Time-space trade-offs in proof complexity
Lecture 2

Jakob Nordström

KTH Royal Institute of Technology

17th Estonian Winter School in Computer Science
Palmse, Estonia
February 26 – March 2, 2012

Goal of Today's Lecture

- Focus on the resolution proof system
- Quick recap of what was said last time
- Brief overview of what is known for proof length and proof space
- Prove length-space trade-offs for resolution (or rather: sketch proofs)
- Discuss extensions to polynomial calculus

Some Notation and Terminology

- Literal a: variable x or its negation $\overline{x}$
- Clause $C = a_1 \lor \ldots \lor a_k$: set of literals
  At most k literals: k-clause
- CNF formula $F = C_1 \land \ldots \land C_m$: set of clauses
  k-CNF formula: CNF formula consisting of k-clauses
- $F \vDash D$: semantical implication, $\alpha(F)$ true $\Rightarrow \alpha(D)$ true
  for all truth value assignments α
- $[n] = \{1, 2, \ldots, n\}$

This course: focus on k-CNF formulas for $k = \mathcal{O}(1)$
(Avoids annoying technicalities, and can always convert to k-CNF anyway)

Resolution Revisited

Last time we talked about a resolution refutations as a sequence of clause configurations $\{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ (snapshots of what's on the board)

For all t, $\mathbb{D}_t$ obtained from $\mathbb{D}_{t-1}$ by one of the following derivation steps:

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

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

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

But if we don't care about space, then we can view a resolution refutation as simply a listing of the clauses (i.e., no erasures)

Resolution Proof System (Ignoring Space)

Resolution derivation $\pi : F \vdash A$ of clause A from F:
Sequence of clauses $\pi = \{D_1, \ldots, D_s\}$ such that $D_s = A$ and each line D_i, $1 \leq i \leq s$, is either

- a clause $C \in F$ (an axiom)
- a resolvent derived from clauses D_j, D_k in π (with $j, k < i$) by the resolution rule

$$\frac{B \vee x \quad C \vee \overline{x}}{B \vee C}$$

  resolving on the variable x

Resolution refutation of CNF formula F:
Derivation of empty clause $\perp$ (clause with no literals) from F

Example Resolution Refutation

$$F = (x \lor z) \land (\overline{z} \lor y) \land (x \lor \overline{y} \lor u) \land (\overline{y} \lor \overline{u}) \land (u \lor v) \land (\overline{x} \lor \overline{v}) \land (\overline{u} \lor w) \land (\overline{x} \lor \overline{u} \lor \overline{w})$$

1.	$x \lor z$	Axiom	9.	$x \lor y$	Res(1, 2)
2.	$\overline{z} \lor y$	Axiom	10.	$x \lor \overline{y}$	Res(3, 4)
3.	$x \lor \overline{y} \lor u$	Axiom	11.	$\overline{x} \lor u$	Res(5, 6)
4.	$\overline{y} \lor \overline{u}$	Axiom	12.	$\overline{x} \lor \overline{u}$	Res(7, 8)
5.	$u \lor v$	Axiom	13.	x	Res(9, 10)
6.	$\overline{x} \lor \overline{v}$	Axiom	14.	$\overline{x}$	Res(11, 12)
7.	$\overline{u} \lor w$	Axiom	15.	$\bot$	Res(13, 14)
8.	$\overline{x} \lor \overline{u} \lor \overline{w}$	Axiom			

Resolution Sound and Complete

Resolution is sound and implicationally complete.

Sound If there is a resolution derivation $\pi : F \vdash A$ then $F \vDash A$

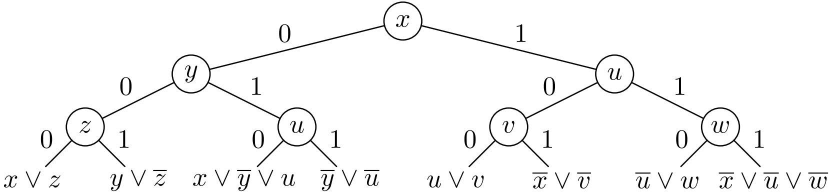
Complete If $F \vDash A$ then there is a resolution derivation $\pi : F \vdash A'$ for some $A' \subseteq A$.

In particular:

$$F \text{ is unsatisfiable} \Leftrightarrow \exists \text{ resolution refutation of } F$$

Completeness of Resolution: Proof by Example

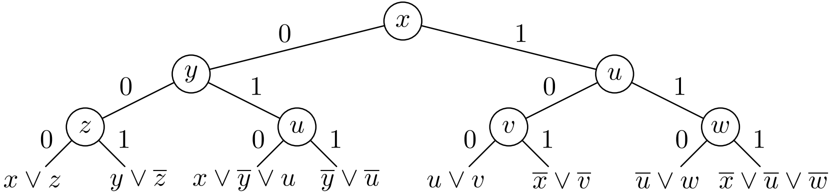
Decision tree:



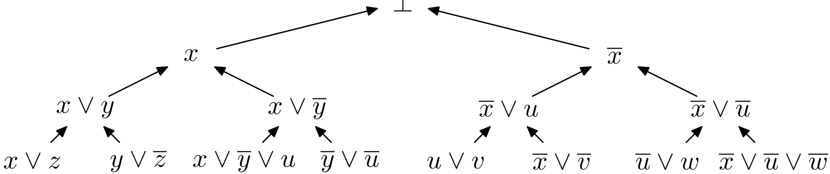
Resulting resolution refutation:

Completeness of Resolution: Proof by Example

Decision tree:



Resulting resolution refutation:



Derivation Graph and Tree-Like Derivations

Derivation graph G_π of a resolution derivation π:
directed acyclic graph (DAG) with

- vertices: clauses of the derivations
- edges: from $B \lor x$ and $C \lor \overline{x}$ to $B \lor C$ for each application of the resolution rule

A resolution derivation π is tree-like if G_π is a tree
(We can make copies of axiom clauses to make G_π into a tree)

Example

Our example resolution proof is tree-like.
(The derivation graph is on the previous slide.)

Derivation Graph and Tree-Like Derivations

Derivation graph G_π of a resolution derivation π:
directed acyclic graph (DAG) with

- vertices: clauses of the derivations
- edges: from $B \lor x$ and $C \lor \overline{x}$ to $B \lor C$ for each application of the resolution rule

A resolution derivation π is tree-like if G_π is a tree
(We can make copies of axiom clauses to make G_π into a tree)

Example

Our example resolution proof is tree-like.
(The derivation graph is on the previous slide.)

Complexity Measures of Interest: Length and Space

- Length: Lower bound on time for SAT solver
  (very straightforward connection)
- Space: Lower bound on memory for SAT solver
  (requires more of an argument — will be happy to elaborate offline)

Length $L_{\mathcal{R}}$

\# clauses written on blackboard counted with repetitions

Space

Several ways of measuring — will mainly be interested in two measures

x
$\overline{y} \lor z$
$\overline{v} \lor \overline{w} \lor y$

Clause space $Sp_{\mathcal{R}}$: 3
Total space $TotSp_{\mathcal{R}}$: 6

Complexity Measures of Interest: Length and Space

- Length: Lower bound on time for SAT solver
  (very straightforward connection)
- Space: Lower bound on memory for SAT solver
  (requires more of an argument — will be happy to elaborate offline)

Length $L_{\mathcal{R}}$

\# clauses written on blackboard counted with repetitions

Space

Several ways of measuring — will mainly be interested in two measures

x

$\overline{y} \lor z$

$\overline{v} \lor \overline{w} \lor y$

Clause space $Sp_{\mathcal{R}}$: 3

Total space $TotSp_{\mathcal{R}}$: 6

Complexity Measures of Interest: Length and Space

- Length: Lower bound on time for SAT solver
  (very straightforward connection)
- Space: Lower bound on memory for SAT solver
  (requires more of an argument — will be happy to elaborate offline)

Length $L_{\mathcal{R}}$
\# clauses written on blackboard counted with repetitions

Space
Several ways of measuring — will mainly be interested in two measures

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

Clause space $Sp_{\mathcal{R}}$: 3
Total space $TotSp_{\mathcal{R}}$: 6

Complexity Measures of Interest: Length and Space

- Length: Lower bound on time for SAT solver
  (very straightforward connection)
- Space: Lower bound on memory for SAT solver
  (requires more of an argument — will be happy to elaborate offline)

Length $L_{\mathcal{R}}$

\# clauses written on blackboard counted with repetitions

Space

Several ways of measuring — will mainly be interested in two measures

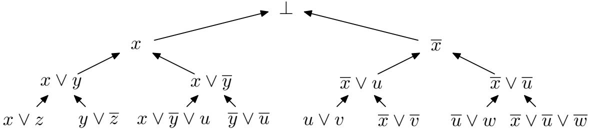
$$x^{1}$$
$$\overline{y}^{2} \lor z^{3}$$
$$\overline{v}^{4} \lor \overline{w}^{5} \lor y^{6}$$

Clause space $Sp_{\mathcal{R}}$: 3

Total space $TotSp_{\mathcal{R}}$: 6

Length and Space Bounds for Resolution (1 / 2)

Let n = size of formula



$\leq n$ variables $\Rightarrow$
decision tree size $\leq 2^{n+1}$ and height $\leq n$

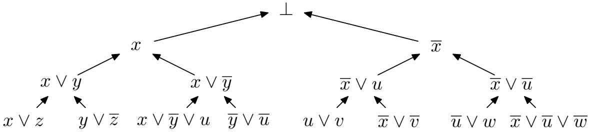
By induction: Clause at root of subtree of height h derivable in space $h + 2$

- Derive left child clause in space $h + 1$ and keep in memory
- Derive right child clause in space $1 + (h + 1)$
- Resolve the two children clauses to get root clause

Hence:

$$L_{\mathcal{R}}(F \vdash \bot) = \exp(\mathcal{O}(n))$$
$$Sp_{\mathcal{R}}(F \vdash \bot) = \mathcal{O}(n)$$

Length and Space Bounds for Resolution (1 / 2)



Let $n =$ size of formula

$\leq n$ variables $\Rightarrow$
decision tree size $\leq 2^{n+1}$ and height $\leq n$

By induction: Clause at root of subtree of height h derivable in space $h + 2$

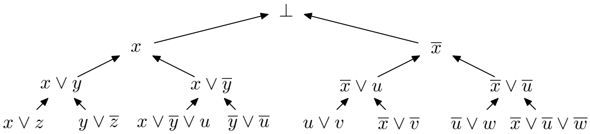
- Derive left child clause in space $h + 1$ and keep in memory
- Derive right child clause in space $1 + (h + 1)$
- Resolve the two children clauses to get root clause

Hence:

$$L_{\mathcal{R}}(F \vdash \bot) = \exp(\mathcal{O}(n))$$
$$Sp_{\mathcal{R}}(F \vdash \bot) = \mathcal{O}(n)$$

Length and Space Bounds for Resolution (1 / 2)

Let n = size of formula



$\leq n$ variables $\Rightarrow$
decision tree size $\leq 2^{n+1}$ and height $\leq n$

By induction: Clause at root of subtree of height h derivable in space $h + 2$

- Derive left child clause in space $h + 1$ and keep in memory
- Derive right child clause in space $1 + (h + 1)$
- Resolve the two children clauses to get root clause

Hence:

$$L_{\mathcal{R}}(F \vdash \bot) = \exp(\mathcal{O}(n))$$
$$Sp_{\mathcal{R}}(F \vdash \bot) = \mathcal{O}(n)$$

Length and Space Bounds for Resolution (2 / 2)

($n =$ size of formula)

Length: at most exponential in n
Matching lower bounds up to constant factors in exponent
[Urquhart '87, Chvátal & Szemerédi '88]

Clause space: at most linear in n
Matching lower bounds up to constant factors
[Torán '99, Alekhnovich et al. '00]

Total space: at most quadratic in n
No better lower bounds than linear in n!?

[Sidenote: space bounds hold even for "magic algorithms" always making optimal choices — so might be much stronger in practice]

Length and Space Bounds for Resolution (2 / 2)

($n =$ size of formula)

Length: at most exponential in n
Matching lower bounds up to constant factors in exponent
[Urquhart '87, Chvátal & Szemerédi '88]

Clause space: at most linear in n
Matching lower bounds up to constant factors
[Torán '99, Alekhnovich et al. '00]

Total space: at most quadratic in n
No better lower bounds than linear in n!?

[Sidenote: space bounds hold even for "magic algorithms" always making optimal choices — so might be much stronger in practice]

Comparing Length and Space

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

- Length exponential in formula size in worst case
- Clause space at most linear
- So natural to compare space to logarithm of length

Length-Space Correlations and/or Trade-offs?

$\exists$ constant space refutation $\Rightarrow$ $\exists$ polynomial length refutation [Atserias & Dalmau '03]

For tree-like resolution: any polynomial length refutation can be carried out in logarithmic space [Esteban & Torán '99]

So essentially no trade-offs for tree-like resolution

Does short length imply small space for general resolution?
Open for quite a while — even no consensus on likely "right answer"

Nothing known about length-space trade-offs for resolution refutations in the general, unrestricted proof system

(Some trade-off results in restricted settings in [Ben-Sasson '02, Nordström '07])

Length-Space Correlations and/or Trade-offs?

$\exists$ constant space refutation $\Rightarrow$ $\exists$ polynomial length refutation [Atserias & Dalmau '03]

For tree-like resolution: any polynomial length refutation can be carried out in logarithmic space [Esteban & Torán '99]

So essentially no trade-offs for tree-like resolution

Does short length imply small space for general resolution?
Open for quite a while — even no consensus on likely "right answer"

Nothing known about length-space trade-offs for resolution refutations in the general, unrestricted proof system

(Some trade-off results in restricted settings in [Ben-Sasson '02, Nordström '07])

Length-Space Correlations and/or Trade-offs?

$\exists$ constant space refutation $\Rightarrow$ $\exists$ polynomial length refutation [Atserias & Dalmau '03]

For tree-like resolution: any polynomial length refutation can be carried out in logarithmic space [Esteban & Torán '99]

So essentially no trade-offs for tree-like resolution

Does short length imply small space for general resolution?
Open for quite a while — even no consensus on likely "right answer"

Nothing known about length-space trade-offs for resolution refutations in the general, unrestricted proof system

(Some trade-off results in restricted settings in [Ben-Sasson '02, Nordström '07])

Length-Space Correlations and/or Trade-offs?

$\exists$ constant space refutation $\Rightarrow$ $\exists$ polynomial length refutation [Atserias & Dalmau '03]

For tree-like resolution: any polynomial length refutation can be carried out in logarithmic space [Esteban & Torán '99]

So essentially no trade-offs for tree-like resolution

Does short length imply small space for general resolution?
Open for quite a while — even no consensus on likely "right answer"

Nothing known about length-space trade-offs for resolution refutations in the general, unrestricted proof system

(Some trade-off results in restricted settings in [Ben-Sasson '02, Nordström '07])

1st result today: An Optimal Length-Space Separation

Length and space in resolution are "completely uncorrelated"

Theorem (Ben-Sasson & Nordström '08)

There are k-CNF formula families of size n with

- *refutation length* $\mathcal{O}(n)$
- *refutation clause space* $\Omega(n/\log n)$

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

2nd result today: Length-Space Trade-offs

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

Results hold for

- resolution
- even stronger proof systems (which we won't go into here)

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

Simple, explicit formulas

(Also some very nice follow-up work in [Beame, Beck & Impagliazzo '12] that we won't have time to go into)

One Example: Robust Trade-offs for Small Space

Theorem (Ben-Sasson & Nordström '11 (informal))

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

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

One Example: Robust Trade-offs for Small Space

Theorem (Ben-Sasson & Nordström '11 (informal))

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

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

One Example: Robust Trade-offs for Small Space

Theorem (Ben-Sasson & Nordström '11 (informal))

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

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

One Example: Robust Trade-offs for Small Space

Theorem (Ben-Sasson & Nordström '11 (informal))

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

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

One Example: Robust Trade-offs for Small Space

Theorem (Ben-Sasson & Nordström '11 (informal))

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

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

And an open problem:

Open Problem

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

Plan for the Rest of This Lecture

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

A Detour into Combinatorial Games

Want to find formulas that

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

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

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

Some quick graph terminology

- DAGs consist of vertices with directed edges between them
- vertices with no incoming edges: sources
- vertices with no outgoing edges: sinks

A Detour into Combinatorial Games

Want to find formulas that

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

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

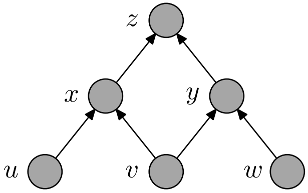
- Time needed for calculation: # pebbling moves
- Space needed for calculation: max # pebbles required

Some quick graph terminology

- DAGs consist of vertices with directed edges between them
- vertices with no incoming edges: sources
- vertices with no outgoing edges: sinks

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

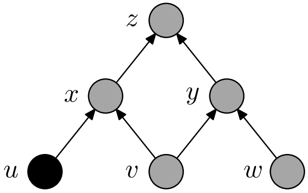


# moves	0
Current # pebbles	0
Max # pebbles so far	0

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

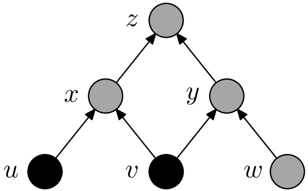


# moves	1
Current # pebbles	1
Max # pebbles so far	1

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

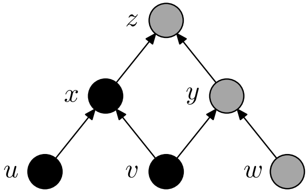


# moves	2
Current # pebbles	2
Max # pebbles so far	2

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

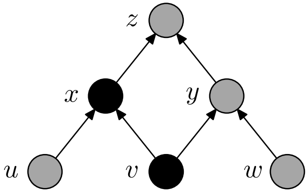


# moves	3
Current # pebbles	3
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

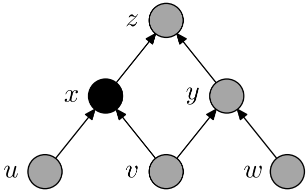


# moves	4
Current # pebbles	2
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

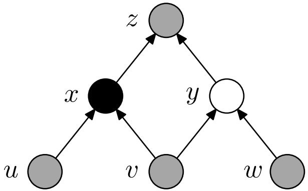


# moves	5
Current # pebbles	1
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

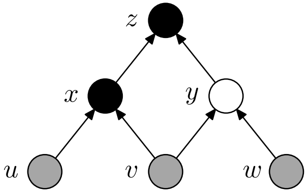


# moves	6
Current # pebbles	2
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

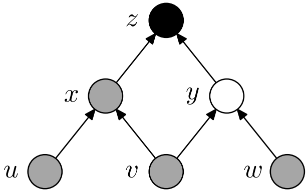


# moves	7
Current # pebbles	3
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

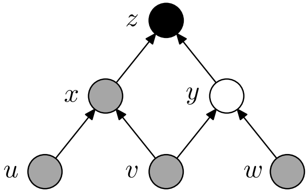


# moves	8
Current # pebbles	2
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

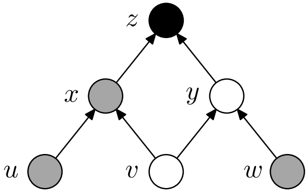


# moves	8
Current # pebbles	2
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

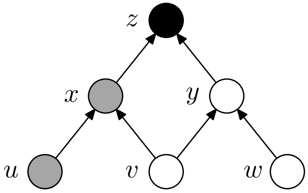


# moves	9
Current # pebbles	3
Max # pebbles so far	3

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

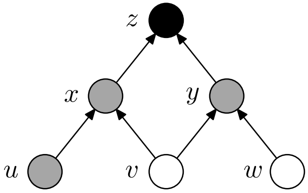


# moves	10
Current # pebbles	4
Max # pebbles so far	4

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

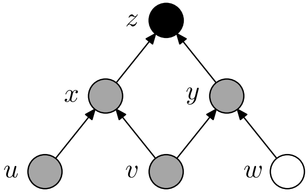


# moves	11
Current # pebbles	3
Max # pebbles so far	4

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)

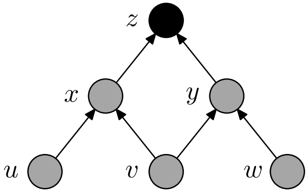


# moves	12
Current # pebbles	2
Max # pebbles so far	4

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

The Black-White Pebble Game

Goal: get single black pebble on sink z of DAG G (with constant fan-in)



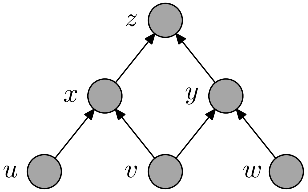
# moves	13
Current # pebbles	1
Max # pebbles so far	4

1. Can place black pebble on (empty) vertex v if all predecessors (vertices with edges to v) 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 if all predecessors have pebbles

Pebbling Contradiction

CNF formula encoding pebble game on DAG G

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



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

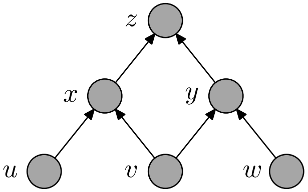
Studied by [Bonet et al. '98, Raz & McKenzie '99, Ben-Sasson & Wigderson '99] and others

We want to show that pebbling properties of DAGs somehow carry over to resolution refutations of pebbling contradictions

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

We want to show that pebbling properties of DAGs somehow carry over to resolution refutations of pebbling contradictions

Interpreting Refutations as Black-White Pebblings

Black-white pebbling models non-deterministic computation (where one can guess partial results and verify later)

- black pebbles ⇔ computed results
- white pebbles ⇔ guesses needing to be verified

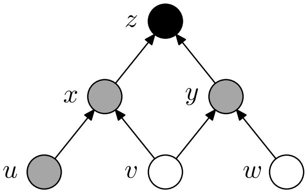
"Know z assuming v, w"

Corresponds to $(v \wedge w) \rightarrow z$, i.e., blackboard clause $\overline{v} \vee \overline{w} \vee z$

So translate clauses to pebbles by:
unnegated variable ⇒ black pebble
negated variable ⇒ white pebble

Black-white pebbling models non-deterministic computation (where one can guess partial results and verify later)

- black pebbles $\Leftrightarrow$ computed results
- white pebbles $\Leftrightarrow$ guesses needing to be verified



"Know z assuming v, w"

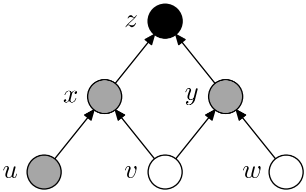
Corresponds to $(v \wedge w) \rightarrow z$, i.e., blackboard clause $\overline{v} \vee \overline{w} \vee z$

So translate clauses to pebbles by:
unnegated variable $\Rightarrow$ black pebble
negated variable $\Rightarrow$ white pebble

Interpreting Refutations as Black-White Pebblings

Black-white pebbling models non-deterministic computation (where one can guess partial results and verify later)

- black pebbles $\Leftrightarrow$ computed results
- white pebbles $\Leftrightarrow$ guesses needing to be verified



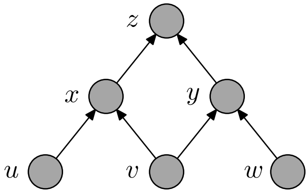
"Know z assuming v, w"

Corresponds to $(v \wedge w) \rightarrow z$, i.e., blackboard clause $\boxed{\overline{v} \vee \overline{w} \vee z}$

So translate clauses to pebbles by:
unnegated variable $\Rightarrow$ black pebble
negated variable $\Rightarrow$ white pebble

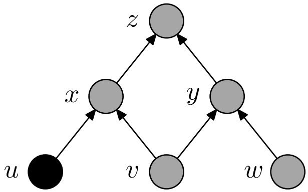
Example of Refutation-Pebbling Correspondence

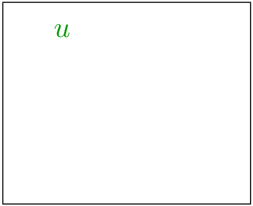
1. u
2. v
3. w
4. $\overline{u} \lor \overline{v} \lor x$
5. $\overline{v} \lor \overline{w} \lor y$
6. $\overline{x} \lor \overline{y} \lor z$
7. $\overline{z}$



Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

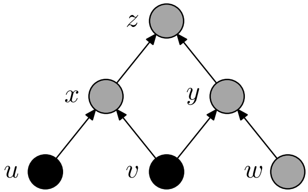




Download axiom 1: u

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

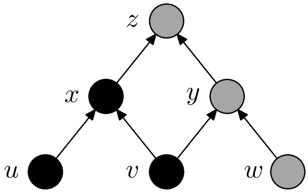
u
v

Download axiom 1: u

Download axiom 2: v

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



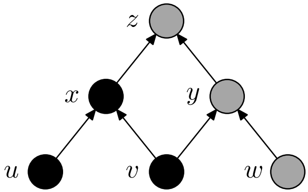
---

u
v
$\overline{u} \lor \overline{v} \lor x$

Download axiom 1: u
Download axiom 2: v
Download axiom 4: $\overline{u} \lor \overline{v} \lor x$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

u

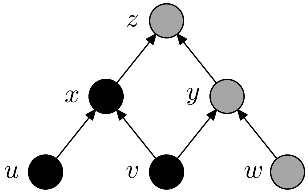
v

$\overline{u} \lor \overline{v} \lor x$

Download axiom 1: u
Download axiom 2: v
Download axiom 4: $\overline{u} \lor \overline{v} \lor x$
Infer $\overline{v} \lor x$ from
u and $\overline{u} \lor \overline{v} \lor x$

Example of Refutation-Pebbling Correspondence

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



---

u

v

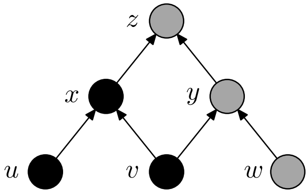
$\overline{u} \vee \overline{v} \vee x$

$\overline{v} \vee x$

Download axiom 1: u
Download axiom 2: v
Download axiom 4: $\overline{u} \vee \overline{v} \vee x$
Infer $\overline{v} \vee x$ from
u and $\overline{u} \vee \overline{v} \vee x$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



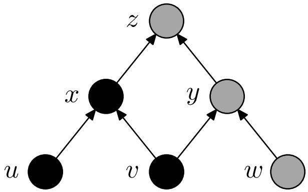
---

u
v
$\overline{u} \lor \overline{v} \lor x$
$\overline{v} \lor x$

Download axiom 2: v
Download axiom 4: $\overline{u} \lor \overline{v} \lor x$
Infer $\overline{v} \lor x$ from
 u and $\overline{u} \lor \overline{v} \lor x$
Erase the clause $\overline{u} \lor \overline{v} \lor x$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

u

v

$\overline{v} \lor x$

Download axiom 2: v

Download axiom 4: $\overline{u} \lor \overline{v} \lor x$

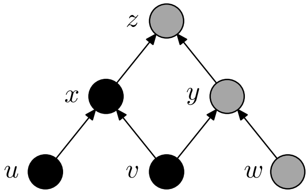
Infer $\overline{v} \lor x$ from

u and $\overline{u} \lor \overline{v} \lor x$

Erase the clause $\overline{u} \lor \overline{v} \lor x$

Example of Refutation-Pebbling Correspondence

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



---

u

v

$\overline{v} \vee x$

Download axiom 4: $\overline{u} \vee \overline{v} \vee x$

Infer $\overline{v} \vee x$ from

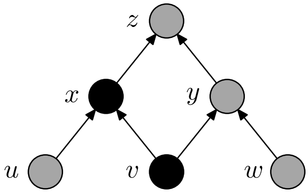
u and $\overline{u} \vee \overline{v} \vee x$

Erase the clause $\overline{u} \vee \overline{v} \vee x$

Erase the clause u

Example of Refutation-Pebbling Correspondence

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



---

v

$\overline{v} \vee x$

Download axiom 4: $\overline{u} \vee \overline{v} \vee x$

Infer $\overline{v} \vee x$ from

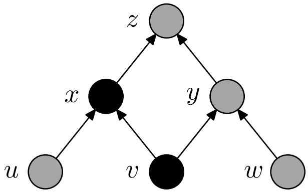
u and $\overline{u} \vee \overline{v} \vee x$

Erase the clause $\overline{u} \vee \overline{v} \vee x$

Erase the clause u

Example of Refutation-Pebbling Correspondence

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



---

v

$\overline{v} \vee x$

u and $\overline{u} \vee \overline{v} \vee x$

Erase the clause $\overline{u} \vee \overline{v} \vee x$

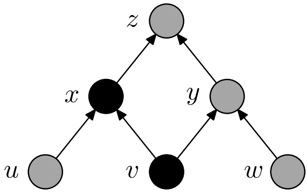
Erase the clause u

Infer x from

v and $\overline{v} \vee x$

Example of Refutation-Pebbling Correspondence

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



---

v
$\overline{v} \vee x$
x

u and $\overline{u} \vee \overline{v} \vee x$

Erase the clause $\overline{u} \vee \overline{v} \vee x$

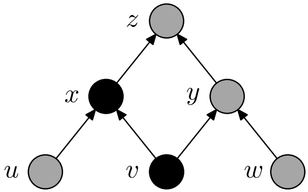
Erase the clause u

Infer x from

v and $\overline{v} \vee x$

Example of Refutation-Pebbling Correspondence

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



---

v

$\overline{v} \vee x$

x

Erase the clause $\overline{u} \vee \overline{v} \vee x$

Erase the clause u

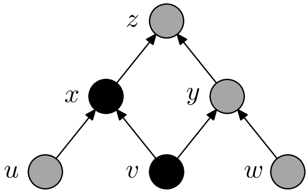
Infer x from

v and $\overline{v} \vee x$

Erase the clause $\overline{v} \vee x$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

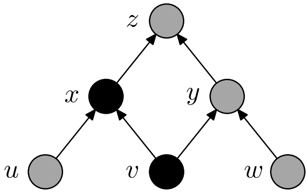
v

x

Erase the clause $\overline{u} \lor \overline{v} \lor x$
Erase the clause u
Infer x from
v and $\overline{v} \lor x$
Erase the clause $\overline{v} \lor x$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



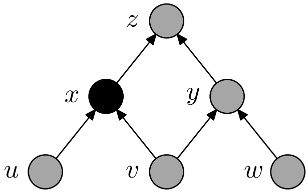
---

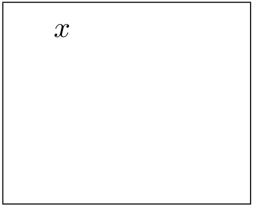
v
x

Erase the clause u
Infer x from
v and $\overline{v} \lor x$
Erase the clause $\overline{v} \lor x$
Erase the clause v

Example of Refutation-Pebbling Correspondence

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

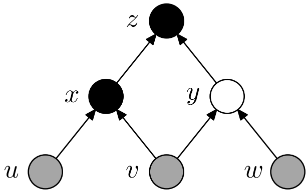




Erase the clause u
Infer x from
v and $\overline{v} \vee x$
Erase the clause $\overline{v} \vee x$
Erase the clause v

Example of Refutation-Pebbling Correspondence

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



---

x

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

Infer x from

v and $\overline{v} \vee x$

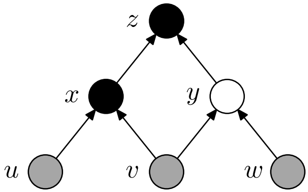
Erase the clause $\overline{v} \vee x$

Erase the clause v

Download axiom 6: $\overline{x} \vee \overline{y} \vee z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

x

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

Erase the clause $\overline{v} \lor x$

Erase the clause v

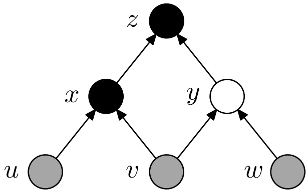
Download axiom 6: $\overline{x} \lor \overline{y} \lor z$

Infer $\overline{y} \lor z$ from

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

Example of Refutation-Pebbling Correspondence

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



x

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

$\overline{y} \vee z$

Erase the clause $\overline{v} \vee x$

Erase the clause v

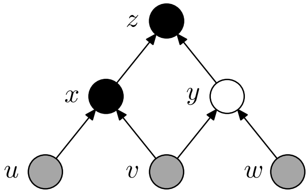
Download axiom 6: $\overline{x} \vee \overline{y} \vee z$

Infer $\overline{y} \vee z$ from

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

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

x

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

$\overline{y} \lor z$

Erase the clause v

Download axiom 6: $\overline{x} \lor \overline{y} \lor z$

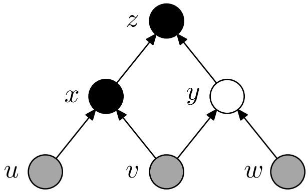
Infer $\overline{y} \lor z$ from

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

Erase the clause $\overline{x} \lor \overline{y} \lor z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

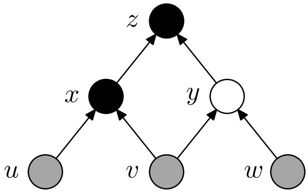
x

$\overline{y} \lor z$

Erase the clause v
Download axiom 6: $\overline{x} \lor \overline{y} \lor z$
Infer $\overline{y} \lor z$ from
x and $\overline{x} \lor \overline{y} \lor z$
Erase the clause $\overline{x} \lor \overline{y} \lor z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

x

$\overline{y} \lor z$

Download axiom 6: $\overline{x} \lor \overline{y} \lor z$

Infer $\overline{y} \lor z$ from

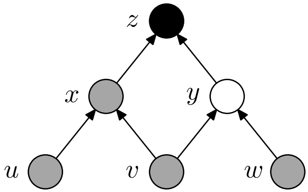
x and $\overline{x} \lor \overline{y} \lor z$

Erase the clause $\overline{x} \lor \overline{y} \lor z$

Erase the clause x

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

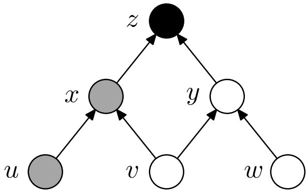


$\overline{y} \lor z$

Download axiom 6: $\overline{x} \lor \overline{y} \lor z$
Infer $\overline{y} \lor z$ from
  x and $\overline{x} \lor \overline{y} \lor z$
Erase the clause $\overline{x} \lor \overline{y} \lor z$
Erase the clause x

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

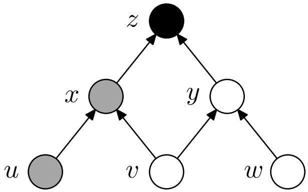


$\overline{y} \lor z$
$\overline{v} \lor \overline{w} \lor y$

Infer $\overline{y} \lor z$ from
x and $\overline{x} \lor \overline{y} \lor z$
Erase the clause $\overline{x} \lor \overline{y} \lor z$
Erase the clause x
Download axiom 5: $\overline{v} \lor \overline{w} \lor y$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



$\overline{y} \lor z$

$\overline{v} \lor \overline{w} \lor y$

Erase the clause $\overline{x} \lor \overline{y} \lor z$

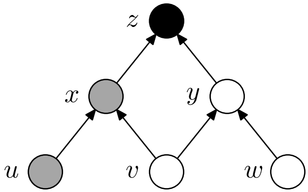
Erase the clause x

Download axiom 5: $\overline{v} \lor \overline{w} \lor y$

Infer $\overline{v} \lor \overline{w} \lor z$ from
$\overline{y} \lor z$ and $\overline{v} \lor \overline{w} \lor y$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

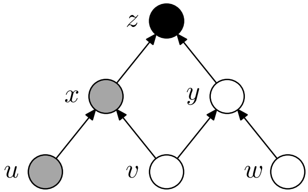


$\overline{y} \lor z$
$\overline{v} \lor \overline{w} \lor y$
$\overline{v} \lor \overline{w} \lor z$

Erase the clause $\overline{x} \lor \overline{y} \lor z$
Erase the clause x
Download axiom 5: $\overline{v} \lor \overline{w} \lor y$
Infer $\overline{v} \lor \overline{w} \lor z$ from
$\overline{y} \lor z$ and $\overline{v} \lor \overline{w} \lor y$

Example of Refutation-Pebbling Correspondence

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

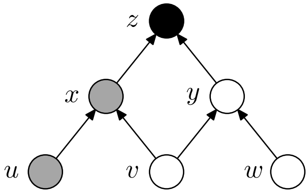


$\overline{y} \vee z$
$\overline{v} \vee \overline{w} \vee y$
$\overline{v} \vee \overline{w} \vee z$

Erase the clause x
Download axiom 5: $\overline{v} \vee \overline{w} \vee y$
Infer $\overline{v} \vee \overline{w} \vee z$ from
$\overline{y} \vee z$ and $\overline{v} \vee \overline{w} \vee y$
Erase the clause $\overline{v} \vee \overline{w} \vee y$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

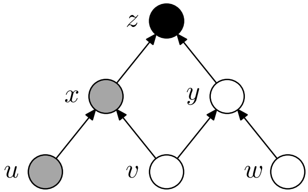


$\overline{y} \lor z$
$\overline{v} \lor \overline{w} \lor z$

Erase the clause x
Download axiom 5: $\overline{v} \lor \overline{w} \lor y$
Infer $\overline{v} \lor \overline{w} \lor z$ from
$\overline{y} \lor z$ and $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{v} \lor \overline{w} \lor y$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



$\overline{y} \lor z$

$\overline{v} \lor \overline{w} \lor z$

Download axiom 5: $\overline{v} \lor \overline{w} \lor y$

Infer $\overline{v} \lor \overline{w} \lor z$ from

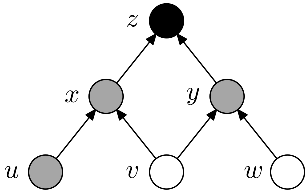
$\overline{y} \lor z$ and $\overline{v} \lor \overline{w} \lor y$

Erase the clause $\overline{v} \lor \overline{w} \lor y$

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

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

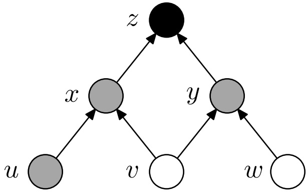


$\overline{v} \lor \overline{w} \lor z$

Download axiom 5: $\overline{v} \lor \overline{w} \lor y$
Infer $\overline{v} \lor \overline{w} \lor z$ from
$\overline{y} \lor z$ and $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{y} \lor z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

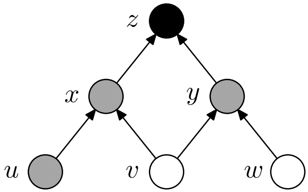


$\overline{v} \lor \overline{w} \lor z$
v

Infer $\overline{v} \lor \overline{w} \lor z$ from
$\overline{y} \lor z$ and $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{y} \lor z$
Download axiom 2: v

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

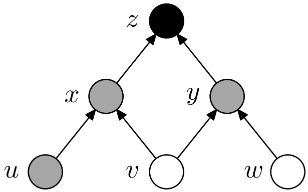


$\overline{v} \lor \overline{w} \lor z$
v
w

$\overline{y} \lor z$ and $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{y} \lor z$
Download axiom 2: v
Download axiom 3: w

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

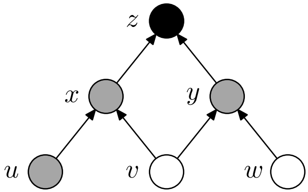


$\overline{v} \lor \overline{w} \lor z$
v
w
$\overline{z}$

Erase the clause $\overline{v} \lor \overline{w} \lor y$
Erase the clause $\overline{y} \lor z$
Download axiom 2: v
Download axiom 3: w
Download axiom 7: $\overline{z}$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

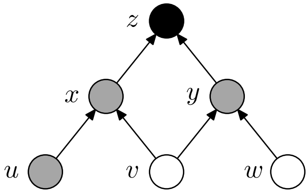


$\overline{v} \lor \overline{w} \lor z$
v
w
$\overline{z}$

Download axiom 2: v
Download axiom 3: w
Download axiom 7: $\overline{z}$
Infer $\overline{w} \lor z$ from
v and $\overline{v} \lor \overline{w} \lor z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

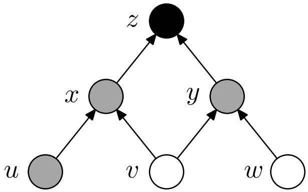


$\overline{v} \lor \overline{w} \lor z$
v
w
$\overline{z}$
$\overline{w} \lor z$

Download axiom 2: v
Download axiom 3: w
Download axiom 7: $\overline{z}$
Infer $\overline{w} \lor z$ from
v and $\overline{v} \lor \overline{w} \lor z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

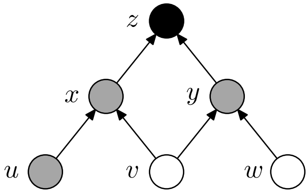


$\overline{v} \lor \overline{w} \lor z$
v
w
$\overline{z}$
$\overline{w} \lor z$

Download axiom 3: w
Download axiom 7: $\overline{z}$
Infer $\overline{w} \lor z$ from
v and $\overline{v} \lor \overline{w} \lor z$
Erase the clause v

Example of Refutation-Pebbling Correspondence

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

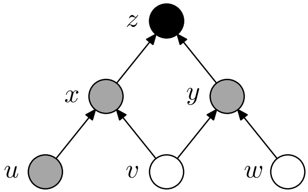


$\overline{v} \vee \overline{w} \vee z$
w
$\overline{z}$
$\overline{w} \vee z$

Download axiom 3: w
Download axiom 7: $\overline{z}$
Infer $\overline{w} \vee z$ from
v and $\overline{v} \vee \overline{w} \vee z$
Erase the clause v

Example of Refutation-Pebbling Correspondence

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

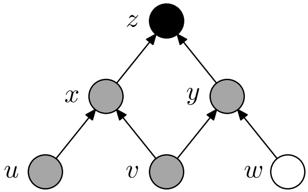


$\overline{v} \vee \overline{w} \vee z$
w
$\overline{z}$
$\overline{w} \vee z$

Download axiom 7: $\overline{z}$
Infer $\overline{w} \vee z$ from
v and $\overline{v} \vee \overline{w} \vee z$
Erase the clause v
Erase the clause $\overline{v} \vee \overline{w} \vee z$

Example of Refutation-Pebbling Correspondence

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



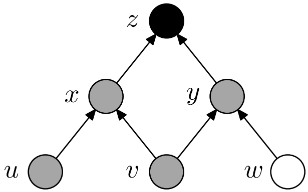
---

w
$\overline{z}$
$\overline{w} \vee z$

Download axiom 7: $\overline{z}$
Infer $\overline{w} \vee z$ from
v and $\overline{v} \vee \overline{w} \vee z$
Erase the clause v
Erase the clause $\overline{v} \vee \overline{w} \vee z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



---

w

$\overline{z}$

$\overline{w} \lor z$

v and $\overline{v} \lor \overline{w} \lor z$

Erase the clause v

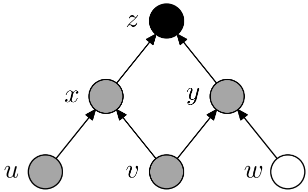
Erase the clause $\overline{v} \lor \overline{w} \lor z$

Infer z from

w and $\overline{w} \lor z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



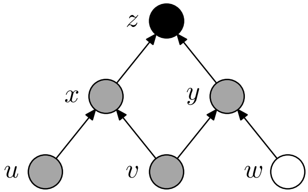
---

w
$\overline{z}$
$\overline{w} \lor z$
z

v and $\overline{v} \lor \overline{w} \lor z$
Erase the clause v
Erase the clause $\overline{v} \lor \overline{w} \lor z$
Infer z from
w and $\overline{w} \lor z$

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



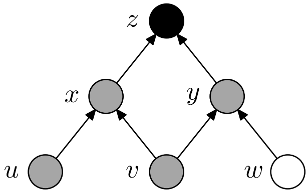
---

w
$\overline{z}$
$\overline{w} \lor z$
z

Erase the clause v
Erase the clause $\overline{v} \lor \overline{w} \lor z$
Infer z from
    w and $\overline{w} \lor z$
Erase the clause w

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$

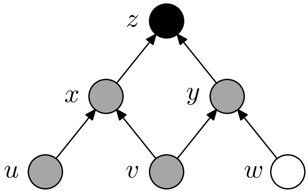


$\overline{z}$
$\overline{w} \lor z$
z

Erase the clause v
Erase the clause $\overline{v} \lor \overline{w} \lor z$
Infer z from
  w and $\overline{w} \lor z$
Erase the clause w

Example of Refutation-Pebbling Correspondence

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. $\overline{z}$



$\overline{z}$

$\overline{w} \lor z$

z

Erase the clause $\overline{v} \lor \overline{w} \lor z$

Infer z from

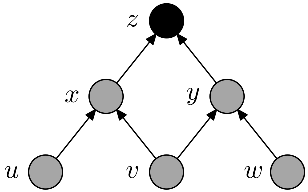
w and $\overline{w} \lor z$

Erase the clause w

Erase the clause $\overline{w} \lor z$

Example of Refutation-Pebbling Correspondence

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



$\overline{z}$
z

Erase the clause $\overline{v} \vee \overline{w} \vee z$

Infer z from

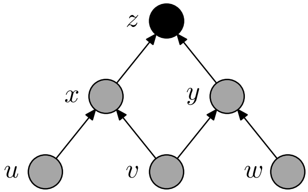
  w and $\overline{w} \vee z$

Erase the clause w

Erase the clause $\overline{w} \vee z$

Example of Refutation-Pebbling Correspondence

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



$\overline{z}$

z

w and $\overline{w} \vee z$

Erase the clause w

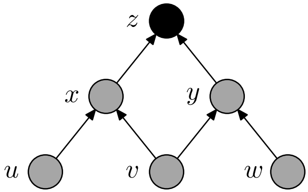
Erase the clause $\overline{w} \vee z$

Infer $\bot$ from

$\overline{z}$ and z

Example of Refutation-Pebbling Correspondence

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



$\overline{z}$
z
$\bot$

w and $\overline{w} \vee z$
Erase the clause w
Erase the clause $\overline{w} \vee z$
Infer $\bot$ from
$\overline{z}$ and z

From Resolution to Pebbling

Theorem (Adapted from [Ben-Sasson '02])

Any resolution refutation translates into black-white pebbling with

- $\#$ *moves* $= \mathcal{O}($*refutation length*$)$
- $\#$ *pebbles* $= \mathcal{O}(\#$ *variables on board*$)$

Proof sketch.

For every clause configuration $\mathbb{D}_t$

- black-pebble vertices with positive literals
- white-pebble vertices with negativt but no positive literals

Argue that for $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, pebbling placements and removals are legal

Download: Always pebbles below new black pebble

Inference: No change in pebbles

Erasure: Only erase after resolution step; only variable resolved over disappears $\Rightarrow$ corresponds to black vertex — OK $\square$

From Resolution to Pebbling

Theorem (Adapted from [Ben-Sasson '02])

Any resolution refutation translates into black-white pebbling with

- *# moves* $= \mathcal{O}($*refutation length*$)$
- *# pebbles* $= \mathcal{O}($*# variables on board*$)$

Proof sketch.

For every clause configuration $\mathbb{D}_t$

- black-pebble vertices with positive literals
- white-pebble vertices with negativt but no positive literals

Argue that for $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, pebbling placements and removals are legal

Download: Always pebbles below new black pebble

Inference: No change in pebbles

Erasure: Only erase after resolution step; only variable resolved over disappears $\Rightarrow$ corresponds to black vertex — OK $\square$

From Resolution to Pebbling

Theorem (Adapted from [Ben-Sasson '02])

Any resolution refutation translates into black-white pebbling with

- $\#$ *moves* $= \mathcal{O}($*refutation length*$)$
- $\#$ *pebbles* $= \mathcal{O}(\#$ *variables on board*$)$

Proof sketch.

For every clause configuration $\mathbb{D}_t$

- black-pebble vertices with positive literals
- white-pebble vertices with negativt but no positive literals

Argue that for $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, pebbling placements and removals are legal

Download: Always pebbles below new black pebble

Inference: No change in pebbles

Erasure: Only erase after resolution step; only variable resolved over disappears $\Rightarrow$ corresponds to black vertex — OK □

From Resolution to Pebbling

Theorem (Adapted from [Ben-Sasson '02])

Any resolution refutation translates into black-white pebbling with

- $\#$ *moves* $= \mathcal{O}($*refutation length*$)$
- $\#$ *pebbles* $= \mathcal{O}(\#$ *variables on board*$)$

Proof sketch.

For every clause configuration $\mathbb{D}_t$

- black-pebble vertices with positive literals
- white-pebble vertices with negativt but no positive literals

Argue that for $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, pebbling placements and removals are legal

Download: Always pebbles below new black pebble

Inference: No change in pebbles

Erasure: Only erase after resolution step; only variable resolved over disappears $\Rightarrow$ corresponds to black vertex — OK □

From Resolution to Pebbling

Theorem (Adapted from [Ben-Sasson '02])

Any resolution refutation translates into black-white pebbling with

- $\#$ *moves* $= \mathcal{O}($*refutation length*$)$
- $\#$ *pebbles* $= \mathcal{O}(\#$ *variables on board*$)$

Proof sketch.

For every clause configuration $\mathbb{D}_t$

- black-pebble vertices with positive literals
- white-pebble vertices with negativt but no positive literals

Argue that for $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, pebbling placements and removals are legal

Download: Always pebbles below new black pebble

Inference: No change in pebbles

Erasure: Only erase after resolution step; only variable resolved over disappears $\Rightarrow$ corresponds to black vertex — OK □

From Resolution to Pebbling

Theorem (Adapted from [Ben-Sasson '02])

Any resolution refutation translates into black-white pebbling with

- $\#$ *moves* $= \mathcal{O}($*refutation length*$)$
- $\#$ *pebbles* $= \mathcal{O}(\#$ *variables on board*$)$

Proof sketch.

For every clause configuration $\mathbb{D}_t$

- black-pebble vertices with positive literals
- white-pebble vertices with negativt but no positive literals

Argue that for $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, pebbling placements and removals are legal

Download: Always pebbles below new black pebble

Inference: No change in pebbles

Erasure: Only erase after resolution step; only variable resolved over disappears $\Rightarrow$ corresponds to black vertex — OK □

From Pebbling to Resolution

Observation (Ben-Sasson et al. '00)

Any black-pebbles-only pebbling translates into resolution refutation with

- *refutation length* $= \mathcal{O}(\#$ *moves*$)$
- *total space* $= \mathcal{O}(\#$ *pebbles*$)$

Proof sketch.

- Invariant: keep clause u in memory for all black-pebbled vertices u
- When source vertex v pebbled, can download source axiom v
- When non-source v is pebbled, all predecessors $u \in pred(v)$ are black
- Download $\bigvee_{u \in pred(v)} \overline{u} \vee v$ and resolve with all clauses u for $u \in pred(v)$ to derive v
- At end of pebbling, z is black-pebbled
- Download sink axiom $\overline{z}$ and resolve with clause z to derive $\perp$ □

From Pebbling to Resolution

Observation (Ben-Sasson et al. '00)

Any black-pebbles-only pebbling translates into resolution refutation with

- *refutation length* $= \mathcal{O}(\#$ *moves*$)$
- *total space* $= \mathcal{O}(\#$ *pebbles*$)$

Proof sketch.

- Invariant: keep clause u in memory for all black-pebbled vertices u
- When source vertex v pebbled, can download source axiom v
- When non-source v is pebbled, all predecessors $u \in pred(v)$ are black
- Download $\bigvee_{u \in pred(v)} \overline{u} \vee v$ and resolve with all clauses u for $u \in pred(v)$ to derive v
- At end of pebbling, z is black-pebbled
- Download sink axiom $\overline{z}$ and resolve with clause z to derive $\bot$ □

From Pebbling to Resolution

Observation (Ben-Sasson et al. '00)

Any black-pebbles-only pebbling translates into resolution refutation with

- *refutation length* $= \mathcal{O}(\#$ *moves*$)$
- *total space* $= \mathcal{O}(\#$ *pebbles*$)$

Proof sketch.

- Invariant: keep clause u in memory for all black-pebbled vertices u
- When source vertex v pebbled, can download source axiom v
- When non-source v is pebbled, all predecessors $u \in pred(v)$ are black
- Download $\bigvee_{u \in pred(v)} \overline{u} \vee v$ and resolve with all clauses u for $u \in pred(v)$ to derive v
- At end of pebbling, z is black-pebbled
- Download sink axiom $\overline{z}$ and resolve with clause z to derive $\perp$ □

From Pebbling to Resolution

Observation (Ben-Sasson et al. '00)

Any black-pebbles-only pebbling translates into resolution refutation with

- *refutation length* $= \mathcal{O}(\#$ *moves*$)$
- *total space* $= \mathcal{O}(\#$ *pebbles*$)$

Proof sketch.

- Invariant: keep clause u in memory for all black-pebbled vertices u
- When source vertex v pebbled, can download source axiom v
- When non-source v is pebbled, all predecessors $u \in pred(v)$ are black
- Download $\bigvee_{u \in pred(v)} \overline{u} \vee v$ and resolve with all clauses u for $u \in pred(v)$ to derive v
- At end of pebbling, z is black-pebbled
- Download sink axiom $\overline{z}$ and resolve with clause z to derive $\perp$ $\square$

From Pebbling to Resolution

Observation (Ben-Sasson et al. '00)

Any black-pebbles-only pebbling translates into resolution refutation with

- *refutation length* $= \mathcal{O}(\#$ *moves*$)$
- *total space* $= \mathcal{O}(\#$ *pebbles*$)$

Proof sketch.

- Invariant: keep clause u in memory for all black-pebbled vertices u
- When source vertex v pebbled, can download source axiom v
- When non-source v is pebbled, all predecessors $u \in pred(v)$ are black
- Download $\bigvee_{u \in pred(v)} \overline{u} \vee v$ and resolve with all clauses u for $u \in pred(v)$ to derive v
- At end of pebbling, z is black-pebbled
- Download sink axiom $\overline{z}$ and resolve with clause z to derive $\perp$ □

From Pebbling to Resolution

Observation (Ben-Sasson et al. '00)

Any black-pebbles-only pebbling translates into resolution refutation with

- *refutation length* $= \mathcal{O}(\#$ *moves*$)$
- *total space* $= \mathcal{O}(\#$ *pebbles*$)$

Proof sketch.

- Invariant: keep clause u in memory for all black-pebbled vertices u
- When source vertex v pebbled, can download source axiom v
- When non-source v is pebbled, all predecessors $u \in pred(v)$ are black
- Download $\bigvee_{u \in pred(v)} \overline{u} \vee v$ and resolve with all clauses u for $u \in pred(v)$ to derive v
- At end of pebbling, z is black-pebbled
- Download sink axiom $\overline{z}$ and resolve with clause z to derive $\perp$ □

From Pebbling to Resolution

Observation (Ben-Sasson et al. '00)

Any black-pebbles-only pebbling translates into resolution refutation with

- *refutation length* $= \mathcal{O}(\#$ *moves*$)$
- *total space* $= \mathcal{O}(\#$ *pebbles*$)$

Proof sketch.

- Invariant: keep clause u in memory for all black-pebbled vertices u
- When source vertex v pebbled, can download source axiom v
- When non-source v is pebbled, all predecessors $u \in pred(v)$ are black
- Download $\bigvee_{u \in pred(v)} \overline{u} \vee v$ and resolve with all clauses u for $u \in pred(v)$ to derive v
- At end of pebbling, z is black-pebbled
- Download sink axiom $\overline{z}$ and resolve with clause z to derive $\perp$ □

But Unfortunately This Totally Doesn't Work...

Unfortunately pebbling contradictions extremely easy w.r.t. clause space!

Theorem (Ben-Sasson '02)

Any pebbling contradiction can be refuted in resolution in linear length and constant clause space simultaneously

Proof sketch.

- Start by resolving $\overline{z}$ and $\bigvee_{u \in pred(z)} \overline{u} \vee z$
- Then, in reverse topological order of vertices v, resolve with pebbling axioms $\bigvee_{u \in pred(v)} \overline{u} \vee v$
- Invariant: One clause in memory; only negative literals; only for vertices preceding v in topological order
- Finally, have one wide clause with negative literals over all sources
- Use source axioms to resolve away these literals one by one □

But Unfortunately This Totally Doesn't Work...

Unfortunately pebbling contradictions extremely easy w.r.t. clause space!

Theorem (Ben-Sasson '02)

Any pebbling contradiction can be refuted in resolution in linear length and constant clause space simultaneously

Proof sketch.

- Start by resolving $\overline{z}$ and $\bigvee_{u \in pred(z)} \overline{u} \vee z$
- Then, in reverse topological order of vertices v, resolve with pebbling axioms $\bigvee_{u \in pred(v)} \overline{u} \vee v$
- Invariant: One clause in memory; only negative literals; only for vertices preceding v in topological order
- Finally, have one wide clause with negative literals over all sources
- Use source axioms to resolve away these literals one by one □

But Unfortunately This Totally Doesn't Work. . .

Unfortunately pebbling contradictions extremely easy w.r.t. clause space!

Theorem (Ben-Sasson '02)

Any pebbling contradiction can be refuted in resolution in linear length and constant clause space simultaneously

Proof sketch.

- Start by resolving $\overline{z}$ and $\bigvee_{u \in pred(z)} \overline{u} \vee z$
- Then, in reverse topological order of vertices v, resolve with pebbling axioms $\bigvee_{u \in pred(v)} \overline{u} \vee v$
- Invariant: One clause in memory; only negative literals; only for vertices preceding v in topological order
- Finally, have one wide clause with negative literals over all sources
- Use source axioms to resolve away these literals one by one □

But Unfortunately This Totally Doesn't Work...

Unfortunately pebbling contradictions extremely easy w.r.t. clause space!

Theorem (Ben-Sasson '02)

Any pebbling contradiction can be refuted in resolution in linear length and constant clause space simultaneously

Proof sketch.

- Start by resolving $\overline{z}$ and $\bigvee_{u \in pred(z)} \overline{u} \vee z$
- Then, in reverse topological order of vertices v, resolve with pebbling axioms $\bigvee_{u \in pred(v)} \overline{u} \vee v$
- Invariant: One clause in memory; only negative literals; only for vertices preceding v in topological order
- Finally, have one wide clause with negative literals over all sources
- Use source axioms to resolve away these literals one by one □

But Unfortunately This Totally Doesn't Work...

Unfortunately pebbling contradictions extremely easy w.r.t. clause space!

Theorem (Ben-Sasson '02)

Any pebbling contradiction can be refuted in resolution in linear length and constant clause space simultaneously

Proof sketch.

- Start by resolving $\overline{z}$ and $\bigvee_{u \in pred(z)} \overline{u} \vee z$
- Then, in reverse topological order of vertices v, resolve with pebbling axioms $\bigvee_{u \in pred(v)} \overline{u} \vee v$
- Invariant: One clause in memory; only negative literals; only for vertices preceding v in topological order
- Finally, have one wide clause with negative literals over all sources
- Use source axioms to resolve away these literals one by one

But Unfortunately This Totally Doesn't Work. . .

Unfortunately pebbling contradictions extremely easy w.r.t. clause space!

Theorem (Ben-Sasson '02)

Any pebbling contradiction can be refuted in resolution in linear length and constant clause space simultaneously

Proof sketch.

- Start by resolving $\overline{z}$ and $\bigvee_{u \in pred(z)} \overline{u} \lor z$
- Then, in reverse topological order of vertices v, resolve with pebbling axioms $\bigvee_{u \in pred(v)} \overline{u} \lor v$
- Invariant: One clause in memory; only negative literals; only for vertices preceding v in topological order
- Finally, have one wide clause with negative literals over all sources
- Use source axioms to resolve away these literals one by one □

Key New Idea: Variable Substitution

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

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

$$\Downarrow$$

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

$$\Downarrow$$

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

Key Technical Result: Substitution Theorem

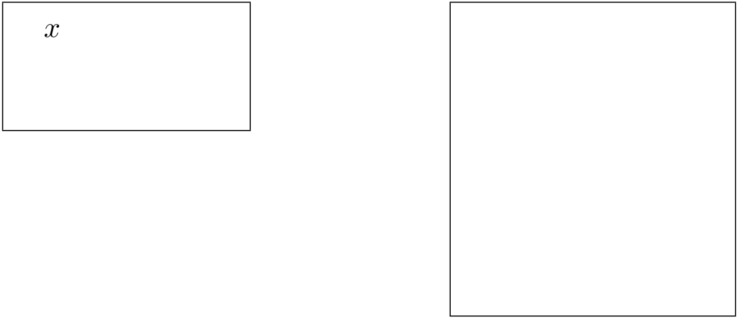
Let $F[\oplus]$ denote formula with XOR $x_1 \oplus x_2$ substituted for x

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

$\overline{x} \vee y$

Key Technical Result: Substitution Theorem

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

Key Technical Result: Substitution Theorem

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

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

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

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

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

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

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

Sketch of Proof of Substitution Theorem

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

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

Putting the Pieces Together

Making variable substitutions in pebbling formulas

- lifts lower bound from number of variables to clause space
- maintains upper bound in terms of total space and length

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 proofs can sometimes do better than black-only pebblings [Nordström '10]

Putting the Pieces Together

Making variable substitutions in pebbling formulas

- lifts lower bound from number of variables to clause space
- maintains upper bound in terms of total space and length

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 proofs can sometimes do better than black-only pebblings [Nordström '10]

Extension to Polynomial Calculus

- Using somewhat different techniques, can extend trade-offs to polynomial calculus [Beck, Nordström & Tang '12]
- Same formulas and much simpler proof, but lose a bit in parameters
- Also, can't get unconditional space lower bounds for polynomial calculus this way
- Will discuss space in polynomial calculus in final two lectures

An Intriguing Open Problem

Recall key technical theorem: amplify space lower bounds through variable substitution

Almost completely oblivious to proof system under study, and has been extended to strictly stronger k-DNF resolution proof systems — maybe can be made to work for other stronger systems as well?

Open Problem

Can the Substitution Theorem be proven for, say, cutting planes or polynomial calculus, thus yielding space lower bounds and time-space trade-offs for these proof systems as well?

Approach in previous papers provably will not work

Partial progress with different techniques in [Huynh & Nordström '12] and [Beck, Nordström & Tang '12] indicate that answer should be "yes"

An Intriguing Open Problem

Recall key technical theorem: amplify space lower bounds through variable substitution

Almost completely oblivious to proof system under study, and has been extended to strictly stronger k-DNF resolution proof systems — maybe can be made to work for other stronger systems as well?

Open Problem

Can the Substitution Theorem be proven for, say, cutting planes or polynomial calculus, thus yielding space lower bounds and time-space trade-offs for these proof systems as well?

Approach in previous papers provably will not work

Partial progress with different techniques in [Huynh & Nordström '12] and [Beck, Nordström & Tang '12] indicate that answer should be "yes"

An Intriguing Open Problem

Recall key technical theorem: amplify space lower bounds through variable substitution

Almost completely oblivious to proof system under study, and has been extended to strictly stronger k-DNF resolution proof systems — maybe can be made to work for other stronger systems as well?

Open Problem

Can the Substitution Theorem be proven for, say, cutting planes or polynomial calculus, thus yielding space lower bounds and time-space trade-offs for these proof systems as well?

Approach in previous papers provably will not work

Partial progress with different techniques in [Huynh & Nordström '12] and [Beck, Nordström & Tang '12] indicate that answer should be "yes"