove black pebble** from vertex
3. Can always **place white pebble** on (empty) vertex
4. Can **remove white pebble** from $v$ if all immediate predecessors have pebbles on them

- # moves: 11
- Current # pebbles: 3
- Max # pebbles so far: 4
The Black-White Pebble Game

Goal: get single black pebble on sink vertex of $G$

1. Can place black pebble on (empty) vertex $v$ if all immediate predecessors have pebbles on them
2. Can always remove black pebble from vertex
3. Can always place white pebble on (empty) vertex
4. Can remove white pebble from $v$ if all immediate predecessors have pebbles on them

# moves | 12
---|---
Current # pebbles | 2
Max # pebbles so far | 4
The Black-White Pebble Game

**Goal:** get single black pebble on sink vertex of $G$

1. Can **place black pebble** on (empty) vertex $v$ if all immediate predecessors have pebbles on them.
2. Can always **remove black pebble** from vertex.
3. Can always **place white pebble** on (empty) vertex.
4. Can **remove white pebble** from $v$ if all immediate predecessors have pebbles on them.

- # moves: 13
- Current # pebbles: 1
- Max # pebbles so far: 4
Pebbling Contradiction

CNF formula encoding pebble game on DAG $G$

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

Studied by [Bonet et al. ’98, Raz & McKenzie ’99, Ben-Sasson & Wigderson ’99] and others
Observation (Ben-Sasson et al. ’00)

Any black-pebbles-only pebbling translates into refutation with
- \( \text{refutation length} \leq \# \text{ moves} \)
- \( \text{total space} \leq \# \text{ pebbles} \)

Theorem (Ben-Sasson ’02)

Any refutation translates into black-white pebbling with
- \( \# \text{ moves} \leq \text{refutation length} \)
- \( \# \text{ pebbles} \leq \text{variable space} \)

Unfortunately extremely easy w.r.t. formula space!
Resolution–Pebbling Correspondence

Observation (Ben-Sasson et al. ’00)

Any black-pebbles-only pebbling translates into refutation with

- refutation length \( \leq \) # moves
- total space \( \leq \) # pebbles

Theorem (Ben-Sasson ’02)

Any refutation translates into black-white pebbling with

- # moves \( \leq \) refutation length
- # pebbles \( \leq \) variable space

Unfortunately extremely easy w.r.t. formula space!
Resolution–Pebbling Correspondence

**Observation (Ben-Sasson et al. ’00)**

Any black-pebbles-only pebbling translates into refutation with

- refutation length \( \leq \) \# moves
- total space \( \leq \) \# pebbles

**Theorem (Ben-Sasson ’02)**

Any refutation translates into black-white pebbling with

- \# moves \( \leq \) refutation length
- \# pebbles \( \leq \) variable space

Unfortunately extremely easy w.r.t. formula space!
Key Idea: Variable Substitution

Make formula harder by substituting $x_1 \oplus x_2$ for every variable $x$:

\[
\begin{align*}
\overline{x} \lor y \\
\Downarrow \\
\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2) \\
\Downarrow \\
(x_1 \lor \overline{x}_2 \lor y_1 \lor y_2) \\
\land (x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2) \\
\land (\overline{x}_1 \lor x_2 \lor y_1 \lor y_2) \\
\land (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2)
\end{align*}
\]
Let $F[\oplus]$ denote formula with $\text{XOR } x_1 \oplus x_2$ substituted for $x$.

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

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

Obvious approach for $F[⊕]$: mimic refutation of $F$
Key Technical Result: Substitution Space Theorem

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

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

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

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

\[
\begin{align*}
x & \\
\overline{x} \lor y & \\
y & 
\end{align*}
\]
Key Technical Result: Substitution Space Theorem

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

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

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

\[
\begin{array}{c}
x_1 \lor x_2 \\
\overline{x}_1 \lor \overline{x}_2 \\
\end{array}
\]
Key Technical Result: Substitution Space Theorem

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

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

\[
\begin{align*}
\quad x \\
\quad \overline{x} \lor y \\
\quad y
\end{align*}
\]

\[
\begin{align*}
\quad x_1 \lor x_2 \\
\quad \overline{x}_1 \lor \overline{x}_2 \\
\quad x_1 \lor \overline{x}_2 \lor y_1 \lor y_2 \\
\quad \overline{x}_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor \overline{y}_2 \\
\quad \overline{x}_1 \lor x_2 \lor y_1 \lor y_2 \\
\quad \overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2
\end{align*}
\]
Key Technical Result: Substitution Space Theorem

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

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

\[
\begin{align*}
x &
\bar{x} \lor y \\
\bar{y} &
\end{align*}
\]

\[
\begin{align*}
x_1 \lor x_2 \\
\bar{x}_1 \lor \bar{x}_2 \\
x_1 \lor \bar{x}_2 \lor y_1 \lor y_2 \\
\bar{x}_1 \lor \bar{x}_2 \lor \bar{y}_1 \lor \bar{y}_2 \\
x_1 \lor x_2 \lor \bar{y}_1 \lor \bar{y}_2 \\
\bar{x}_1 \lor x_2 \lor y_1 \lor y_2 \\
y_1 \lor y_2 \\
\bar{y}_1 \lor \bar{y}_2
\end{align*}
\]
Key Technical Result: Substitution Space Theorem

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

Obvious approach for \( F[\oplus] \): mimic refutation of \( F \)

\[
\begin{align*}
  &x \\
  &\overline{x} \lor y \\
  &y
\end{align*}
\]

For such refutation of \( F[\oplus] \):

- length \( \geq \) length for \( F \)
- formula space \( \geq \) variable space for \( F \)

\[
\begin{align*}
  &x_1 \lor x_2 \\
  &\overline{x}_1 \lor \overline{x}_2 \\
  &x_1 \lor \overline{x}_2 \lor y_1 \lor y_2 \\
  &\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2 \\
  &y_1 \lor y_2 \\
  &\overline{y}_1 \lor \overline{y}_2
\end{align*}
\]
Key Technical Result: Substitution Space Theorem

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

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

For such refutation of $F[⊕]$: 
- $\text{length} \geq \text{length for } F$
- $\text{formula space} \geq \text{variable space for } F$

Prove that this is (sort of) best one can do for $F[⊕]$!
### Sketch of Proof of Substitution Space 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. $¬(x_1 ⊕ x_2) ∨ (y_1 ⊕ y_2)$…</td>
<td>write $\overline{x} ∨ y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations…</td>
<td>can get between corresponding shadow blackboards by legal derivation steps</td>
</tr>
<tr>
<td>… (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
## Sketch of Proof of Substitution Space Theorem

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 derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
### Sketch of Proof of Substitution Space Theorem

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 derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
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 derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
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 derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
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. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)\ldots$</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 derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
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 derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
### Sketch of Proof of Substitution Space 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. $\neg(x_1 \oplus x_2) \lor (y_1 \oplus y_2)\ldots$</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 derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
## Sketch of Proof of Substitution Space 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. $¬(x_1 ⊕ x_2) ∨ (y_1 ⊕ y_2)$...</td>
<td>write $\overline{x} ∨ y$ on shadow blackboard</td>
</tr>
<tr>
<td>For consecutive XOR blackboard configurations...</td>
<td>can get between corresponding shadow blackboards by legal derivation steps</td>
</tr>
<tr>
<td>... (sort of) upper-bounded by XOR derivation length</td>
<td>Length 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>
Applying Substitution to Pebbling Formulas

Making variable substitutions in pebbling formulas

- lifts lower bound from variable space to formula space
- maintains upper bound in terms of total space and length

Substitution with XOR over \( k + 1 \) variables works against \( k \)-DNF resolution

Get our results by

- using known pebbling results from literature of 70s and 80s
- proving a couple of new pebbling results
- to get tight trade-offs, showing that resolution proofs can sometimes do better than black-only pebblings
Applying Substitution to Pebbling Formulas

Making variable substitutions in pebbling formulas
- lifts lower bound from variable space to formula space
- maintains upper bound in terms of total space and length

Substitution with XOR over $k + 1$ variables works against $k$-DNF resolution

Get our results by
- using known pebbling results from literature of 70s and 80s
- proving a couple of new pebbling results
- to get tight trade-offs, showing that resolution proofs can sometimes do better than black-only pebblings
Applying Substitution to Pebbling Formulas

Making variable substitutions in pebbling formulas

- lifts lower bound from variable space to formula space
- maintains upper bound in terms of total space and length

Substitution with XOR over \( k + 1 \) variables works against \( k \)-DNF resolution

Get our results by

- using known pebbling results from literature of 70s and 80s
- proving a couple of new pebbling results
- to get tight trade-offs, showing that resolution proofs can sometimes do better than black-only pebblings
Stronger Results for $k$-DNF resolution?

Gap of $(k+1)\text{st root}$ between upper and lower bounds for $k$-DNF resolution

Open Question

*Can the loss of a $(k+1)\text{st root}$ in the $k$-DNF resolution lower bounds be diminished? Or even eliminated completely?*

Conceivable that same bounds as for resolution could hold

However, any improvement beyond $k\text{th root}$ requires fundamentally different approach [Nordström & Razborov ’09]
Stronger Results for $k$-DNF resolution?

Gap of $(k+1)$st root between upper and lower bounds for $k$-DNF resolution

Open Question

*Can the loss of a $(k+1)$st root in the $k$-DNF resolution lower bounds be diminished? Or even eliminated completely?*

Conceivable that same bounds as for resolution could hold

However, any improvement beyond $k$th root requires fundamentally different approach [Nordström & Razborov ’09]
Stronger Length-Space Trade-offs than from Pebbling?

Open Question

Are there superpolynomial trade-offs for formulas refutable in constant space?

Open Question

Are there formulas with trade-offs in the range space > formula size? Or can every proof be carried out in at most linear space?

Pebbling formulas cannot answer these questions—can impossibly have such strong trade-offs.
Summing up

- **Strong time-space trade-offs** for resolution and $k$-DNF resolution for wide range of parameters

- **Strict space hierarchy** for $k$-DNF resolution

- **Many remaining open questions** about space in resolution

Thank you for your attention!