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

Jakob Nordström

Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Propositional Proof Complexity: Theory and Practice
Federated Logic Conference (FLoC '10)
University of Edinburgh, UK
July 9, 2010

Joint work with Eli Ben-Sasson

Executive Summary of Talk

- SATISFIABILITY: NP-complete and so probably intractable in worst case
- But enormous progress on applied algorithms last 10-15 years
- Best known algorithms today based on resolution (DPLL-algorithms augmented with clause learning)
- Key resources for SAT-solvers: time and space
- What are the connections between these resources? Time-space correlations? Trade-offs?
- What can proof complexity say about this? (For resolution and more powerful k-DNF resolution proof systems)

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

$\boxed{x}$

Write down axiom 1: x

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

x
$\overline{y} \lor z$

Write down axiom 1: x
Write down axiom 3: $\overline{y} \lor z$

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

x
$\overline{y} \vee z$

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

x

$\overline{y} \vee z$

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

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

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

Write down axiom 1: x
Write down axiom 3: $\overline{y} \vee z$
Combine x and $\overline{y} \vee z$
to get $(x \wedge \overline{y}) \vee z$
Erase the line x

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

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

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

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

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

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

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

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

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

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

$$\boxed{\begin{array}{l} (x \wedge \overline{y}) \vee z \\ \overline{x} \vee y \\ \end{array}}$$

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

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

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

$$\boxed{\begin{array}{l} (x \land \overline{y}) \lor z \\ \overline{x} \lor y \\ z \end{array}}$$

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

$\overline{x} \vee y$

z

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

$\overline{x} \vee y$

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

Erase the line $\overline{x} \vee y$

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

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

Erase the line $\overline{x} \vee y$

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

z
$\overline{z}$

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 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

z
$\overline{z}$

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

Example 2-DNF Resolution Refutation

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 $k = 2$)
- Details about derivation rules won't matter for us

---

z
$\overline{z}$
0

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 $\approx$ Lower bound on time for SAT-solver
- Space $\approx$ Lower bound on memory for SAT-solver

Length

\# formulas written on blackboard counted with repetitions

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 SAT-solver
- Space ≈ Lower bound on memory for SAT-solver

Length

\# formulas written on blackboard counted with repetitions

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 $\approx$ Lower bound on time for SAT-solver
- Space $\approx$ Lower bound on memory for SAT-solver

Length

formulas written on blackboard counted with repetitions

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 $\approx$ Lower bound on time for SAT-solver
- Space $\approx$ Lower bound on memory for SAT-solver

Length

\# formulas written on blackboard counted with repetitions

Space

Somewhat less straightforward — several ways of measuring

1. x
2. $\overline{y} \vee z$
3. $(x \wedge \overline{y}) \vee z$

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

Complexity Measures of Interest: Length and Space

- Length $\approx$ Lower bound on time for SAT-solver
- Space $\approx$ Lower bound on memory for SAT-solver

Length

formulas written on blackboard counted with repetitions

Space

Somewhat less straightforward — several ways of measuring

$$\boxed{\begin{array}{l} x^{1} \\ \overline{y}^{2} \lor z^{3} \\ (x^{4} \land \overline{y})^{5} \lor z^{6} \end{array}}$$

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

Complexity Measures of Interest: Length and Space

- Length $\approx$ Lower bound on time for SAT-solver
- Space $\approx$ Lower bound on memory for SAT-solver

Length

formulas written on blackboard counted with repetitions

Space

Somewhat less straightforward — several ways of measuring

$$\boxed{\begin{array}{l} x^{1} \\ \overline{y}^{2} \lor z^{3} \\ (x \land \overline{y}) \lor z \end{array}}$$

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

Length and Space Bounds for (1-DNF) 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 $\Omega(n)$!?

Notice formula space lower bounds can be at most linear — but these are **nondeterministic bounds!** (So might be much stronger in practice)

Length and Space Bounds for (1-DNF) 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 $\Omega(n)$!?

Notice formula space lower bounds can be at most linear — but these are **nondeterministic bounds!** (So might be much stronger in practice)

Length-Space Trade-offs for Resolution?

For restricted system of so-called tree-like resolution ($\Leftrightarrow$ original DLL algorithm): length and space strongly correlated [Esteban & Torán '99, Atserias & Dalmau '03]

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

- explicit formulas in
- general, unrestricted resolution

(Results in restricted settings in [Ben-Sasson '02, Nordström '07])

Length-Space Trade-offs for Resolution?

For restricted system of so-called tree-like resolution ($\Leftrightarrow$ original DLL algorithm): length and space strongly correlated [Esteban & Torán '99, Atserias & Dalmau '03]

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

- explicit formulas in
- general, unrestricted resolution

(Results in restricted settings in [Ben-Sasson '02, Nordström '07])

Length-Space Trade-offs for Resolution?

For restricted system of so-called tree-like resolution ($\Leftrightarrow$ original DLL algorithm): length and space strongly correlated [Esteban & Torán '99, Atserias & Dalmau '03]

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

- explicit formulas in
- general, unrestricted resolution

(Results in restricted settings in [Ben-Sasson '02, Nordström '07])

Previous Work on k-DNF Resolution ($k \geq 2$)

Upper bounds carry over from resolution

Length: lower bound $\exp\bigl(\Omega\bigl(n^{1-o(1)}\bigr)\bigr)$ [Segerlind et al. '04, 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 ($k \geq 2$)

Upper bounds carry over from resolution

Length: lower bound $\exp\left(\Omega\left(n^{1-o(1)}\right)\right)$ [Segerlind et al. '04, 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 ($k \geq 2$)

Upper bounds carry over from resolution

Length: lower bound $\exp\left(\Omega\left(n^{1-o(1)}\right)\right)$ [Segerlind et al. '04, 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: Length-Space Trade-offs

We prove collection of length-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 that have

- linear length (and constant width) refutations of high space complexity, but for which
- any small space complexity refutation must be (very) long

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, $k \leq K$, 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, $k \leq K$, 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, $k \leq K$, 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, $k \leq K$, 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, $k \leq K$, in formula space $\lesssim n^{1/3(k+1)}$ requires superpolynomial length*

Some Quick Technical Remarks

Upper bounds hold for

- total space (# literals) — larger measure
- standard syntactic rules

Lower bounds hold for

- formula space (# lines) — smaller measure
- semantic rules — exponentially stronger than syntactic

Space definition reminder

$$\boxed{\begin{array}{l} x \\ \overline{y} \lor z \\ (x \land \overline{y}) \lor z \end{array}}$$

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

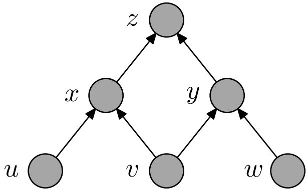
How to Get a Handle on Time-Space Relations?

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

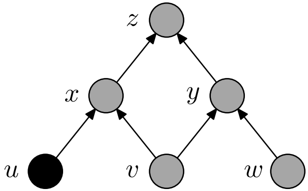


# moves	0
Current # pebbles	0
Max # pebbles so far	0

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

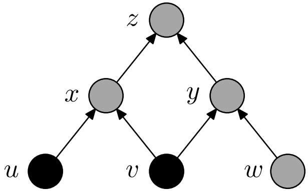


# moves	1
Current # pebbles	1
Max # pebbles so far	1

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

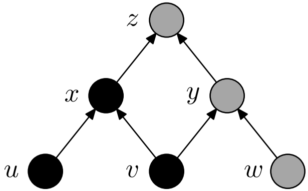


# moves	2
Current # pebbles	2
Max # pebbles so far	2

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

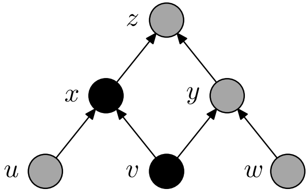


# moves	3
Current # pebbles	3
Max # pebbles so far	3

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

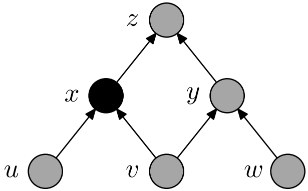


# moves	4
Current # pebbles	2
Max # pebbles so far	3

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

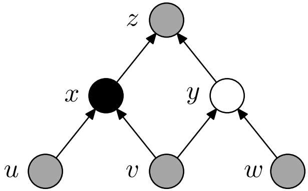


# moves	5
Current # pebbles	1
Max # pebbles so far	3

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

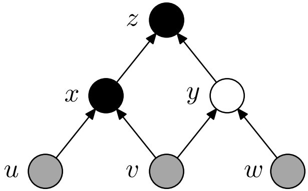


# moves	6
Current # pebbles	2
Max # pebbles so far	3

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

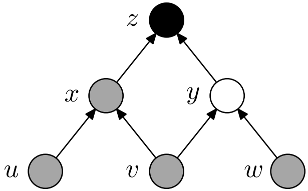


# moves	7
Current # pebbles	3
Max # pebbles so far	3

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

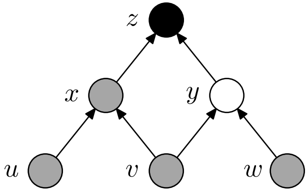


# moves	8
Current # pebbles	2
Max # pebbles so far	3

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

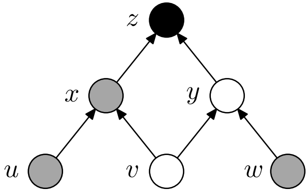


# moves	8
Current # pebbles	2
Max # pebbles so far	3

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

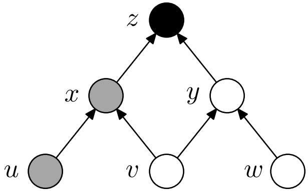


# moves	9
Current # pebbles	3
Max # pebbles so far	3

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

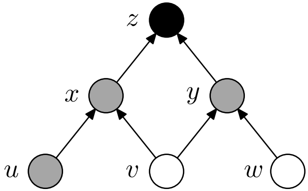


# moves	10
Current # pebbles	4
Max # pebbles so far	4

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

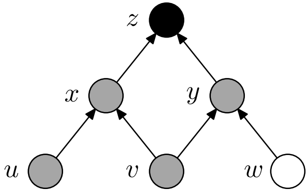


# moves	11
Current # pebbles	3
Max # pebbles so far	4

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

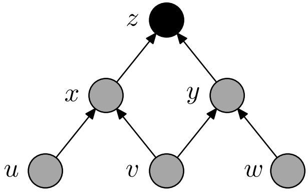


# moves	12
Current # pebbles	2
Max # pebbles so far	4

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



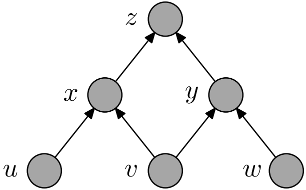
# moves	13
Current # pebbles	1
Max # pebbles so far	4

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 (also works for other Boolean functions with "right" properties):

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

$$\Downarrow$$

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

$$\Downarrow$$

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

Key Technical Result: Substitution Theorem

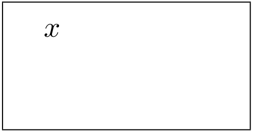
Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

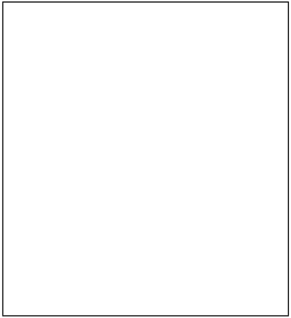
Obvious approach for refuting $F[\oplus]$: mimic refutation of F

Key Technical Result: Substitution Theorem

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

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





Key Technical Result: Substitution Theorem

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

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

x

$\overline{x} \lor y$

Key Technical Result: Substitution Theorem

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

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

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

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

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

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

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

Sketch of Proof of Substitution Theorem

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

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

Pieces Together: Substitution + 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 [Nordström '10]
- to get tight trade-offs, showing that resolution can sometimes do better than black-only pebbling [Nordström '10]

Pieces Together: Substitution + 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 [Nordström '10]
- to get tight trade-offs, showing that resolution can sometimes do better than black-only pebbling [Nordström '10]

Pieces Together: Substitution + 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 [Nordström '10]
- to get tight trade-offs, showing that resolution can sometimes do better than black-only pebbling [Nordström '10]

Some Open Problems

- Many remaining open (theoretical) questions about space in proof complexity
- See recent survey *Pebble Games, Proof Complexity, and Time-Space Trade-offs* at my webpage for details
- In this talk, want to focus on main applied question

Is Tractability Captured by Space Complexity?

Open Question

Do our trade-off phenomena show up in real life for state-of-the-art SAT-solvers run on pebbling contradictions?

That is, does space complexity capture hardness?

Space suggested as hardness measure in [Ansótegui et al.'08]

Some results in [Sabharwal et al.'03] indicate pebbling formulas hard for SAT-solvers at that time

Note that pebbling formulas are always extremely easy with respect to length (and width), so hardness in practice would be intriguing

Is Tractability Captured by Space Complexity?

Open Question

Do our trade-off phenomena show up in real life for state-of-the-art SAT-solvers run on pebbling contradictions?

That is, does space complexity capture hardness?

Space suggested as hardness measure in [Ansótegui et al.'08]

Some results in [Sabharwal et al.'03] indicate pebbling formulas hard for SAT-solvers at that time

Note that pebbling formulas are always extremely easy with respect to length (and width), so hardness in practice would be intriguing

Is Tractability Captured by Space Complexity?

Open Question

Do our trade-off phenomena show up in real life for state-of-the-art SAT-solvers run on pebbling contradictions?

That is, does space complexity capture hardness?

Space suggested as hardness measure in [Ansótegui et al.'08]

Some results in [Sabharwal et al.'03] indicate pebbling formulas hard for SAT-solvers at that time

Note that pebbling formulas are always extremely easy with respect to length (and width), so hardness in practice would be intriguing

Summing up

- Strong resolution time-space trade-offs for wide range of parameters
- Results also extend to stronger k-DNF resolution proof systems
- Main (applied) open question: tractability $\approx$ space complexity?

Thank you for your attention!