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

Resolution proof system

Given UNSAT CNF formula F: $\underbrace{(\overline{y} \vee \overline{z})} \wedge \underbrace{(\overline{x} \vee \overline{z})} \wedge \underbrace{(x \vee y)} \wedge \underbrace{(x \vee \overline{y} \vee z)} \wedge \underbrace{(\overline{x} \vee z)}$

clauses/axioms

Resolution proof system

Given UNSAT CNF formula F: $\underbrace{(\overline{y} \vee \overline{z})} \wedge \underbrace{(\overline{x} \vee \overline{z})} \wedge \underbrace{(x \vee y)} \wedge \underbrace{(x \vee \overline{y} \vee z)} \wedge \underbrace{(\overline{x} \vee z)}$

clauses/axioms

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

Refutation: Derivation of empty clause $\perp$

Resolution proof system

Given UNSAT CNF formula F: $\underbrace{(\overline{y} \vee \overline{z})} \wedge \underbrace{(\overline{x} \vee \overline{z})} \wedge \underbrace{(x \vee y)} \wedge \underbrace{(x \vee \overline{y} \vee z)} \wedge \underbrace{(\overline{x} \vee z)}$

clauses/axioms

Resolution rule: $\dfrac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$

Refutation: Derivation of empty clause $\perp$

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

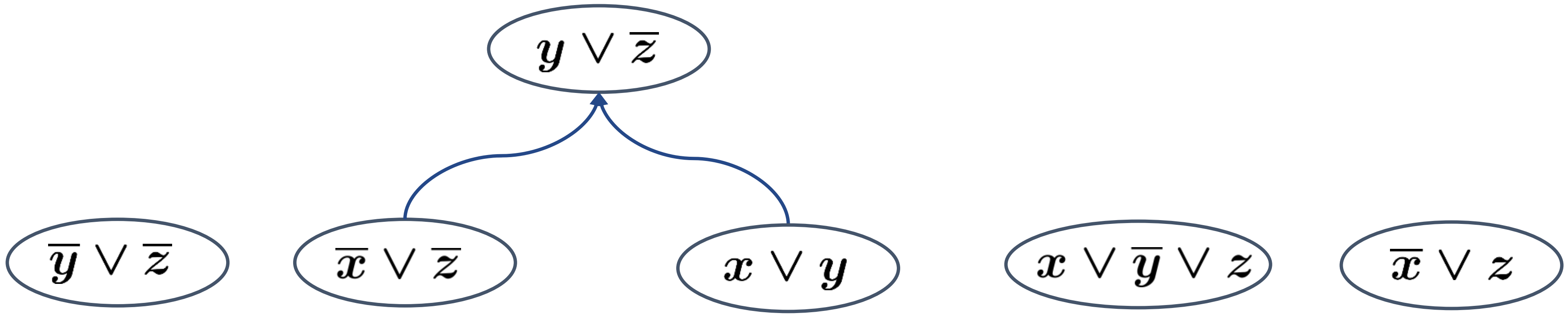
Resolution proof system

Given UNSAT CNF formula F: $\underbrace{(\overline{y} \vee \overline{z})}\wedge \underbrace{(\overline{x} \vee \overline{z})} \wedge \underbrace{(x \vee y)} \wedge \underbrace{(x \vee \overline{y} \vee z)} \wedge \underbrace{(\overline{x} \vee z)}$

clauses/axioms

Resolution rule: $\dfrac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$

Refutation: Derivation of empty clause $\perp$



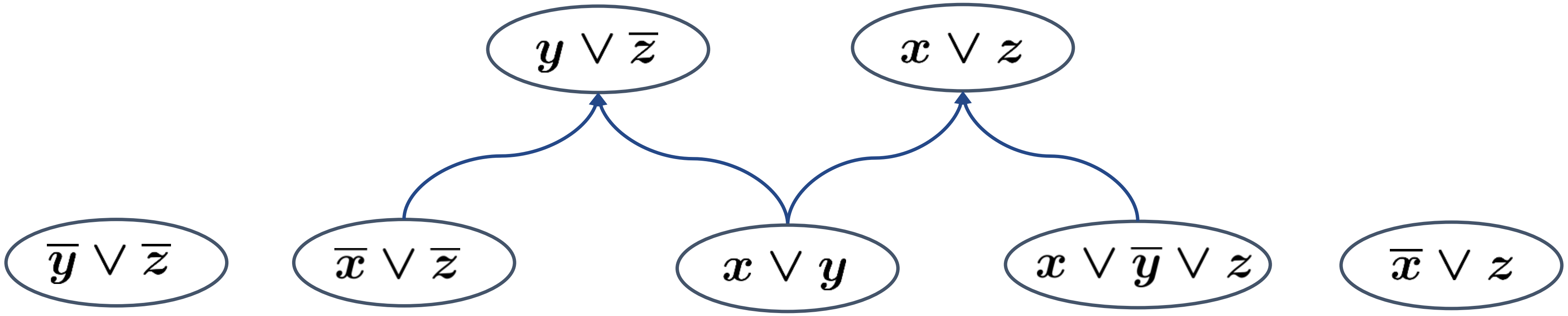
Resolution proof system

Given UNSAT CNF formula F: $\underbrace{(\overline{y} \vee \overline{z})} \wedge \underbrace{(\overline{x} \vee \overline{z})} \wedge \underbrace{(x \vee y)} \wedge \underbrace{(x \vee \overline{y} \vee z)} \wedge \underbrace{(\overline{x} \vee z)}$

clauses/axioms

Resolution rule: $\dfrac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$

Refutation: Derivation of empty clause $\perp$



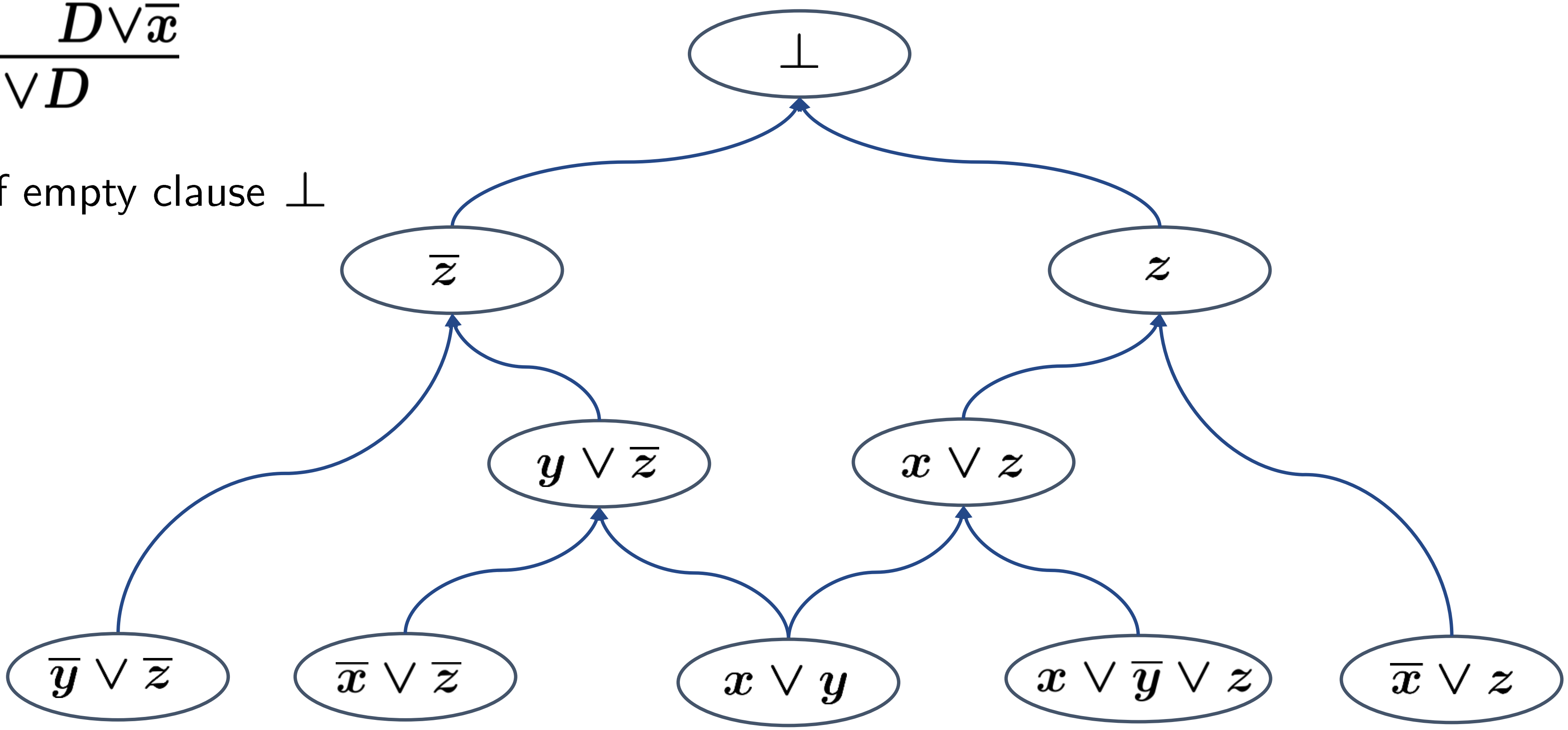
Resolution proof system

Given UNSAT CNF formula F: $\underbrace{(\overline{y} \vee \overline{z})} \wedge \underbrace{(\overline{x} \vee \overline{z})} \wedge \underbrace{(x \vee y)} \wedge \underbrace{(x \vee \overline{y} \vee z)} \wedge \underbrace{(\overline{x} \vee z)}$

clauses/axioms

Resolution rule: $\dfrac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$

Refutation: Derivation of empty clause $\perp$



Resolution proof system

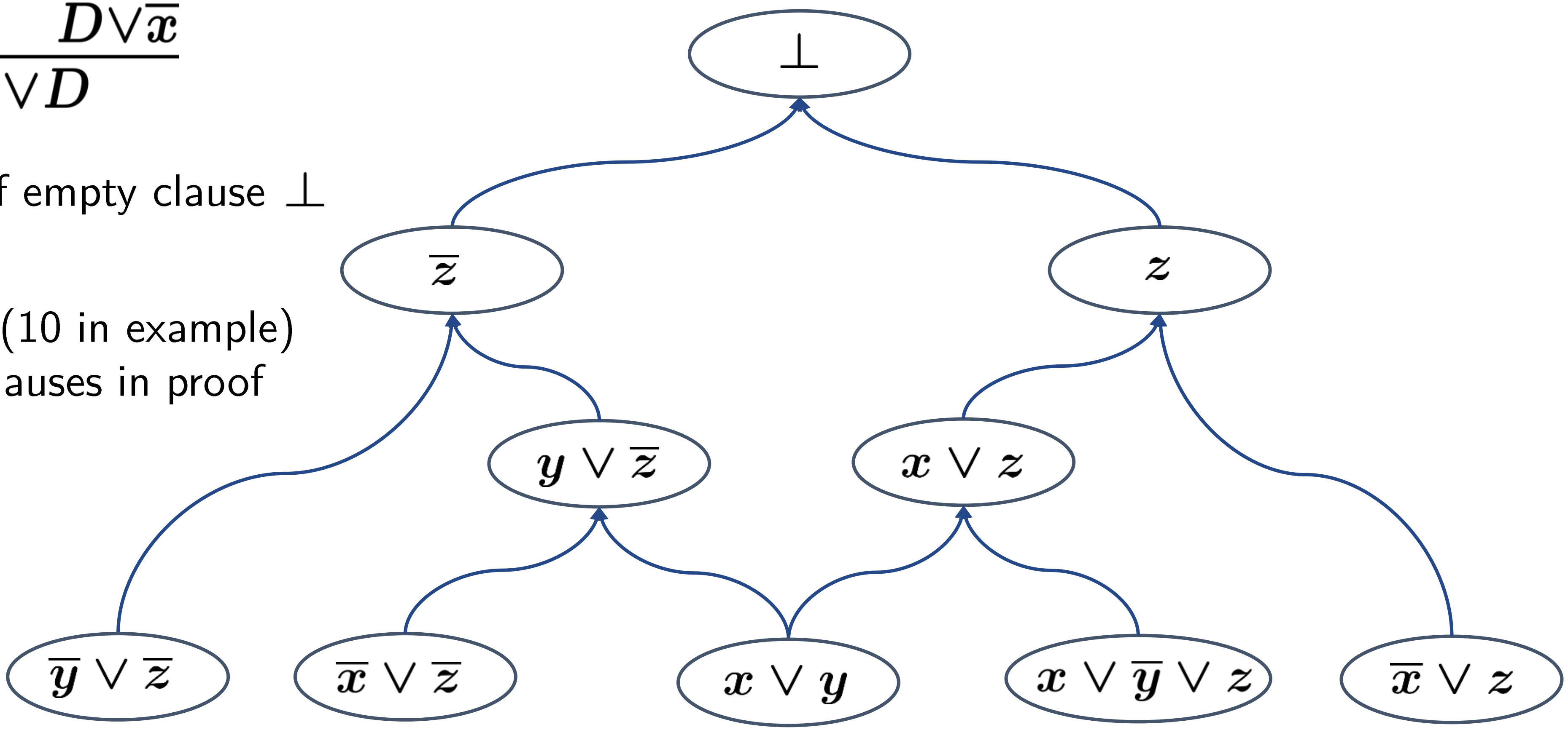
Given UNSAT CNF formula $\boldsymbol{F}$: $(\overline{y} \vee \overline{z}) \wedge (\overline{x} \vee \overline{z}) \wedge (x \vee y) \wedge (x \vee \overline{y} \vee z) \wedge (\overline{x} \vee z)$

clauses/axioms

Resolution rule: $\dfrac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$

Refutation: Derivation of empty clause $\perp$

Size: # clauses in proof (10 in example)
Width: *max* # literals/clauses in proof



Resolution proof system

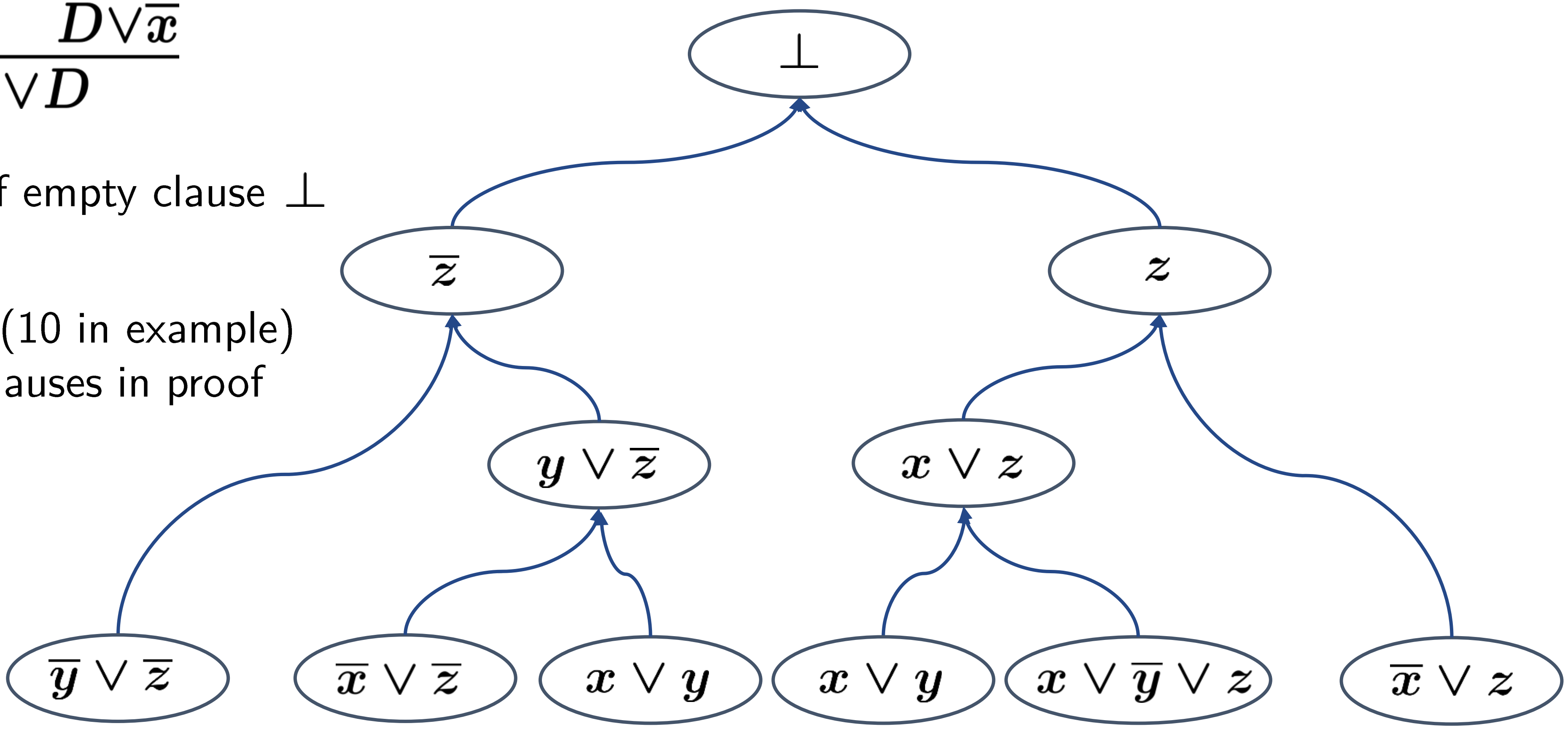
Given UNSAT CNF formula F: $\underbrace{(\overline{y} \vee \overline{z})} \wedge \underbrace{(\overline{x} \vee \overline{z})} \wedge \underbrace{(x \vee y)} \wedge \underbrace{(x \vee \overline{y} \vee z)} \wedge \underbrace{(\overline{x} \vee z)}$

clauses/axioms

Resolution rule: $\dfrac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$

Refutation: Derivation of empty clause $\perp$

Size: # clauses in proof (10 in example)
Width: *max* # literals/clauses in proof



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 $\boldsymbol{F}$ 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 $\boldsymbol{F}$ 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 $\boldsymbol{F}$
outputs $\mathcal{P}$-refutation of $\boldsymbol{F}$ in time $f(n)$

Automatability [BPR'97]

How hard is it to **find** proofs/refutations?

Suppose unsat CNF $\boldsymbol{F}$ 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 $\boldsymbol{F}$
outputs $\mathcal{P}$-refutation of $\boldsymbol{F}$ in time $f(n)$

size of smallest $\mathcal{P}$-refutation of $\boldsymbol{F}$
plus the size of $\boldsymbol{F}$

Atserias-Müller ‘19

If resolution is automatable:

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

Atserias-Müller '19

If resolution is automatable:

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

Generalizations

1. Cutting planes [GKMP'20]
2. Res($\boldsymbol{k}$) [Gar'20]
3. Algebraic proof systems (NS, PC, SA) [dRGNPRS'21]

Tree-like resolution is automatable in time $n^{O(\log n)}$ [BP'96]

Tree-like resolution is automatable in time $s^{O(\log n)}$ [BP'96]

\# variables

size of smallest refutation

Tree-like resolution is automatable in time $s^{O(\log n)}$ [BP'96]

\# variables

size of smallest refutation

Given UNSAT CNF formula F

Tree-like resolution is automatable in time $s^{O(\log n)}$ [BP'96]

\# variables

size of smallest refutation

Given UNSAT CNF formula $\boldsymbol{F}$

Suppose $\boldsymbol{s}$ is known

Tree-like resolution is automatable in time $s^{O(\log n)}$ [BP'96]

\# variables

size of smallest refutation

Given UNSAT CNF formula F

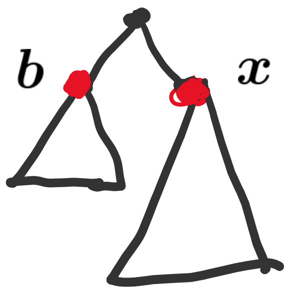
Suppose s is known

R^* : size of smallest tree-like resolution refutation

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

$\perp$

$x = b$ $\quad$ $x = 1 - b$



Tree-like resolution is automatable in time $s^{O(\log n)}$ [BP'96]

\# variables

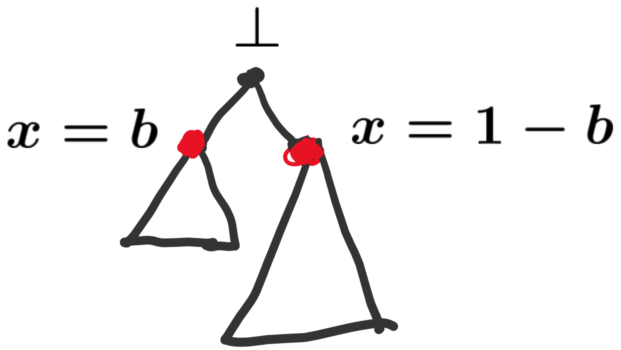
size of smallest refutation

Given UNSAT CNF formula F

Suppose s is known

R^* : size of smallest tree-like resolution refutation

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



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

$$T(n, s) = s^{O(\log n)}$$

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 $\mathbf{poly}(n)$ then W[P] = 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 $\mathbf{poly}(n)$ then W[P] $=$ FPT [AR'01]
2. in time $n^{O(\log^{1/7-\epsilon} \log n)}$ 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 $\mathbf{poly}(n)$ then W[P] = 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 $\mathbf{poly}(n)$ then W[P] $=$ 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 NP $\subseteq$ 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 $\mathbf{poly}(n)$ then W[P] $=$ 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 NP $\subseteq$ DTIME$(2^{O(n^{1-\epsilon/2})})$
3. in time $\mathbf{poly}(n)$ then W[P] $=$ 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 $\mathsf{NP} \subseteq \mathsf{DTIME}(2^{O(n^{1-\epsilon/2})})$
3. in time $\mathrm{poly}(n)$ then $\mathsf{W[P]} = \mathsf{FPT}$

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

Main Theorem

$\exists$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

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 $\mathsf{NP} \subseteq \mathsf{DTIME}(2^{O(n^{1-\epsilon/2})})$

$\exists$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

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 $\mathsf{NP} \subseteq \mathsf{DTIME}(2^{O(n^{1-\epsilon/2})})$

$\exists$ 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 $\mathbb{A}$ 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 $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{O(n^{1-\epsilon/2})})$

$\exists$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

size of smallest refutation of F plus the size of F

Proof. Suppose $\mathbb{A}$ 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 $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{O(n^{1-\epsilon/2})})$

$\exists$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

size of smallest refutation of F plus the size of F

Proof. Suppose $\mathbb{A}$ automatates 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 $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{O(n^{1-\epsilon/2})})$

$\exists$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

size of smallest refutation of F plus the size of F

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

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

$\mathcal{A}(F)$: # 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 NP $\subseteq$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

size of smallest refutation of F plus the size of F

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

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

$\mathcal{A}(F)$: # 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 NP $\subseteq$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

size of smallest refutation of F plus the size of F

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

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

$\mathcal{A}(F)$: # 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

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 $\text{NP} \subseteq \text{DTIME}(2^{O(n^{1-\epsilon/2})})$

$\exists$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

size of smallest refutation of F plus the size of F

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

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

$\mathcal{A}(F)$: # 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

obs. $o(\log N) = o(\sqrt{n})$

Atserias-Müller

Automating Resolution is NP-Hard – Simplified [dRGNPRS'21]

Atserias-Müller '19

$\exists$ 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

$\exists$ 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 $\Rightarrow R(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

Main Theorem (tree-like resolution)

$\exists$ 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 $\Rightarrow R^*(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

Atserias-Müller '19

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

$\mathrm{Ref}(F)$

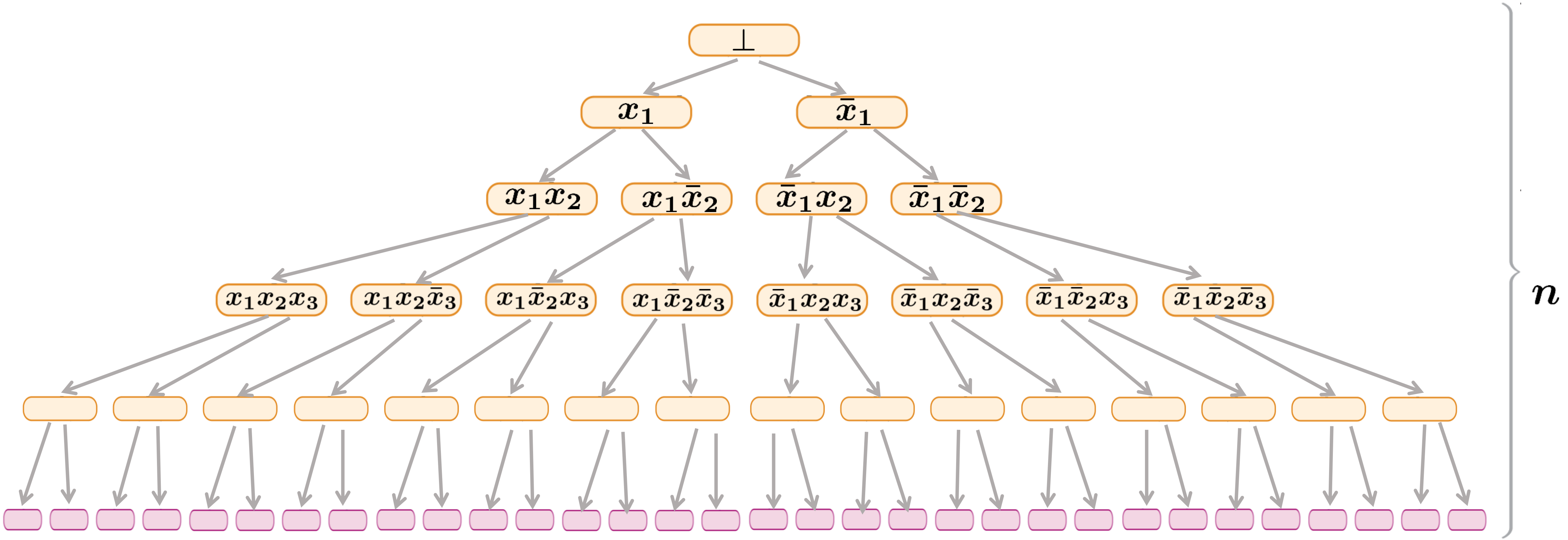
1. F is SAT $\Rightarrow R(\mathcal{A}(F)) \leq n^{O(1)}$
2. F is UNSAT $\Rightarrow R(\mathcal{A}(F)) \geq 2^{\Omega(n)}$

Main Theorem (tree-like resolution)

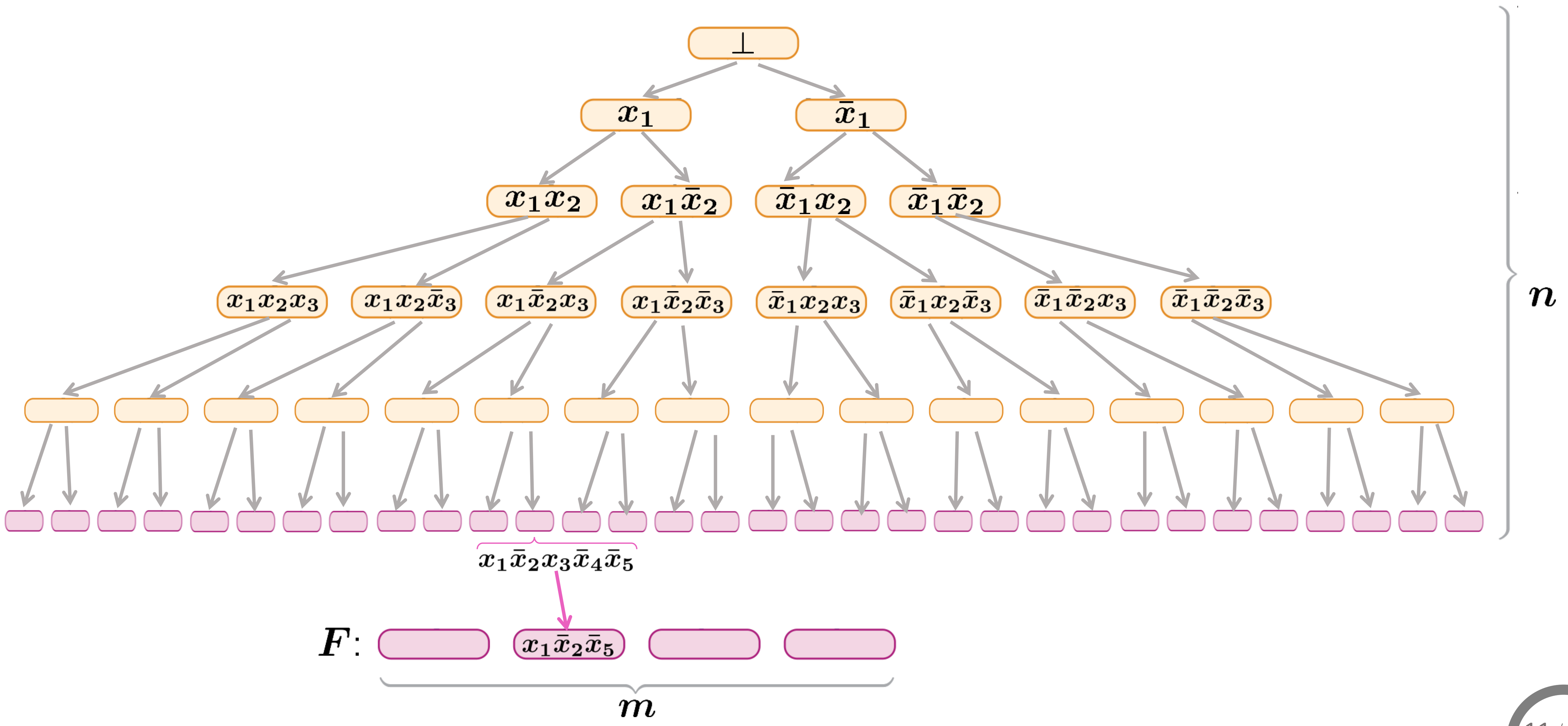
$\exists$ algorithm that given n-variate F outputs $\mathcal{A}(F)$ in time $2^{O(\sqrt{n})}$ s.t.

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

"Universal refutation" (complete Binary tree)

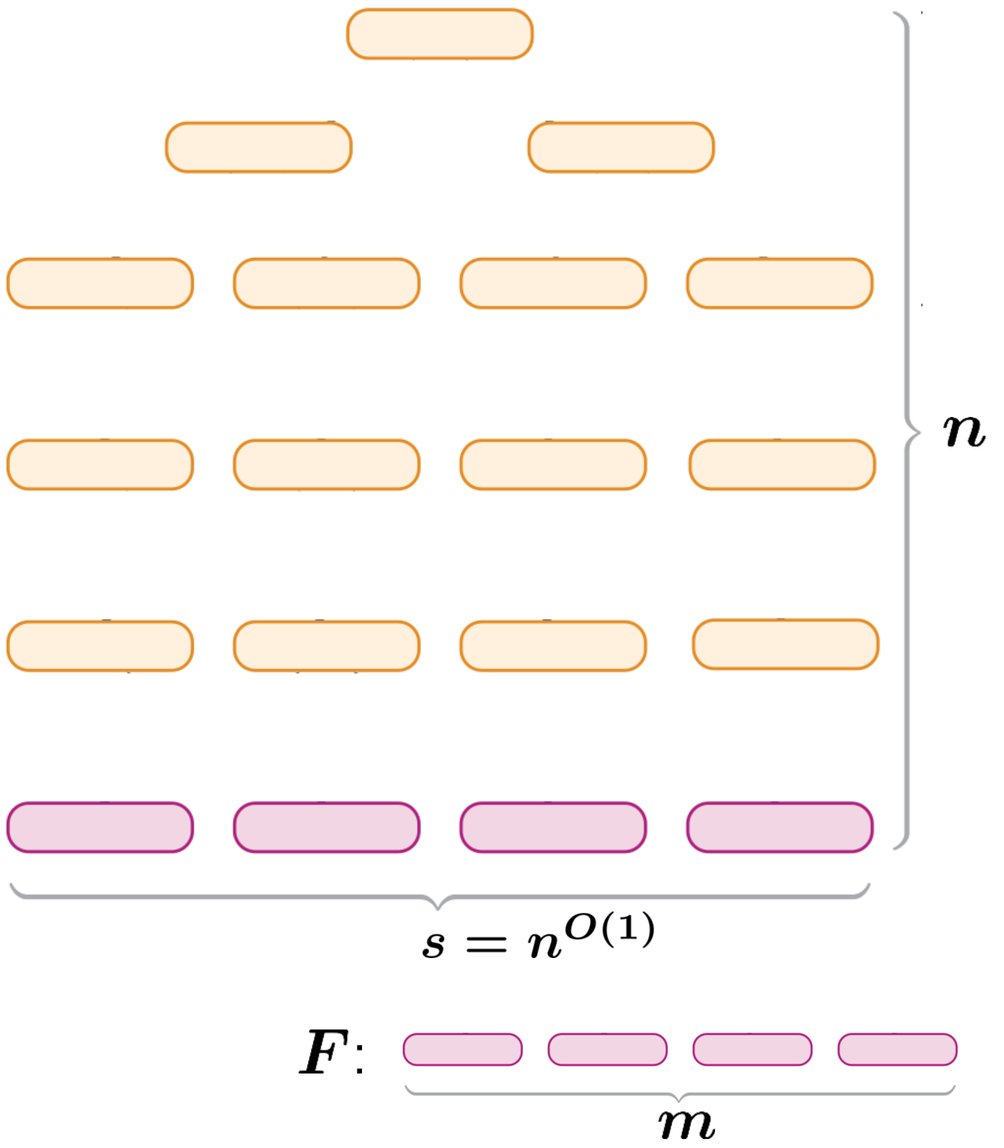


"Universal refutation" (complete Binary tree)



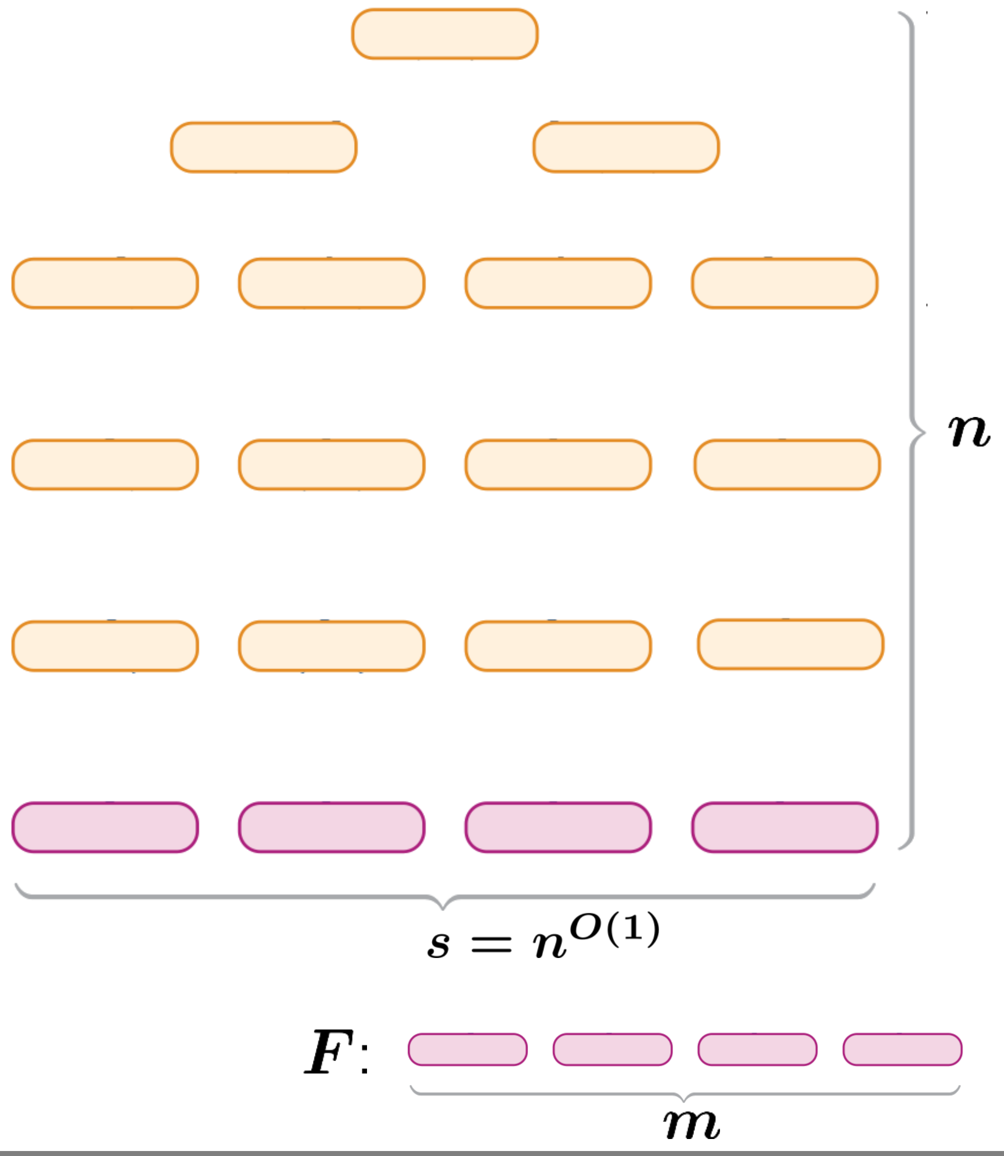
$\mathrm{Ref}(F)$

Encodes "F has short resolution refutation"



$\mathrm{Ref}(F)$

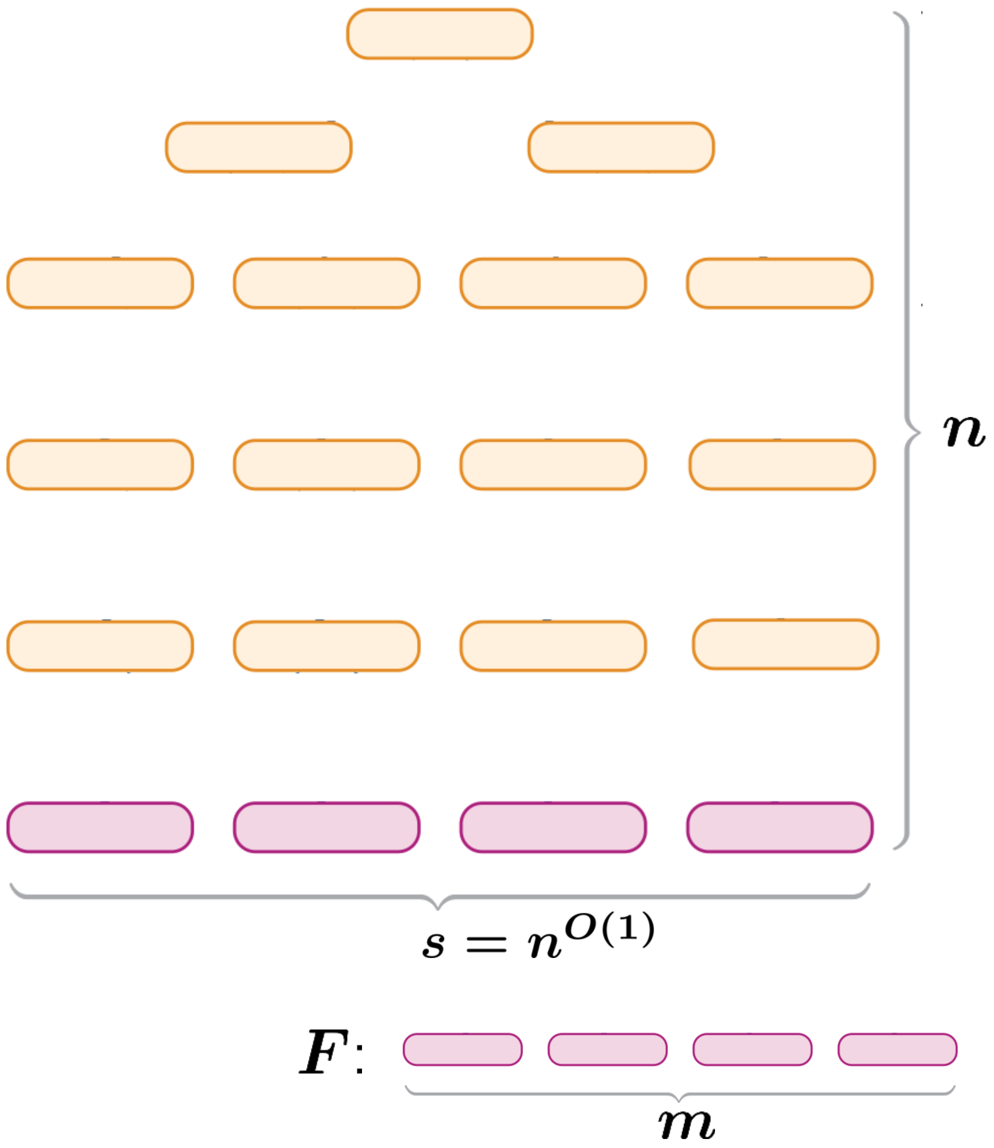
Encodes "F has short $\overbrace{\text{ resolution}}^{\text{structured}}$ refutation"



$\mathrm{Ref}(F)$

Encodes "F has $\underbrace{\text{short resolution}}$ refutation"
structured

Variables for each block B:

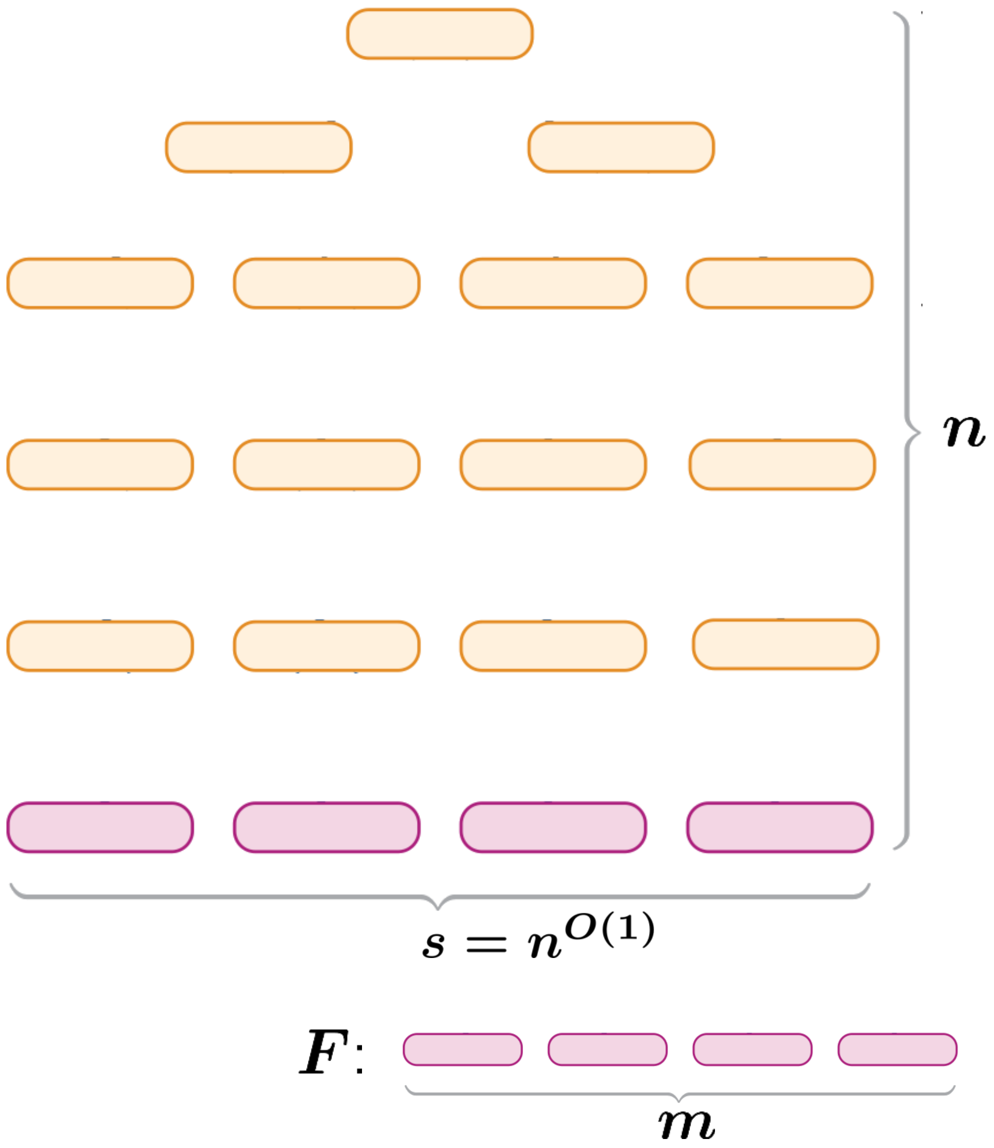


$\mathrm{Ref}(F)$

Encodes "F has short structured resolution refutation"

Variables for each block B:

- $2n$ variables (1 per literal): indicates clause

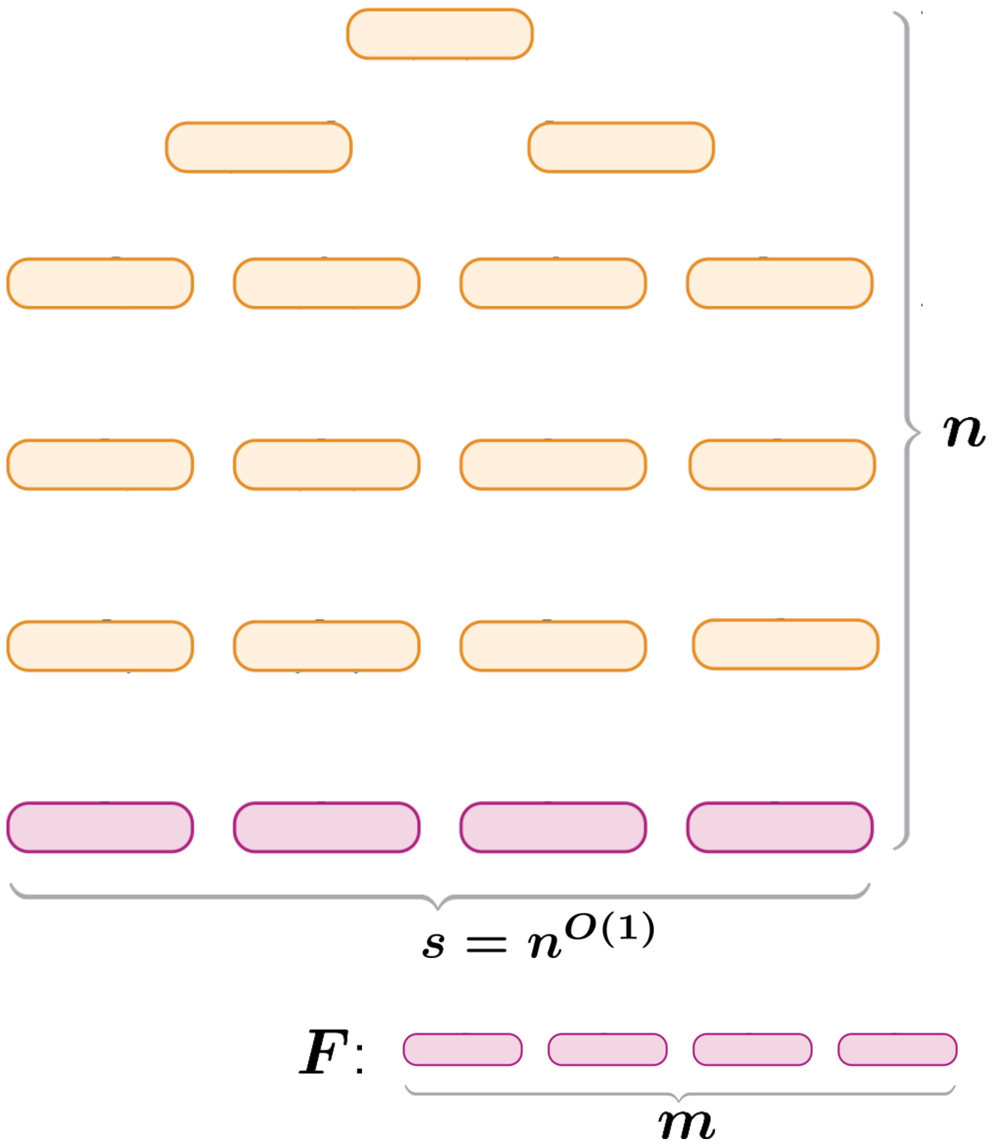


$\mathbf{Ref}(F)$

Encodes "F has short structured resolution refutation"

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"

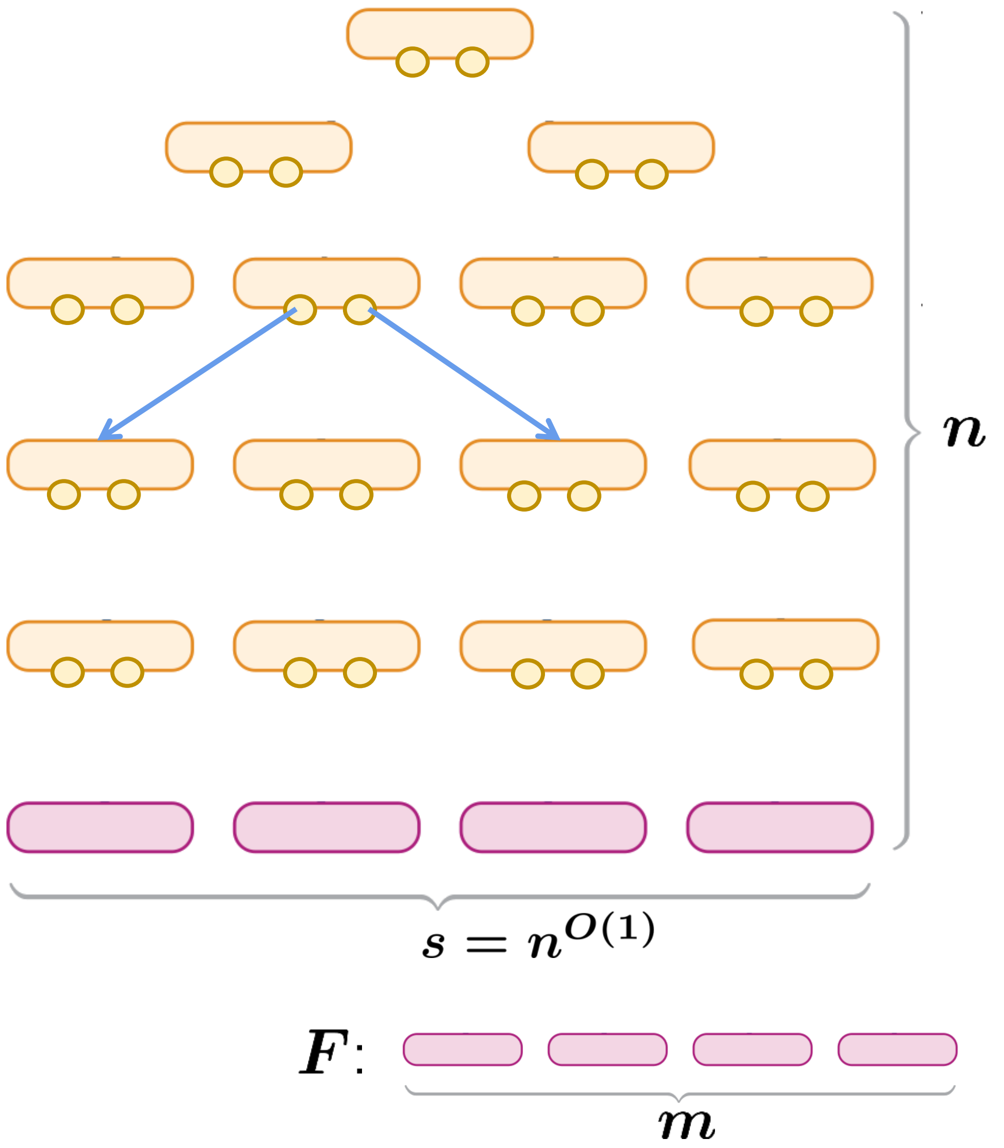


$\mathbf{Ref}(F)$

Encodes "F has short structured resolution refutation"

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"

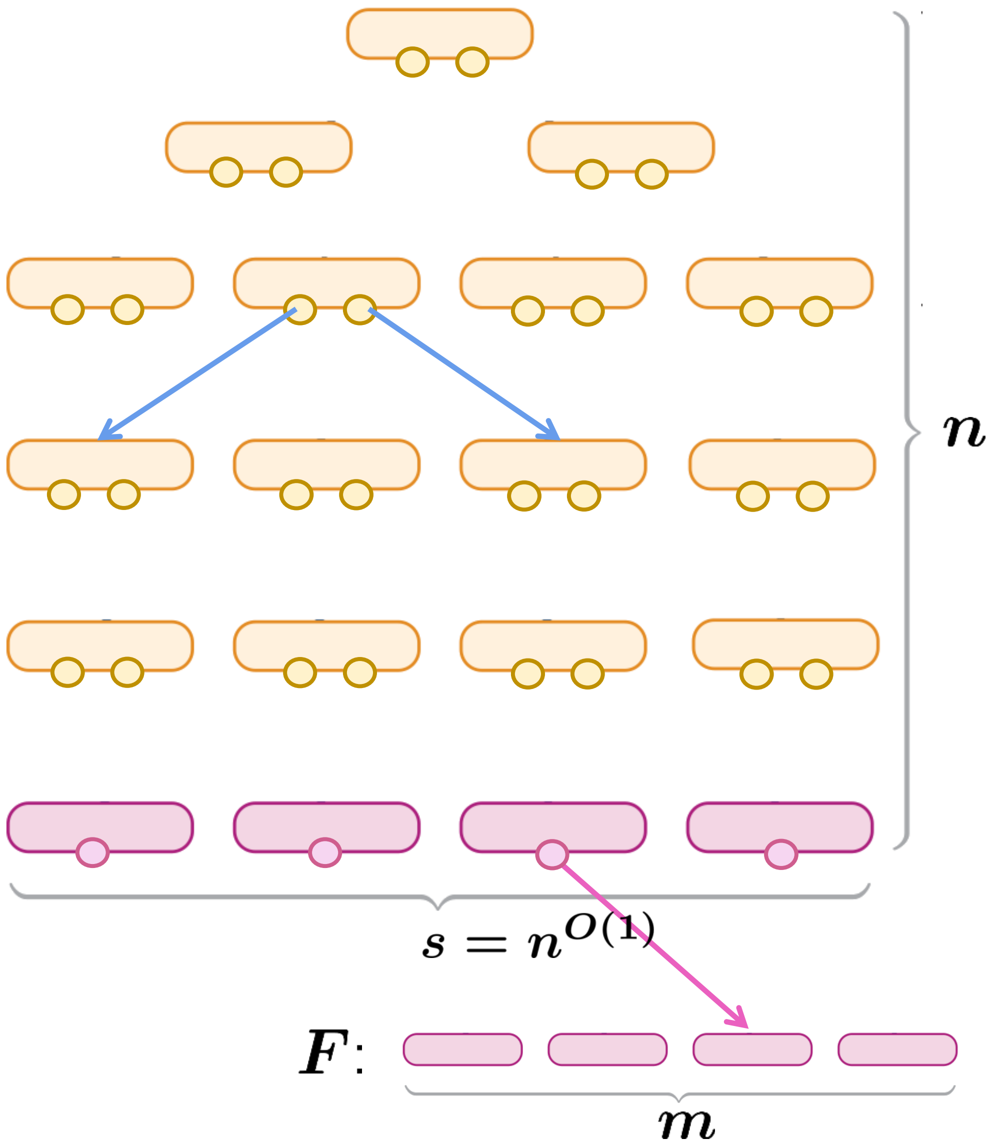


$\mathrm{Ref}(F)$

Encodes "F has short $\underbrace{\text{resolution refutation}}$" structured

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 $j \in [m]$



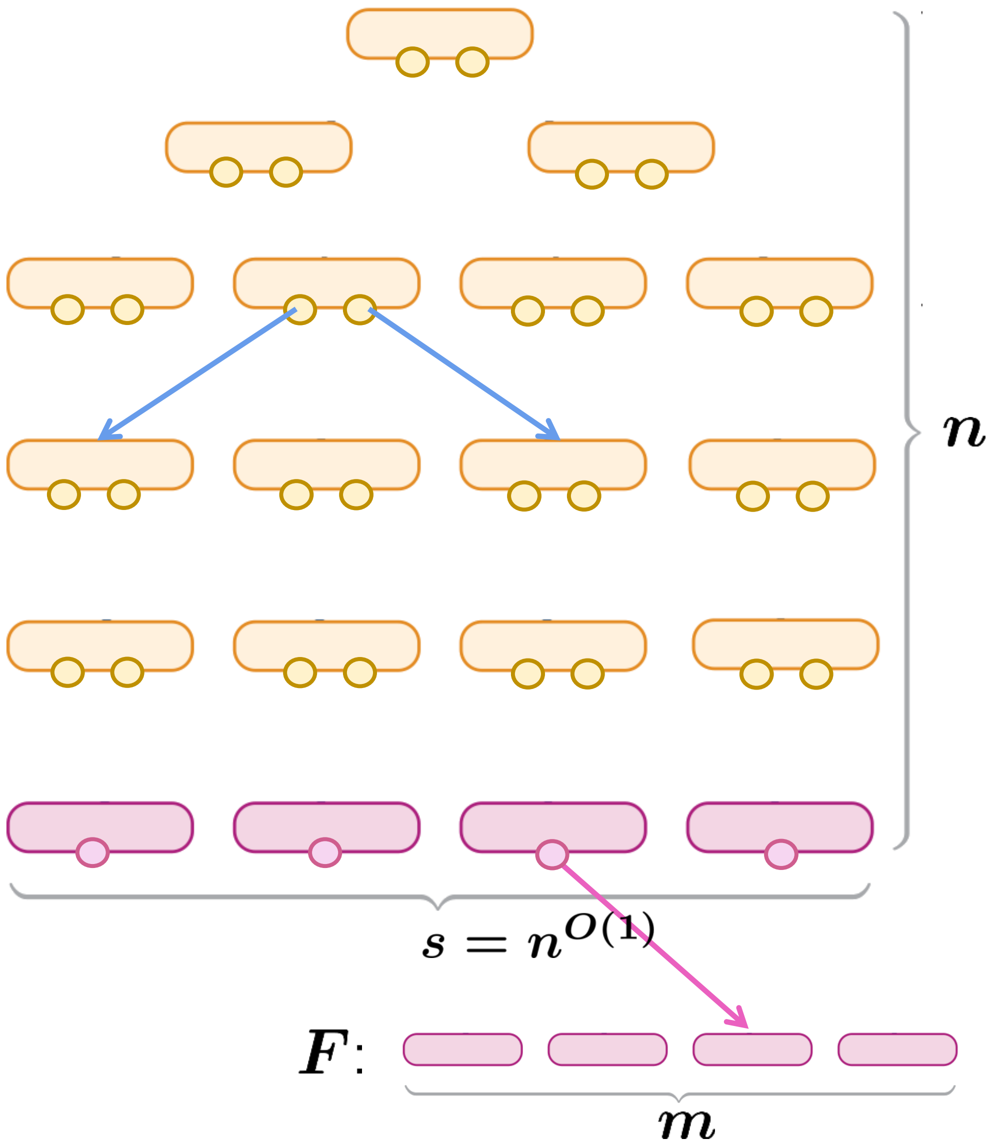
$\mathrm{Ref}(F)$

Encodes "F has short structured resolution refutation"

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 $j \in [m]$

$\boxed{O(n^2 s) \text{ variables}}$



$\mathbf{Ref}(F)$

structured
Encodes "F has short resolution refutation"

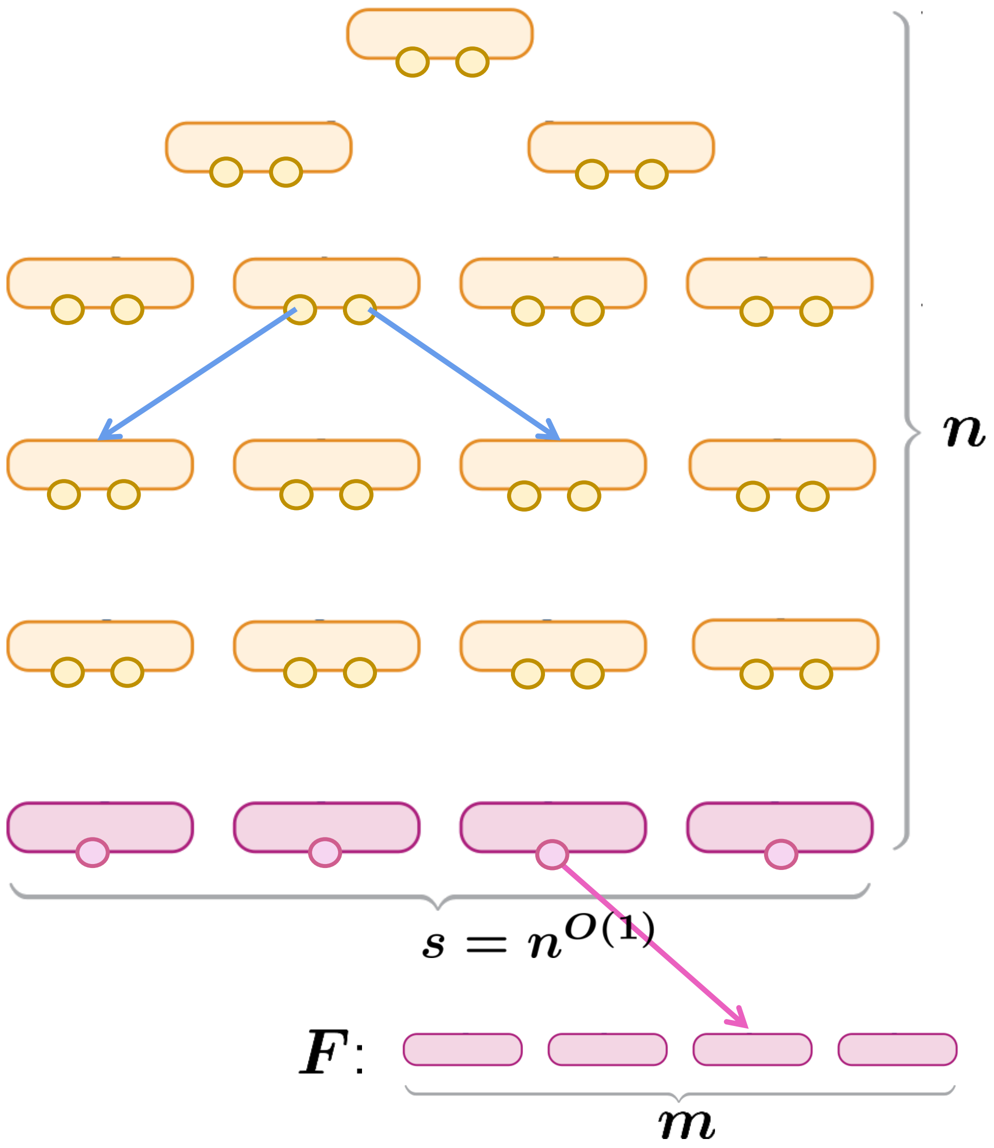
Variables for each block B:

- $2n$ 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 $j \in [m]$

$\boxed{O(n^2 s) \text{ variables}}$

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

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



$\mathbf{Ref}(F)$

structured

Encodes "F has short resolution refutation"

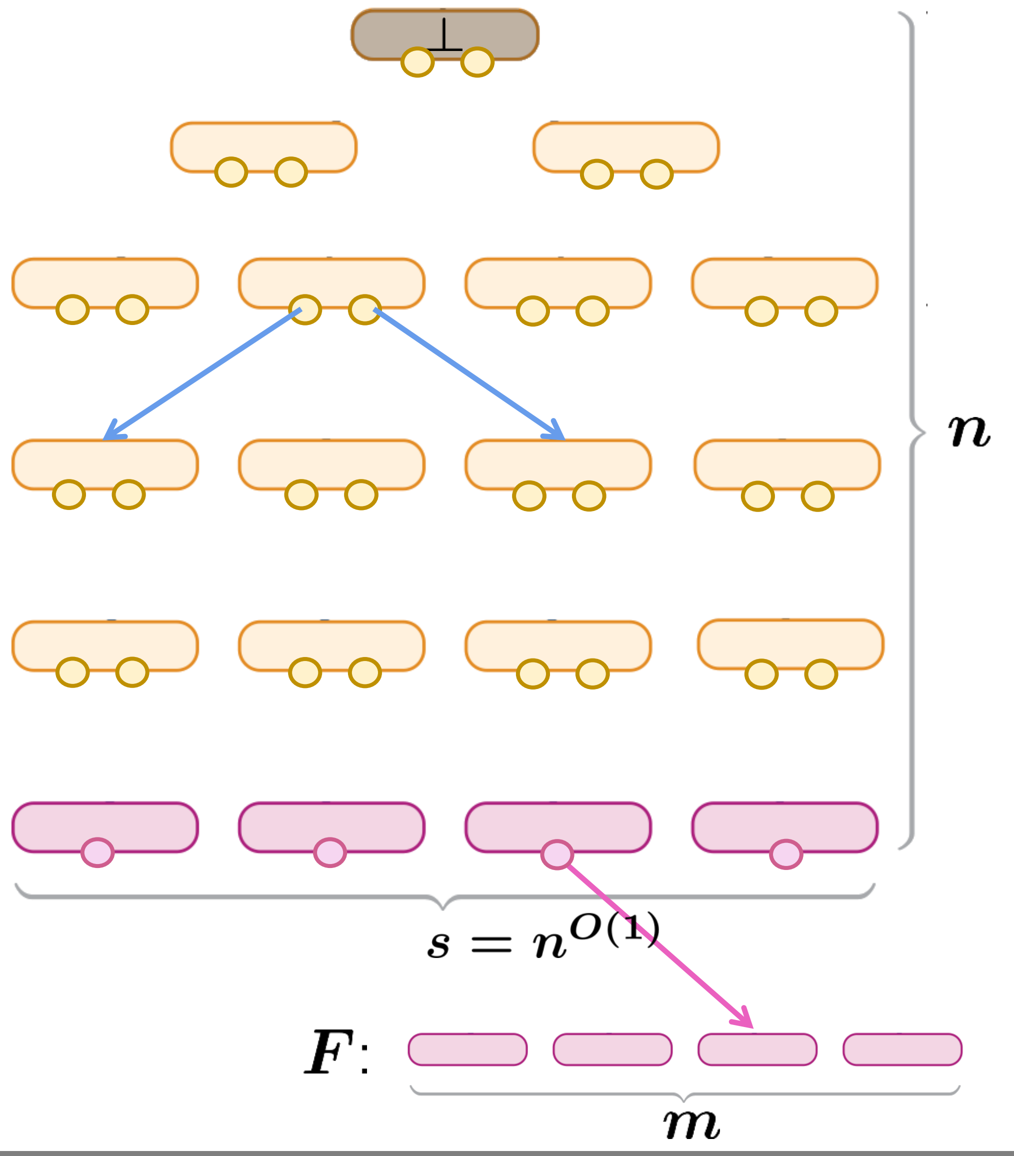
Variables for each block B:

- $2n$ 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 $j \in [m]$

$\boxed{O(n^2 s) \text{ variables}}$

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

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



$\mathbf{Ref}(F)$

Encodes "F has short $\underbrace{\text{resolution}}$ refutation"
structured

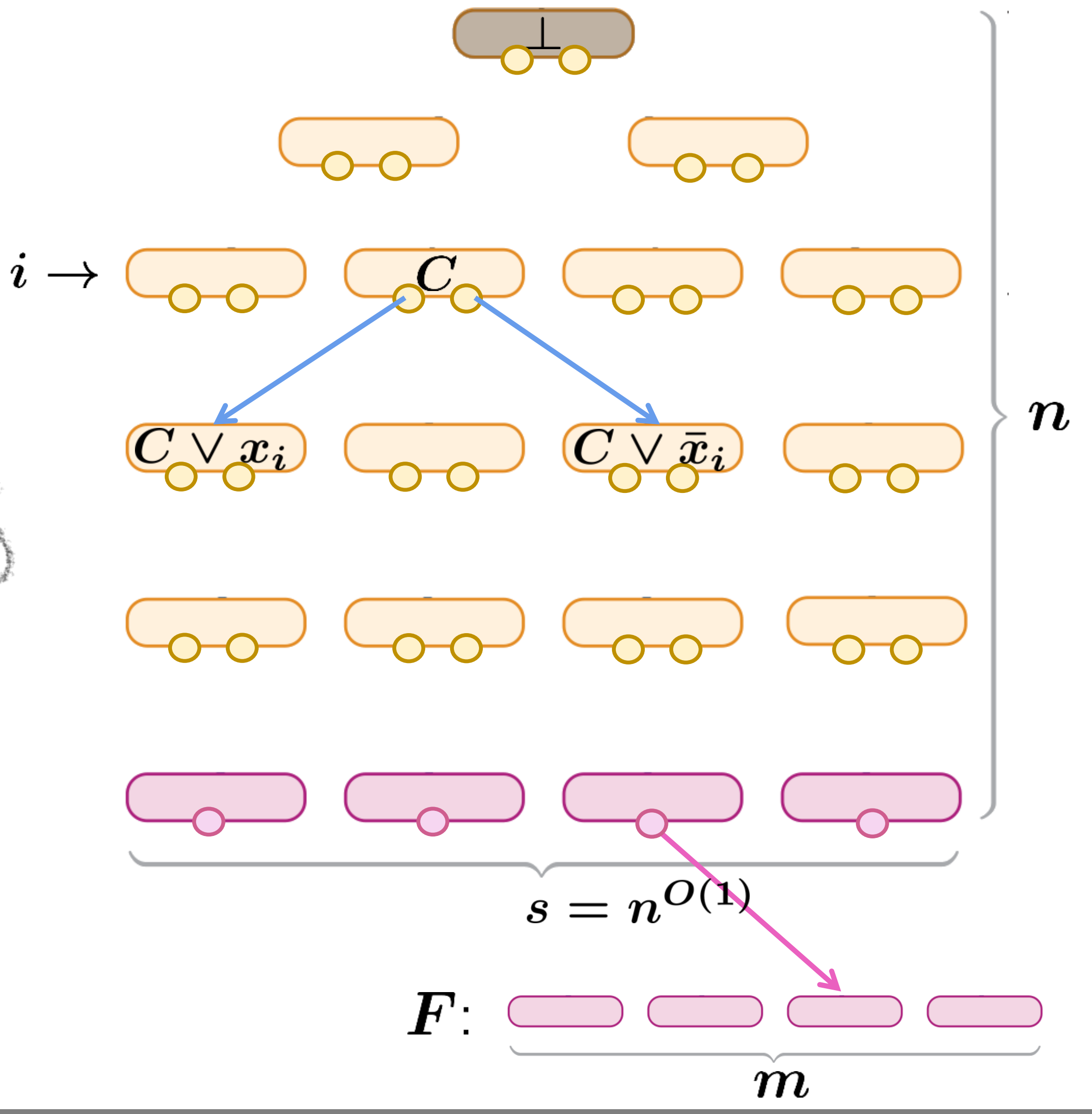
Variables for each block B:

- $2n$ 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 $j \in [m]$

$\boxed{O(n^2 s) \text{ variables}}$

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

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



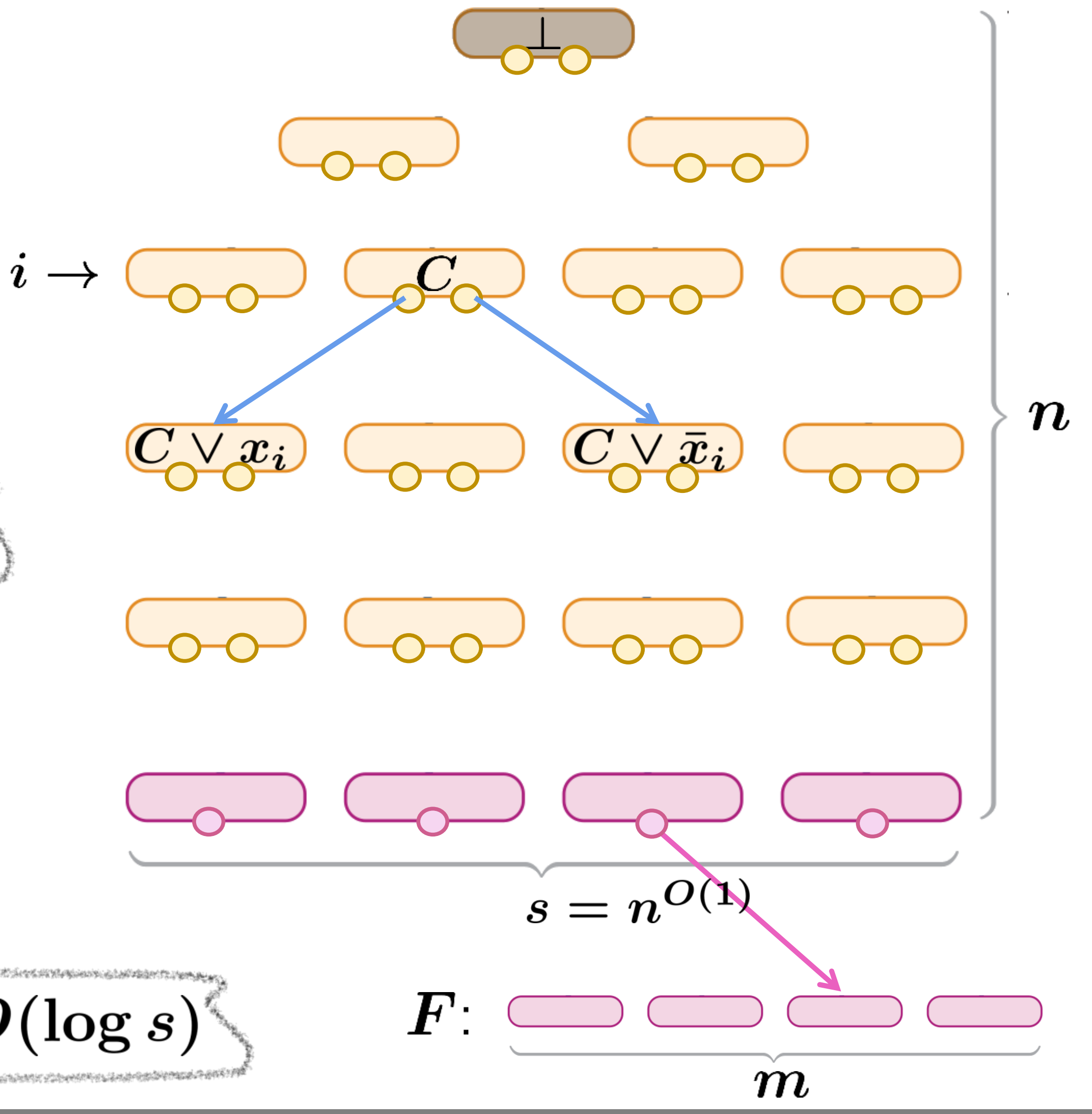
$\mathbf{Ref}(F)$

Encodes "F has short structured resolution refutation"

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 $j \in [m]$

$\boxed{O(n^2 s) \text{ variables}}$



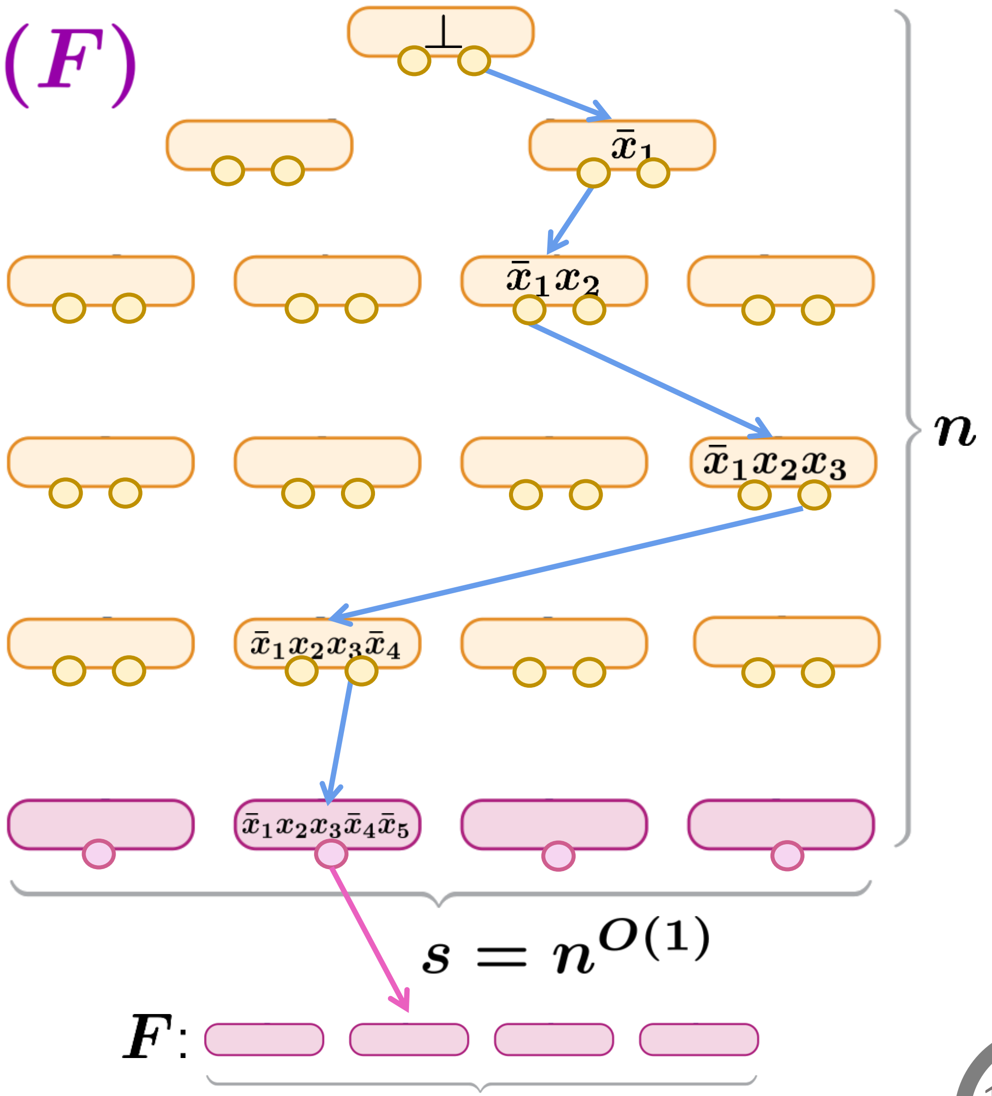
Axioms of $\mathbf{Ref}(F)$:

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

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

(1) $\boldsymbol{F}$ is **SAT** $\Rightarrow \mathbf{Ref}(\boldsymbol{F})$ has size-$\boldsymbol{n}^{\boldsymbol{O}(\mathbf{1})}$ resolution refutation

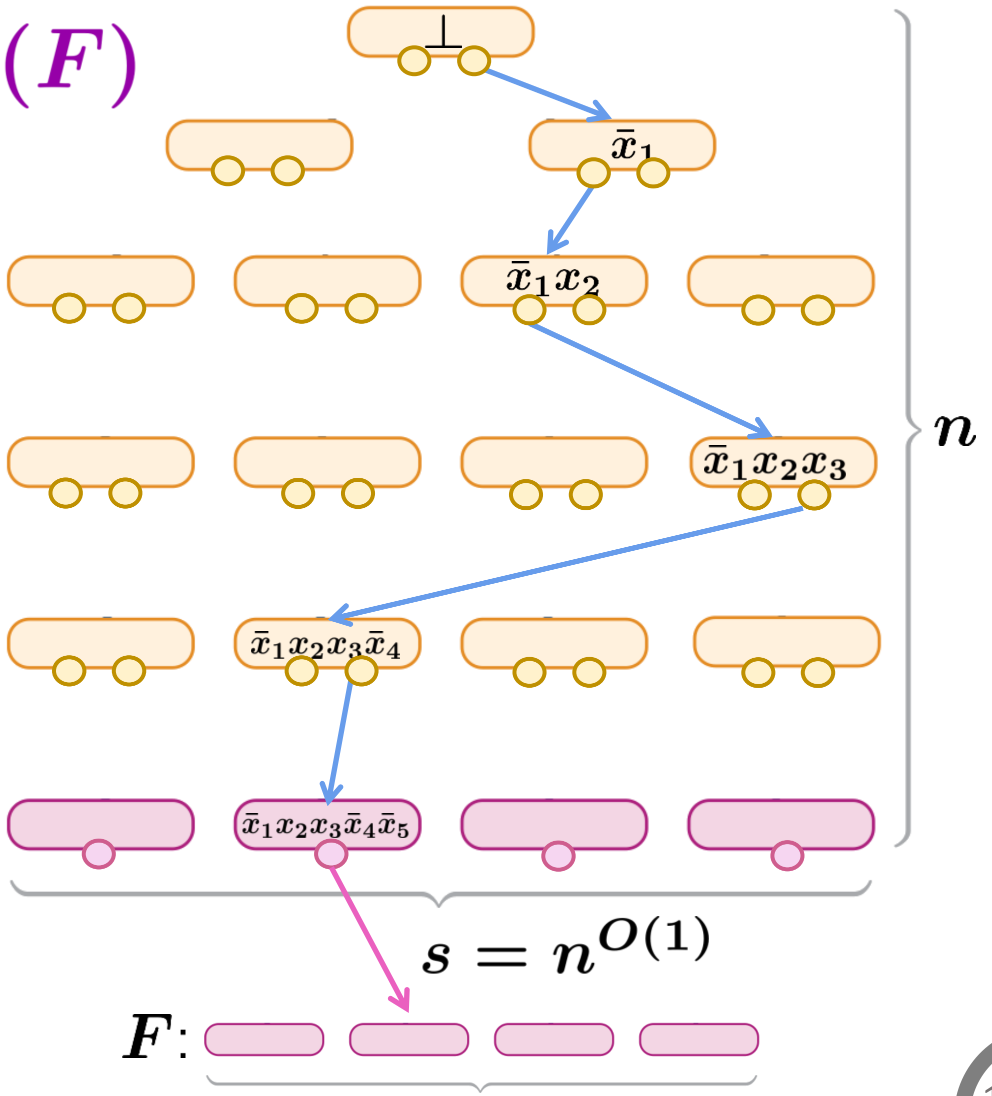
Read-once branching program for $\mathbf{Ref}(\boldsymbol{F})$



(1) $\boldsymbol{F}$ is **SAT** $\Rightarrow$ $\mathbf{Ref}(\boldsymbol{F})$ has size-$\boldsymbol{n}^{\boldsymbol{O}(1)}$ resolution refutation

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

$\boldsymbol{x}^*$ satisfying assignment for $\boldsymbol{F}$

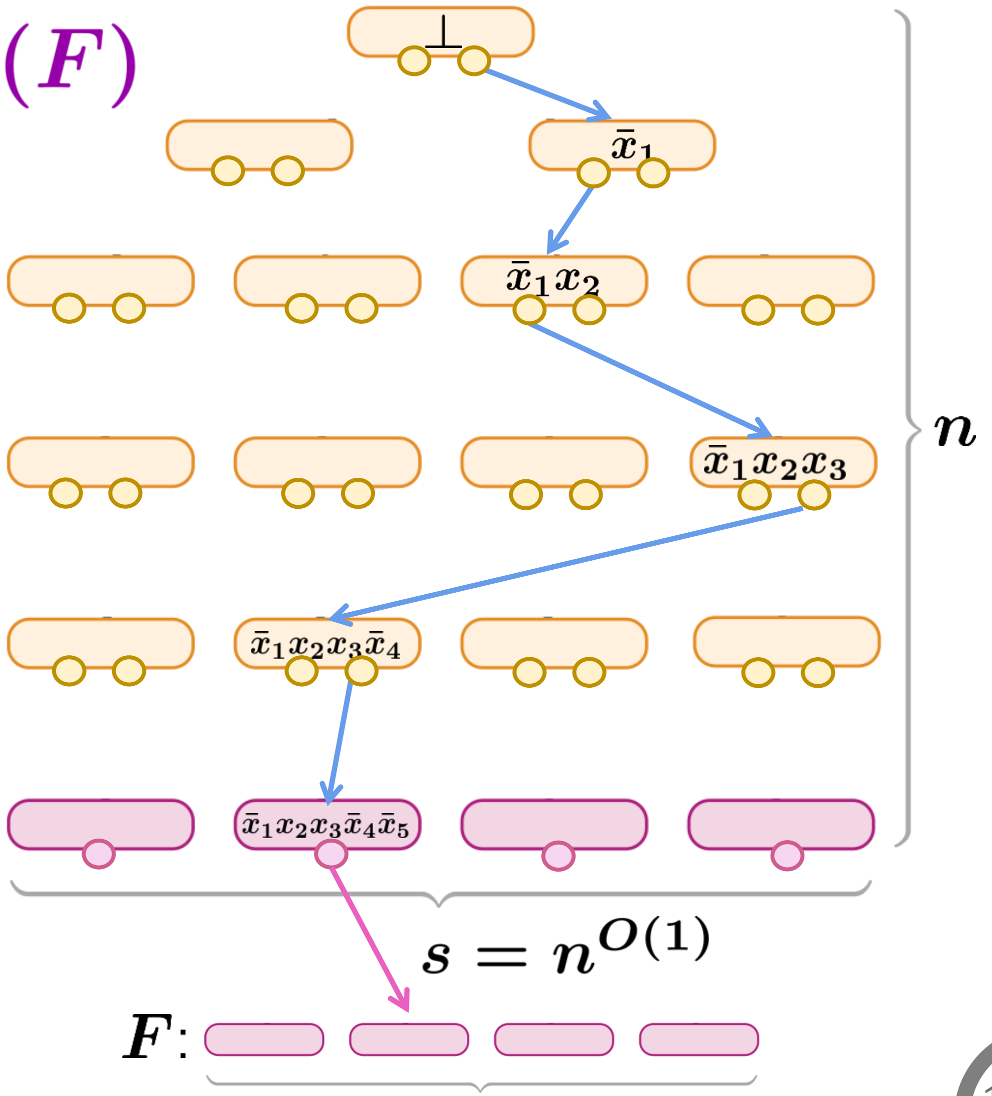


(1) $\boldsymbol{F}$ is **SAT** $\Rightarrow$ $\mathbf{Ref}(\boldsymbol{F})$ has size-$n^{O(1)}$ resolution refutation

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

$\boldsymbol{x}^*$ satisfying assignment for $\boldsymbol{F}$

invariant: $\boldsymbol{x}^*$ falsifies clause in current block $\boldsymbol{B}$



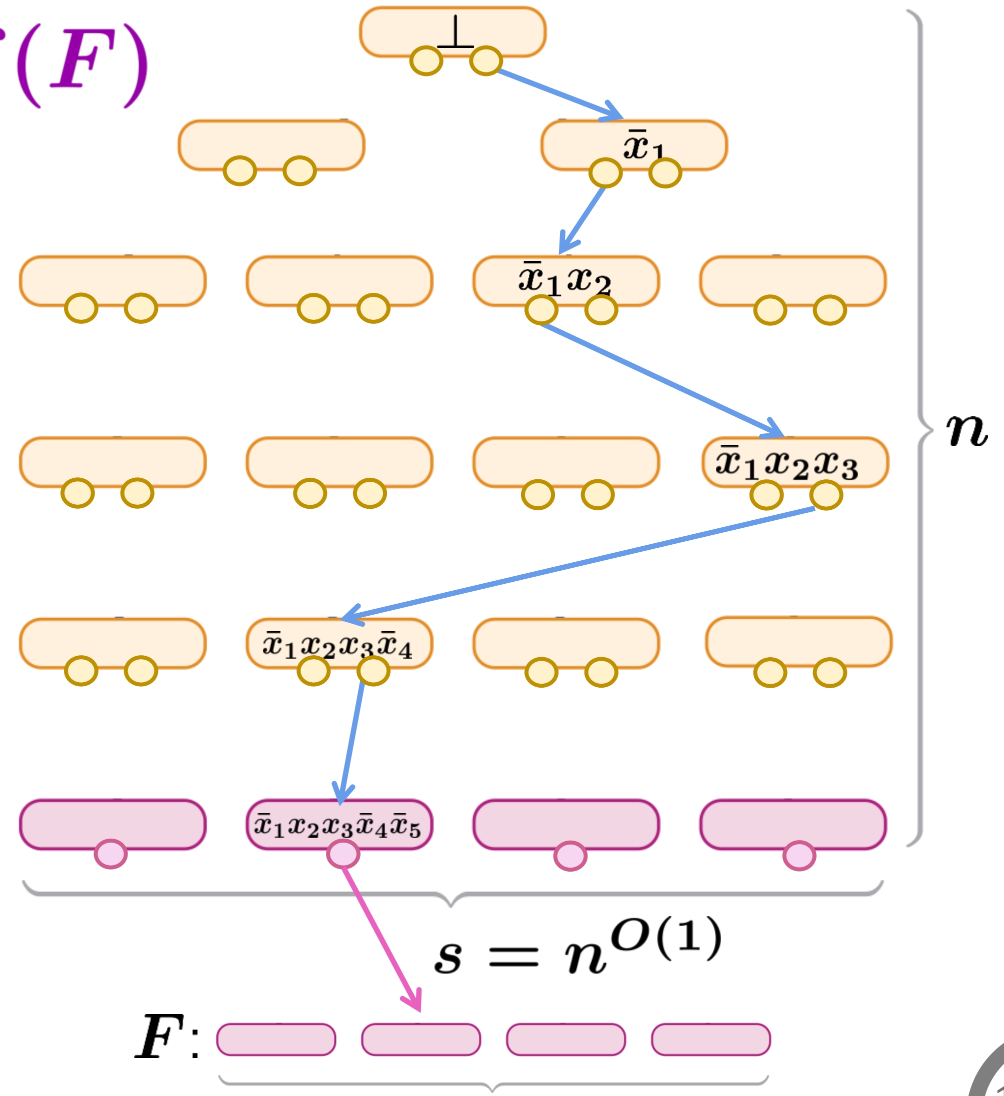
(1) F is **SAT** $\Rightarrow$ $\mathrm{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation

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

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)



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

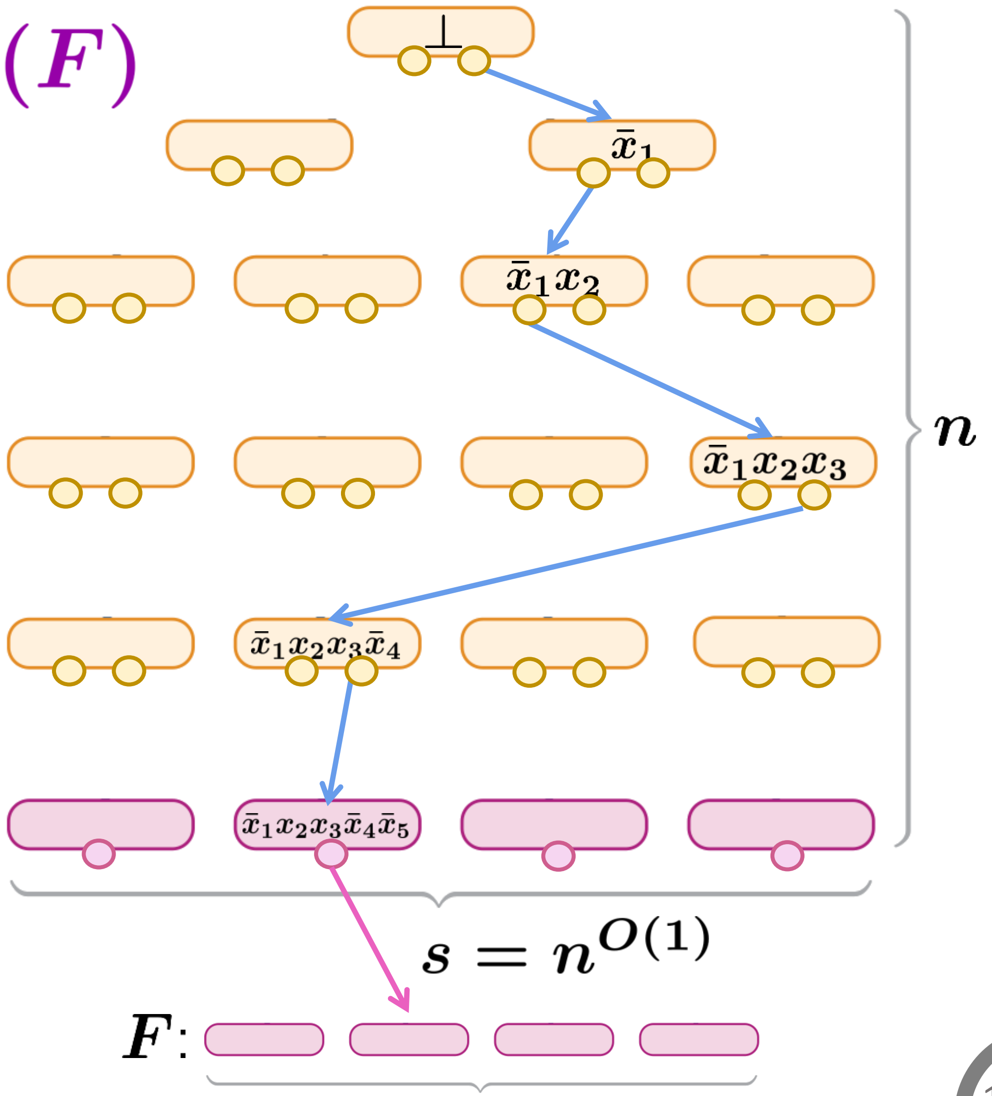
Read-once branching program for $\mathrm{Ref}(F)$

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) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{Ref}(\boldsymbol{F})$ requires size $\mathbf{2}^{\boldsymbol{\Omega}(\boldsymbol{n})}$ resolution refutation

$\boxed{\text{(2) } F \text{ is UNSAT} \Rightarrow \mathrm{Ref}(F) \text{ requires size } 2^{\Omega(n)} \text{ resolution refutation}}$

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

  [dRGNPRS'21]

$$\text{(2) } \boldsymbol{F} \text{ is } \textbf{UNSAT} \Rightarrow \mathbf{Ref}(\boldsymbol{F}) \text{ requires size } \mathbf{2}^{\boldsymbol{\Omega(n)}} \text{ resolution refutation}$$

- $$\boldsymbol{w}(\mathbf{Ref}(\boldsymbol{F}) \vdash \perp) \geq \widetilde{\boldsymbol{\Omega}}(\boldsymbol{w}(\mathbf{PHP}^{\boldsymbol{2s}}_{\boldsymbol{s}} \vdash \perp)/\boldsymbol{n}) \geq \widetilde{\boldsymbol{\Omega}}(\boldsymbol{s}/\boldsymbol{n})$$

  [dRGNPRS'21]

(2) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{Ref}(\boldsymbol{F})$ requires size $\mathbf{2}^{\boldsymbol{\Omega(n)}}$ resolution refutation

- $\boldsymbol{w(\mathrm{Ref}(F) \vdash \bot) \geq \widetilde{\Omega}(w(\mathrm{PHP}_s^{2s} \vdash \bot)/n) \geq \widetilde{\Omega}(s/n)}$
  [dRGNPRS'21]
- [BW'01] size of resolution refutation of $\varphi \geq \exp\left(\boldsymbol{\Omega}\left(\frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\#\mathrm{var}}\right)\right)$

(2) F is **UNSAT** $\Rightarrow$ $\mathbf{Ref}(F)$ requires size $2^{\Omega(n)}$ resolution refutation

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

(2) F is **UNSAT** $\Rightarrow$ $\mathrm{Ref}(F)$ requires size $2^{\Omega(n)}$ resolution refutation

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

  [dRGNPRS'21]

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

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

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

(2) F is **UNSAT** $\Rightarrow$ $\mathrm{Ref}(F)$ requires size $2^{\Omega(n)}$ resolution refutation

- $w(\mathrm{Ref}(F) \vdash \bot) \geq \widetilde{\Omega}(w(\mathrm{PHP}_s^{2s} \vdash \bot)/n) \geq \widetilde{\Omega}(s/n)$
  [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: $\mathrm{Ref}(F)$ has $O(n^2 s)$ variables and width $O(\log s)$

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

choose $s \geq n^5$

$\boxed{\text{(2) } F \text{ is } \textbf{UNSAT} \Rightarrow \mathrm{Ref}(F) \text{ requires size } 2^{\Omega(n)} \text{ resolution refutation}}$

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

  [dRGNPRS'21]

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

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

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

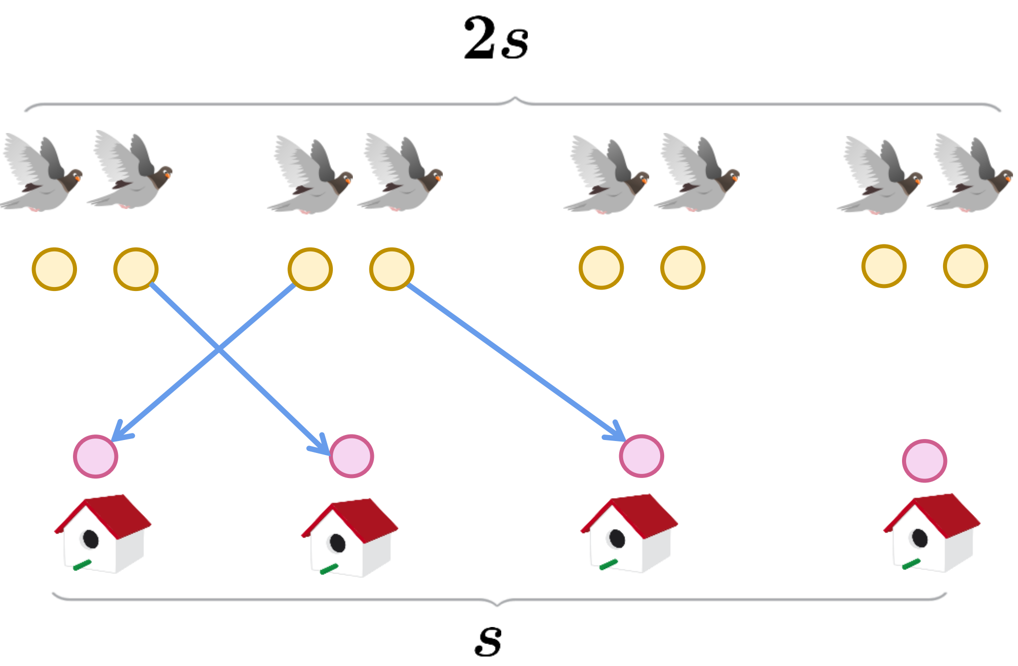
$\boxed{\text{choose } s \geq n^5}$

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

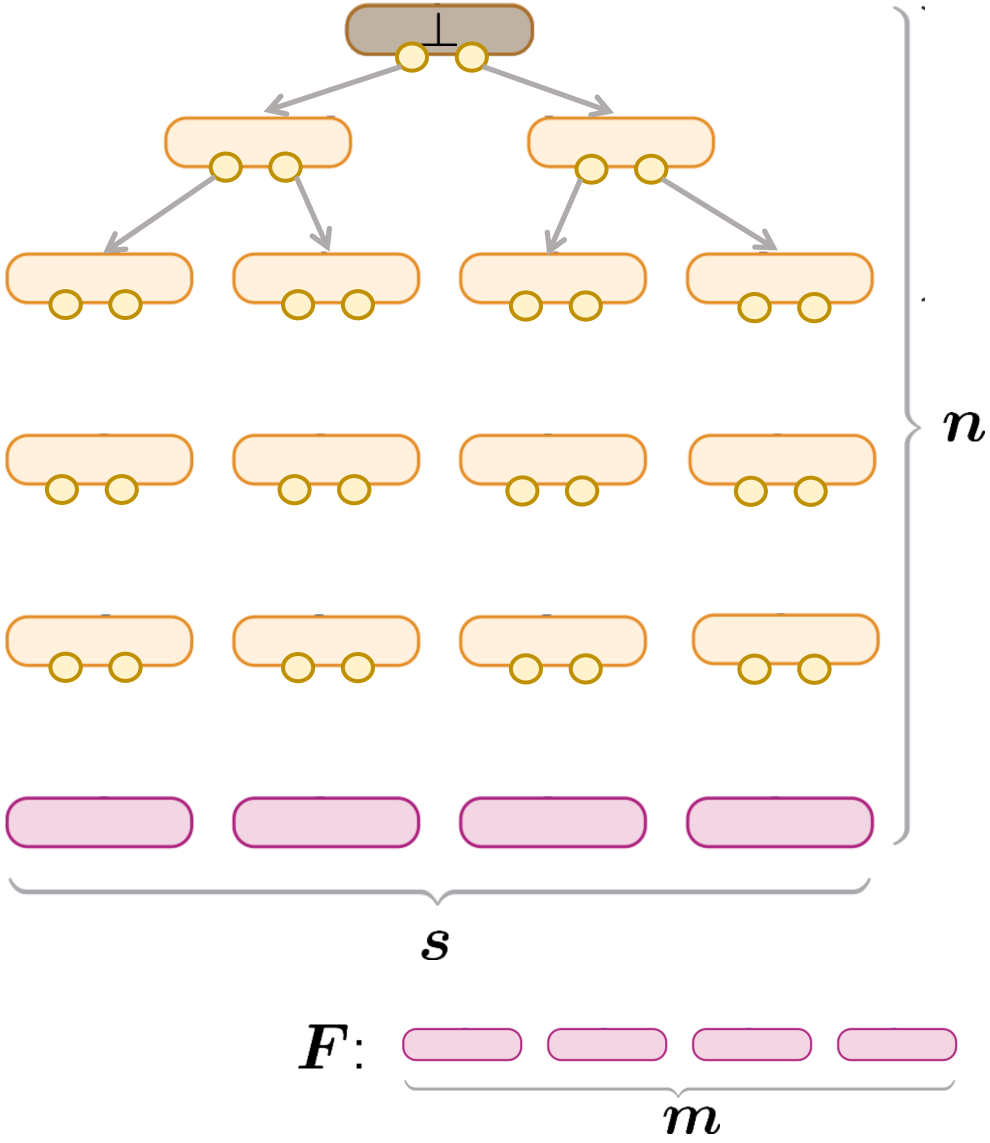
- degree lower bound for PHP_s^{2s}
- similar size-degree relation

$$\mathbf{PHP}_{\boldsymbol{s}}^{\boldsymbol{2s}} \leq_{\widetilde{\boldsymbol{O}}(\boldsymbol{n})} \mathbf{Ref}(\boldsymbol{F})$$

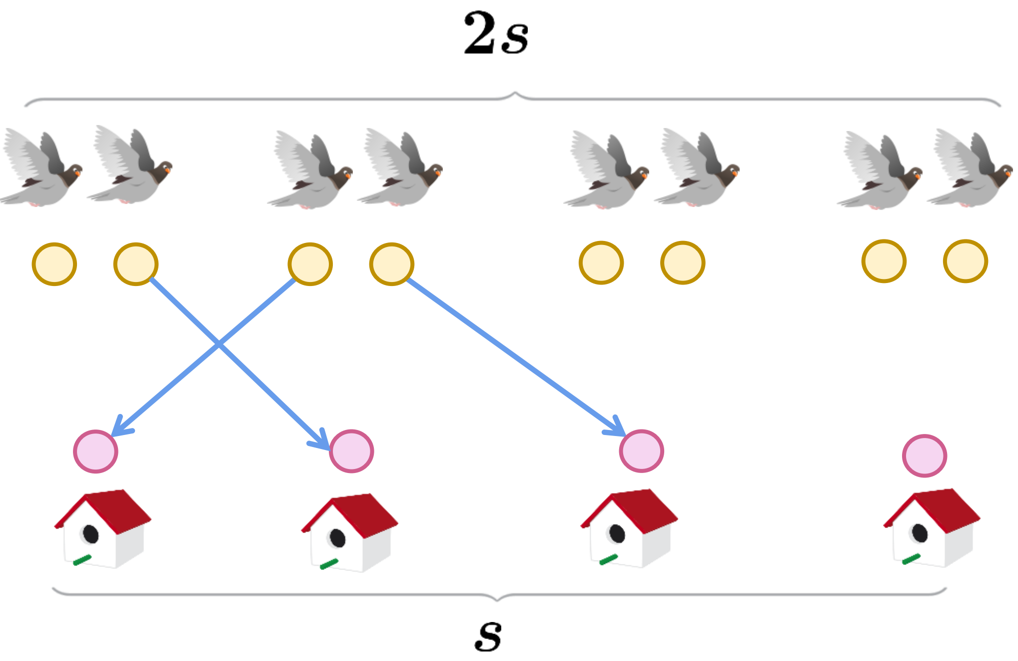
$\mathrm{PHP}_s^{2s} \leq_{\widetilde{O}(n)} \mathrm{Ref}(F)$



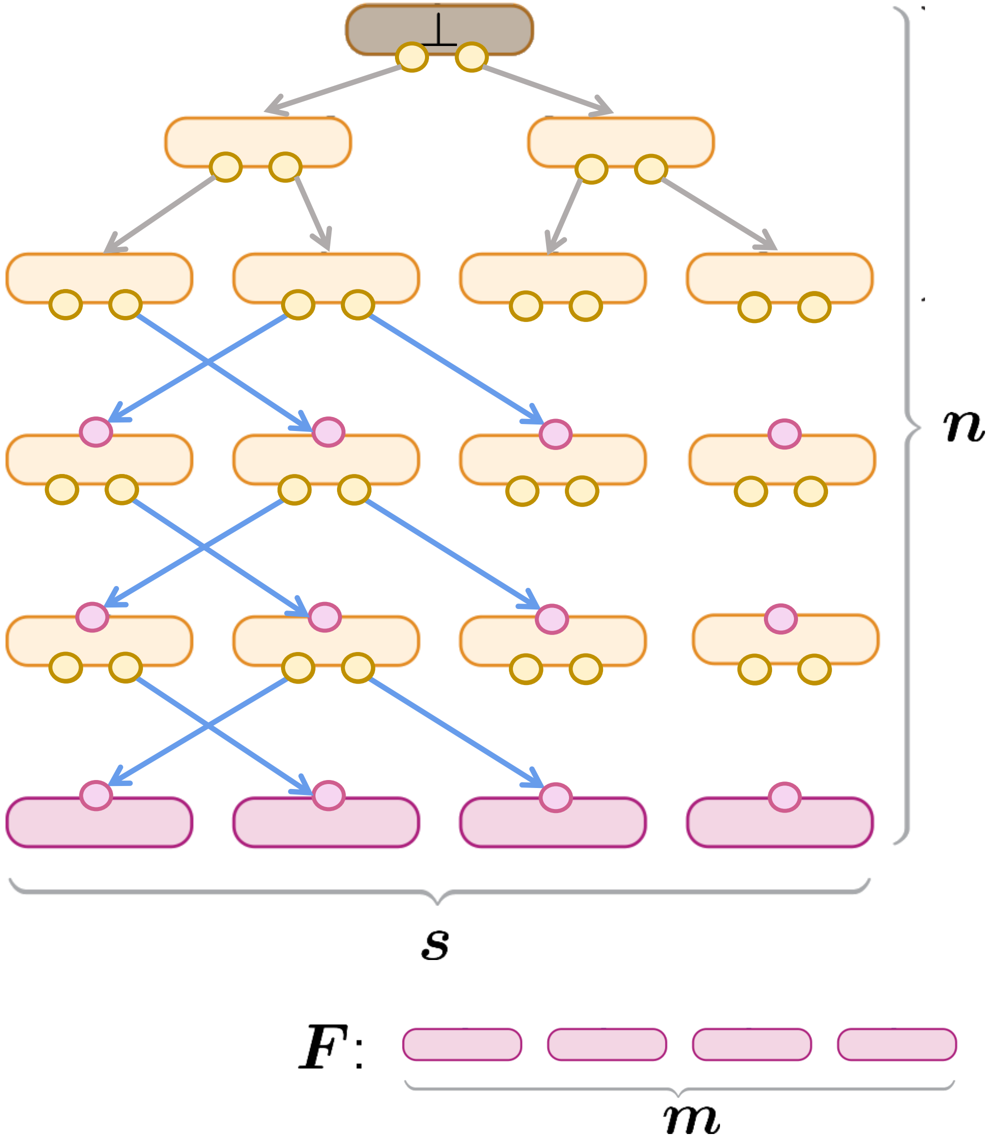
$\leq_{\widetilde{O}(n)}$



$\mathrm{PHP}_s^{2s} \leq_{\widetilde{O}(n)} \mathrm{Ref}(F)$



$\leq_{\widetilde{O}(n)}$

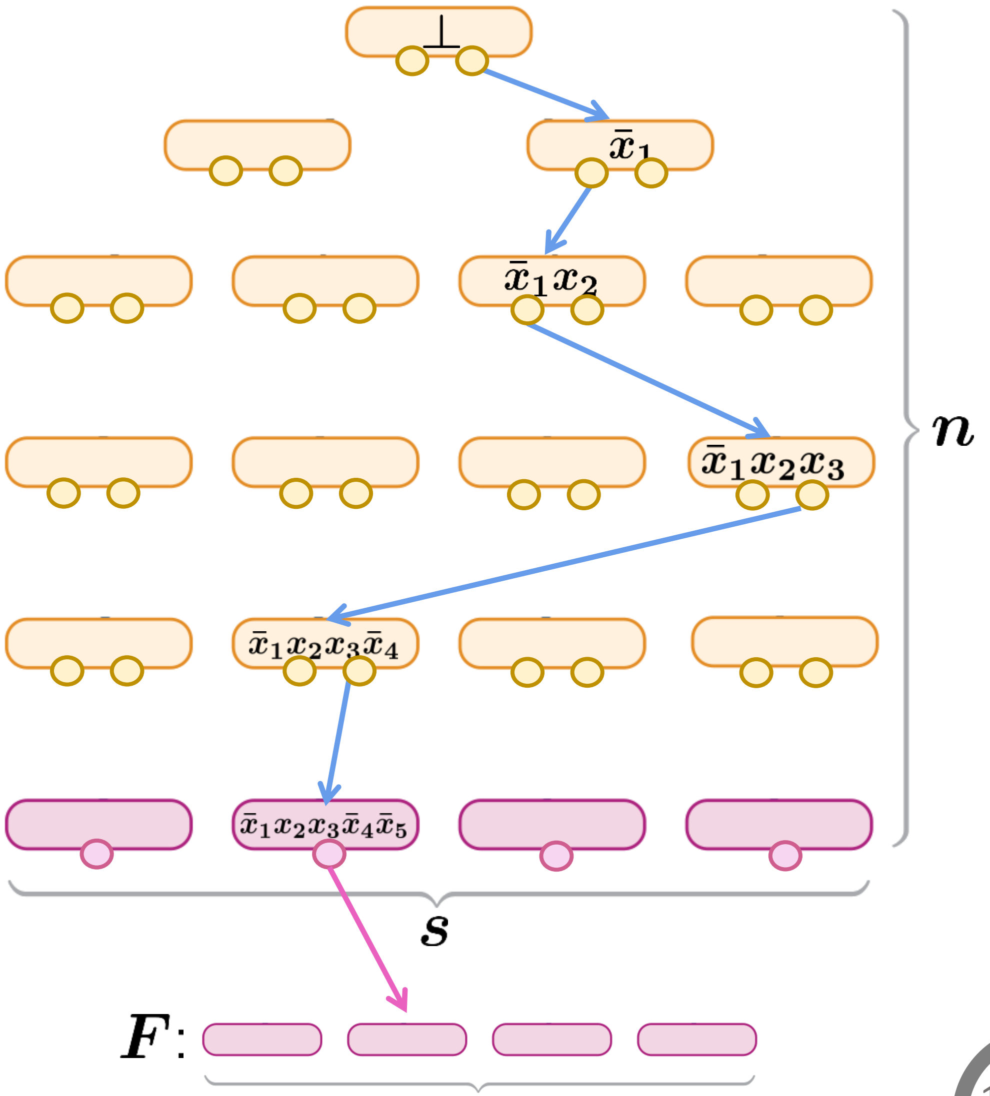


Tree-Like Resolution

(1) $\boldsymbol{F}$ is **SAT** $\Rightarrow$ $\mathbf{Ref}(\boldsymbol{F})$ has size-$\boldsymbol{n}^{\boldsymbol{O(1)}}$ resolution refutation

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

size: $\mathbf{poly}(\boldsymbol{n})$

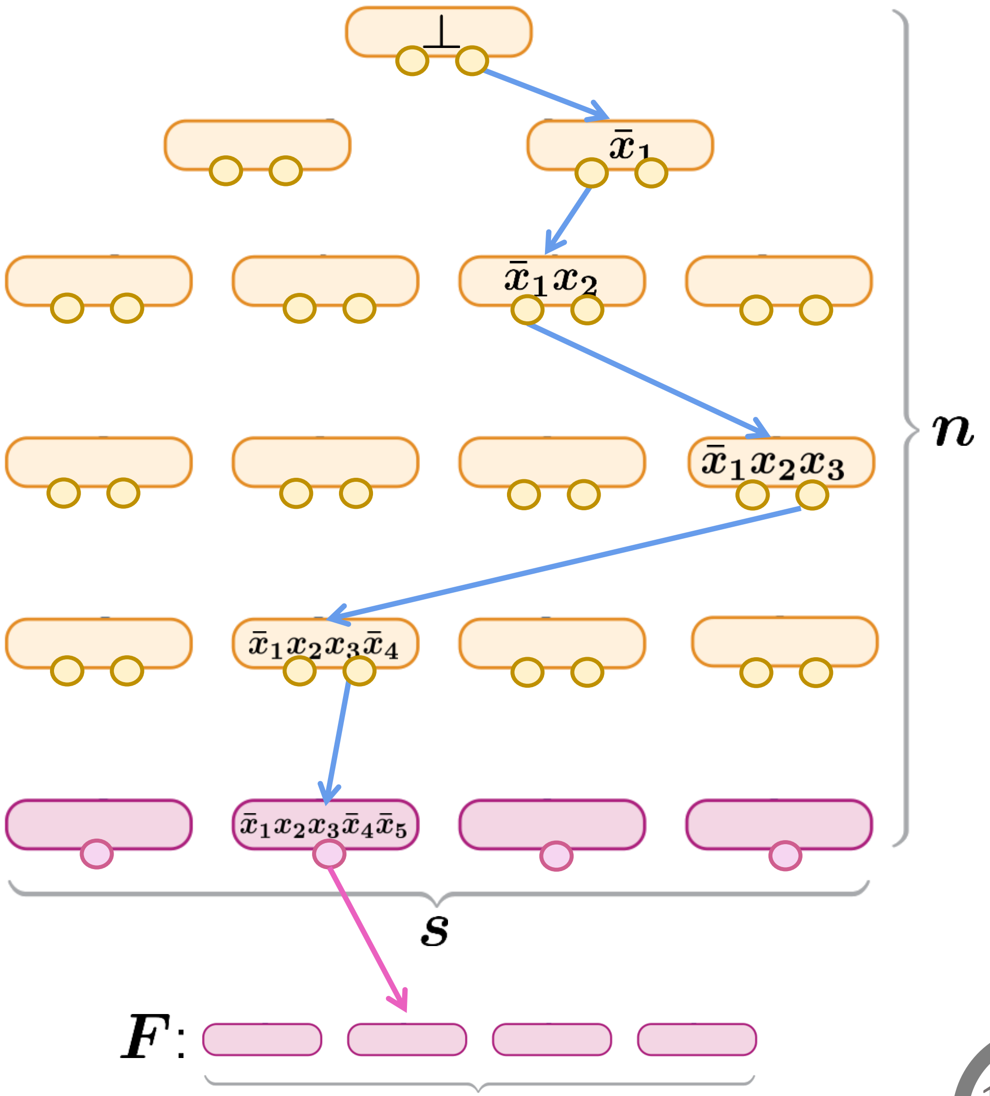


(1) F is **SAT** $\Rightarrow$ $\mathrm{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation

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

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

Decision tree for $\mathrm{Ref}(F)$



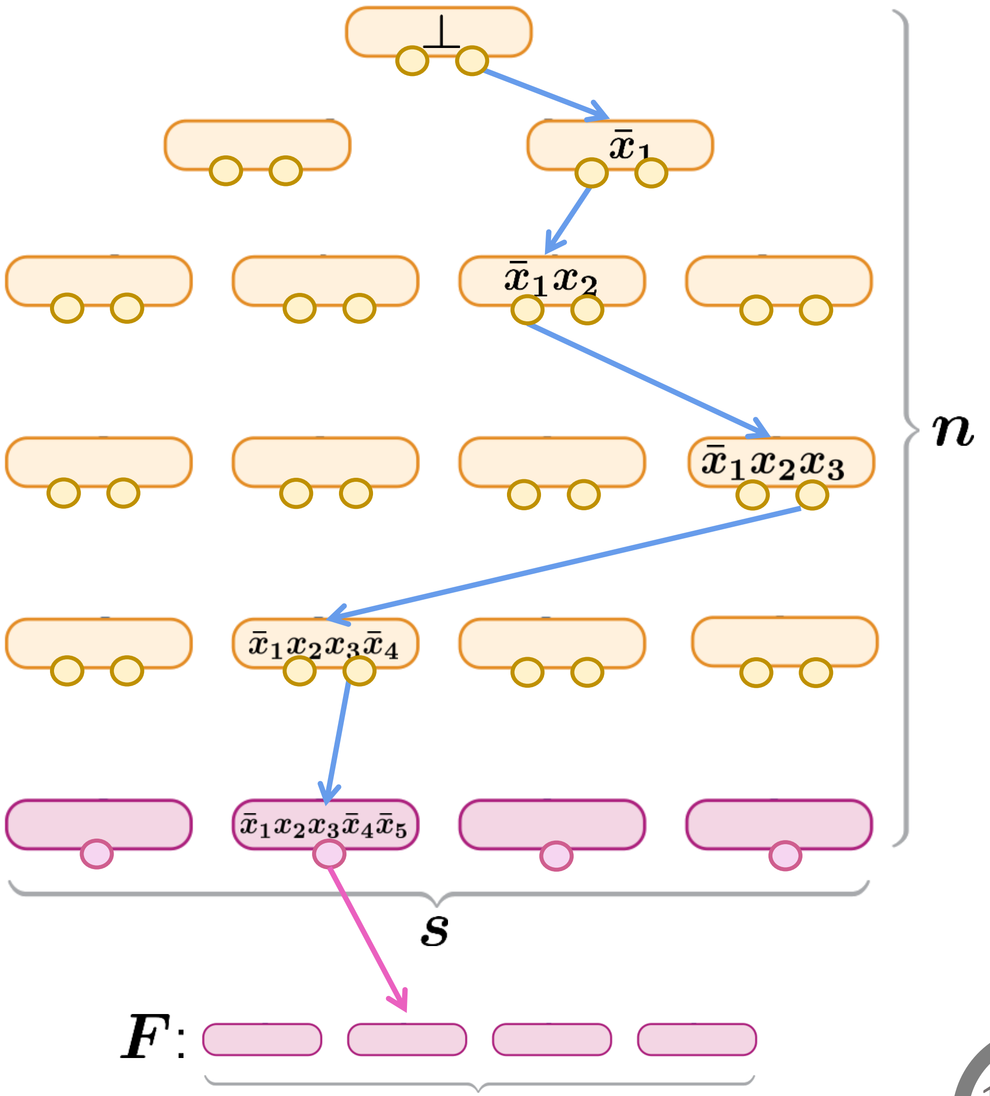
(1) F is **SAT** $\Rightarrow$ $\mathrm{Ref}(F)$ has size-$n^{O(1)}$ resolution refutation

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

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

Decision tree for $\mathrm{Ref}(F)$

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



(1) $\boldsymbol{F}$ is **SAT** $\Rightarrow$ $\mathbf{Ref}(\boldsymbol{F})$ has size-$\boldsymbol{n}^{O(1)}$ resolution refutation

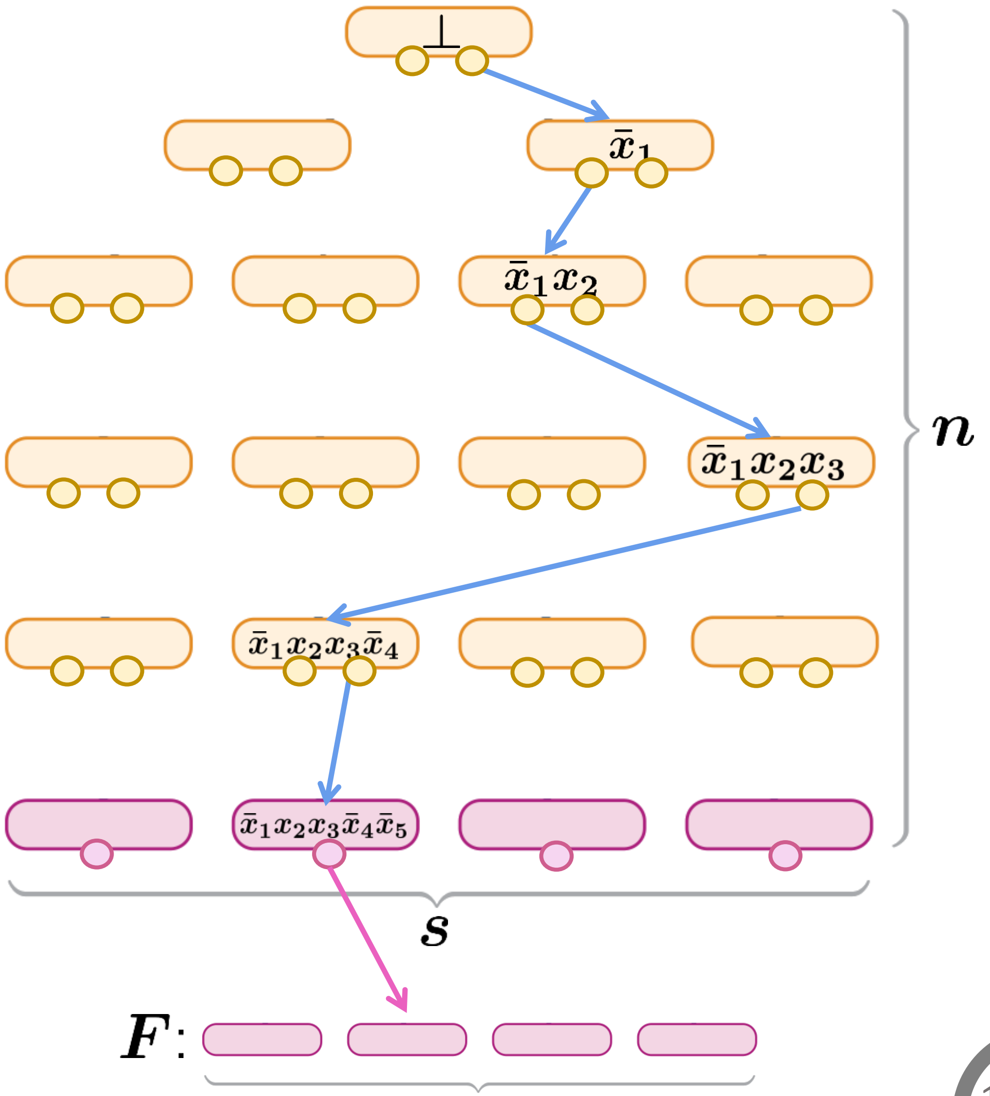
Read-once branching program for $\mathbf{Ref}(\boldsymbol{F})$

size: $\mathbf{poly}(\boldsymbol{n})$

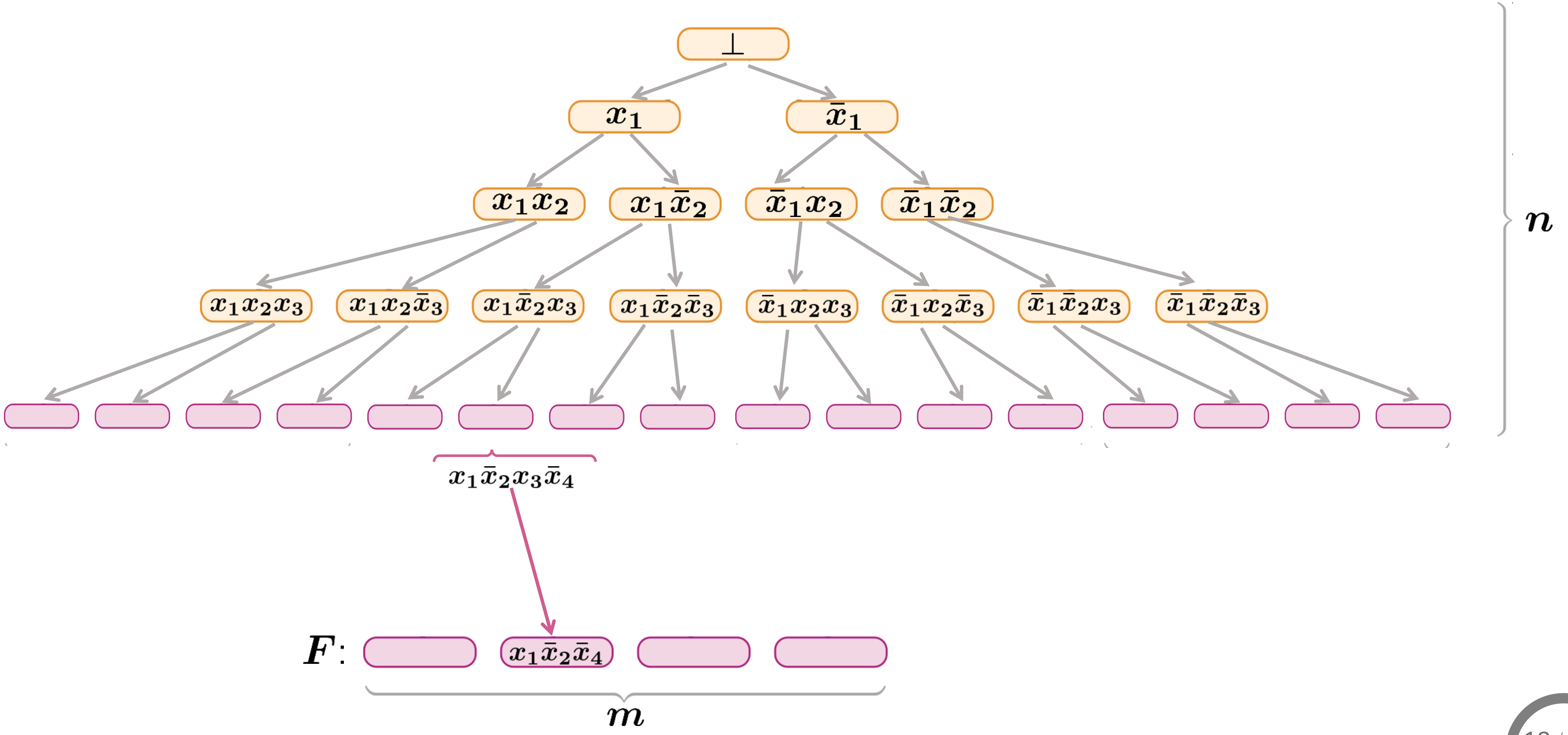
Decision tree for $\mathbf{Ref}(\boldsymbol{F})$

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

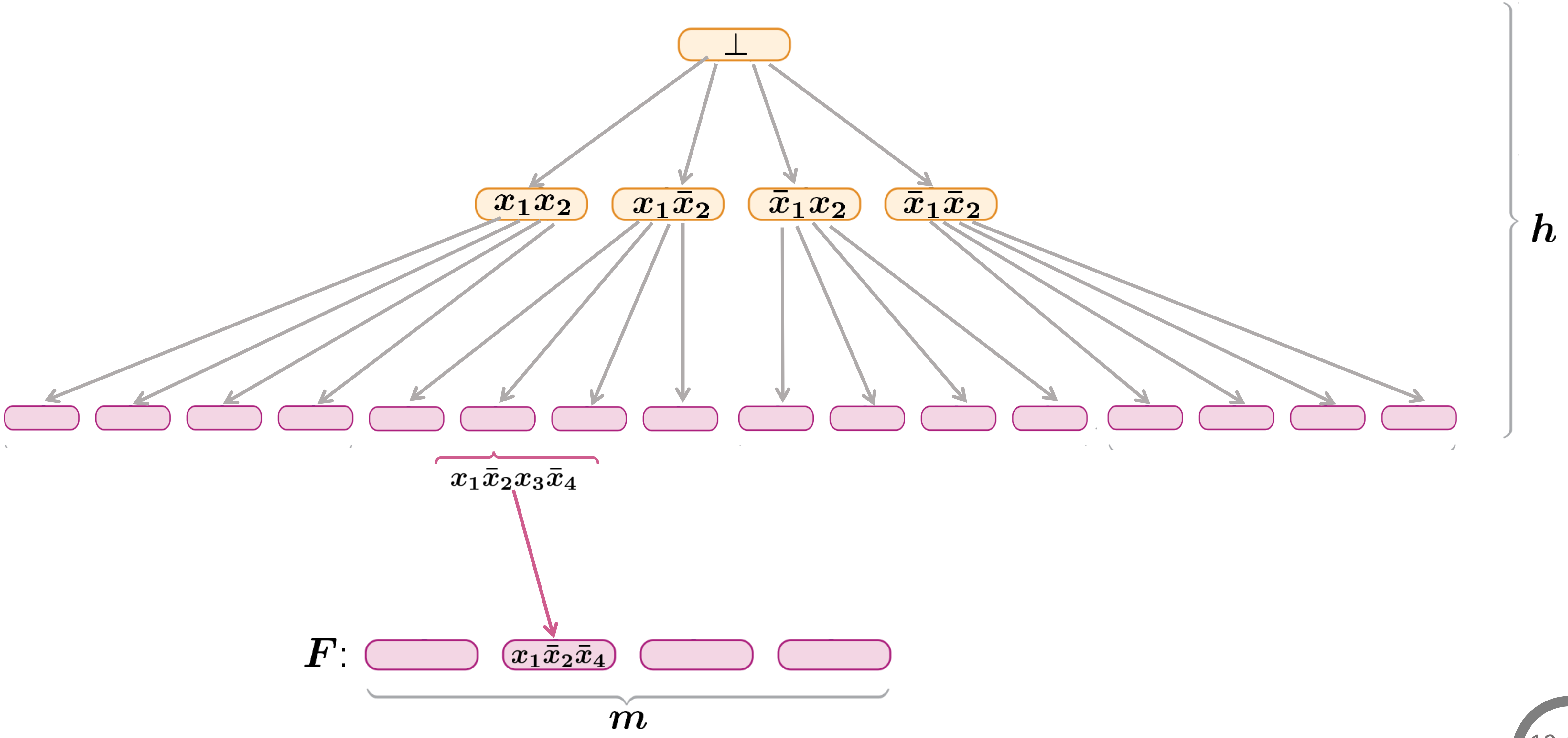
(don't expect upper bound to hold: would imply NP $\subseteq$ QP)



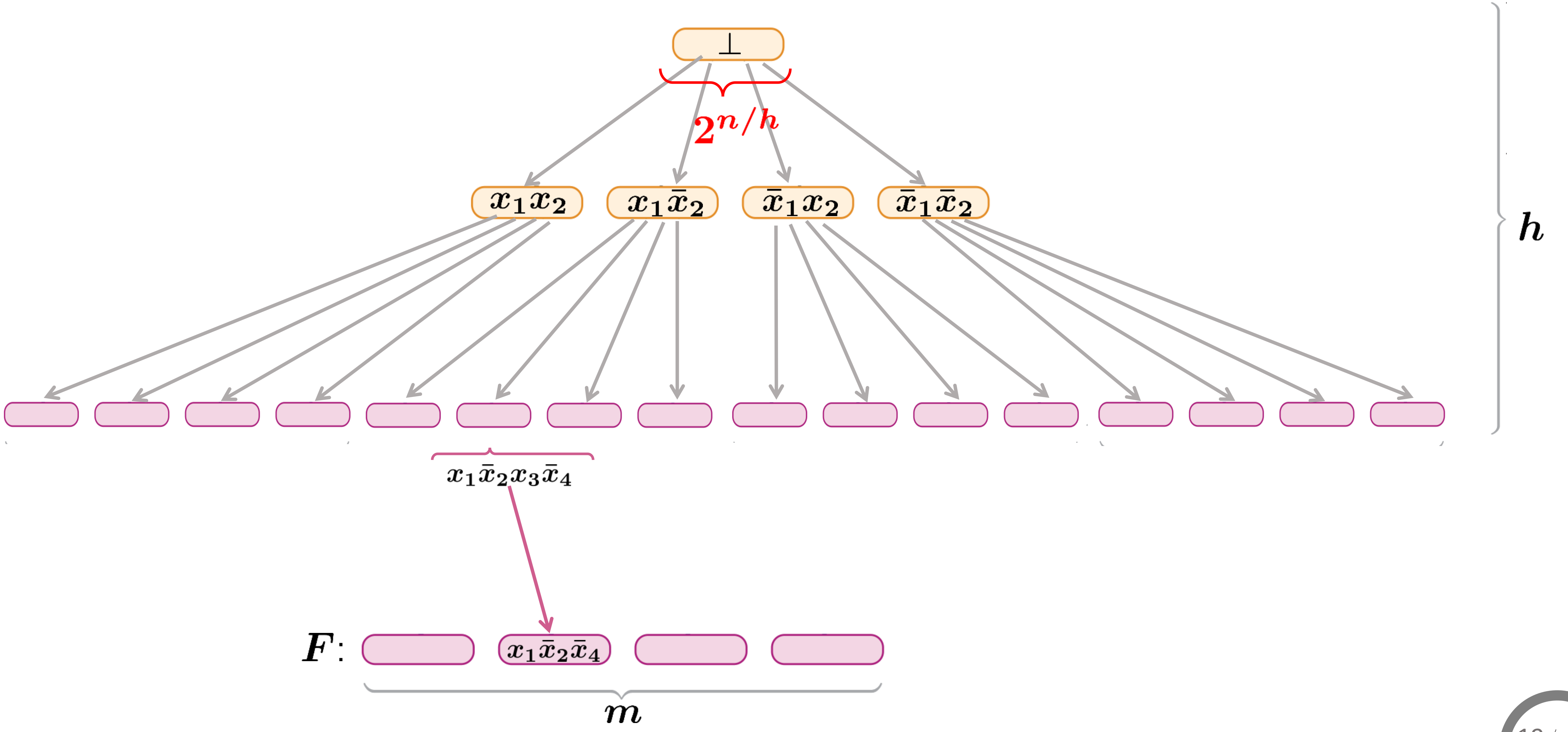
“Universal refutation” (complete Binary tree: depth n)



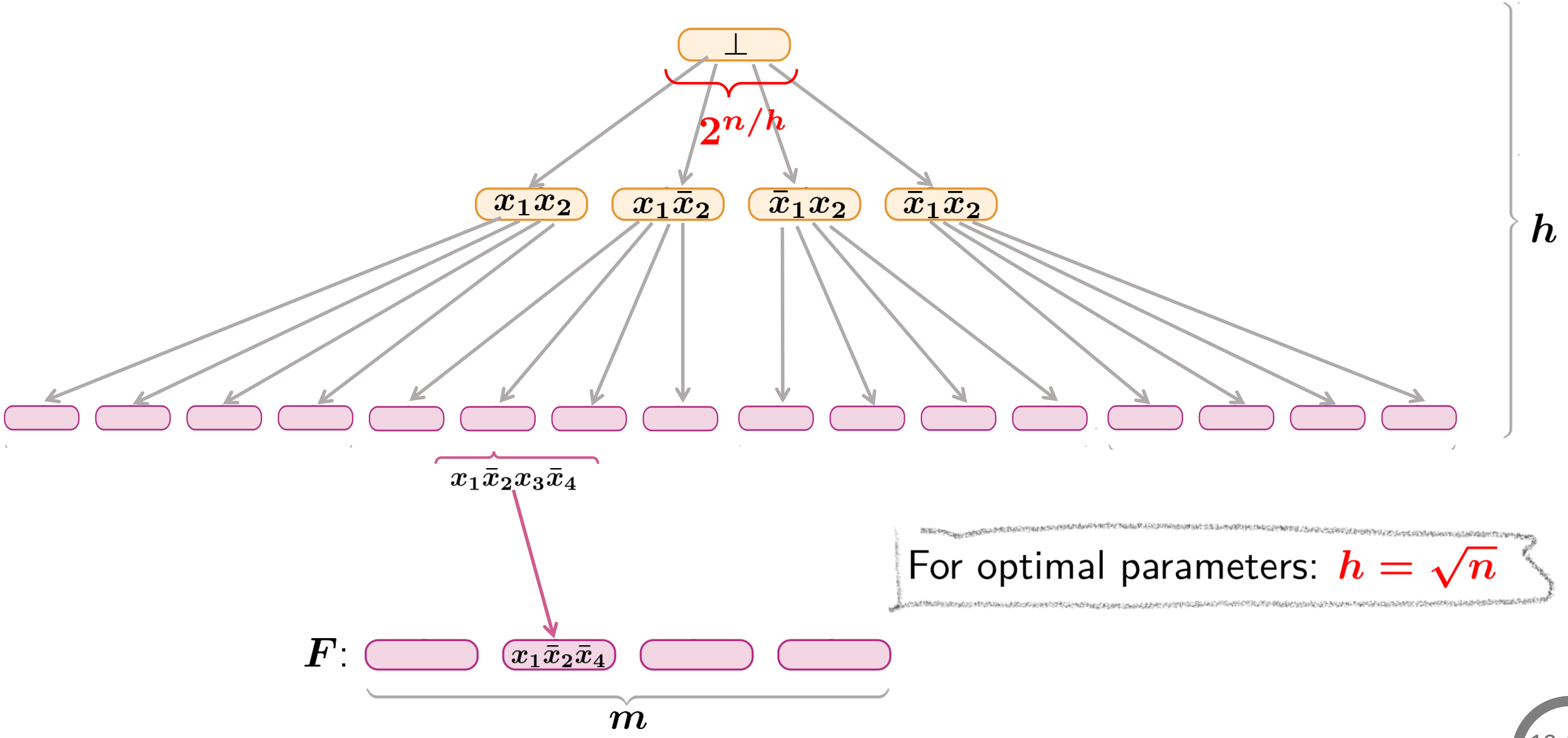
“Universal refutation” (complete tree: depth $h \ll n$)



“Universal refutation” (complete tree: depth $h \ll n$)



“Universal refutation” (complete tree: depth $h \ll n$)



For optimal parameters: $h = \sqrt{n}$

ShallowRef(F)

Variables for each block B:

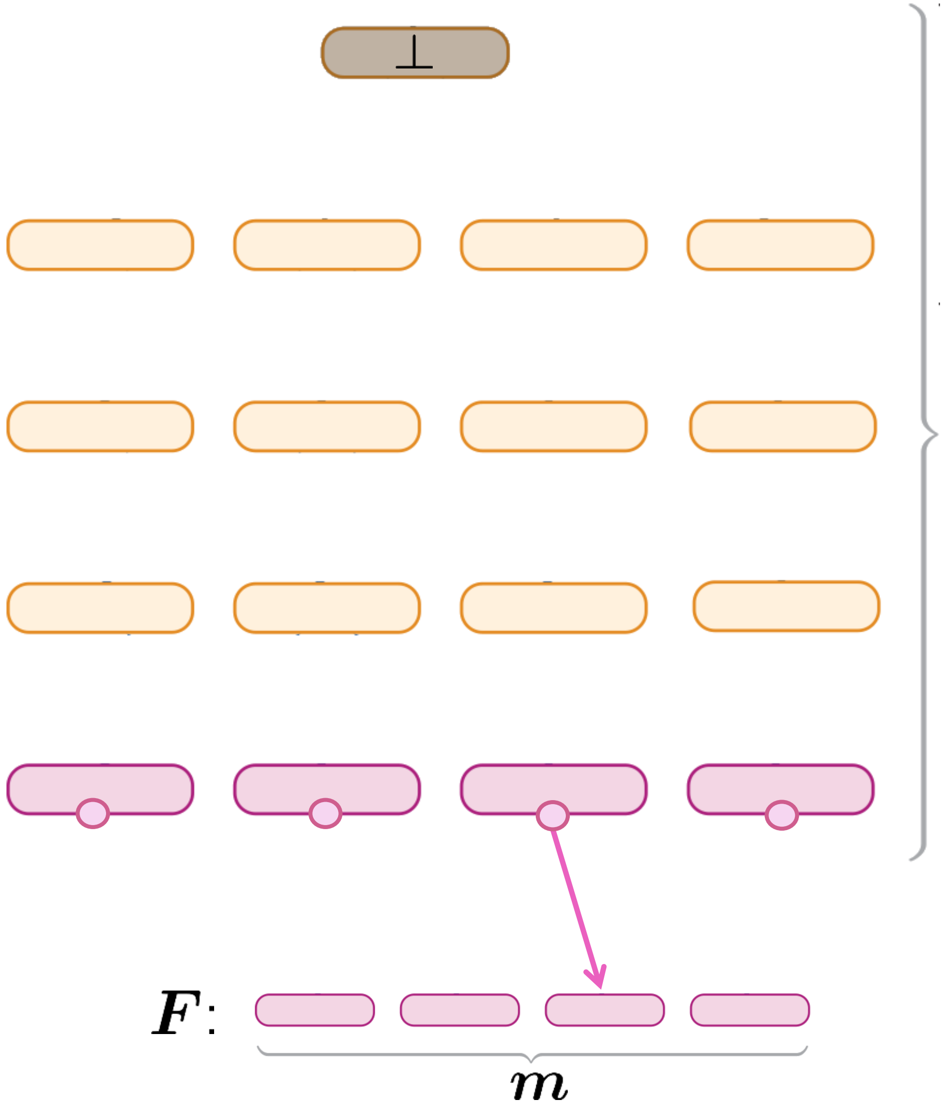
- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children "derived from B_i and B_j"
- $\log m$ variables: axiom-index $j \in [m]$

$O(n^2 s)$ variables

Axioms:

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

clauses of width $O(\log s)$



$\mathbf{ShallowRef}(F)$

Variables for each block B:

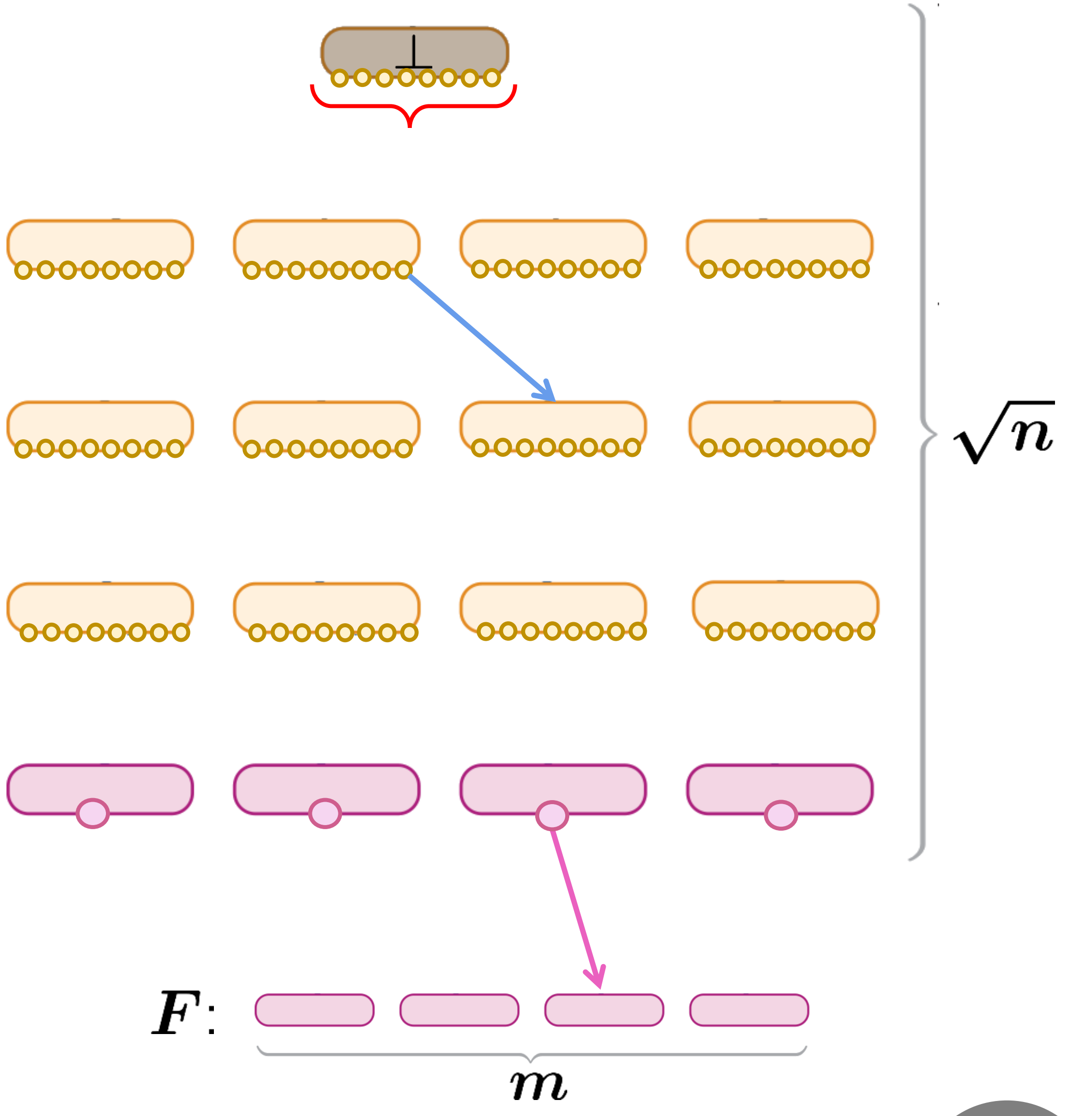
- $2n$ variables (1 per literal): indicates clause
- $2(\log s)$ variables: 2 pointers to children "derived from B_i and B_j"
- $\log m$ variables: axiom-index $j \in [m]$

$O(n^2 s)$ variables

Axioms:

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

clauses of width $O(\log s)$



ShallowRef(F)

Variables for each block B:

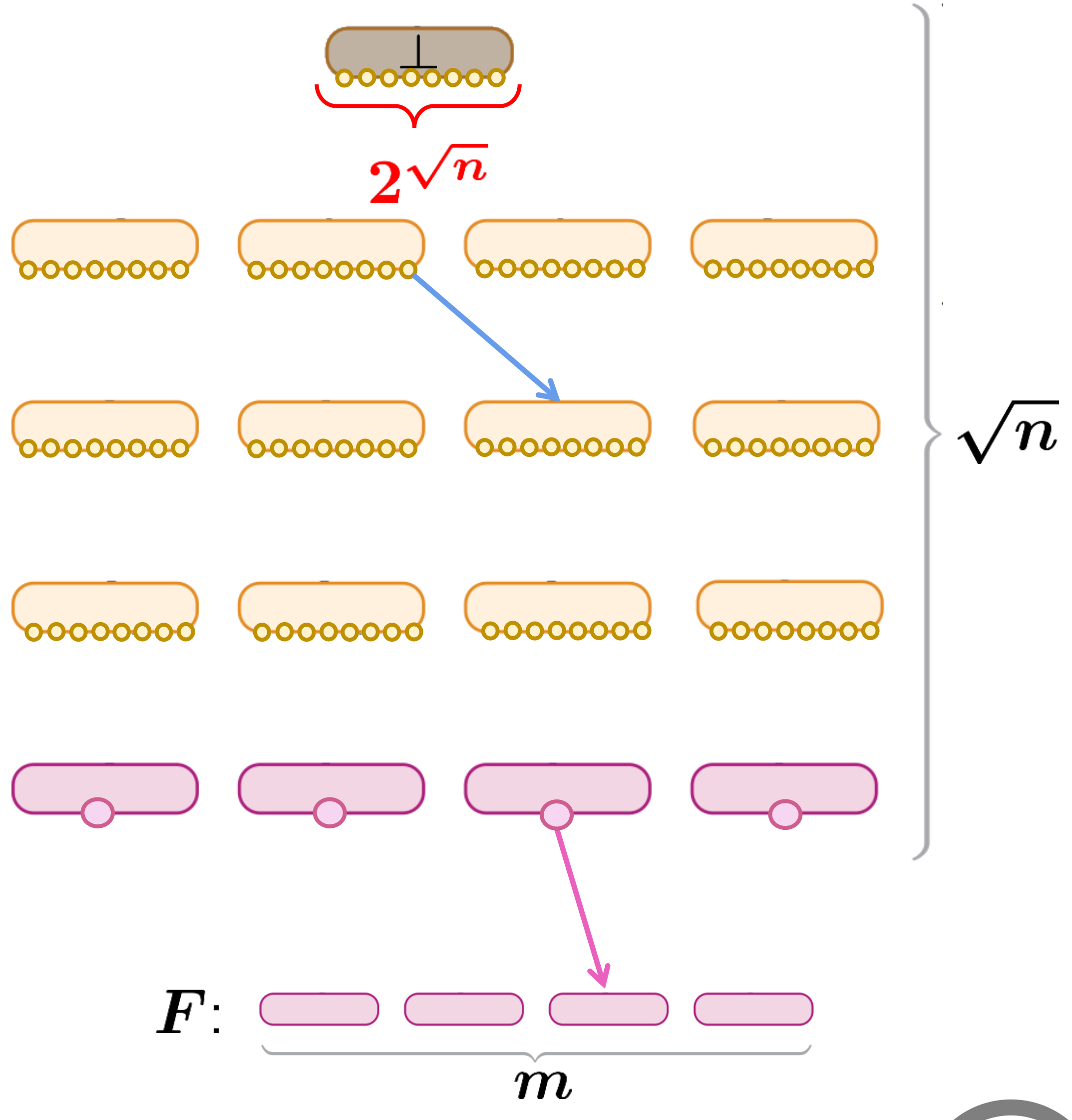
- $2n$ variables (1 per literal): indicates clause
- $O(2^{\sqrt{n}})$ variables: $2^{\sqrt{n}}$ pointers to children
  "derived from all children"
- $\log m$ variables: axiom-index $j \in [m]$

$\boxed{2^{O(\sqrt{n})} \text{ variables}}$

Axioms:

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

$\boxed{\text{clauses of width } O(\log s)}$



$\mathbf{ShallowRef}(F)$

Variables for each block B:

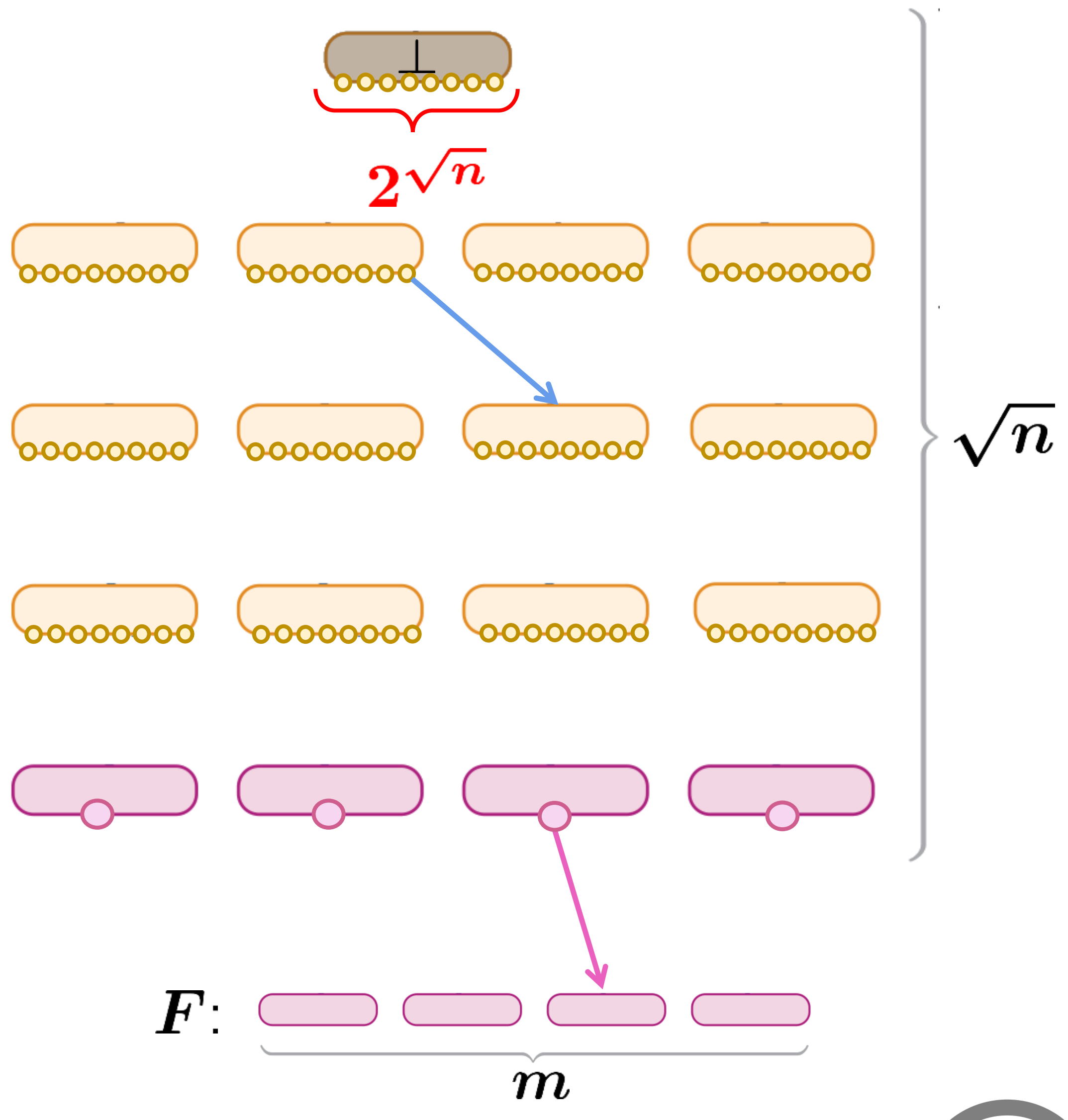
- $2n$ variables (1 per literal): indicates clause
- $O(2^{\sqrt{n}})$ variables: $2^{\sqrt{n}}$ pointers to children
  "derived from all children"
- $\log m$ variables: axiom-index $j \in [m]$

$2^{O(\sqrt{n})}$ variables

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

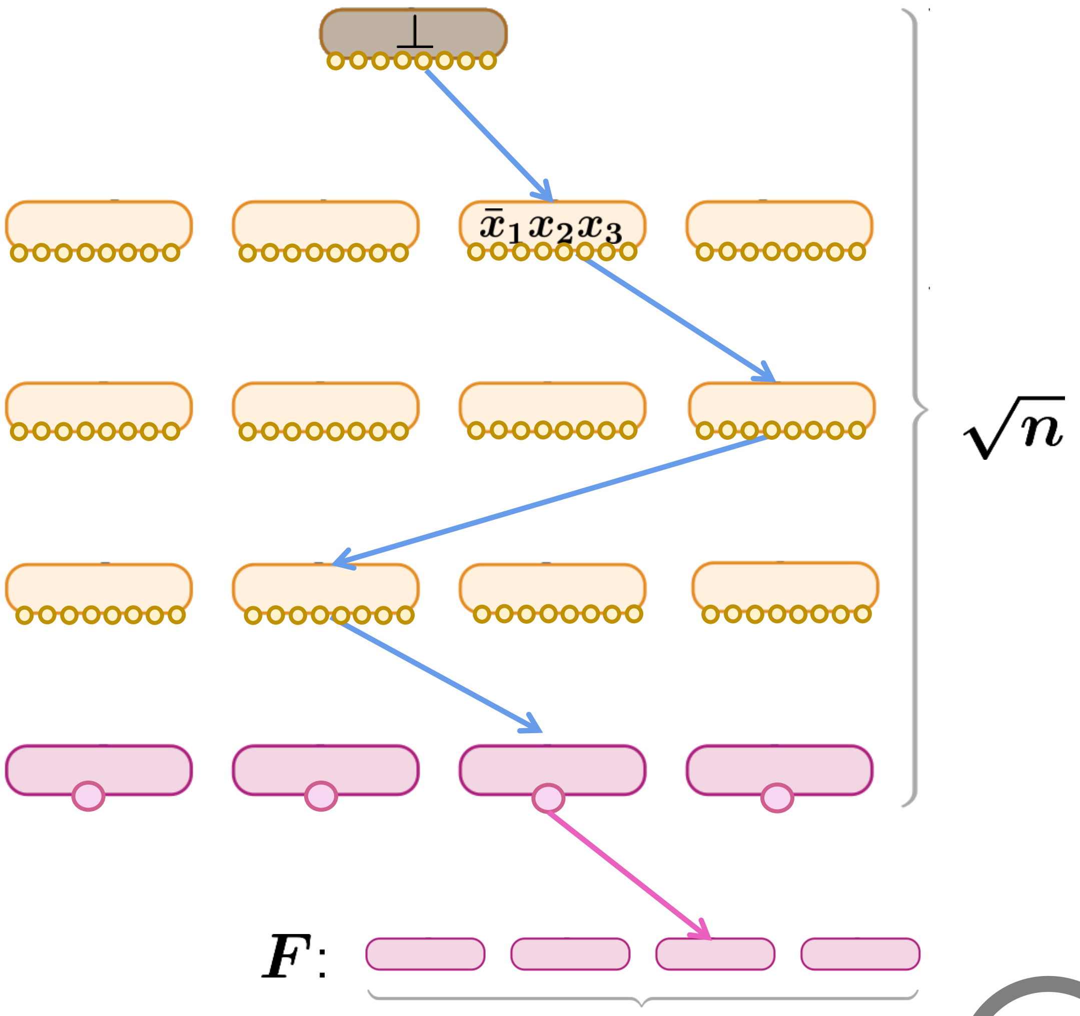
- Root: $\perp$ clause
- Derived: valid resolution step
- Axiom: weakening of axiom

clauses of width $O(\log s)$



(1) $\boldsymbol{F}$ is **SAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ has size-$\mathbf{2}^{O(\sqrt{n})}$ tree-like resolution refutation

Decision tree for $\mathbf{ShallowRef}(\boldsymbol{F})$



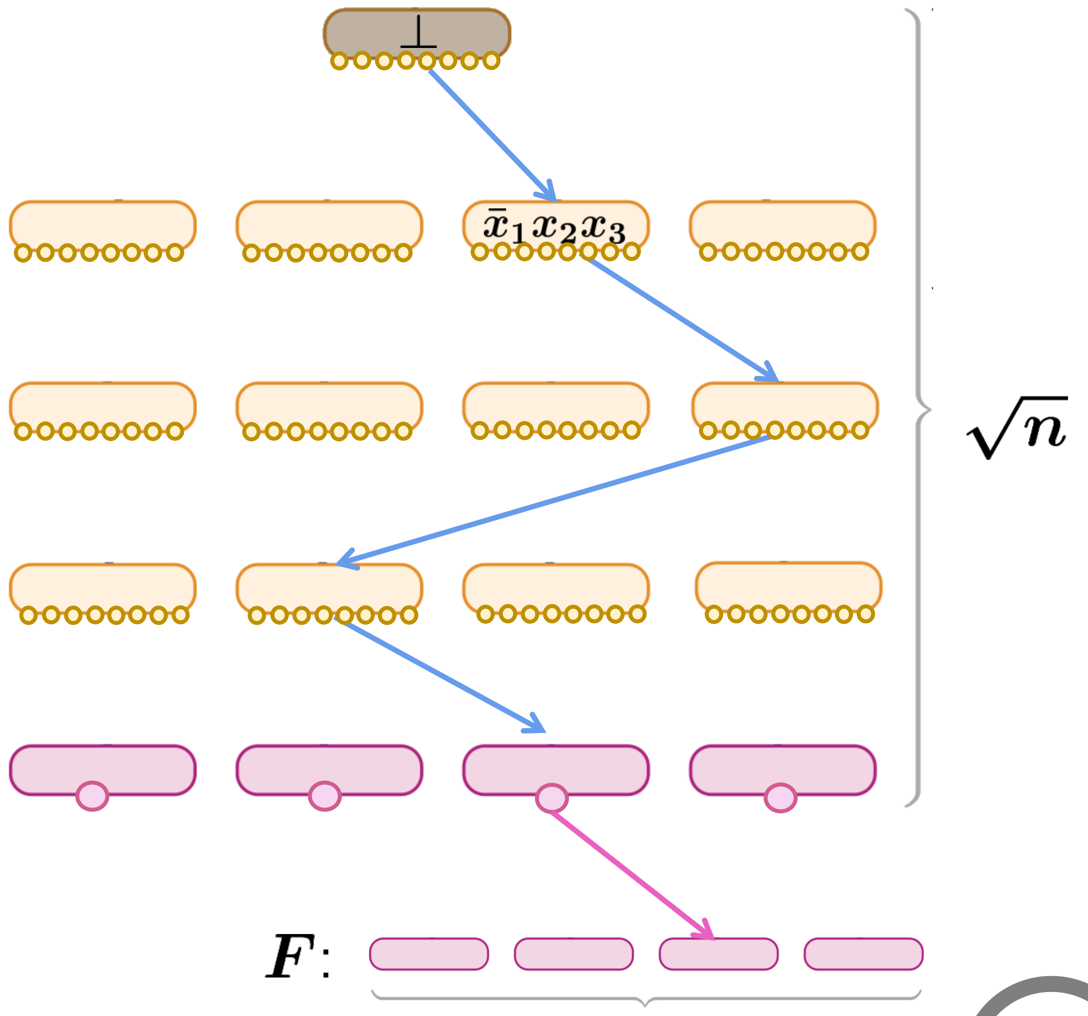
(1) $\boldsymbol{F}$ is **SAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ has size-$\mathbf{2}^{O(\sqrt{n})}$ tree-like resolution refutation

Decision tree for $\mathbf{ShallowRef}(\boldsymbol{F})$

$\boldsymbol{x}^*$ satisfying assignment for $\boldsymbol{F}$

invariant: $\boldsymbol{x}^*$ falsifies clause in block $\boldsymbol{B}$

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



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

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

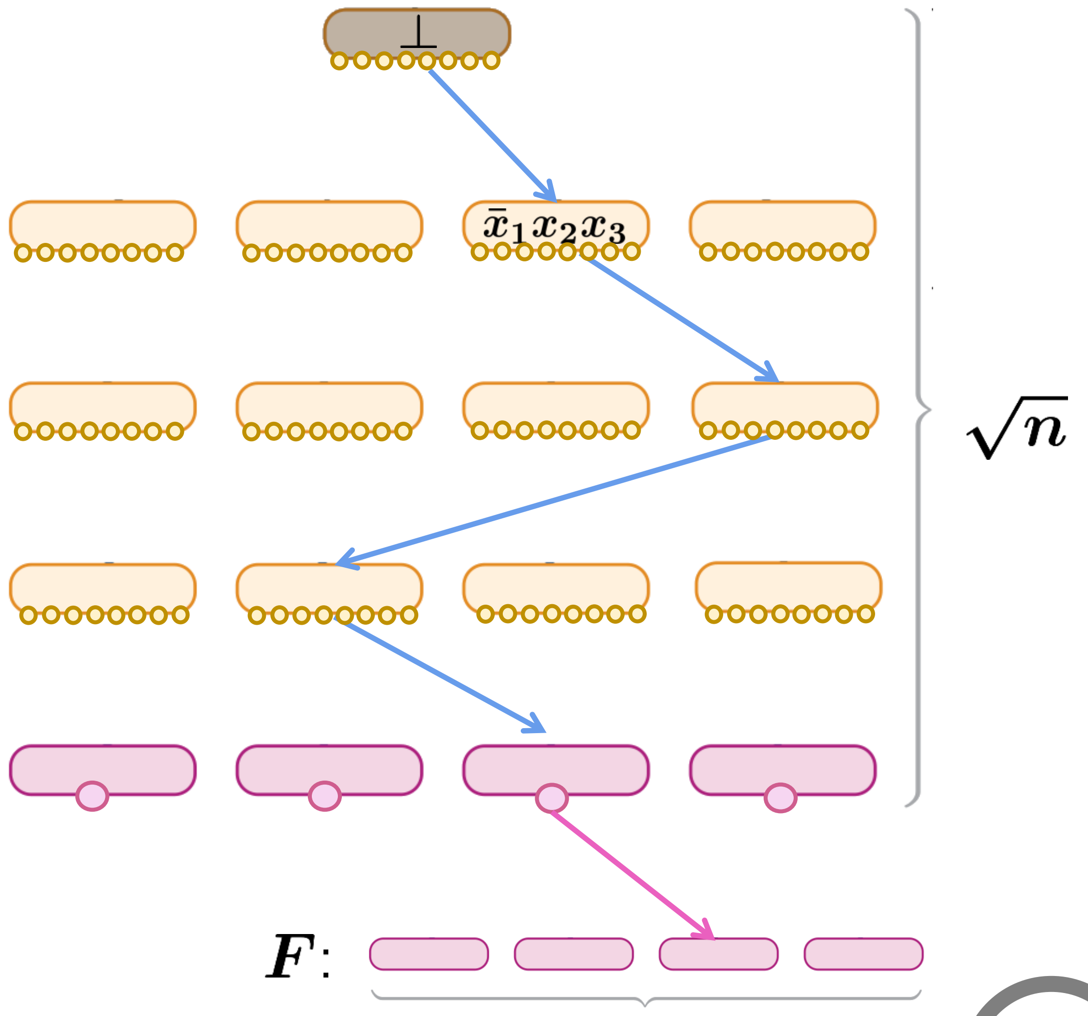
x^* satisfying assignment for F

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 Δ expander

tree-like refutation size $\approx \Delta^{\sqrt{n}} = 2^{O(\sqrt{n})}$



(2) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ requires tree-like res refutations size $\mathbf{2}^{\boldsymbol{\Omega(n)}}$

(2) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ requires tree-like res refutations size $\mathbf{2}^{\boldsymbol{\Omega(n)}}$

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

(2) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ requires tree-like res refutations size $\mathbf{2^{\Omega(n)}}$

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

(2) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ requires tree-like res refutations size $\mathbf{2}^{\boldsymbol{\Omega(n)}}$

- $\boldsymbol{w}(\mathbf{ShallowRef}(\boldsymbol{F}) \vdash \perp) \geq \boldsymbol{\Omega}(\boldsymbol{w}(\mathbf{GPHP}^{\mathbf{2}^{(\boldsymbol{c}+\mathbf{1})\sqrt{\boldsymbol{n}}}}_{\mathbf{2}^{\boldsymbol{c}\sqrt{\boldsymbol{n}}}} \vdash \perp)/\boldsymbol{n}) \geq \boldsymbol{\Omega}(\mathbf{2}^{\boldsymbol{c}\sqrt{\boldsymbol{n}}}/\boldsymbol{n})$
- [BW'01] size of tree-like resolution refutation of $\boldsymbol{\varphi} \geq \mathbf{2}^{\boldsymbol{w}(\boldsymbol{\varphi} \vdash \perp) - \boldsymbol{w}(\boldsymbol{\varphi})}$

(2) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ requires tree-like res refutations size $\mathbf{2}^{\boldsymbol{\Omega(n)}}$

- $w(\mathbf{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\mathbf{GPHP}^{2^{(c+1)\sqrt{n}}}_{2^{c\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$
- [BW'01] size of tree-like resolution refutation of $\varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)}$
- $\mathbf{ShallowRef}(F)$ has width $O(1)$ and $\#$ variables: $\mathbf{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

(2) F is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(F)$ requires tree-like res refutations size $2^{\Omega(n)}$

- $w(\mathbf{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\mathbf{GPHP}^{2^{(c+1)\sqrt{n}}}_{2^{c\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$
- [BW'01] size of tree-like resolution refutation of $\varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)}$
- $\mathbf{ShallowRef}(F)$ has width $O(1)$ and # variables: $\mathbf{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

$\mathbf{ShallowRef}(F)$ requires tree-like res refutation size $2^{\Omega(2^{c\sqrt{n}}/n)} \geq 2^{\Omega(n)}$

- $w(\mathrm{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\mathrm{GPHP}^{2^{(c+1)\sqrt{n}}}_{2^{c\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$
- [BW'01] size of tree-like resolution refutation of $\varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)}$
- $\mathrm{ShallowRef}(F)$ has width $O(1)$ and # variables: $\mathrm{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

$\mathrm{ShallowRef}(F)$ requires tree-like res refutation size $2^{\Omega(2^{c\sqrt{n}}/n)} \geq 2^{\Omega(n)}$

(2) F is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(F)$ requires dag-like res refutations size $\mathbf{2^{\Omega(n)}}$

- $w(\mathbf{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\mathbf{GPHP}^{2^{(c+1)\sqrt{n}}}_{2^{c\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$
- [BW'01] size of tree-like resolution refutation of $\varphi \geq 2^{w(\varphi \vdash \bot) - w(\varphi)}$
- $\mathbf{ShallowRef}(F)$ has width $O(1)$ and # variables: $\mathbf{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

$\mathbf{ShallowRef}(F)$ requires tree-like res refutation size $2^{\Omega(2^{c\sqrt{n}}/n)} \geq 2^{\Omega(n)}$

(2) F is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(F)$ requires dag-like res refutations size $2^{\Omega(n)}$

- $w(\mathbf{ShallowRef}(F) \vdash \perp) \geq \Omega(w(\mathbf{GPHP}^{2^{(c+1)\sqrt{n}}}_{2^{c\sqrt{n}}} \vdash \perp)/n) \geq \Omega(2^{c\sqrt{n}}/n)$
- [BW'01] size of resolution refutation of $\varphi \geq \exp\left(\Omega\left(\frac{(w(\varphi \vdash \perp) - w(\varphi))^2}{\#\text{var}}\right)\right)$
- $\mathbf{ShallowRef}(F)$ has width $O(1)$ and $\#$ variables: $\mathbf{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

$\mathbf{ShallowRef}(F)$ requires tree-like res refutation size $2^{\Omega(2^{c\sqrt{n}}/n)} \geq 2^{\Omega(n)}$

(2) $\boldsymbol{F}$ is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(\boldsymbol{F})$ requires dag-like res refutations size $\mathbf{2}^{\boldsymbol{\Omega(n)}}$

- $\boldsymbol{w(\mathbf{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\mathbf{GPHP}^{2^{(c+1)\sqrt{n}}}_{2^{c\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)}$
- [BW'01] size of resolution refutation of $\varphi \geq \mathbf{exp}\left(\boldsymbol{\Omega}\left(\frac{(w(\varphi\vdash\bot)-w(\varphi))^2}{\#\text{var}}\right)\right)$
- $\mathbf{ShallowRef}(\boldsymbol{F})$ has width $\boldsymbol{O(1)}$ and $\#$ variables: $\mathbf{poly}(\boldsymbol{n}) \cdot \mathbf{2}^{\boldsymbol{(c+1)\sqrt{n}}}$

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

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

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

choose $c = 2$

(2) F is **UNSAT** $\Rightarrow$ $\mathbf{ShallowRef}(F)$ requires dag-like res refutations size $2^{\Omega(n)}$

- $w(\mathbf{ShallowRef}(F) \vdash \bot) \geq \Omega(w(\mathbf{GPHP}^{2^{(c+1)\sqrt{n}}}_{2^{c\sqrt{n}}} \vdash \bot)/n) \geq \Omega(2^{c\sqrt{n}}/n)$
- [BW'01] size of resolution refutation of $\varphi \geq \exp\left(\Omega\left(\frac{(w(\varphi \vdash \bot) - w(\varphi))^2}{\#\text{var}}\right)\right)$
- $\mathbf{ShallowRef}(F)$ has width $O(1)$ and # variables: $\mathbf{poly}(n) \cdot 2^{(c+1)\sqrt{n}}$

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

choose $c = 2$

OBS. this same lower bound holds for PC:

- degree lower bound for $\mathbf{GPHP}^{\mathbf{poly}(s)}_{s}$
- similar size-degree relation

Generalization

If tree-like resolution is automatable:

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

Generalization

If tree-like resolution is automatable:

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

Generalization

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 $\mathrm{poly}(m) \cdot 2^{o(n)}$ then W[P] = FPT.

Generalization

Classical result in parameterized complexity [ADF'95]

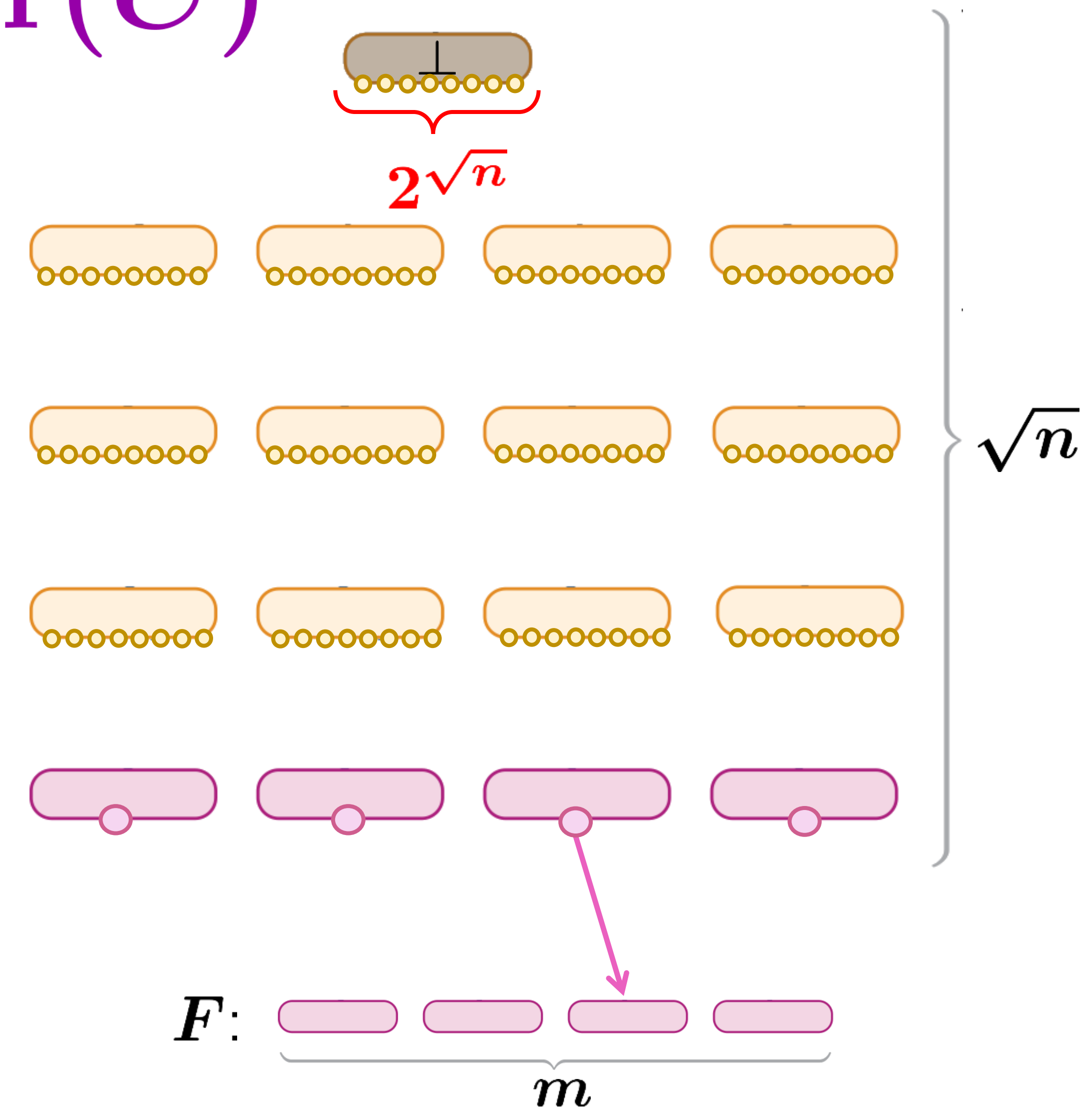
If $\exists$ algorithm that given n-variate circuit C of size m decides if C is satisfiable in time $\mathbf{poly}(m) \cdot \mathbf{2}^{o(n)}$ then W[P] = FPT.

Main Theorem

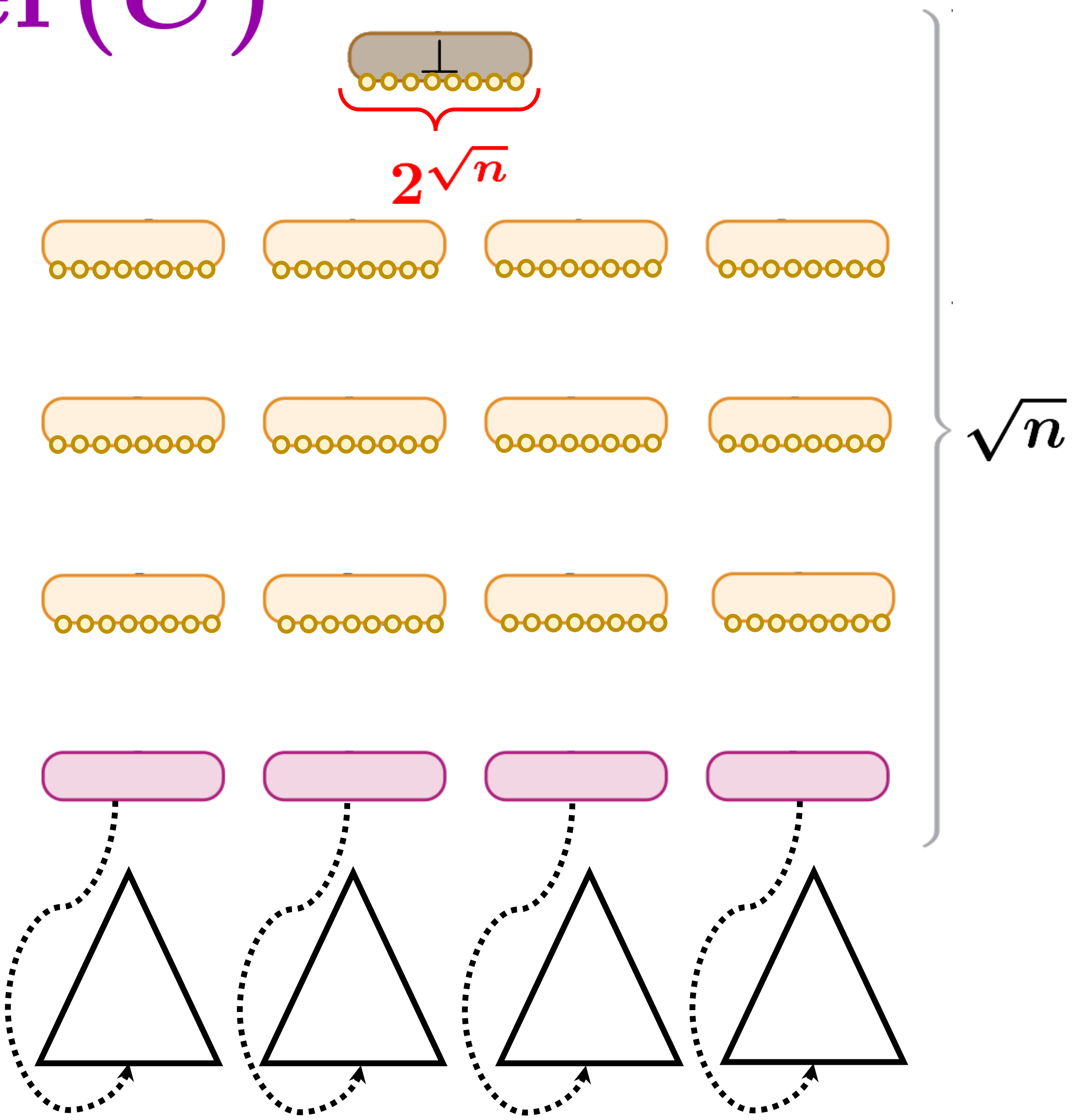
$\exists$ algorithm that given n-variate circuit C of size m outputs CNF $\mathcal{A}(C)$ in time $\mathbf{poly}(m) \cdot \mathbf{2}^{O(\sqrt{n})}$ s.t.

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

$\mathbf{ShallowRef}(C)$



$\mathbf{ShallowRef}(C)$



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($\boldsymbol{k}$), Res($\bigoplus$), 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($\boldsymbol{k}$), Res($\bigoplus$), 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: Res($\boldsymbol{k}$), Res($\bigoplus$), CP
- Weak automatability?

Thanks!