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

Executive Summary of Talk

- 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 $C = a_1 \vee \cdots \vee a_k$: disjunction of literals
- Term $T = a_1 \wedge \cdots \wedge a_k$: conjunction of literals
- CNF formula $F = C_1 \wedge \cdots \wedge C_m$: conjunction of clauses
  k-CNF formula: CNF formula with clauses of size $\leq k$
- DNF formula $D = T_1 \vee \cdots \vee T_m$: 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. $\overline{x} \vee y$
3. $\overline{y} \vee z$
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. $\overline{x} \vee y$
3. $\overline{y} \vee z$
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. $\overline{z}$

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

Write down axiom 1: 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. $\overline{z}$

x $\overline{y} \lor z$

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. $\overline{x} \vee y$
3. $\overline{y} \vee z$
4. $\overline{z}$

x
$\overline{y} \vee z$

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: $\overline{y} \vee z$
Combine x and $\overline{y} \vee z$
to get $(x \wedge \overline{y}) \vee z$

Example k-DNF Resolution Refutation ($k = 2$)

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

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

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

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: $\overline{y} \vee z$
Combine x and $\overline{y} \vee z$
to get $(x \wedge \overline{y}) \vee z$

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. $\overline{z}$

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

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. $\overline{z}$

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

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. $\overline{z}$

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

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 $\overline{y} \lor z$

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. $\overline{z}$

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

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 $\overline{y} \lor z$

Example k-DNF Resolution Refutation ($k = 2$)

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

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

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

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 $\overline{y} \vee z$
to get $(x \wedge \overline{y}) \vee z$
Erase the line x
Erase the line $\overline{y} \vee z$
Write down axiom 2: $\overline{x} \vee y$

Example k-DNF Resolution Refutation ($k = 2$)

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

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

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

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 $\overline{y} \vee z$
Write down axiom 2: $\overline{x} \vee y$
Infer z from
$\overline{x} \vee y$ and $(x \wedge \overline{y}) \vee z$

Example k-DNF Resolution Refutation ($k = 2$)

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

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

$(x \wedge \overline{y}) \vee z$
$\overline{x} \vee y$
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 $\overline{y} \vee z$
Write down axiom 2: $\overline{x} \vee y$
Infer z from
$\overline{x} \vee y$ and $(x \wedge \overline{y}) \vee z$

Example k-DNF Resolution Refutation ($k = 2$)

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

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

$(x \wedge \overline{y}) \vee z$
$\overline{x} \vee y$
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 $\overline{y} \vee z$
Write down axiom 2: $\overline{x} \vee y$
Infer z from
$\overline{x} \vee y$ and $(x \wedge \overline{y}) \vee z$
Erase the line $(x \wedge \overline{y}) \vee z$

Example k-DNF Resolution Refutation ($k = 2$)

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

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

$\overline{x} \vee y$
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 $\overline{y} \vee z$
Write down axiom 2: $\overline{x} \vee y$
Infer z from
$\overline{x} \vee y$ and $(x \wedge \overline{y}) \vee z$
Erase the line $(x \wedge \overline{y}) \vee z$

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. $\overline{z}$

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

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: $\overline{x} \vee y$
Infer z from
$\overline{x} \vee y$ and $(x \wedge \overline{y}) \vee z$
Erase the line $(x \wedge \overline{y}) \vee z$
Erase the line $\overline{x} \vee y$

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. $\overline{z}$

z
$\overline{z}$

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. $\overline{z}$

z
$\overline{z}$

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. $\overline{x} \vee y$
3. $\overline{y} \vee z$
4. $\overline{z}$

z
$\overline{z}$
0

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

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

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

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

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

Length and Space Bounds for Resolution

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]

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])

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

New Results 1: Time-Space Trade-offs

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 $\mathcal{O}(n)$

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

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

Formula space:	3
Total space:	6
Variable space:	3

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 $\mathcal{O}(n)$

- *refutable in $(k+1)$-DNF resolution in formula space $\mathcal{O}(1)$ but such that*
- *any k-DNF resolution refutation requires formula space $\Omega\left(\sqrt[k+1]{n/\log n}\right)$*

Rest of This Talk

- Study old combinatorial game from the 1970s
- Prove new theorem about variable substitution and proof space
- Combine the two

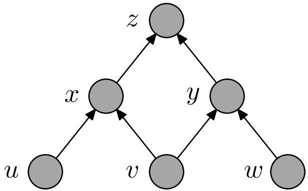
How to Get a Handle on Time-Space Relations?

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

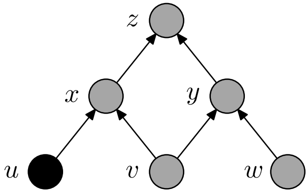


# moves	0
Current # pebbles	0
Max # pebbles so far	0

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

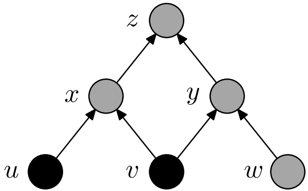


# moves	1
Current # pebbles	1
Max # pebbles so far	1

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

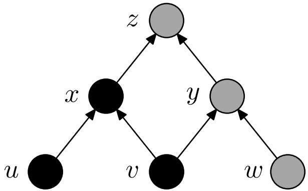


# moves	2
Current # pebbles	2
Max # pebbles so far	2

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

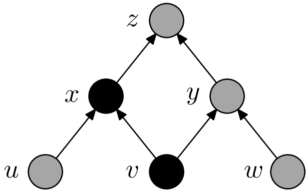


# moves	3
Current # pebbles	3
Max # pebbles so far	3

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*

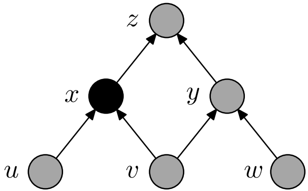


# moves	4
Current # pebbles	2
Max # pebbles so far	3

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*

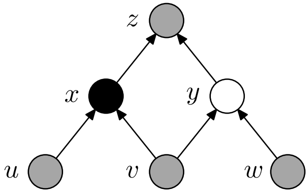


# moves	5
Current # pebbles	1
Max # pebbles so far	3

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

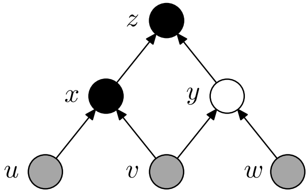


# moves	6
Current # pebbles	2
Max # pebbles so far	3

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*

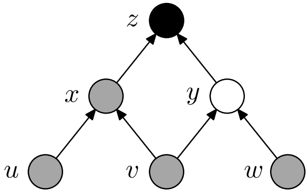


# moves	7
Current # pebbles	3
Max # pebbles so far	3

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*

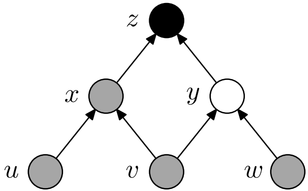


# moves	8
Current # pebbles	2
Max # pebbles so far	3

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*

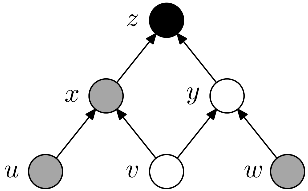


# moves	8
Current # pebbles	2
Max # pebbles so far	3

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

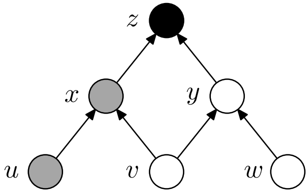


# moves	9
Current # pebbles	3
Max # pebbles so far	3

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*

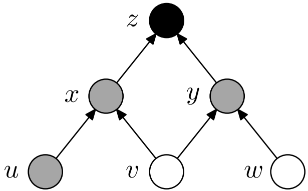


# moves	10
Current # pebbles	4
Max # pebbles so far	4

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*

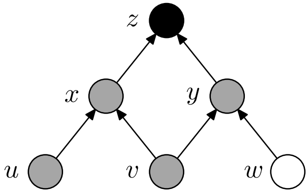


# moves	11
Current # pebbles	3
Max # pebbles so far	4

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

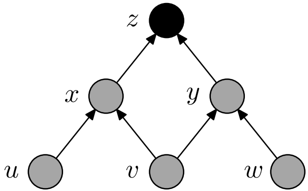


# moves	12
Current # pebbles	2
Max # pebbles so far	4

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*



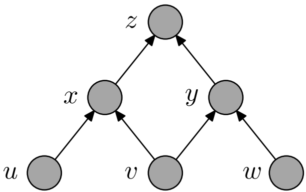
# moves	13
Current # pebbles	1
Max # pebbles so far	4

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

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

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!

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:

$$\overline{x} \vee y$$

$$\Downarrow$$

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

$$\Downarrow$$

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

Key Technical Result: Substitution Space Theorem

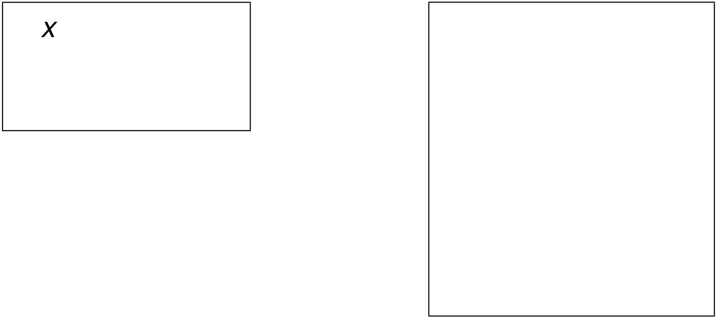
Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

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

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



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

x
$\overline{x} \lor y$

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

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

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

x
$\overline{x} \vee y$
y

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

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

x
$\overline{x} \vee y$
y

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

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

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

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

x
$\overline{x} \vee y$
y

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

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

- length $\geq$ length for F
- formula space $\geq$ variable space for F

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

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

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

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

- length $\geq$ length for F
- formula space $\geq$ variable space for F

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

Sketch of Proof of Substitution Space Theorem

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

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

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 $(k+1)$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 kth 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 kth 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!