Automating Tree-Like Resolution in Time $n^{o(\log n)}$ Is ETH-Hard

Susanna F. de Rezende

Institute of Mathematics of the Czech Academy of Sciences

February 2021
Resolution proof system
Resolution proof system

Given **UNSAT** CNF formula $F$:  

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

Given **UNSAT** CNF formula $F$:  

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

clauses/axioms
Resolution proof system

Given \textbf{UNSAT} CNF formula $F$: $$(\overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{z}) \land (x \lor \overline{y}) \land (x \lor \overline{y} \lor z) \land (\overline{x} \lor z)$$

Resolution rule: $$\frac{C \lor x}{C \lor D} \frac{D \lor \overline{x}}{C \lor D}$$

Refutation: Derivation of empty clause $\bot$
Resolution proof system

Given UNSAT CNF formula $F$: $(\overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{z}) \land (x \lor y) \land (x \lor \overline{y} \lor z) \land (\overline{x} \lor z)$

Resolution rule: $\frac{C \lor x \quad D \lor \overline{x}}{C \lor D}$

Refutation: Derivation of empty clause $\bot$
Resolution proof system

Given **UNSAT** CNF formula $F$:

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

Resolution rule: \[
\frac{C \lor x}{C \lor D}
\]

Refutation: Derivation of empty clause $\bot$
Resolution proof system

Given UNSAT CNF formula $F$: $(\overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{z}) \land (x \lor y) \land (x \lor \overline{y} \lor z) \land (\overline{x} \lor z)$

Resolution rule: \[
\frac{C \lor x \quad D \lor \overline{x}}{C \lor D}
\]

Refutation: Derivation of empty clause $\bot$
Resolution proof system

Given UNSAT CNF formula $F$: $(\overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{z}) \land (x \lor y) \land (x \lor \overline{y} \lor z) \land (\overline{x} \lor z)$

Resolution rule: $\frac{C \lor x \ D \lor \overline{x}}{C \lor D}$

Refutation: Derivation of empty clause $\bot$
Resolution proof system

Given **UNSAT** CNF formula $F$: 

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

Resolution rule: 

$$\frac{C \lor x \quad D \lor \overline{x}}{C \lor D}$$

Refutation: Derivation of empty clause $\bot$

**Size:** # clauses in proof (10 in example)

**Width:** max # literals/clauses in proof
Resolution proof system

Given **UNSAT** CNF formula $F$: 

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

Resolution rule: \[
\frac{C \lor x}{C \lor D} \quad \frac{D \lor \overline{x}}{C \lor D}
\]

Refutation: Derivation of empty clause $\perp$

Size: $\#$ clauses in proof (10 in example)

Width: $\max \ # \ literals/\text{clauses \ in \ proof}$

Automating Tree-Like Resolution in Time $n^{o(\log n)}$ Is ETH-Hard
Automatability [BPR’97]

How hard is it to find proofs/refutations?
Automatability [BPR’97]
How hard is it to find proofs/refutations?

Suppose unsat CNF $F$ has poly-size refutations. Can you find one in poly-time?
Automatability [BPR’97]

How hard is it to find proofs/refutations?

Suppose unsat CNF $F$ has poly-size refutations. Can you find one in poly-time?

Proof system $\mathcal{P}$ is automatable in time $f(n)$ if $\exists$ algorithm $\mathcal{A}$ that given unsat CNF $F$ outputs $\mathcal{P}$-refutation of $F$ in time $f(n)$.
Automatability [BPR’97]
How hard is it to find proofs/refutations?

Suppose unsat CNF $F$ has poly-size refutations. Can you find one in poly-time?

Proof system $\mathcal{P}$ is automatable in time $f(n)$ if $\exists$ algorithm $A$ that given unsat CNF $F$ outputs $\mathcal{P}$-refutation of $F$ in time $f(n)$

size of smallest $\mathcal{P}$-refutation of $F$ plus the size of $F$
Atserias–Müller ‘19

If resolution is automatable:

1. in time \( \text{poly}(n) \) then \( \text{NP} \subseteq \text{P} \)
2. in time \( \text{quasipoly}(n) \) then \( \text{NP} \subseteq \text{QP} \)
3. in time \( \text{subexp}(n) \) then \( \text{NP} \subseteq \text{SUBEXP} \)
If resolution is automatable:

1. in time $\text{poly}(n)$ then $\text{NP} \subseteq \text{P}$
2. in time $\text{quasipoly}(n)$ then $\text{NP} \subseteq \text{QP}$
3. in time $\text{subexp}(n)$ then $\text{NP} \subseteq \text{SUBEXP}$

Generalizations

1. Cutting planes [GKMP’20]
2. $\text{Res}(k)$ [Gar’20]
3. Algebraic proof systems (NS, PC, SA) [dRGNPRS’21]
Tree-like resolution is automatable in time $n^{O(\log n)}$ \cite{BP96}
Tree-like resolution is automatable in time $s^{O(\log n)} \notin \# \text{ variables}$ [BP’96] size of smallest refutation
Tree-like resolution is automatable in time $O(\log n)$ \cite{BP96}.

Given UNSAT CNF formula $F$
Tree-like resolution is automatable in time $s^{O(\log n)}$ \([BP’96]\)

Given **UNSAT** CNF formula $F$

Suppose $s$ is known
Tree-like resolution is automatable in time $s^{O(\log n)}$ \cite{BP'96}.

Given \textbf{UNSAT} CNF formula $F$.

Suppose $s$ is known.

\[ \exists x \text{ s.t. } R^*(F|_{x=0}) \leq s/2 \text{ or } R^*(F|_{x=1}) \leq s/2 \]

\[ x = b \quad x = 1 - b \]
Tree-like resolution is automatable in time $s^{O\left(\log n\right)}$ [BP’96]

Given \text{UNSAT} CNF formula $F$

Suppose $s$ is known

$\exists x \text{ s.t. } R^*(F|_{x=0}) \leq s/2 \text{ or } R^*(F|_{x=1}) \leq s/2$

$T(n, s) = 2n \cdot T(n-1, s/2) + T(n-1, s) + O(1)$

$T(n, s) = s^{O\left(\log n\right)}$
Tree-like resolution is automatable in time $n^{O(\log n)}$ [BP’96]
Tree-like resolution is automatable in time $n^{O(\log n)}$ [BP’96]

If tree-like resolution is automatable:

1. in time $\text{poly}(n)$ then $W[P] = \text{FPT}$ [AR’01]
Tree-like resolution is automatable in time $n^{O(\log n)}$ [BP’96]

If tree-like resolution is automatable:

1. in time $\text{poly}(n)$ then $W[P] = \text{FPT}$ [AR’01]

2. in time $n^{O(\log^{1/7-\epsilon} 1 \log n)}$ then ETH is false [MPW’19]
Tree-like resolution is automatable in time $n^{O(\log n)}$ [BP’96]

If tree-like resolution is automatable:

1. in time $\text{poly}(n)$ then $W[P] = \text{FPT}$ [AR’01]
2. in time $n^{O(\log^{1/7-\epsilon} \log n)}$ then ETH is false [MPW’19]

Theorem 1

If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false
Tree-like resolution is automatable in time $n^{O(\log n)}$ [BP’96]

If tree-like resolution is automatable:

1. in time $\text{poly}(n)$ then $\text{W}[P] = \text{FPT}$ [AR’01]

2. in time $n^{O(\log^{1/7-\epsilon} \log n)}$ then ETH is false [MPW’19]

**Theorem 1**

If tree-like resolution is automatable:

1. in time $n^{O(\log n)}$ then ETH is false

2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$
Tree-like resolution is automatable in time $n^{O(\log n)}$ [BP’96]

If tree-like resolution is automatable:

1. in time $\text{poly}(n)$ then $W[P] = \text{FPT}$ [AR’01]

2. in time $n^{O(\log^{1/7-\varepsilon} \log n)}$ then ETH is false [MPW’19]

**Theorem 1**

If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false

2. in time $n^{O(\log^{1-\varepsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\varepsilon/2})})$

3. in time $\text{poly}(n)$ then $W[P] = \text{FPT}$
Theorem 1

If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$
3. in time $\text{poly}(n)$ then $\text{W}[P] = \text{FPT}$
Theorem 1

If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$
3. in time $\text{poly}(n)$ then $\text{W}[P] = \text{FPT}$

Main Theorem

$\exists$ algorithm that given $n$-variate $F$ outputs $A(F)$ in time $2^{O(\sqrt{n})}$ s.t.

1. $F$ is SAT $\Rightarrow R^*(A(F')) \leq 2^{O(\sqrt{n})}$
2. $F$ is UNSAT $\Rightarrow R^*(A(F)) \geq 2^{\Omega(n)}$
If tree-like resolution is automatable:

1. in time \( n^{O(\log n)} \) then ETH is false
2. in time \( n^{O(\log^{1-\epsilon} n)} \) then \( \text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})}) \)

\( \exists \) algorithm that given \( n \)-variate \( F \) outputs \( A(F) \) in time \( 2^{O(\sqrt{n})} \) s.t.

1. \( F \) is SAT \( \Rightarrow R^*(A(F)) \leq 2^{O(\sqrt{n})} \)
2. \( F \) is UNSAT \( \Rightarrow R^*(A(F)) \geq 2^{\Omega(n)} \)
If tree-like resolution is automatable:

1. in time \( n^{O(\log n)} \) then ETH is false
2. in time \( n^{O(\log^{1-\epsilon} n)} \) then \( \text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})}) \)

\( \exists \) algorithm that given \( n \)-variate \( F \) outputs \( \mathcal{A}(F) \) in time \( 2^{O(\sqrt{n})} \) s.t.

1. \( F \) is SAT \( \Rightarrow R^*(\mathcal{A}(F)) \leq 2^{O(\sqrt{n})} \)
2. \( F \) is UNSAT \( \Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)} \)

Proof. Suppose \( \mathcal{A} \) automatates tree-like Res in time \( f(N) = N^{O(\log N)} \)
If tree-like resolution is automatable:
1. in time $n^{O(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$

∃ algorithm that given $n$-variate $F$ outputs $\mathcal{A}(F)$ in time $2^{O(\sqrt{n})}$ s.t.
1. $F$ is SAT ⇒ $R^*(\mathcal{A}(F)) \leq 2^{O(\sqrt{n})}$
2. $F$ is UNSAT ⇒ $R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

Proof. Suppose $\mathcal{A}$ automataxes tree-like Res in time $f(N) = N^{O(\log N)}$

size of smallest refutation of $F$ plus the size of $F$
If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$

\exists \text{ algorithm that given } n\text{-variate } F \text{ outputs } A(F) \text{ in time } 2^{O(\sqrt{n})} \text{ s.t.}

1. $F$ is SAT $\Rightarrow R^*(A(F)) \leq 2^{O(\sqrt{n})}$
2. $F$ is UNSAT $\Rightarrow R^*(A(F)) \geq 2^{\Omega(n)}$

Proof. Suppose $A$ automataates tree-like Res in time $f(N) = N^{o(\log N)}$

Want to decide if 3-CNF $F$ is SAT or UNSAT
If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$

 Exists algorithm that given $n$-variate $F$ outputs $\mathcal{A}(F)$ in time $2^{O(\sqrt{n})}$ s.t.

1. $F$ is SAT $\implies R^*(\mathcal{A}(F)) \leq 2^{O(\sqrt{n})}$
2. $F$ is UNSAT $\implies R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

Proof. Suppose $\mathcal{A}$ automatates tree-like Res in time $f(N) = N^{o(\log N)}$

Want to decide if 3-CNF $F$ is SAT or UNSAT

$\mathcal{A}(F) : \# \text{ var } 2^{O(\sqrt{n})}$
If tree-like resolution is automatable:
1. in time $n^{o(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$

$\exists$ algorithm that given $n$-variate $F$ outputs $\mathcal{A}(F)$ in time $2^{O(\sqrt{n})}$ s.t.
1. $F$ is SAT $\Rightarrow R^*(\mathcal{A}(F)) \leq 2^{O(\sqrt{n})}$
2. $F$ is UNSAT $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

Proof. Suppose $\mathcal{A}$ automatates tree-like Res in time $f(N) = N^{o(\log N)}$

Want to decide if 3-CNF $F$ is SAT or UN SAT

$\mathcal{A}(F)$: $\#$ var $2^{O(\sqrt{n})}$

and if $F$ is SAT $N = 2^{O(\sqrt{n})}$
If tree-like resolution is automatable:
1. in time $n^{O(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$

∃ algorithm that given $n$-variate $F$ outputs $A(F)$ in time $2^{O(\sqrt{n})}$ s.t.
1. $F$ is SAT $\Rightarrow R^*(A(F)) \leq 2^{O(\sqrt{n})}$
2. $F$ is UNSAT $\Rightarrow R^*(A(F)) \geq 2^{\Omega(n)}$

Proof. Suppose $A$ automatates tree-like Res in time $f(N) = N^{O(\log N)}$

Want to decide if 3-CNF $F$ is SAT or UNSAT

$A(F)$: # var $2^{O(\sqrt{n})}$
and if $F$ is SAT $N = 2^{O(\sqrt{n})}$

Run $A$ on $A(F)$ for $f(2^{O(\sqrt{n})}) = 2^{o(n)}$ steps

obs. $o(\log N) = o(\sqrt{n})$
If tree-like resolution is automatable:

1. in time $n^{O(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$

∃ algorithm that given $n$-variate $F$ outputs $\mathcal{A}(F)$ in time $2^{O(\sqrt{n})}$ s.t.

1. $F$ is SAT $\Rightarrow R^*(\mathcal{A}(F)) \leq 2^{O(\sqrt{n})}$
2. $F$ is UNSAT $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

Proof. Suppose $\mathbb{A}$ automates tree-like Res in time $f(N) = N^{O(\log N)}$

Want to decide if 3-CNF $F$ is SAT or UNSAT

$\mathcal{A}(F)$: $\# \text{ var } 2^{O(\sqrt{n})}$ and if $F$ is SAT $N = 2^{O(\sqrt{n})}$

Run $\mathbb{A}$ on $\mathcal{A}(F)$ for $f(2^{O(\sqrt{n})}) = 2^{o(n)}$ steps

$\mathbb{A}$ returns refutation $\Leftrightarrow F$ is SAT
Atserias-Müller

Automating Resolution is NP-Hard – Simplified [dRGNPRS’21]
exists algorithm that given \( n \)-variate \( F \) outputs \( \mathcal{A}(F) \) in time \( n^{O(1)} \) s.t.

1. \( F \) is SAT \( \Rightarrow R(\mathcal{A}(F)) \leq n^{O(1)} \)
2. \( F \) is UNSAT \( \Rightarrow R(\mathcal{A}(F)) \geq 2^{\Omega(n)} \)
Atserias-Müller ’19

∃ algorithm that given \( n \)-variate \( F \) outputs \( \mathcal{A}(F) \) in time \( n^{O(1)} \) s.t.

1. \( F \) is SAT \( \Rightarrow R(\mathcal{A}(F)) \leq n^{O(1)} \)
2. \( F \) is UNSAT \( \Rightarrow R(\mathcal{A}(F)) \geq 2^{\Omega(n)} \)

Main Theorem (tree-like resolution)

∃ algorithm that given \( n \)-variate \( F \) outputs \( \mathcal{A}(F) \) in time \( 2^{O(\sqrt{n})} \) s.t.

1. \( F \) is SAT \( \Rightarrow R^*(\mathcal{A}(F)) \leq 2^{O(\sqrt{n})} \)
2. \( F \) is UNSAT \( \Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)} \)
### Atserias–Müller ’19

∃ algorithm that given \( n \)-variate \( F \) outputs \( \mathcal{A}(F) \) in time \( n^{O(1)} \) s.t.

1. \( F \) is SAT \( \Rightarrow R(\mathcal{A}(F)) \leq n^{O(1)} \)
2. \( F \) is UNSAT \( \Rightarrow R(\mathcal{A}(F)) \geq 2^{\Omega(n)} \)

### Main Theorem (tree-like resolution)

∃ algorithm that given \( n \)-variate \( F \) outputs \( \mathcal{A}(F) \) in time \( 2^{O(\sqrt{n})} \) s.t.

1. \( F \) is SAT \( \Rightarrow R^*(\mathcal{A}(F)) \leq 2^{O(\sqrt{n})} \)
2. \( F \) is UNSAT \( \Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)} \)
“Universal refutation” (complete Binary tree)
“Universal refutation” (complete Binary tree)
Ref($F$)

Enodes “$F$ has short resolution refutation”

$n$

$s = n^{O(1)}$

$F$: $m$
Ref($F$)

Encodes "$F$ has short resolution refutation"

$n$

$s = n^{O(1)}$

$F$: 

$\text{structured}$
Ref($F$)

Variables for each block $B$:

\[ \text{Encodes "} F \text{ has short resolution refutation"} \]

\[ s = n^{O(1)} \]

\[ m \]

\[ n \]
\textbf{Ref}(F)

Variables for each block $B$:

- \textbf{2n} variables (1 per literal): indicates clause

Encodes \textit{“F has short resolution refutation”}

$s = n^{O(1)}$

$m$

Susanna F. de Rezende

Automating Tree-Like Resolution in Time $n^{o(\log n)}$ Is ETH-Hard
$\text{Ref}(F)$

Variables for each block $B$:
- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children “derived from $B_i$ and $B_j$”

Encodes “$F$ has short resolution refutation”
Ref($F$)

Variables for each block $B$: 

- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children “derived from $B_i$ and $B_j$”

Encodes “$F$ has short resolution refutation”
Ref$(F)$

Variables for each block $B$:

- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children “derived from $B_i$ and $B_j$”
- $\log m$ variables: axiom-index $j \in [m]$

Encodes “$F$ has short resolution refutation”
$\text{Ref}(F)$

Variables for each block $B$:
- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children “derived from $B_i$ and $B_j$”
- $\log m$ variables: axiom-index $j \in [m]$

$O(n^2s)$ variables

Encodes “$F$ has short resolution refutation”

$n$

$s = n^{O(1)}$

$F:$

$s = n^{O(1)}$

$n$

$m$
Ref($F$)

Variables for each block $B$:

- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children “derived from $B_i$ and $B_j$”
- $\log m$ variables: axiom-index $j \in [m]$

$O(n^2 s)$ variables

Axioms of $\text{Ref}(F)$:

- Root: $\bot$ clause
- Derived: valid resolution step
- Axiom: weakening of axiom
Ref($F$)

Variables for each block $B$:

- **$2n$** variables (1 per literal): indicates clause
- **$2(\log s)$** variables: 2 pointers to children “derived from $B_i$ and $B_j$”
- **$\log m$** variables: axiom-index $j \in [m]$

$O(n^2s)$ variables

Axioms of Ref($F$):

- **Root**: $\bot$ clause
- **Derived**: valid resolution step
- **Axiom**: weakening of axiom

Encodes “$F$ has short resolution refutation”

$s = n^{O(1)}$

$F$: $m$
Ref($F$)

Variables for each block $B$:
- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children “derived from $B_i$ and $B_j$”
- $\log m$ variables: axiom-index $j \in [m]$

$O(n^2s)$ variables

Axioms of Ref($F$):
- Root: $\bot$ clause
- Derived: valid resolution step
- Axiom: weakening of axiom

Encodes “$F$ has short resolution refutation”

$F$: $\bot$

$s = n^{O(1)}$

$n$

$m$
Ref(\(F\))

Variables for each block \(B\):

- \(2n\) variables (1 per literal): indicates clause
- \(2(\log s)\) variables: 2 pointers to children “derived from \(B_i\) and \(B_j\)”
- \(\log m\) variables: axiom-index \(j \in [m]\)

\(O(n^2s)\) variables

Axioms of Ref(\(F\)):

- **Root**: \(\bot\) clause
- **Derived**: valid resolution step
- **Axiom**: weakening of axiom

\(\text{poly}(n)\) clauses of width \(O(\log s)\)

\(s = n^{O(1)}\)

\(F:\)

\(m\)

Encodes “\(F\) has short resolution refutation”
(1) $F$ is $\text{SAT} \Rightarrow \text{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation

Read-once branching program for $\text{Ref}(F)$
(1) $F$ is SAT $\Rightarrow$ Ref($F$) has size-$n^{O(1)}$ resolution refutation

Read-once branching program for Ref($F$)

$x^*$ satisfying assignment for $F$

$$s = n^{O(1)}$$

$F$: 

$$\text{Susanna F. de Rezende}$$

Automating Tree-Like Resolution in Time $n^{o(\log n)}$ Is ETH-Hard
(1) $F$ is $\textbf{SAT} \Rightarrow \text{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation

Read-once branching program for $\text{Ref}(F)$

$x^*$ satisfying assignment for $F$

invariant: $x^*$ falsifies clause in current block $B$
(1) $F$ is $\text{SAT} \implies \text{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation

Read-once branching program for $\text{Ref}(F)$

$x^*$ satisfying assignment for $F$

invariant: $x^*$ falsifies clause in current block $B$

Start at root and keep invariant

until detect non-valid derivation step or

until reach leaf (cannot be weakening of axiom)
\begin{enumerate}
\item $F$ is $\text{SAT} \Rightarrow \text{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation
\end{enumerate}

Read-once branching program for $\text{Ref}(F)$

- $x^*$ satisfying assignment for $F$

  invariant: $x^*$ falsifies clause in current block $B$

Start at root and keep invariant until detect non-valid derivation step or until reach leaf (cannot be weakening of axiom)

refutation size: $\approx (\#\text{blocks})^2 = n^{O(1)}$
(2) $F$ is UNSAT $\Rightarrow$ Ref($F$) requires size $2^\Omega(n)$ resolution refutation
(2) $F$ is UNSAT $\Rightarrow \text{Ref}(F)$ requires size $2^{\Omega(n)}$ resolution refutation

- $w(\text{Ref}(F) \vdash \bot) \geq \tilde{\Omega}(w(\text{PHP}_{s}^{2s} \vdash \bot)/n)$

[drgnprs’21]
(2) $F$ is UNSAT $\Rightarrow \text{Ref}(F)$ requires size $2^\Omega(n)$ resolution refutation

- $w(\text{Ref}(F) \vdash \bot) \geq \tilde{\Omega}(w(\text{PHP}_s^{2s} \vdash \bot)/n) \geq \tilde{\Omega}(s/n)$
  
  [dRGNPRS’21]
(2) $F$ is **UNSAT** $\Rightarrow$ Ref$(F)$ requires size $2^{\Omega(n)}$ resolution refutation

- $w(\text{Ref}(F) \vdash \bot) \geq \tilde{\Omega}(w(\text{PHP}_s^{2s} \vdash \bot)/n) \geq \tilde{\Omega}(s/n)$
  
  [dRGNPRS'21]

- [BW'01] size of resolution refutation of $\varphi \geq \exp \left( \Omega \left( \frac{w(\varphi \vdash \bot) - w(\varphi)^2}{\#\text{var}} \right) \right)$
(2) $F$ is **UNSAT** $\Rightarrow \text{Ref}(F)$ requires size $2^{\Omega(n)}$ resolution refutation

- $w(\text{Ref}(F) \vdash \perp) \geq \tilde{\Omega}(w(\text{PHP}_s^{2s} \vdash \perp)/n) \geq \tilde{\Omega}(s/n)$
  
  [dRGNPRS’21]

- [BW’01] size of resolution refutation of $\varphi \geq \exp \left( \Omega \left( \frac{w(\varphi \vdash \perp) - w(\varphi)}{\#\text{var}} \right) \right)$

- Recall: $\text{Ref}(F)$ has $O(n^2 s)$ variables and width $O(\log s)$
(2) $F$ is **UNSAT** $\Rightarrow \text{Ref}(F)$ requires size $2^{\Omega(n)}$ resolution refutation

- $w(\text{Ref}(F) \vdash \bot) \geq \tilde{\Omega}(w(\text{PHP}^{2s}_s \vdash \bot)/n) \geq \tilde{\Omega}(s/n)$
  [dRGNPRS’21]
- [BW’01] size of resolution refutation of $\varphi \geq \exp \left( \Omega \left( \frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\#\text{var}} \right) \right)$
- Recall: $\text{Ref}(F)$ has $O(n^2 s)$ variables and width $O(\log s)$

$\text{Ref}(F')$ requires size $\exp \left( \tilde{\Omega} \left( \frac{(s/n)^2}{n^2 s} \right) \right) \geq \exp \left( \tilde{\Omega} \left( \frac{s}{n^4} \right) \right)$
(2) $F$ is UNSAT $\Rightarrow$ Ref($F$) requires size $2^{\Omega(n)}$ resolution refutation

- $w(\text{Ref}(F) \vdash \bot) \geq \tilde{\Omega}(w(\text{PHP}_s^{2s} \vdash \bot)/n) \geq \tilde{\Omega}(s/n)$
  \[\text{[dRGNPRS'21]}\]

- $[\text{BW'01}]$ size of resolution refutation of $\varphi \geq \exp \left( \Omega \left( \frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\#\text{var}} \right) \right)$

- Recall: Ref($F$) has $O(n^2 s)$ variables and width $O(\log s)$

  Ref($F$) requires size $\exp \left( \tilde{\Omega} \left( \frac{(s/n)^2}{n^2 s} \right) \right) \geq \exp \left( \tilde{\Omega} \left( \frac{s}{n^4} \right) \right)$

  choose $s \geq n^5$
(2) \( F \) is **UNSAT** \( \Rightarrow \) \( \text{Ref}(F) \) requires size \( 2^{\Omega(n)} \) resolution refutation

- \( w(\text{Ref}(F) \vdash \bot) \geq \tilde{\Omega}(w(\text{PHP}^{2s}_s \vdash \bot)/n) \geq \tilde{\Omega}(s/n) \)
  
  [dRGNPRS’21]

- \([\text{BW}’01]\) size of resolution refutation of \( \varphi \) \( \geq \exp \left( \Omega \left( \frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\#\text{var}} \right) \right) \)

- Recall: \( \text{Ref}(F) \) has \( O(n^2s) \) variables and width \( O(\log s) \)

\[ \text{Ref}(F) \text{ requires size } \exp \left( \tilde{\Omega} \left( \frac{(s/n)^2}{n^2s} \right) \right) \geq \exp \left( \tilde{\Omega} \left( \frac{s}{n^4} \right) \right) \]

**OBS.** this same lower bound proof works for PC, SA:

- degree lower bound for \( \text{PHP}^{2s}_s \)
- similar size-degree relation

\[ \text{choose } s \geq n^5 \]
\( \text{PHP}^{2s}_s \leq \tilde{O}(n) \text{ Ref}(F) \)
\[
\text{PHP}^2_s \leq \tilde{\mathcal{O}}(n) \quad \text{Ref}(F)
\]
\[ \text{PHP}_s^{2s} \leq \tilde{O}(n) \text{ Ref}(F) \]

Diagram showing a tree-like structure with nodes and edges illustrating the relationship between the PHP and Ref complexity class, bounded by \( \tilde{O}(n) \).
Tree-Like Resolution
(1) $F$ is SAT $\Rightarrow$ Ref($F$) has size-$n^{O(1)}$ resolution refutation

Read-once branching program for Ref($F$)

size: poly($n$)
(1) $F$ is $\text{SAT} \Rightarrow \text{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation

Read-once branching program for $\text{Ref}(F)$

size: $\text{poly}(n)$

Decision tree for $\text{Ref}(F)$
(1) $F$ is **SAT** $\Rightarrow$ **Ref**($F$) has size-$n^{O(1)}$ resolution refutation

Read-once branching program for **Ref**($F$)

size: $\text{poly}(n)$

Decision tree for **Ref**($F$)

size $\approx \# \text{ root-to-leaf paths} \approx s^n$
(1) $F$ is SAT $\Rightarrow$ Ref($F$) has size-$n^{O(1)}$ resolution refutation

Read-once branching program for Ref($F$)

size: poly($n$)

Decision tree for Ref($F$)

size $\approx$ # root-to-leaf paths $\approx s^n$

(don’t expect upper bound to hold: would imply $NP \subseteq QP$)
“Universal refutation” (complete Binary tree: depth $n$)

$F$: $x_1 \bar{x}_2 x_4$
“Universal refutation” (complete tree: depth $h \ll n$)
"Universal refutation" (complete tree: depth $h \ll n$)

Automating Tree-Like Resolution in Time $n^{o(\log n)}$ is ETH-Hard
“Universal refutation” (complete tree: depth $h \ll n$)

\[
\frac{2^n}{h}
\]

\[
x_1x_2 \quad x_1\bar{x}_2 \quad \bar{x}_1x_2 \quad \bar{x}_1\bar{x}_2
\]

For optimal parameters: $h = \sqrt{n}$
ShallowRef\((F)\)

Variables for each block \(B\):
- \(2n\) variables (1 per literal): indicates clause
- \(2(\log s)\) variables: 2 pointers to children
  "derived from \(B_i\) and \(B_j\)"
- \(\log m\) variables: axiom-index \(j \in [m]\)

\(O(n^2s)\) variables

Axioms:
- Root: \(\perp\) clause
- Derived: valid resolution step
- Axiom: weakening of axiom

clauses of width \(O(\log s)\)

\(F:\)
ShallowRef\( (F) \)

Variables for each block \( B \):
- 2\( n \) variables (1 per literal): indicates clause
- 2(\( \log s \)) variables: 2 pointers to children “derived from \( B_i \) and \( B_j \)”
- \( \log m \) variables: axiom-index \( j \in [m] \)

\( O(n^2s) \) variables

Axioms:
- Root: \( \bot \) clause
- Derived: valid resolution step
- Axiom: weakening of axiom clauses of width \( O(\log s) \)

\( \sqrt{n} \)

\( F: \)
**ShallowRef**(\(F\))

Variables for each block \(B\):

- 2\(n\) variables (1 per literal): indicates clause
- \(O(2\sqrt{n})\) variables: \(2\sqrt{n}\) pointers to children “derived from all children”
- \(\log m\) variables: axiom-index \(j \in [m]\)

Axioms:

- Root: \(\bot\) clause
- Derived: valid resolution step
- Axiom: weakening of axiom

clauses of width \(O(\log s)\)
ShallowRef\((F)\)

Variables for each block \(B\):

- \(2n\) variables (1 per literal): indicates clause
- \(O(2\sqrt{n})\) variables: \(2\sqrt{n}\) pointers to children “derived from all children”
- \(\log m\) variables: axiom-index \(j \in [m]\)

\[2^{O(\sqrt{n})}\] variables

OBS. Bounded degree expander between layers (requires \(2^{\Omega(\sqrt{n})}\) blocks/layer)

- Root: \(\bot\) clause
- Derived: valid resolution step
- Axiom: weakening of axiom

clauses of width \(O(\log s)\)
(1) \( F \) is \( \text{SAT} \) \( \Rightarrow \) \( \text{ShallowRef}(F) \) has size-\( 2^O(\sqrt{n}) \) tree-like resolution refutation

Decision tree for \( \text{ShallowRef}(F) \)
(1) $F$ is \textbf{SAT} $\Rightarrow$ ShallowRef($F$) has size-$2^{O(\sqrt{n})}$ tree-like resolution refutation

Decision tree for \textbf{ShallowRef($F$)}

- $x^*$ satisfying assignment for $F$
- invariant: $x^*$ falsifies clause in block $B$

Start at root and keep invariant
until detect non-valid derivation step or
until reach leaf (cannot be weakening of axiom)
\( F \) is \( \text{SAT} \Rightarrow \text{ShallowRef}(F) \) has size-\( 2^{O(\sqrt{n})} \) tree-like resolution refutation

Decision tree for \( \text{ShallowRef}(F) \)

- \( x^* \) satisfying assignment for \( F \)
- invariant: \( x^* \) falsifies clause in block \( B \)

Start at root and keep invariant until detect non-valid derivation step or until reach leaf (cannot be weakening of axiom)

OBS. Bounded degree \( \Delta \) expander

\[
\text{tree-like refutation size} \approx \Delta \sqrt{n} = 2^{O(\sqrt{n})}
\]
(2) $F$ is **UNSAT** $\Rightarrow$ \texttt{ShallowRef}(F) requires tree-like res refutations size $2^{\Omega(n)}$
(2) $F$ is UNSAT $\Rightarrow$ ShallowRef$(F)$ requires tree-like res refutations size $2^{\Omega(n)}$

- $w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2^{c(1+\sqrt{n})}} \vdash \bot)/n)$
(2) \( F \) is UNSAT \( \Rightarrow \) ShallowRef(\( F \)) requires tree-like res refutations size \( 2^{\Omega(n)} \)

\[
\omega(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(\omega(\text{GPHP}_{2^{c\sqrt{n}}}^{2^{(c+1)\sqrt{n}}}) \vdash \bot)/n \geq \Omega(2^{c\sqrt{n}}/n)
\]
(2) \( F \) is UNSAT \( \Rightarrow \) ShallowRef\((F)\) requires tree-like res refutations size \( 2^{\Omega(n)} \)

- \( w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(\frac{w(\text{GPHP}_{2^{c\sqrt{n}}} \vdash \bot)}{n}) \geq \Omega(\frac{2^{c\sqrt{n}}}{n}) \)

- [BW’01] size of tree-like resolution refutation of \( \varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)} \)
(2) $F$ is UNSAT $\Rightarrow$ ShallowRef$(F)$ requires tree-like res refutations size $2^{\Omega(n)}$

- $w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2^{c+1}\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$

- [BW’01] size of tree-like resolution refutation of $\varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)}$

- ShallowRef$(F)$ has width $O(1)$ and $\#$ variables: $\text{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$
(2) $F$ is $\text{UNSAT} \Rightarrow \text{ShallowRef}(F)$ requires tree-like res refutations size $2^{\Omega(n)}$

- $w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2^{(c+1)\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$

- [BW’01] size of tree-like resolution refutation of $\varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)}$

- $\text{ShallowRef}(F)$ has width $O(1)$ and # variables: $\text{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

ShallowRef$(F)$ requires tree-like res refutation size $2^{\Omega(2^{c\sqrt{n}}/n)} \geq 2^{\Omega(n)}$
\[ w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2^{(c+1)\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n) \]

\[ [\text{BW'01}] \text{ size of tree-like resolution refutation of } \varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)} \]

\[ \text{ShallowRef}(F) \text{ has width } O(1) \text{ and } \# \text{ variables: } \text{poly}(n) \cdot 2^{(c+1)\sqrt{n}} \]

\text{ShallowRef}(F) \text{ requires tree-like res refutation size } 2^{\Omega(2^{c\sqrt{n}}/n)} \geq 2^\Omega(n)
(2) $F$ is **UNSAT** $\Rightarrow$ **ShallowRef**($F$) requires dag-like res refutations size $2^{\Omega(n)}$

$w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2^{2c\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$

[BW’01] size of tree-like resolution refutation of $\varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)}$

**ShallowRef**($F$) has width $O(1)$ and $\#$ variables: $\text{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

**ShallowRef**($F$) requires tree-like res refutation size $2^{\Omega(2^{c\sqrt{n}}/n)} \geq 2^{\Omega(n)}$
(2) \( F \) is UNSAT \( \Rightarrow \) ShallowRef\((F)\) requires dag-like res refutations size \( 2^{\Omega(n)} \)

- \( w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2^{(c+1)\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^c \sqrt{n} / n) \)

- [BW’01] size of resolution refutation of \( \varphi \) \( \geq \exp \left( \Omega \left( \frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\# \text{var}} \right) \right) \)

- ShallowRef\((F)\) has width \( O(1) \) and \# variables: \( \text{poly}(n) \cdot 2^{(c+1)\sqrt{n}} \)

ShallowRef\((F)\) requires tree-like res refutation size \( 2^{\Omega(2^c \sqrt{n} / n)} \geq 2^{\Omega(n)} \)
(2) $F$ is UNSAT $\implies$ ShallowRef($F$) requires dag-like res refutations size $2^{\Omega(n)}$

- $w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2^{(c+1)\sqrt{n}}/n}) \vdash \bot)/n) \geq \Omega(2^c\sqrt{n}/n)$

- [BW’01] size of resolution refutation of $\varphi \geq \exp \left( \Omega \left( \frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\# \text{var}} \right) \right)$

- ShallowRef($F$) has width $O(1)$ and \# variables: $\text{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

**ShallowRef($F$) requires dag-like res refutation size $2^{\tilde{\Omega}(2^{2c-(c+1}\sqrt{n})} \geq 2^{\Omega(n)}$**
(2) \( F \) is UNSAT \( \Rightarrow \) ShallowRef\( (F) \) requires dag-like res refutations size \( 2^{\Omega(n)} \)

- \( w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(\frac{w(\text{GPHP}_{2c\sqrt{n}} \vdash \bot)}{n}) \geq \Omega(\frac{2^{c\sqrt{n}}}{n}) \)

- [BW’01] size of resolution refutation of \( \varphi \) \( \geq \exp\left(\Omega\left(\frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\#\text{var}}\right)\right) \)

- ShallowRef\( (F) \) has width \( O(1) \) and \# variables: poly\( (n) \cdot 2^{(c+1)\sqrt{n}} \)

ShallowRef\( (F) \) requires dag-like res refutation size \( 2^{\tilde{\Omega}(2^{2(c-(c+1))\sqrt{n}})} \geq 2^{\Omega(n)} \)

choose \( c = 2 \)
(2) $F$ is \textbf{UNSAT} $\Rightarrow$ \textbf{ShallowRef}(F) requires dag-like res refutations size $2^{\Omega(n)}$

- $w(\text{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\text{GPHP}_{2c\sqrt{n}}^{2(c+1)\sqrt{n}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$

- [BW'01] size of resolution refutation of $\varphi \geq \exp \left( \Omega \left( \frac{w(\varphi \vdash \bot) - w(\varphi)^2}{\#\text{var}} \right) \right)$

- \textbf{ShallowRef}(F) has width $O(1)$ and \# variables: $\textbf{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

\textbf{ShallowRef}(F) requires dag-like res refutation size $\tilde{2^{\Omega\left(2^{2c-(c+1)\sqrt{n}}\right)}} \geq 2^{\Omega(n)}$

OBS. this same lower bound holds for PC:
- degree lower bound for $\textbf{GPHP}_s^{\text{poly}(s)}$
- similar size-degree relation

---

choose $c = 2$
Generalization

If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\epsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$
3. in time $\text{poly}(n)$ then $W[\text{P}] = \text{FPT}$
Generalization

If tree-like resolution is automatable:

1. in time $n^{o(\log n)}$ then ETH is false
2. in time $n^{O(\log^{1-\varepsilon} n)}$ then $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\varepsilon}/2)})$
3. in time $\text{poly}(n)$ then $W[\text{P}] = \text{FPT}$

Classical result in parameterized complexity [ADF’95]

If $\exists$ algorithm that given $n$-variate circuit $C$ of size $m$ decides if $C$ is satisfiable in time $\text{poly}(m) \cdot 2^{o(n)}$ then $W[\text{P}] = \text{FPT}$. 
Generalization

Classical result in parameterized complexity [ADF’95]

If there exists an $n$-variate circuit $C'$ of size $m$ that decides if $C'$ is satisfiable in time $\text{poly}(m) \cdot 2^{o(n)}$ then $W[1] = \text{FPT}$. 
Generalization

Classical result in parameterized complexity [ADF’95]

If there exists an algorithm that given \( n \)-variate circuit \( C \) of size \( m \) decides if \( C \) is satisfiable in time \( \text{poly}(m) \cdot 2^{o(n)} \), then \( W[P] = \text{FPT} \).

Main Theorem

\( \exists \) algorithm that given \( n \)-variate circuit \( C \) of size \( m \) outputs CNF \( \mathcal{A}(C) \) in time \( \text{poly}(m) \cdot 2^{O(\sqrt{n})} \) s.t.

1. \( C \) is SAT \( \Rightarrow R^{*}(\mathcal{A}(C)) \leq \text{poly}(m) \cdot 2^{O(\sqrt{n})} \)
2. \( C \) is UNSAT \( \Rightarrow PC(\mathcal{A}(C)) \geq 2^{\Omega(n)} \)
ShallowRef\((C)\)

\[ F : \]

\[ m \]

\[ \sqrt{n} \]

\[ 2^{\sqrt{n}} \]
ShallowRef$(C)$
Proof summary

- **Upper bound**: follow the path given by satisfying assignment
- **Lower bound**: width/degree lower bound (from PHP), size-width/degree relation
Proof summary

- **Upper bound**: follow the path given by satisfying assignment
- **Lower bound**: width/degree lower bound (from PHP), size-width/degree relation
- **Lower bound**: width/degree lower bound (from PHP), lift/relativize (needed for CP)
Proof summary

- **Upper bound**: follow the path given by satisfying assignment
- **Lower bound**: width/degree lower bound (from PHP), size-width/degree relation
- **Lower bound**: width/degree lower bound (from PHP), lift/relativize (needed for CP)

Open problems

- Other proof systems: SOS, bounded-depth Frege, stabbing planes
Proof summary

- **Upper bound**: follow the path given by satisfying assignment
- **Lower bound**: width/degree lower bound (from PHP), size-width/degree relation
- **Lower bound**: width/degree lower bound (from PHP), lift/relativize (needed for CP)

Open problems

- Other proof systems: SOS, bounded-depth Frege, stabbing planes
- Tree-like proof systems: Res\((k)\), Res\((\oplus)\), CP
Proof summary

- **Upper bound**: follow the path given by satisfying assignment
- **Lower bound**: width/degree lower bound (from PHP), size-width/degree relation
- **Lower bound**: width/degree lower bound (from PHP), lift/relativize (needed for CP)

Open problems

- Other proof systems: SOS, bounded-depth Frege, stabbing planes
- Tree-like proof systems: Res($k$), Res($\oplus$), CP
- Weak automatability?
Proof summary

- **Upper bound**: follow the path given by satisfying assignment
- **Lower bound**: width/degree lower bound (from PHP), size-width/degree relation
- **Lower bound**: width/degree lower bound (from PHP), lift/relativize (needed for CP)

Open problems

- Other proof systems: SOS, bounded-depth Frege, stabbing planes
- Tree-like proof systems: $\text{Res}(k)$, $\text{Res}(\oplus)$, CP
- Weak automatability?

Thanks!