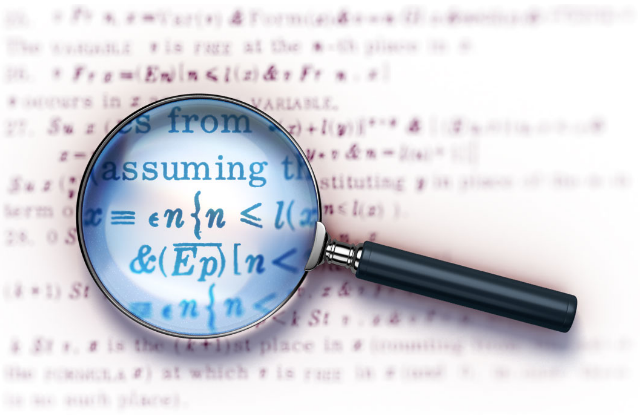


Automated Proof Search: The Aftermath

Susanna de Rezende, Mika Göös, Sajin Koroth, Ian Mertz, Jakob Nordström, Toni Pitassi, Robert Robere, Dmitry Sokolov

Me me me!

Me me me!

Here is a problem in
Proof Complexity

Me me me!

Here is a problem in
Proof Complexity

But I wanna work on
Communication :(

Me me me!

Lifting for dag-like models?

Me me me!

Lifting for dag-like models?

Great idea!

Me me me!

Lifting for dag-like models?

Great idea!

Takeaway: **Monotone circuits** for XOR-SAT can simulate **Resolution**

Me me me!

Lifting for dag-like models?

Great idea!

Takeaway: **Monotone circuits** for XOR-SAT can simulate **Resolution**

. . . we proved the converse [GGKS'18]

Me me me!

Lifting for dag-like models?

Great idea!

Takeaway: **Monotone circuits** for Xor-Sat can simulate **Resolution**

…we proved the converse [GGKS'18]

$\Longrightarrow$ Proof complexity is **cool**!

This talk: Results on

Hardness of Automatability

- **Simpler proof** for Resolution [Atserias–Müller'19]
- **Generalises better:** **NP**-hardness for
  - Nullstellensatz . . . *previously* [Galesi–Lauria'10]
  - Polynomial Calculus . . . *previously* [Galesi–Lauria'10]
  - Sherali–Adams
  - Cutting Planes (requires more work)

This talk: Results on

Hardness of Automatability

- **Simpler proof** for Resolution [Atserias–Müller'19]
- **Generalises better:** **NP**-hardness for
  - Nullstellensatz ...*previously* [Galesi–Lauria'10]
  - Polynomial Calculus ...*previously* [Galesi–Lauria'10]
  - Sherali–Adams
  - Cutting Planes (requires more work)

  - ***Still open:*** *Sum-of-Squares*

Simple proof of
[Atserias–Müller'19]

Atserias–Müller

There is polytime reduction $\mathcal{A}$ that maps n-variate CNF F to unsatisfiable CNF $\mathcal{A}(F)$:

$$F \text{ is } \textbf{SAT} \implies \mathcal{A}(F) \text{ has Resolution length } n^{O(1)}$$
$$F \text{ is } \textbf{UNSAT} \implies \mathcal{A}(F) \text{ has Resolution length } 2^{n^{\Omega(1)}}$$

Atserias–Müller

There is polytime reduction $\mathcal{A}$ that maps n-variate CNF F to unsatisfiable CNF $\mathcal{A}(F)$:

$$F \text{ is } \textbf{SAT} \implies \mathcal{A}(F) \text{ has Resolution length } n^{O(1)}$$
$$F \text{ is } \textbf{UNSAT} \implies \mathcal{A}(F) \text{ has Resolution length } 2^{n^{\Omega(1)}}$$

Overview of $\mathcal{A}$:

Input: F that is SAT-vs-UNSAT

1. Construct $\mathsf{Ref}(F)$ of block-width $O(1)$-vs-$n^{\Omega(1)}$
2. Output $\mathsf{Lifted\text{-}Ref}(F)$ of Res-length $n^{O(1)}$-vs-$2^{n^{\Omega(1)}}$

Atserias–Müller

There is polytime reduction $\mathcal{A}$ that maps n-variate CNF F to unsatisfiable CNF $\mathcal{A}(F)$:

$$F \text{ is } \mathbf{SAT} \implies \mathcal{A}(F) \text{ has Resolution length } n^{O(1)}$$
$$F \text{ is } \mathbf{UNSAT} \implies \mathcal{A}(F) \text{ has Resolution length } 2^{n^{\Omega(1)}}$$

Overview of $\mathcal{A}$:

We simplify

Input: F that is SAT-vs-UNSAT

1. Construct Ref(F) of block-width $O(1)$-vs-$n^{\Omega(1)}$
2. Output Lifted-Ref(F) of Res-length $n^{O(1)}$-vs-$2^{n^{\Omega(1)}}$

Key: Reduction from PHP

(When F is UNSAT)

$$\underset{\text{width}}{\text{PHP}} \leq \underset{\text{block-width}}{\text{Ref}(F)}$$

Key: Reduction from PHP

(When F is UNSAT)

$$\underset{\text{width}}{\mathrm{PHP}} \leq \underset{\text{block-width}}{\mathrm{Ref}(F)}$$

Reduction via **Tree-Resolution**

Key: Reduction from PHP

(When F is UNSAT)

$$\underset{\text{width}}{\mathrm{PHP}} \leq \underset{\text{block-width}}{\mathrm{Ref}(F)}$$

Reduction via **Tree-Resolution**
…in depth n^{ε} (surprising!)

Key: Reduction from PHP

(When F is UNSAT)

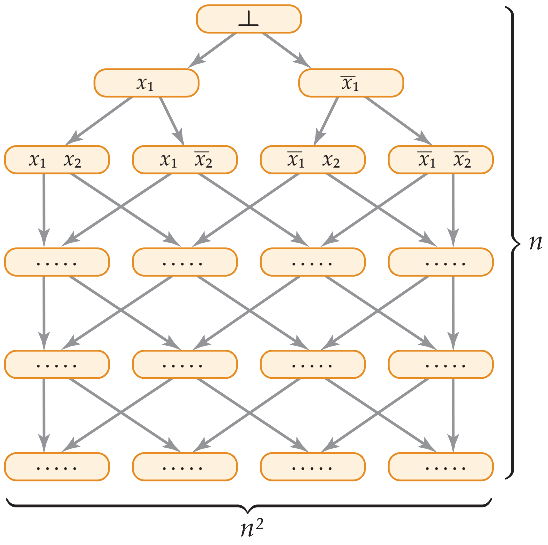
$$\underset{\text{width}}{\mathrm{PHP}} \quad \le \quad \underset{\text{block-width}}{\mathrm{Ref}(F)}$$

Reduction via **Tree-Resolution**
. . . in depth n^{ε} (surprising!)

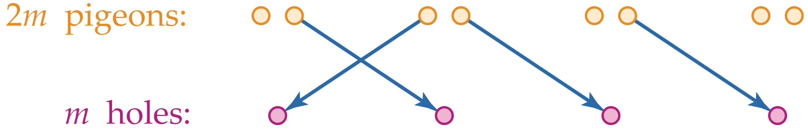
$$\text{width}(\mathrm{PHP})/n^{\varepsilon} \;\le\; \text{block-width}(\mathrm{Ref}(F))$$

Ref(F)

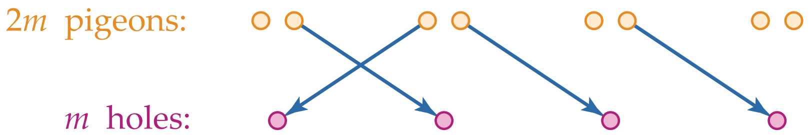
- Encoding of "F admits short Resolution proof"
- Consists of **blocks**
  n layers of n^2 blocks
- **Blocks** encode clauses
  - Indicators for literals
  - Pointers to children
  - Name of axiom of F
- **Important:** Children picked from lower layer
  $\Longrightarrow$ **Dag!**



PHP: Weak bit-encoded invertible function



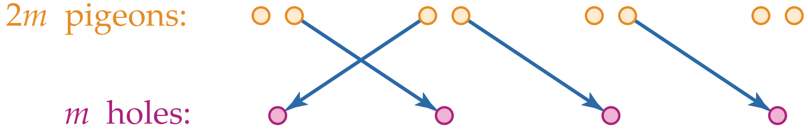
PHP: Weak bit-encoded invertible function



Bit + inv: Each pigeon (hole) associated with $O(\log m)$ variables that *name* one hole (pigeon)

$i \to j$ *iff* i names j and vice versa

PHP: Weak bit-encoded invertible function

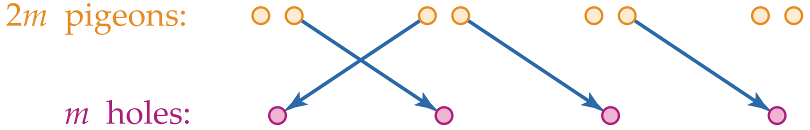


Bit + inv: Each pigeon (hole) associated with $O(\log m)$ variables that *name* one hole (pigeon)

$i \rightarrow j$ *iff* i names j and vice versa

Function: Require every pigeon maps to hole (mapping need not be *onto*)

PHP: Weak bit-encoded invertible function



Lower bounds

PHP_m^{2m} requires degree $\Omega(m)$ for

- Polynomial Calculus [Razborov'98]
- Sherali–Adams [Georgiou–Magen'08]

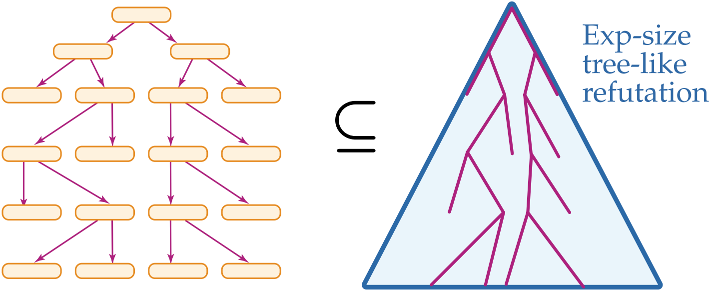
(Using unary encodings. Easy for SoS)

$$\underset{\text{width}}{\mathrm{PHP}} \leq \underset{\text{block-width}}{\mathrm{Ref}(F)}$$

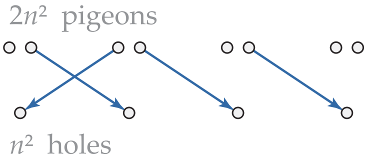
Intution: $\mathrm{Ref}(F)$ looks locally like **full binary tree**

$$\underset{\text{width}}{\mathrm{PHP}} \leq \underset{\text{block-width}}{\mathrm{Ref}(F)}$$

Intution: $\mathrm{Ref}(F)$ looks locally like **full binary tree**



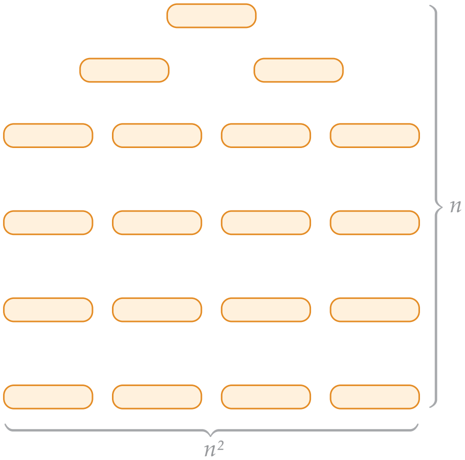
PHP



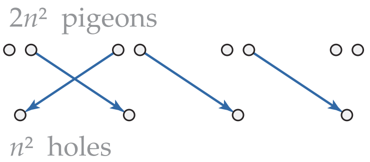
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

$\mathrm{Ref}(F)$



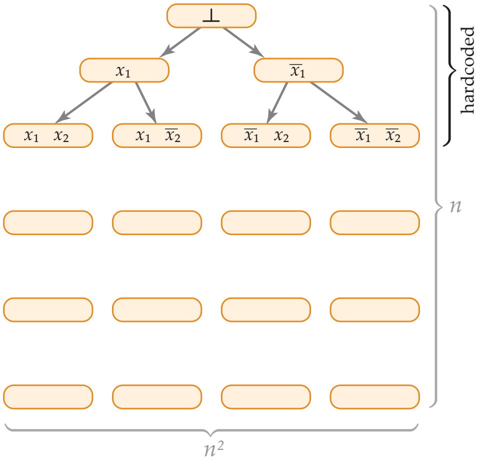
PHP



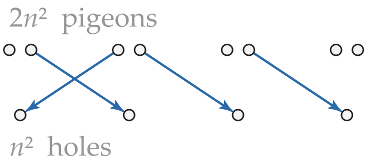
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

$\mathrm{Ref}(F)$



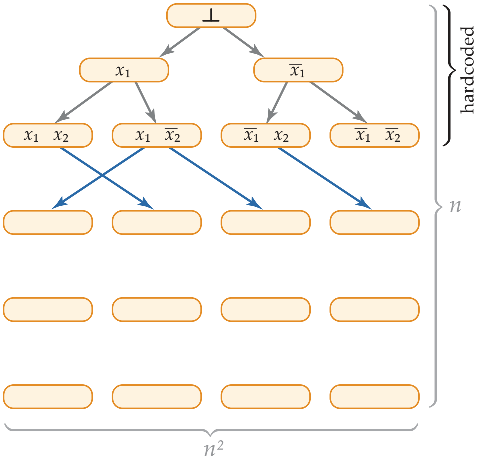
PHP



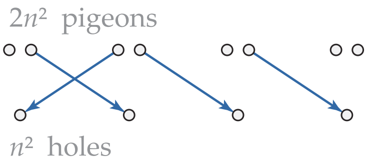
Rules of the game

- Each var of Ref(F) is decision tree of vars of PHP
- Each axiom of Ref(F) is implied by axioms of PHP

Ref(F)



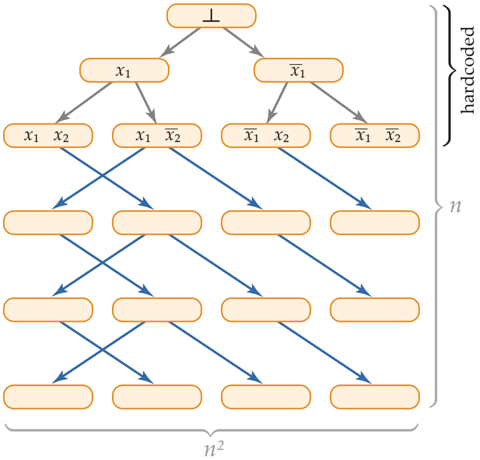
PHP



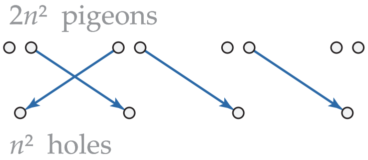
Rules of the game

- Each var of Ref(F) is decision tree of vars of PHP
- Each axiom of Ref(F) is implied by axioms of PHP

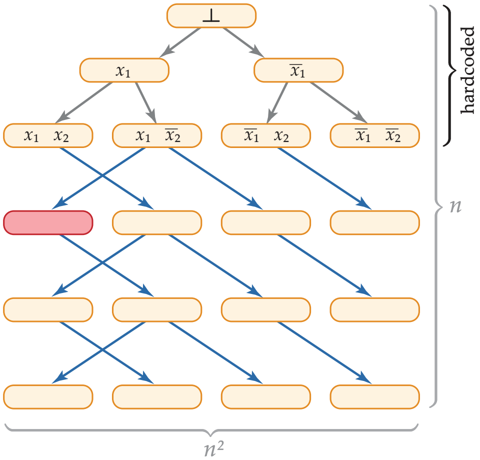
Ref(F)



PHP



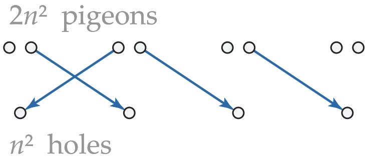
$\mathrm{Ref}(F)$



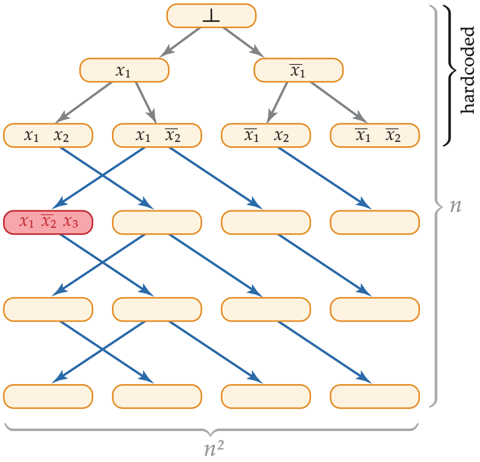
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

PHP



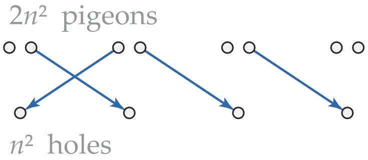
$\mathrm{Ref}(F)$



Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

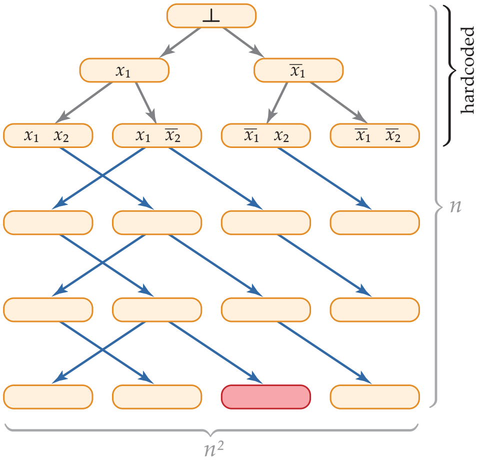
PHP



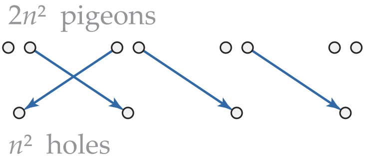
Rules of the game

- Each var of Ref(F) is decision tree of vars of PHP
- Each axiom of Ref(F) is implied by axioms of PHP

Ref(F)



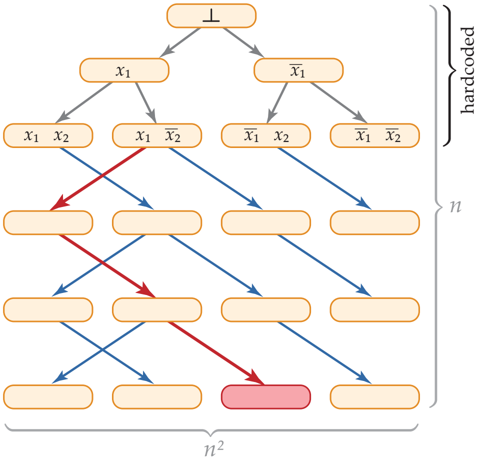
PHP



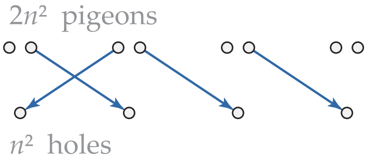
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

$\mathrm{Ref}(F)$



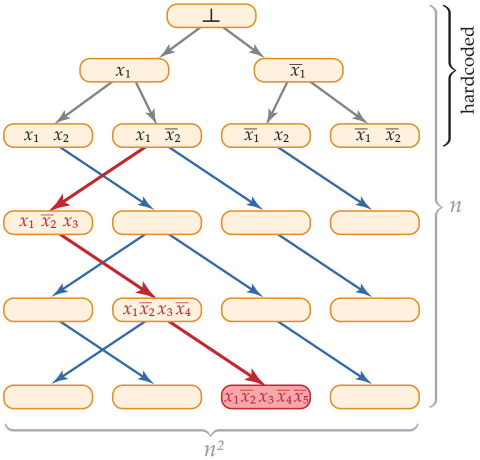
PHP



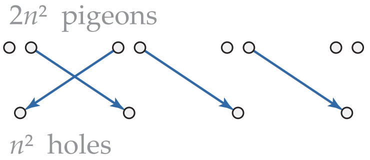
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

$\mathrm{Ref}(F)$



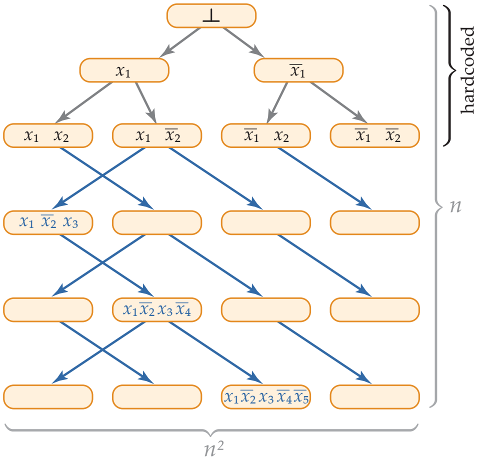
PHP



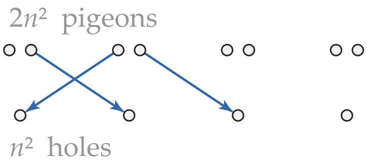
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

$\mathrm{Ref}(F)$



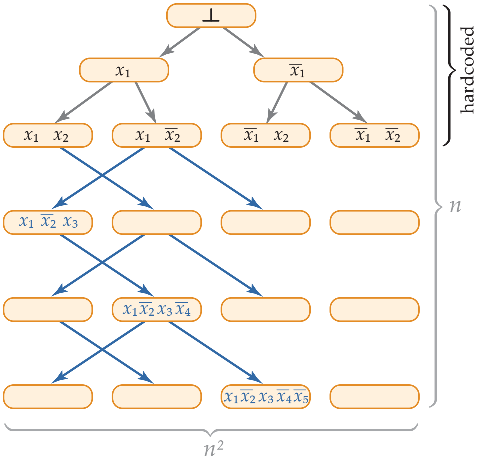
PHP



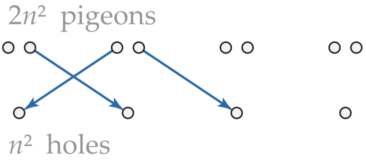
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

$\mathrm{Ref}(F)$



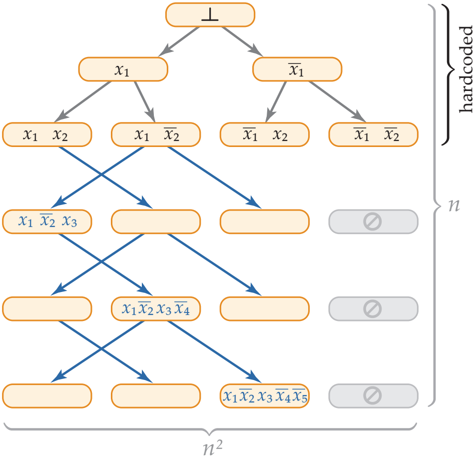
PHP



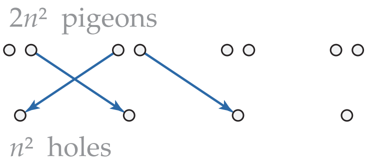
Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

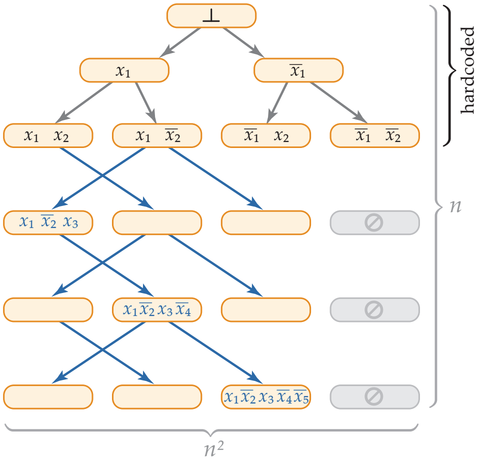
$\mathrm{Ref}(F)$



PHP



$\mathrm{Ref}(F)$



Rules of the game

- Each var of $\mathrm{Ref}(F)$ is decision tree of vars of PHP
- Each axiom of $\mathrm{Ref}(F)$ is implied by axioms of PHP

Conclusion: $\text{block-width}(\mathrm{Ref}(F)) \cdot n \;\geq \text{width}(\textbf{PHP}) = \Omega(n^2)$

$\Longrightarrow \quad \text{block-width}(\mathrm{Ref}(F)) \geq \Omega(n)$

When F is UNSAT

$$\underset{\text{width}}{\mathrm{PHP}} \quad \leq \quad \underset{\text{block-width}}{\mathrm{Ref}(F)}$$

We showed: $\mathrm{Ref}(F)$ has block-width $n^{\Omega(1)}$

Apply lifting: Lifted-$\mathrm{Ref}(F)$ has Resolution size $2^{n^{\Omega(1)}}$

Atserias–Müller

There is polytime reduction $\mathcal{A}$:

$$F \text{ is } \textbf{SAT} \implies \mathcal{A}(F) \text{ has } \textbf{Res} \text{ size } n^{O(1)}$$
$$F \text{ is } \textbf{UNSAT} \implies \mathcal{A}(F) \text{ has } \textbf{Res} \text{ size } 2^{n^{\Omega(1)}}$$

Atserias–Müller

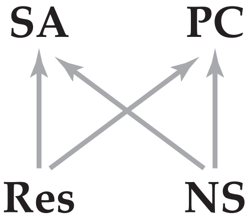
There is polytime reduction $\mathcal{A}$:

$$\begin{aligned} F \text{ is } \textbf{SAT} &\implies \mathcal{A}(F) \text{ has } \textbf{Res} \text{ size } n^{O(1)} \\ F \text{ is } \textbf{UNSAT} &\implies \mathcal{A}(F) \text{ has } \textbf{Res} \text{ size } 2^{n^{\Omega(1)}} \end{aligned}$$

Our extension

There is polytime reduction $\mathcal{A}$:

$$\begin{aligned} F \text{ is } \textbf{SAT} &\implies \mathcal{A}(F) \text{ has } \textbf{Res} \text{ and } \textbf{NS} \text{ size } n^{O(1)} \\ F \text{ is } \textbf{UNSAT} &\implies \mathcal{A}(F) \text{ has } \textbf{PC} \text{ and } \textbf{SA} \text{ size } 2^{n^{\Omega(1)}} \end{aligned}$$



Our extension

There is polytime reduction $\mathcal{A}$:

$$\begin{aligned} F \text{ is } \textbf{SAT} \quad &\Longrightarrow \quad \mathcal{A}(F) \text{ has } \textbf{Res} \text{ and } \textbf{NS} \text{ size } n^{O(1)} \\ F \text{ is } \textbf{UNSAT} \quad &\Longrightarrow \quad \mathcal{A}(F) \text{ has } \textbf{PC} \text{ and } \textbf{SA} \text{ size } 2^{n^{\Omega(1)}} \end{aligned}$$

Upper bound for NS

$\mathrm{Ref}(F)$

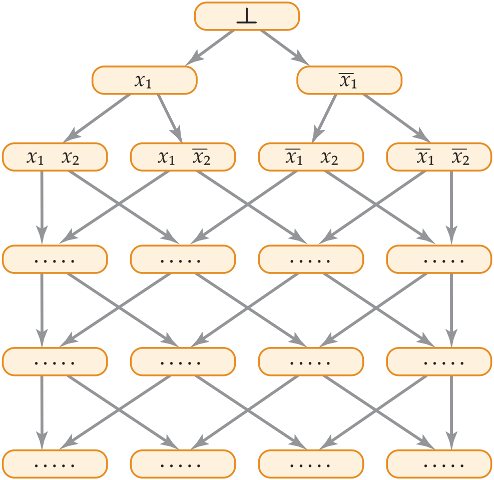
F satisfied by $x \in \{0,1\}^n$

Easy for Resolution since

$\mathrm{Ref}(F) \leq$ Pebbling

Pebbling

- Root of DAG pebbled
- If node is pebbled, then ≥ 1 children is pebbled



$\mathrm{Ref}(F)$

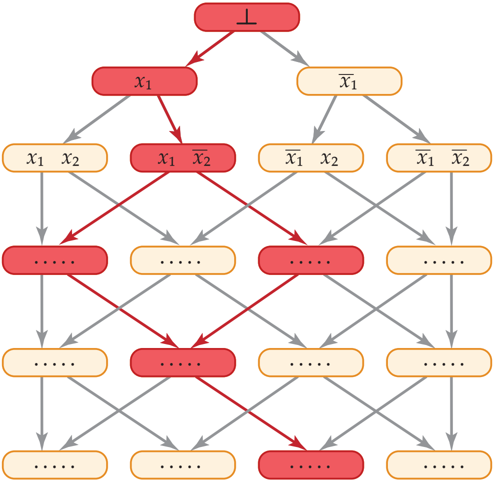
F satisfied by $x \in \{0,1\}^n$

Easy for Resolution since

$\mathrm{Ref}(F) \leq$ Pebbling

“Pebbled” block

$=$ falsified by x



$\mathrm{Ref}(F)$

F satisfied by $x \in \{0,1\}^n$

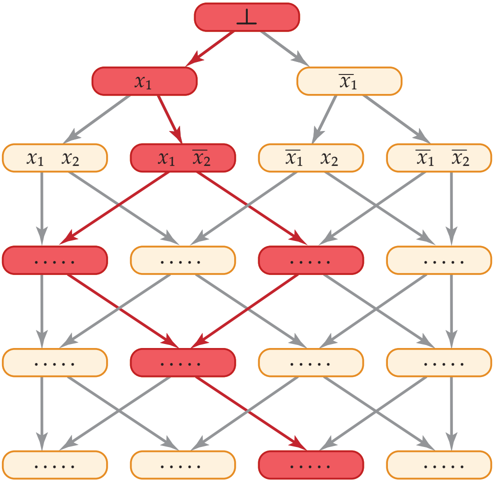
Easy for Resolution since

$\mathrm{Ref}(F) \leq$ Pebbling

"Pebbled" block

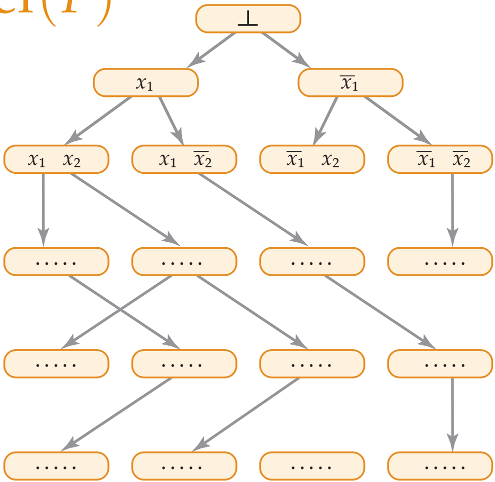
$=$ falsified by x

…but Pebbling is
hard for **NS**!



Solution: TreeRef(F)

F satisfied by $x \in \{0,1\}^n$



Solution: TreeRef(F)

F satisfied by $x \in \{0,1\}^n$

Easy for NS since

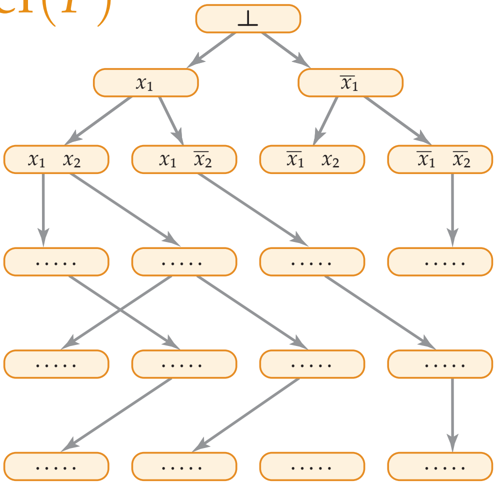
TreeRef(F)

$\leq$ End-of-Line

aka Onto-PHP

End-of-Line

- Root pebbled
- If node is pebbled, then unique child and unique parent pebbled



Solution: TreeRef(F)

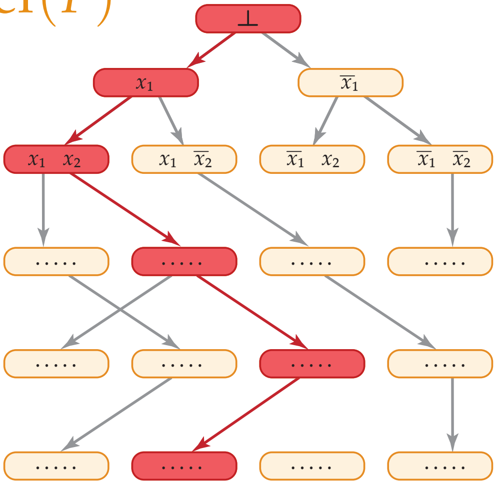
F satisfied by $x \in \{0,1\}^n$

Easy for NS since

TreeRef(F)

$\leq$ End-of-Line

aka Onto-PHP



End-of-Line

- Root pebbled
- If node is pebbled, then unique child and unique parent pebbled

Solution: TreeRef(F)

F satisfied by $x \in \{0,1\}^n$

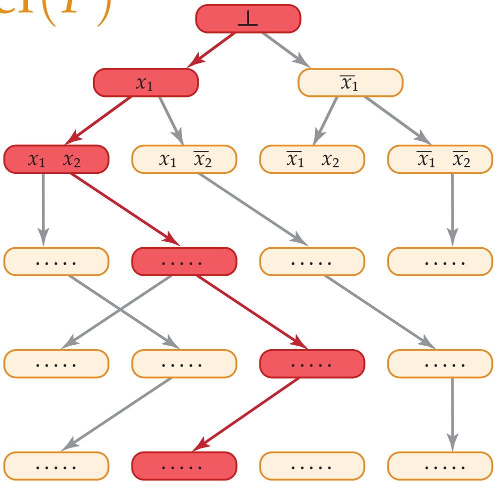
Easy for NS since

TreeRef(F)

$\leq$ End-of-Line

aka Onto-PHP

Hardness still works!



Cutting Planes

Result for Cutting Planes

There is polytime reduction $\mathcal{A}$:

$$F \text{ is } \mathbf{SAT} \implies \mathcal{A}(F) \text{ has } \mathbf{CP} \text{ length } n^{O(1)}$$
$$F \text{ is } \mathbf{UNSAT} \implies \mathcal{A}(F) \text{ has } \mathbf{CP} \text{ length } 2^{n^{\Omega(1)}}$$

Highlights

- [GGKS'18]: F has width $w \Rightarrow F \circ g$ has CP length $2^{\Omega(w)}$
- Instead: need **block**-lifting
- Bypass monotone circuits (first such technique?)

Papers

2019: Automating Resolution is **NP**-hard
Atserias, Müller

Papers

2019: Automating Resolution is **NP**-hard
Atserias, Müller

2020: Automating Cutting Planes is **NP**-hard
Göös, Koroth, Pitassi, Mertz

Papers

2019: Automating Resolution is **NP**-hard
Atserias, Müller

2020: Automating Cutting Planes is **NP**-hard
Göös, Koroth, Pitassi, Mertz

2021: Automating Algebraic Proof Systems is **NP**-hard
de Rezende, Göös, Nordström, Pitassi, Robere, Sokolov

Papers

2019: Automating Resolution is **NP**-hard
Atserias, Müller

2020: Automating Cutting Planes is **NP**-hard
Göös, Koroth, Pitassi, Mertz

2021: Automating Algebraic Proof Systems is **NP**-hard
de Rezende, Göös, Nordström, Pitassi, Robere, Sokolov

2022: Automating Sum-of-Squares is **NP**-hard

Papers

2019: Automating Resolution is **NP**-hard
Atserias, Müller

2020: Automating Cutting Planes is **NP**-hard
Göös, Koroth, Pitassi, Mertz

2021: Automating Algebraic Proof Systems is **NP**-hard
de Rezende, Göös, Nordström, Pitassi, Robere, Sokolov

2022: Automating Sum-of-Squares is **NP**-hard
You?!

Papers

2019: Automating Resolution is **NP**-hard
Atserias, Müller

2020: Automating Cutting Planes is **NP**-hard
Göös, Koroth, Pitassi, Mertz

2021: Automating Algebraic Proof Systems is **NP**-hard
de Rezende, Göös, Nordström, Pitassi, Robere, Sokolov

2022: Automating Sum-of-Squares is **NP**-hard
You?!

Cheers!


EPFL