Automatability: the Proof Search Problem

Susanna F. de Rezende

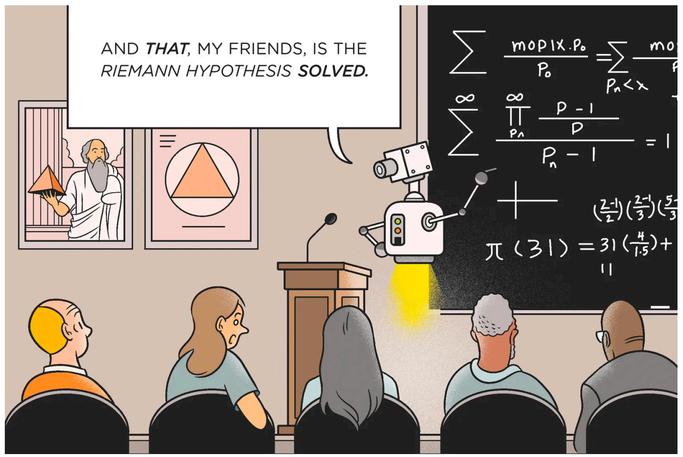
Lund University

Circuits, Communication and Proofs — ICTS-TIFR

14 August 2026

How hard is it to find proofs?

- Can a machine substitute a mathematician?
- Gödel's letter to von Neumann (20 March 1956)



Source: https://www.wired.co.uk/article/marcus-du-sautoy-maths-proofs

One can obviously easily construct a Turing machine, which for every formula F in first order predicate logic and every natural number n, allows one to decide if there is a proof of F of length n (length = number of symbols).

Let $\psi(F, n)$ be the number of steps the machine requires for this and let $\phi(n) = \max_F \psi(F, n)$.

The question is how fast $\phi(n)$ grows for an optimal machine. One can show that $\phi(n) \geq k \cdot n$.

If there really were a machine with $\phi(n) \sim k \cdot n$ (or even $\sim k \cdot n^2$), this would have consequences of the greatest importance. Namely, it would obviously mean that in spite of the undecidability of the Entscheidungsproblem, the mental work of a mathematician concerning Yes-or-No questions could be completely replaced by a machine. After all, one would simply have to choose the natural number n so large that when the machine does not deliver a result, it makes no sense to think more about the problem.

How hard is it to find proofs?

$\exists$ algorithm?

Polynomial time (on the size of the input) $+ s)$

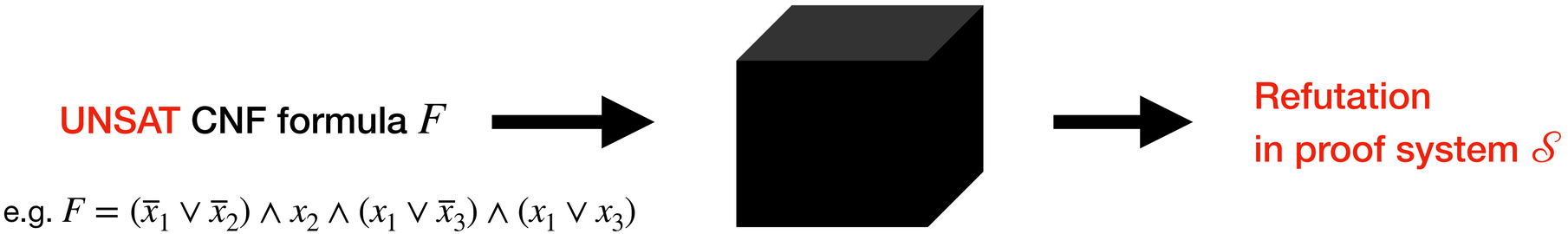
s = size of smallest proof

May not exist proof of polynomial size

Automatability

- Proof system $\mathcal{S}$ is ***automatable*** if $\exists$ algorithm: (in time $f(N)$)

[Bonet, Pitassi, Raz '97]



time $\text{poly}(|F| + s)$

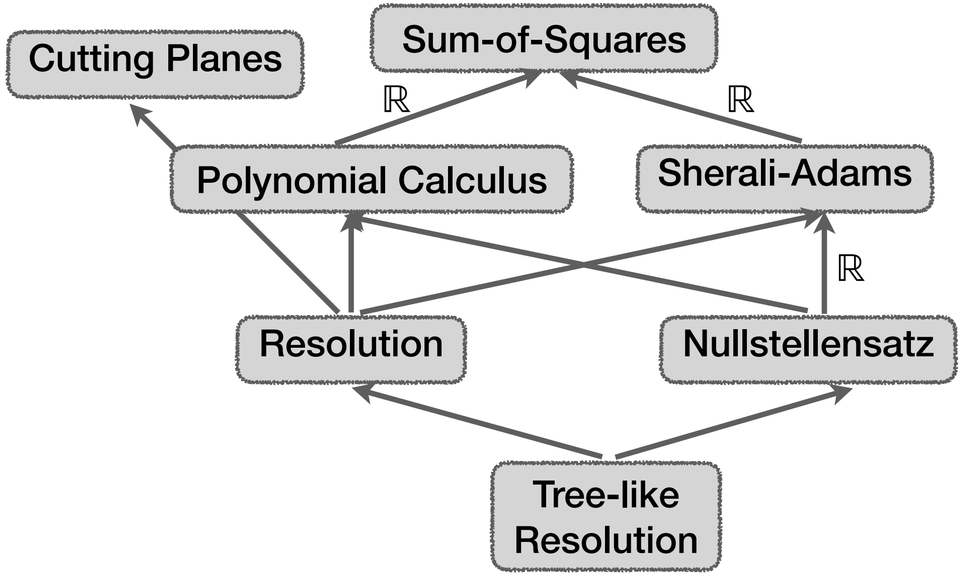
time $f(|F| + s)$

where s is the size of smallest refutation of F in proof system $\mathcal{S}$

- Best running time we can hope for $|F| + s$
- Here we are asking for time $\text{poly}(|F| + s)$

Why should a proof system be automatable?

- Weak proof system: can only prove simple facts ⇒ proofs should be easy to find
- There are algorithms that search for proofs:
  - DPLL: searching for **tree-like Resolution** proofs
  - CDCL: essentially searching for **Resolution** proofs
  - Pseudo-boolean SAT solvers: **Cutting Planes**
  - Linear algebra algorithms for **Nullstellensatz**
  - Gröbner basis algorithms: **Polynomial Calculus**
  - LP solving: **Sherali-Adams** LP relaxations
  - Ellipsoid method: **Sum-of-Squares** SDP relaxations

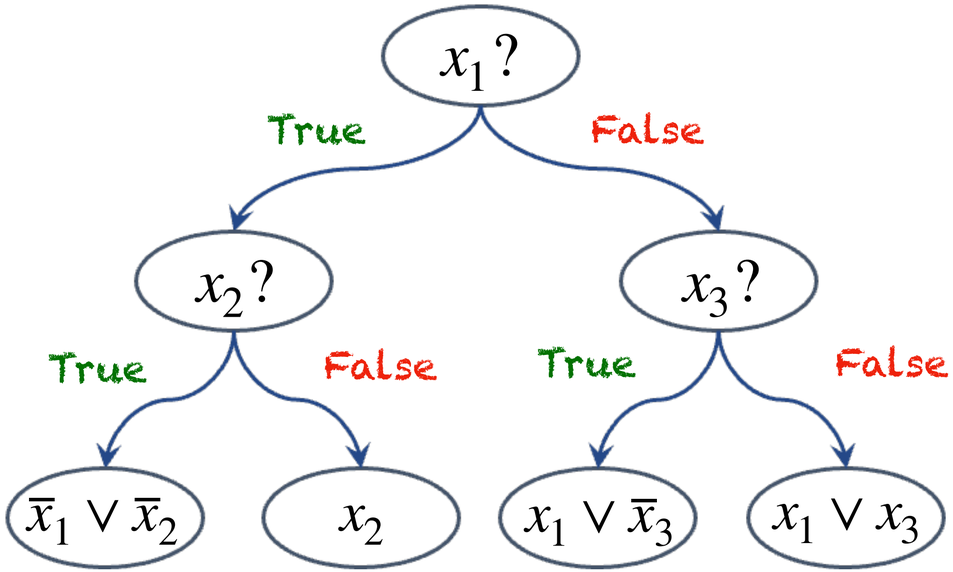


Tree-like resolution automatable in time $n^{O(\log s)}$

[Beame, Pitassi '96]

- Given UNSAT CNF formula F on n variables
- Suppose s, size of smallest refutation, is known
- $\exists x$ s.t. $\text{Size}_{\text{tree-like Res}}(F|_{x=0} \vdash \bot) \leq s/2$ or $\text{Size}_{\text{tree-like Res}}(F|_{x=1} \vdash \bot) \leq s/2$
- $T(n, s) = 2n \cdot T(n-1, s/2) + T(n-1, s) + n^{O(1)}$

  $T(n, s) = n^{O(\log s)}$
- Since s not known, repeat for $s = 1,2,3,\ldots$
- Can we do better than this?



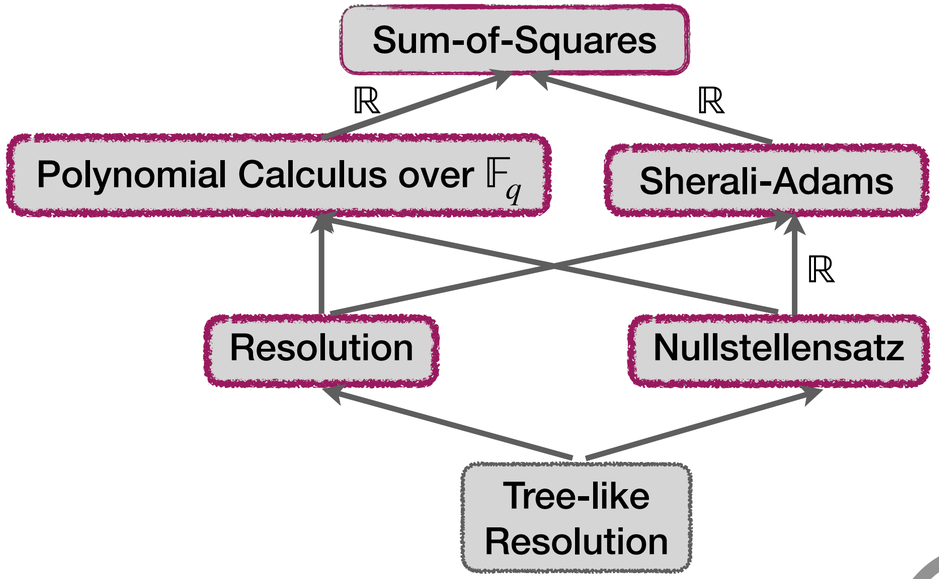
$F = (\bar{x}_1 \vee \bar{x}_2) \wedge x_2 \wedge (x_1 \vee \bar{x}_3) \wedge (x_1 \vee x_3)$

Proof search for bounded width / degree

- *In some cases*:* width/degree-d refutation can be found in time $n^{O(d)}$:
  - **Resolution, Nullstellensatz, Sherali-Adams**
  - **Polynomial Calculus over finite fields**
  - **Bounded coefficients Sum-of-Squares / Polynomial Calculus over $\mathbb{R}$**

* [Hakoniemi '21]

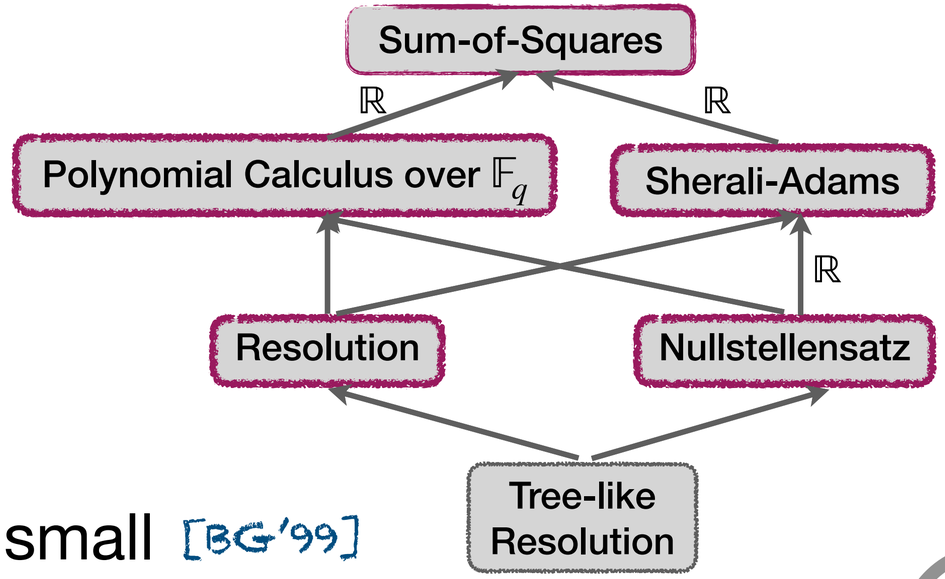
Is SoS degree automatable? Are there degree-coefficient trade-offs for CNF formulas?



Proof search in terms of size

- Size-s refutation → width/degree-$O(\sqrt{n \log s})$ refutation for:
  - **Resolution** [BW'99]
  - **Polynomial Calculus** [IPS'99]
  - **Sherali-Adams, Sum-of-Squares** [AH'19]

- Automatability in time $n^{O(\sqrt{n \log s})}$
  - Also for Nullstellensatz



- Can we do better?
  - Not via width: can be ~$\sqrt{n}$ even if proofs small [BG'99]

Why should a proof system **not** be automatable?

- Personal experience: finding proofs require insights
- Automatability → feasible interpolation [BPR 97 - Impagliazzo]
  - $\mathscr{A}$: automating algorithm, time $T(N)$
  - Find proof for $A(x,z) \wedge B(y,z)$ of size s (running $\mathscr{A}$)
  - Given assignment α to z: if $A(x,\alpha)$ is SAT, then $\exists$ size-s proof for $B(y,\alpha)$
  - Run $\mathscr{A}$ on $B(y,\alpha)$ for $T(s + |B|)$ steps: if it finds proof, output 1, otherwise 0

$\mathscr{P}$ admits feasible interpolation if short $\mathscr{P}$-proof for $A(x,z) \wedge B(y,z)$ → small circuit that given z outputs if A or B is SAT

One-way injective function f
If short $\mathscr{P}$-proof of injectivity: $(f(x0) = z) \wedge (f(y1) = z)$, then $\mathscr{P}$ admits feasible interpolation → can invert f bit by bit

Non-automatability of strong proof systems

not automatable unless RSA can be broken by poly-size circuits

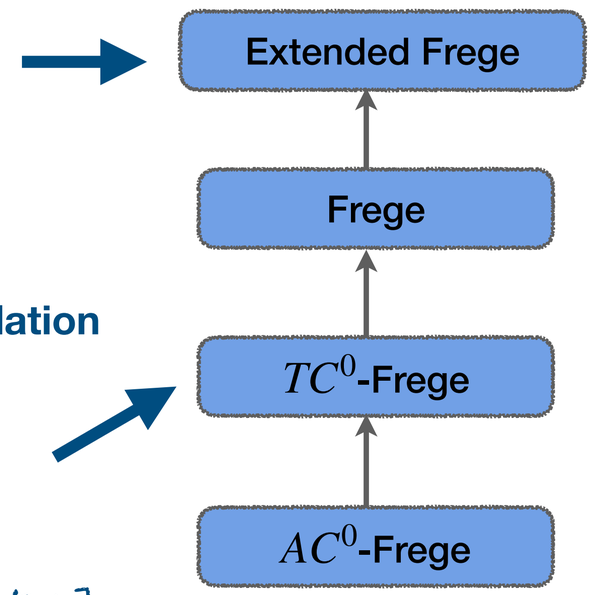
[Krajíček, Pudlák 98]

automatability → feasible interpolation

[BPR 97 - Impagliazzo]

not quantum automatable under LWE

[Arteche, Carenini, and Gray '24]



not automatable unless Diffie-Hellman procedure can be broken by small circuits TC^0 (poly size); AC^0 (sub-exp size)

[Bonet, Pitassi, Raz 97]

[Bonet, Domingo, Gavaldà, Maciel, Pitassi 99]

Some early results

- Determining if ∃ proof of size s is NP-complete
  - for certain Frege systems [Buss '95]
  - for Resolution (up to an additive $n^{O(1)}$) [Iwama '97]
- Length of shortest proof: NP-hard to approximate *within linear factor*
  - for several proof systems (including Resolution and Frege systems)

[Alekhnovich, Buss, Moran, Pitassi '98]

- Is it hard to find a proof of size poly(s)?

Non-automatability of resolution

- Cannot be automated in:
  - linear time unless P=NP [Alekhnovich, Buss, Moran, Pitassi '98]
  - polynomial time unless W[P] = FPT* [Alekhnovich, Razborov '01] [Eickmeyer, Grohe, Grüber '08]

"The main problem left open by this paper is whether general resolution is quasi-automatizable"

- time $N^{(\log \log N)^{1/7}}$ unless ETH is false* [Mertz, Pitassi, Wei '19]

* Also applies to tree-like resolution

[Atserias, Müller '19]

If resolution is automatable:
1. in time $\mathrm{poly}(N)$ then NP ⊆ P
2. in time $\mathrm{quasipoly}(N)$ then NP ⊆ QP
3. in time $\mathrm{subexp}(N)$ then NP ⊆ SUBEXP

$\mathrm{poly}(N) = N^{O(1)}$
$\mathrm{quasipoly}(N) = \exp(\log^{O(1)} N)$
$\mathrm{subexp}(N) = \exp(N^{o(1)})$

- Actually, "if and only if"

Main ideas behind proof and related questions

Reflection principle

- "if a sentence is **provable** then it is **true**"
- "if F has a $\mathscr{P}$-**refutation** (of size s) it is not **satisfiable**"
- Reflection principle: $\mathsf{Ref}_{\mathscr{P},s}(F) \rightarrow \neg\mathsf{sat}(F)$
  - How hard is it to prove lower bounds for $\mathscr{P}$? (Prove $\mathsf{Ref}_{\mathscr{P},s}(F)$ is unsat.)
  - Can AC^0-Frege prove superpoly Resolution lower bounds? [Pudlák '20]
  - If F is SAT then $\mathsf{Ref}_{\mathrm{Res},s}(F)$ has poly-size resolution refutation [Pudlák '01]

Proof of Atserias-Müller ’19

If resolution is automatable: in time $\mathrm{poly}(N)$ then NP $\subseteq$ P

Use: $\mathscr{A}(F)$ → ■ → Refutation π to decide if F is **SAT** or **UNSAT** in time $n^{O(1)}$

Formula $\mathscr{A}(F)$ s.t.

(1) F is **SAT** $\Rightarrow$ $\mathscr{A}(F)$ has refutations of size $n^{O(1)}$

(2) F is **UNSAT** $\Rightarrow$ $\mathscr{A}(F)$ requires refutations of size 2^n

1. Given F $\Rightarrow$ construct $\mathscr{A}(F)$
2. Run for $n^{O(1)}$ steps: $\mathscr{A}(F)$ → ■ → Refutation π
3. Answer:
   - F is **SAT** if π is a refutation of $\mathscr{A}(F)$
   - F is **UNSAT** otherwise

$\mathscr{A}(F) = \mathrm{Ref}_s(F)$

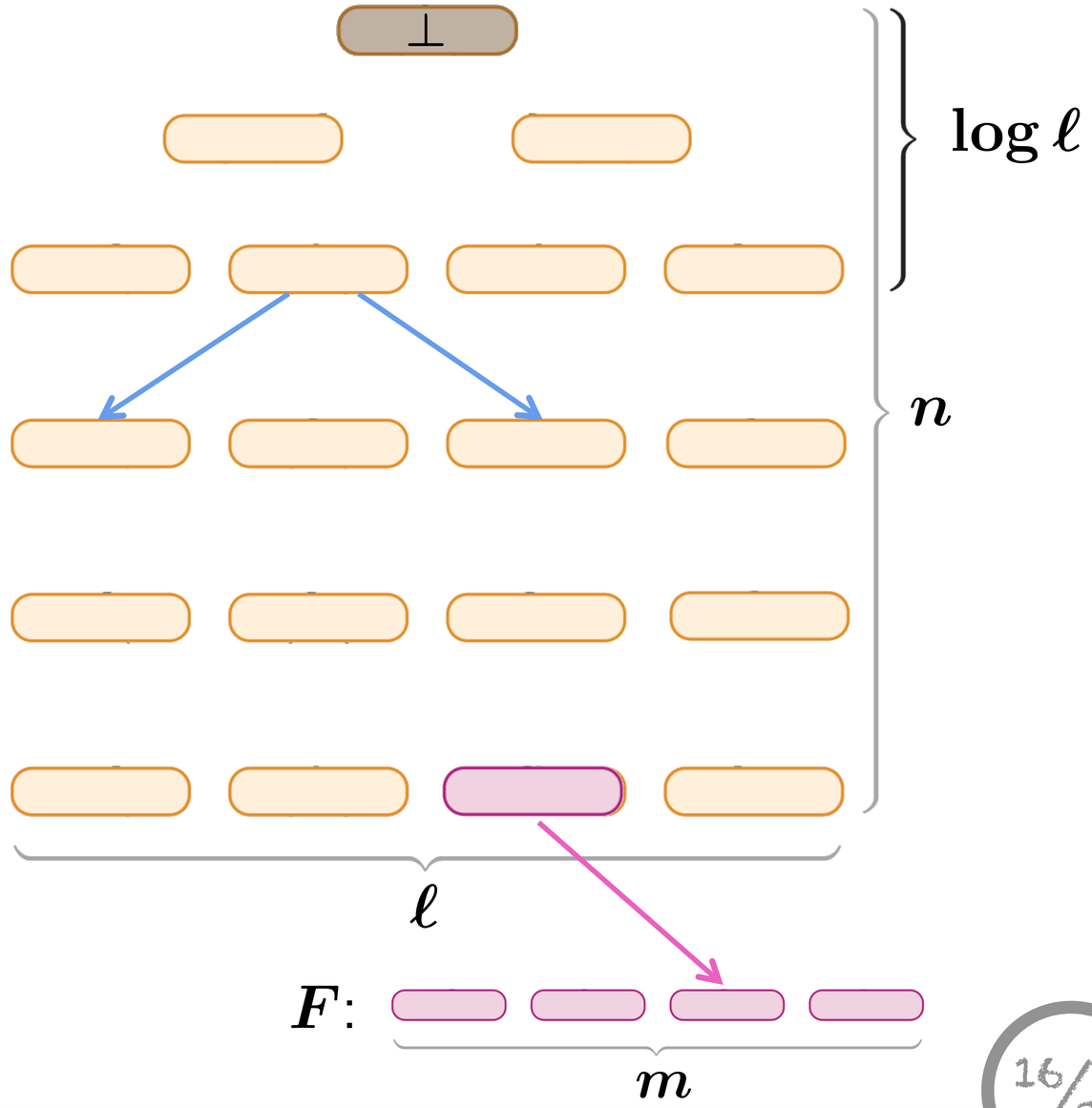
Variables for each block B:

- $2n$ variables (1 per literal): indicates clause
- 1 variable encoding if axiom or derived
- $\log m$ variables: axiom-index $j \in [m]$
- $O(\log n)$ variables: pointers to children
  "derived from B_i and B_j by resolving x_k"

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

- Root: $\bot$ clause
- Derived: from layer below, valid resolution step
- Leaf: implied by some axiom

Encodes "F has short resolution proof"



(1) F is **SAT** $\Rightarrow \mathrm{Ref}_s(F)$ has refutations of size $n^{O(1)}$

Prover–Adversary game for $\mathbf{Ref}(F)$

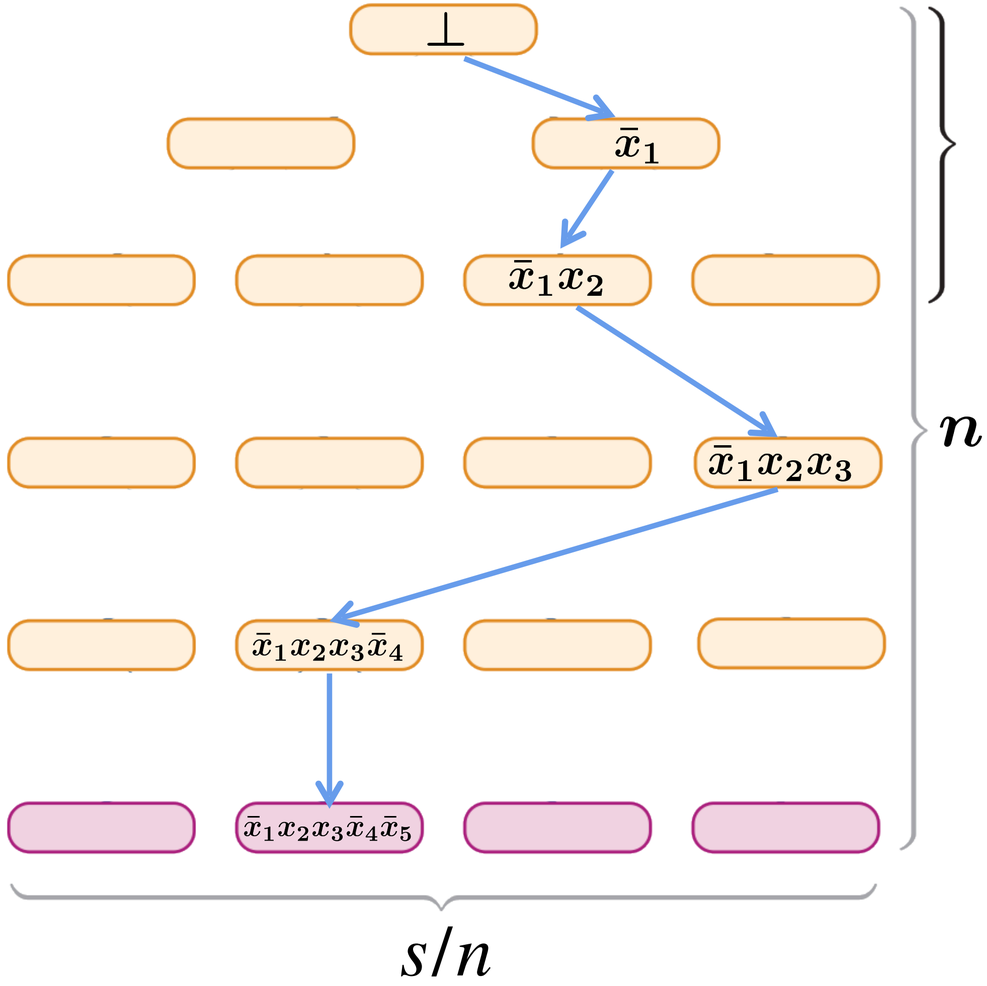
x^* satisfying assignment for F

e.g. $x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 1$

ρ : clause in block B doesn't contain any of $x_1, \bar{x}_2, \bar{x}_3, x_4, x_5$

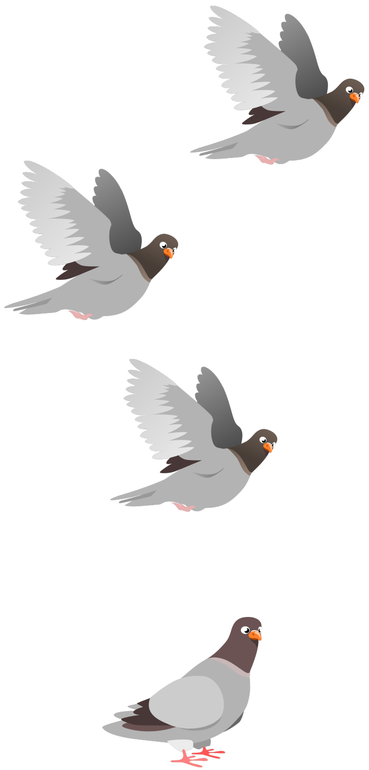
Start at root and keep invariant
until detect non-valid derivation step or
until reach leaf (cannot be weakening of axiom)

Remember only current block: $n^{O(1)}$ states

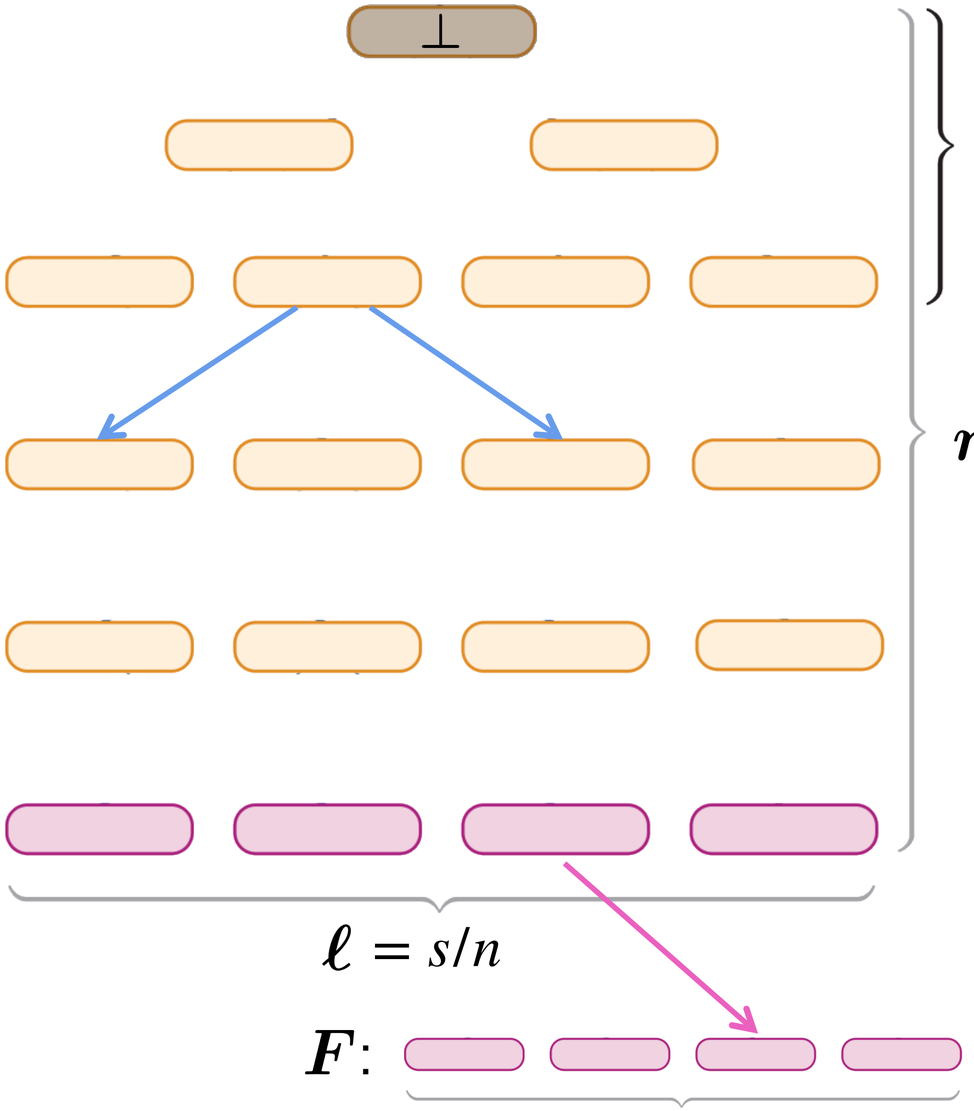


(2) F is **UNSAT** $\Rightarrow \mathrm{Ref}_s(F)$ requires refutations of width s/n

$$\mathbf{PHP} \leq \mathbf{Ref}(F)$$

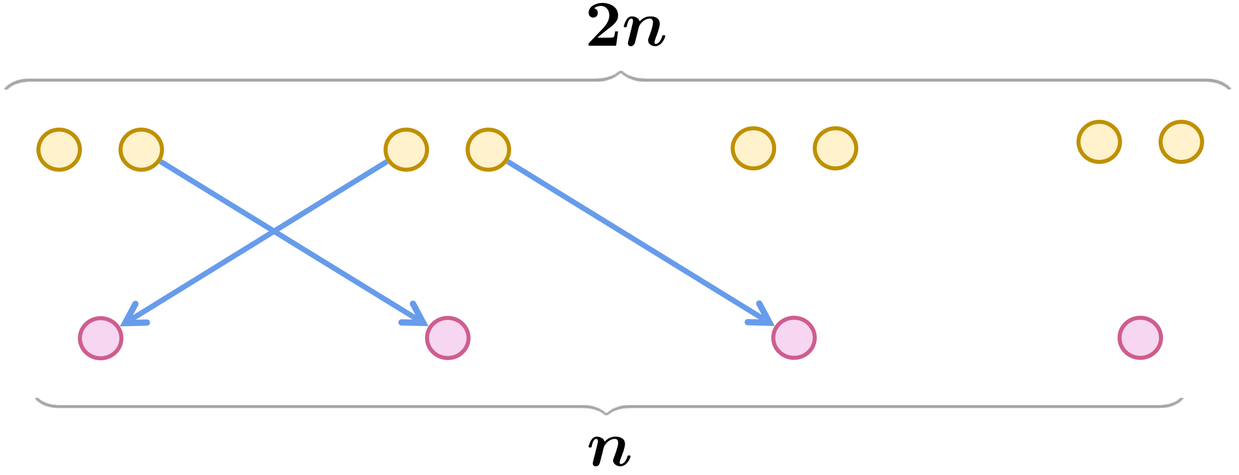


$\leq$



rPHP: retraction weak pigeonhole principle

$\mathbf{rPHP}_n$



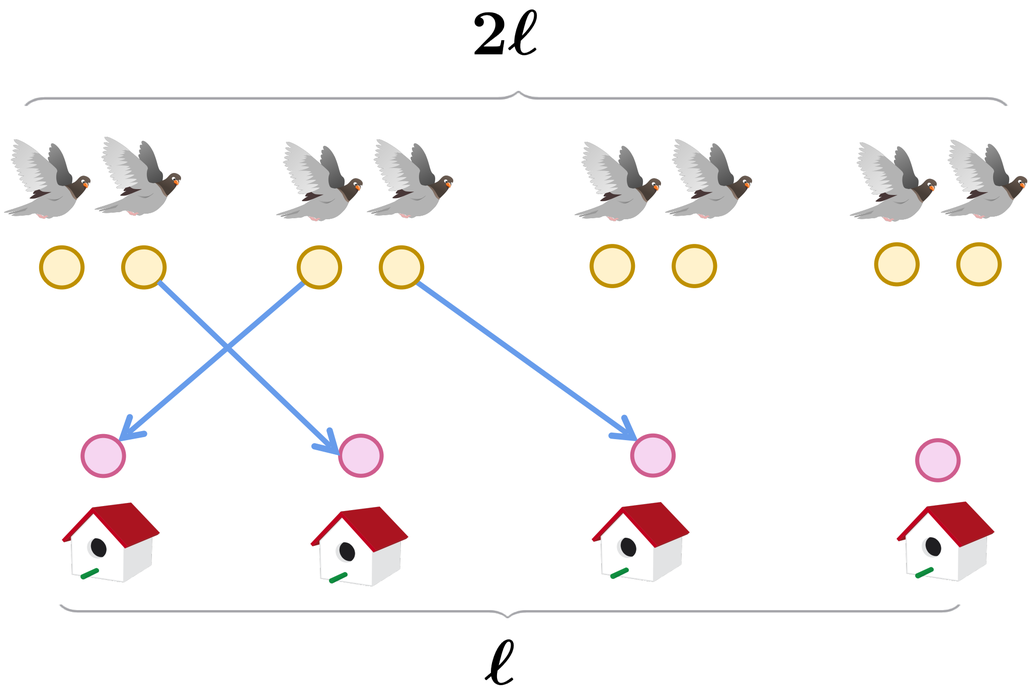
Variables:

- For pigeon $i \in [2n]$, variables f_{ik} for $k \in [\log n]$: indicates hole $f(i)$
- For hole $j \in [n]$, variables g_{jk} for $k \in [\log 2n]$: indicates pigeon $g(j)$

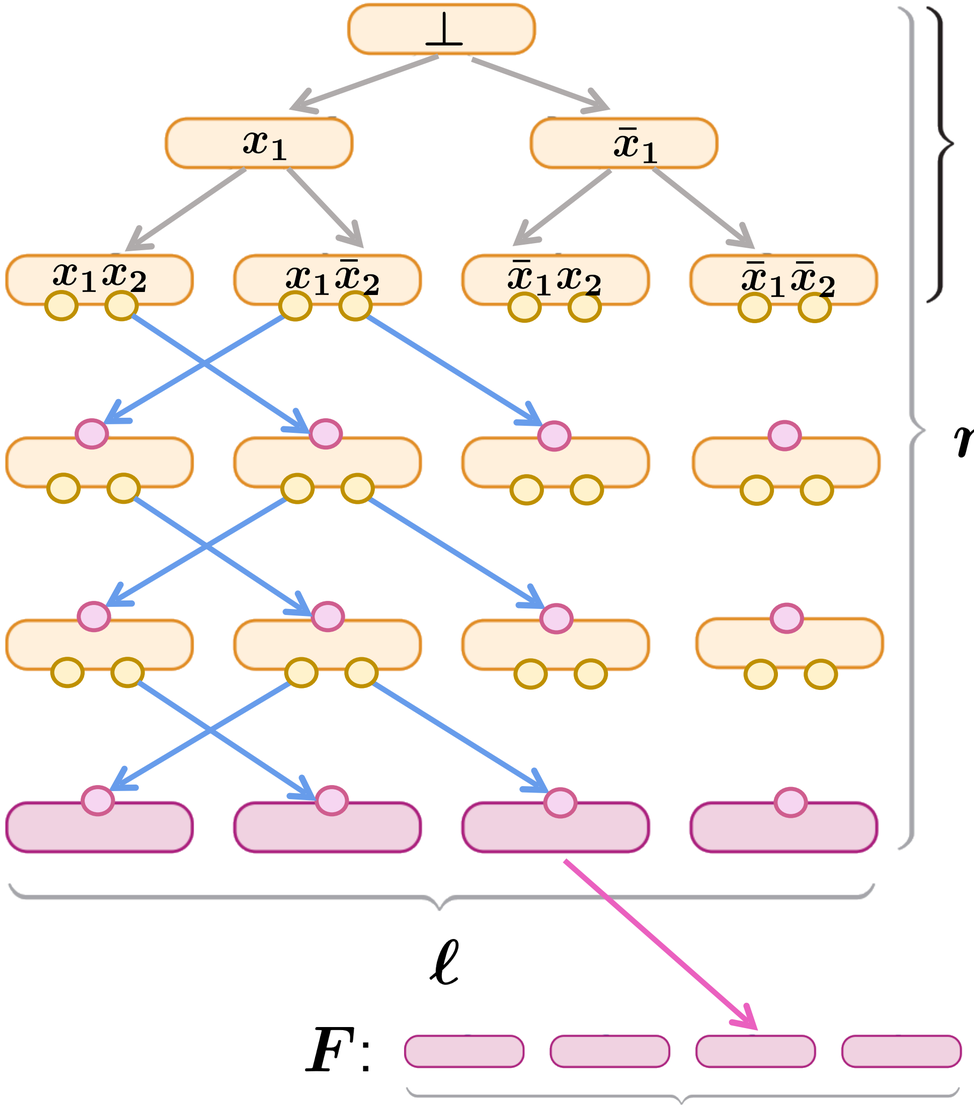
Axioms:

- $f(i) = j \Rightarrow g(j) = i$

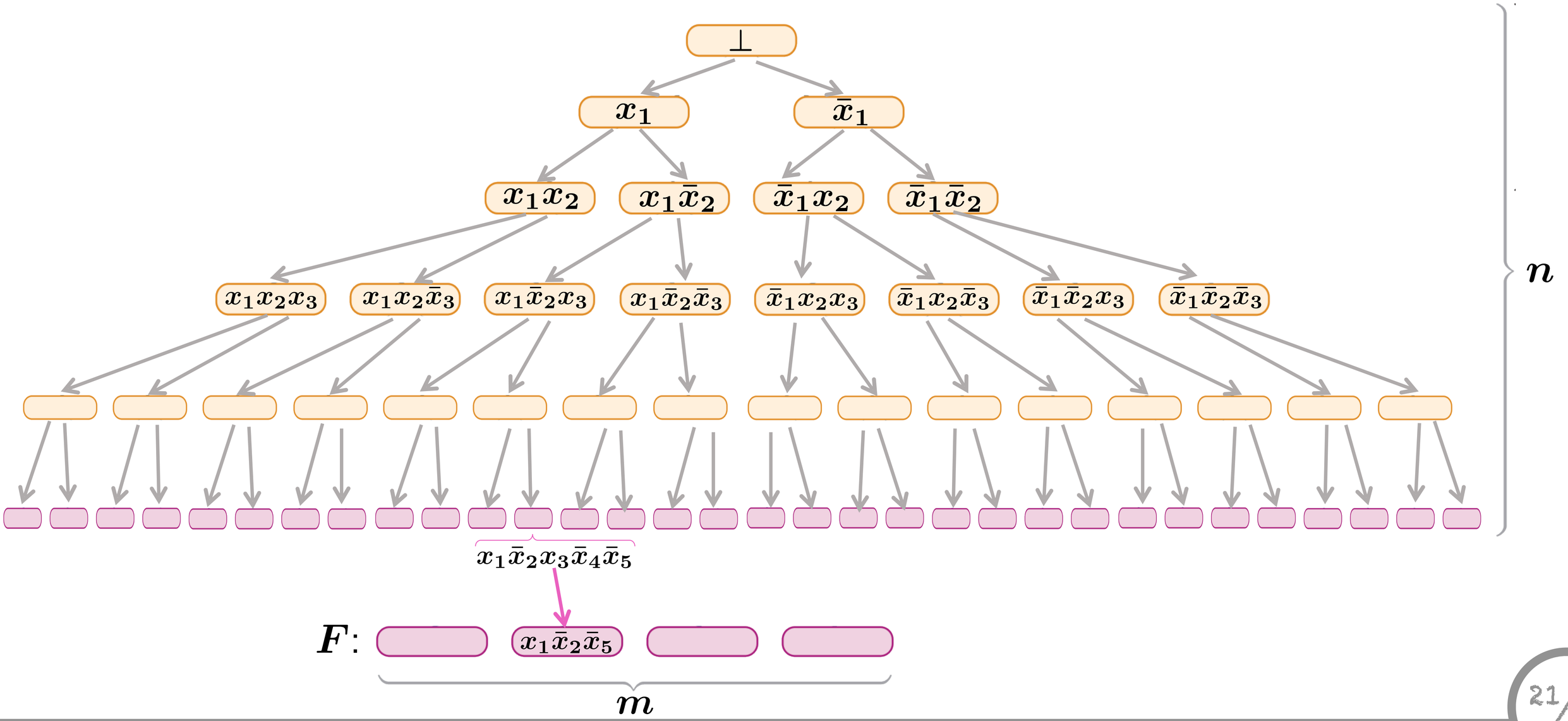
$\mathbf{rPHP} \leq \mathbf{Ref}(F)$



$\leq$



Universal proof (complete Binary tree)



(Simplified) proof of main theorem [AM '19]

- Define $\mathscr{A}(F) := \mathsf{Ref}_s(F) := \mathsf{Ref}_{\mathsf{Res},s}(F)$
- 1. F is SAT $\Rightarrow$ $\mathsf{Ref}_s(F)$ has poly-size resolution refutation [Pudlák '01]
  - Idea: use a satisfying assignment to find invalid step of purported refutation
- 2. F is UNSAT $\Rightarrow$ $\mathsf{Ref}_s(F)$ requires resolution refutation of size $2^{\Omega(n)}$
  - Encode $\mathsf{Ref}_s(F)$ with $O(sn)$ variables and $O(\log n)$ width
  - Prove width lower bound $w(\mathsf{Ref}_s(F) \vdash \bot) \geq \Omega(s/n)$
  - Apply size-width relation [BW '01] and conclude that size is at least $\exp(\Omega(s/n^3))$
  - Choose $s \geq n^4$

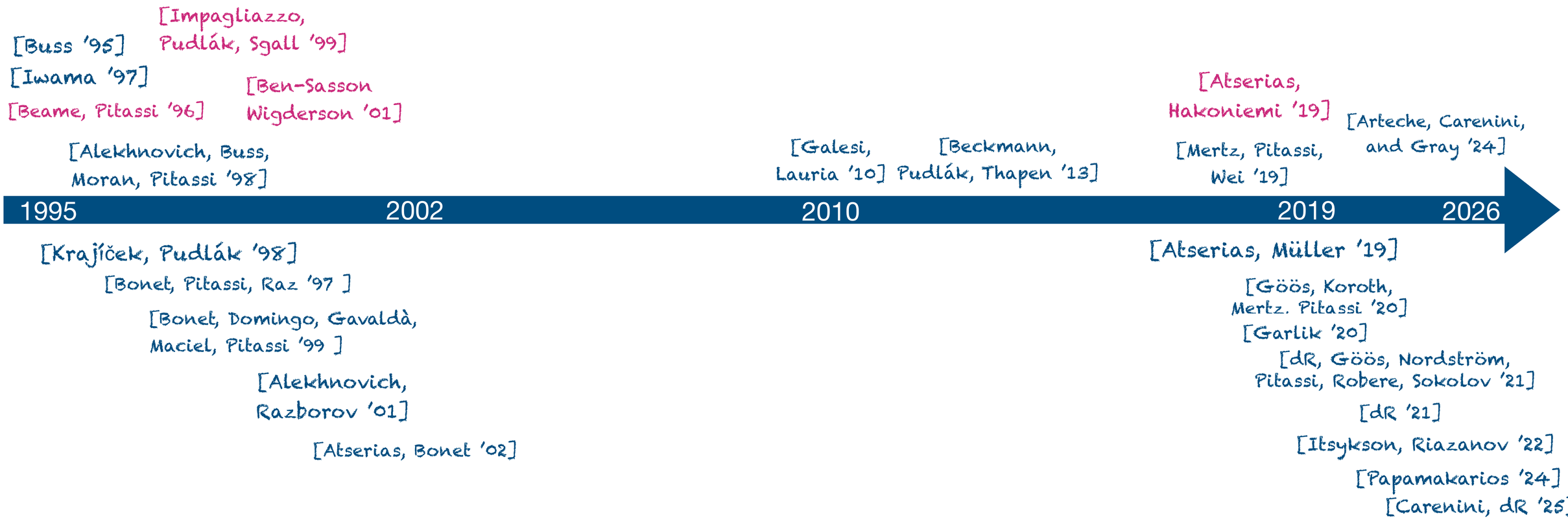
More robust proof

- Define $\mathscr{A}(F) := \text{Lift}(\text{Ref}_s(F))$ s.t. upper bound 1. still holds
- 2. F is UNSAT $\Rightarrow$ $\text{Ref}_s(F)$ requires Resolution refutation of size $2^{\Omega(n)}$
  - Prove block-width/degree lower bound $bw(\text{Ref}_s(F) \vdash \bot\ ) \geq \Omega(s/n)$
  - $\text{Lift}(\text{Ref}_s(F))$ requires Resolution refutation of size $\exp(\Omega(s/n))$ [Atserias, Müller 19]
  - $\text{Lift}_{\text{CP}}(\text{Ref}_s(F))$ requires CP refutation of size $\exp(\Omega(s/n))$ [Göös, Koroth, Mertz, Pitassi 19]
  - $\text{Lift}_k(\text{Ref}_s(F))$ requires Res(k) refutation of size $\exp(\Omega(s/n^k))$ [Garlik 20]
  - $\text{Lift}(\text{Ref}_s(F))$ requires NS and PC refutation of size $\exp(\Omega(s/n))$ [dR, Göös, Nordström, Pitassi, Robere, Sokolov 21]
  - $\text{Lift}_d(\text{Ref}_s(F))$ requires depth-d Frege refutation of exponential size* [Papamakarios '24]

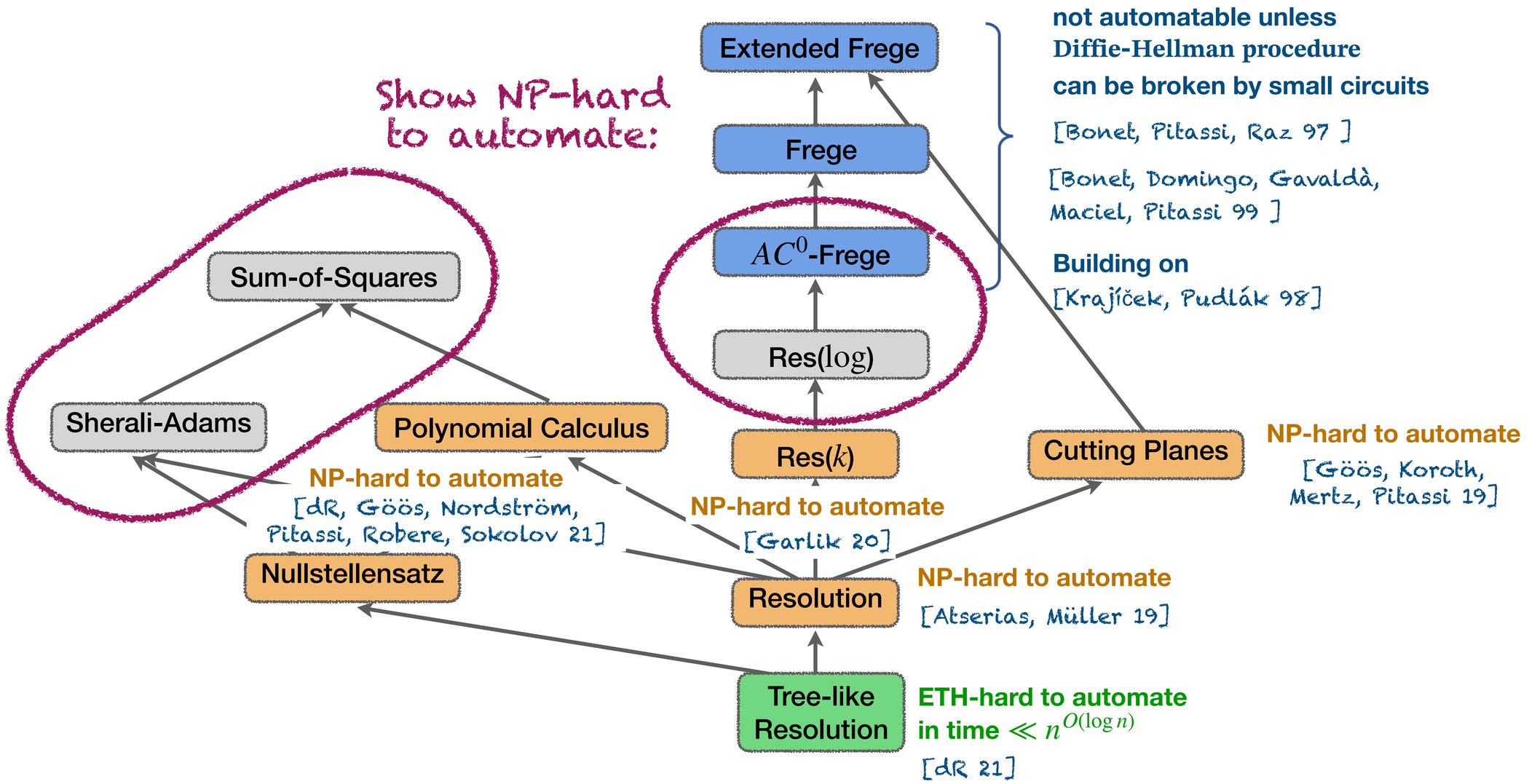
Follow-up results

- Cannot be automated in poly time unless P = NP, building on [Atserias, Müller '19]
  - **Regular, ordered Resolution** [Bell '20]
  - **Cutting Planes** [Göös, Koroth, Mertz, Pitassi '19]
  - **Res(k)** for small k [Garlik '20]
  - **Nullstellensatz and Polynomial Calculus** [dR, Göös, Nordström, Pitassi, Robere, Sokolov '21]
  - **Tree-like Resolution** (in time $n^{o(\log s)}$ unless rETH is false) [dR '21]
  - **OBDD** [Itsykson, Riazanov '22]
  - **Frege (non CNFs)** [Papamakarios '24]
  - **Tree-like Res(k)** (in time $n^{o(\frac{k}{\log k}\log s)}$ unless rETH is false) [Carenini, dR '25]

Timeline


[Buss '95]
[Iwama '97]
[Beame, Pitassi '96]
[Impagliazzo, Pudlák, Sgall '99]
[Ben-Sasson Wigderson '01]
[Alekhnovich, Buss, Moran, Pitassi '98]
[Galesi, Lauria '10]
[Beckmann, Pudlák, Thapen '13]
[Atserias, Hakoniemi '19]
[Mertz, Pitassi, Wei '19]
[Arteche, Carenini, and Gray '24]
1995
2002
2010
2019
2026
[Krajíček, Pudlák '98]
[Bonet, Pitassi, Raz '97]
[Bonet, Domingo, Gavaldà, Maciel, Pitassi '99]
[Alekhnovich, Razborov '01]
[Atserias, Bonet '02]
[Atserias, Müller '19]
[Göös, Koroth, Mertz. Pitassi '20]
[Garlik '20]
[dR, Göös, Nordström, Pitassi, Robere, Sokolov '21]
[dR '21]
[Itsykson, Riazanov '22]
[Papamakarios '24]
[Carenini, dR '25]

Summary


Show NP-hard to automate:
Extended Frege
Frege
AC^0-Frege
Res(log)
Res(k)
Sum-of-Squares
Sherali-Adams
Polynomial Calculus
Nullstellensatz
Resolution
Tree-like Resolution
Cutting Planes
not automatable unless Diffie-Hellman procedure can be broken by small circuits
[Bonet, Pitassi, Raz 97]
[Bonet, Domingo, Gavaldà, Maciel, Pitassi 99]
Building on
[Krajíček, Pudlák 98]
NP-hard to automate
[dR, Göös, Nordström, Pitassi, Robere, Sokolov 21]
NP-hard to automate
[Garlik 20]
NP-hard to automate
[Göös, Koroth, Mertz, Pitassi 19]
NP-hard to automate
[Atserias, Müller 19]
ETH-hard to automate in time $\ll n^{O(\log n)}$
[dR 21]

Weak automatability

$\mathscr{P}$ **is weakly automatable if**
$\exists$ **algorithm that given unsat CNF** F
returns $\mathscr{Q}$**-refutation of** F **in time poly**$(|F| + s)$

where s is the size of the shortest $\mathscr{P}$-refutation of F

- $\mathscr{P}$ is weakly automatable $\iff$ Canonical pair of $\mathscr{P}$ is poly separable [Pudlák '01]
- If $\mathscr{P}$ proves its own soundness: $\mathscr{P}$ is weakly aut. $\iff$ $\mathscr{P}$ has feasible interpolation [Pudlák '01]
- Resolution weakly aut. $\iff$ Res(k) weakly aut. $\iff$ Res(k) has feasible interpolation [Atserias, Bonet '02]
- Resolution weakly automatable $\Rightarrow$ parity games decided in poly time [Beckmann, Pudlák, Thapen '13]
  quasi-poly upper bound have been proved [CJKLS '17]

Minimum proof size problem (MPSP)

Now it seems to me, however, to be completely within the realm of possibility that $\phi(n)$ grows that slowly. Since it seems that $\phi(n) \geq k \cdot n$ is the only estimation which one can obtain by a generalization of the proof of the undecidability of the Entscheidungsproblem and after all $\phi(n) \sim k \cdot n$ (or $\sim k \cdot n^2$) only means that the number of steps as opposed to trial and error can be reduced from N to $\log N$ (or $(\log N)^2$). […]

It would be interesting to know […] how strongly in general the number of steps in finite combinatorial problems can be reduced with respect to simple exhaustive search.

- Exhaustive search for proof of size s takes time 2^s. Can we do better?
- Can finding short proofs require more time than finding long ones? (2^s vs 2^n)

Input: Parameter $s < 2^n$ and trivial proof of size 2^n
Output: Is there a proof of size s?

Can MPSP be solved in time poly$(s + 2^n)$? (For tree-like resolution: yes!)

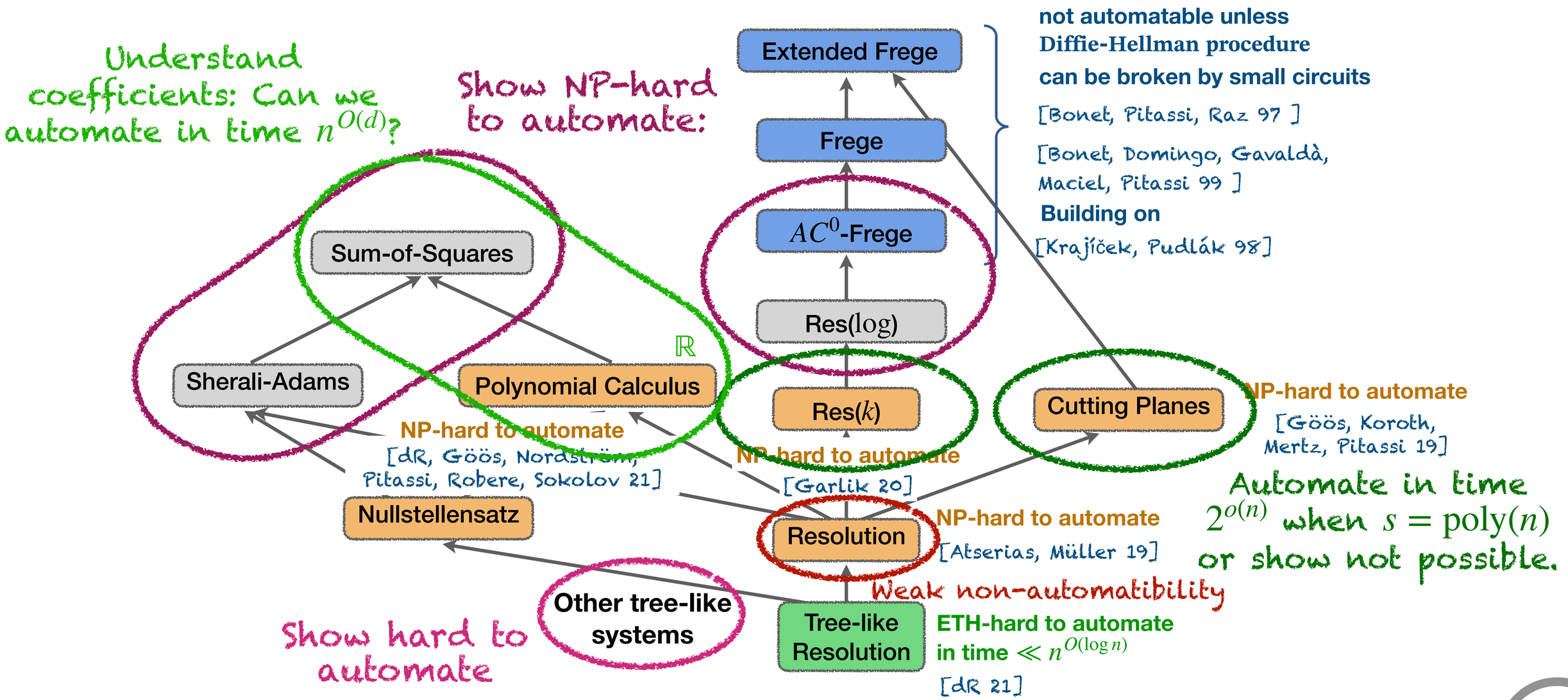
Proof analysis problem (PAP) [Arteche, Atserias, dR, Khaniki '25]

- π short Resolution refutation of $\mathrm{Ref}(F)$ then F is **SAT**
- Can we extract from π a satisfying assignment for F?

- If we can formalize the proof of AM lower bound (in PV / extended Frege): yes!
- Witnessing lemma
- How efficient is the algorithm that extracts assignments? [Arteche, dR, Khaniki, Risse '26]

- What if π is an AC^0-Frege refutation? A PC refutation? Can we extract assignment?
- What systems can prove Resolution lower bounds?

Open problems



Links to references

- Gödel's lost letter https://rjlipton.wpcomstaging.com/the-gdel-letter/
- [Alekhnovich, Buss, Moran, Pitassi '98]
  Minimum propositional proof length is NP-hard to linearly approximate https://doi.org/10.2307/2694916
- [Alekhnovich, Razborov '01]
  Resolution Is Not Automatizable Unless W[P] Is Tractable https://doi.org/10.1137/06066850X
- [Arteche, Atserias, dR, Khaniki '25] The Proof Analysis Problem https://doi.org/10.1109/FOCS63196.2025.00133
- [Arteche, Carenini, and Gray '24] Quantum Automating TC^0-Frege Is LWE-Hard https://doi.org/10.4230/LIPIcs.CCC.2024.15
- [Arteche, dR, Khaniki, Risse '25] The Proof Analysis Problem for Constant-Depth Frege
  https://doi.org/10.1109/FOCS63196.2025.00133
- [Atserias, Bonet '02] On the automatizability of resolution and related propositional proof systems
  https://eccc.weizmann.ac.il/report/2002/010/ and https://doi.org/10.1016/j.ic.2003.10.004
- [Atserias, Müller '19] Automating Resolution is NP-Hard https://doi.org/10.1145/3409472
- [AH'19] [Atserias, Hakoniemi '19]
  Size-degree trade-offs for sums-of-squares and positivstellensatz proofs https://doi.org/10.4230/LIPIcs.CCC.2019.24
- [Beame, Pitassi '96] Simplified and improved resolution lower bounds https://doi.org/10.1109/SFCS.1996.548486
- [Beckmann, Pudlák, Thapen '13] Parity Games and Propositional Proofs https://doi.org/10.1145/2579822
- [Bell '20] Automating Regular or Ordered Resolution is NP-Hard https://eccc.weizmann.ac.il/report/2020/105/
- [Bonet, Domingo, Gavaldà, Maciel, Pitassi '99]
  Non-automatizability of bounded-depth Frege proofs https://doi.org/10.1007/s00037-004-0183-5
- [BPR'97] [Bonet, Pitassi, Raz '97]
  On Interpolation and Automatization for Frege Systems https://doi.org/10.1137/S0097539798353230
- [Buss '95] On Gödel's Theorems on Lengths of Proofs II: Lower Bounds for Recognizing k Symbol Provability
  https://doi.org/10.1007/978-1-4612-2566-9_4
- [BW '01] Short Proofs are Narrow — Resolution made Simple https://doi.org/10.1145/375827.375835

Links to references

- [Carenini, dR '25] On the Automatability of Tree-Like k-DNF Resolution https://doi.org/10.4230/LIPIcs.CCC.2025.14
- [CJKLS '17] Deciding Parity Games in Quasi-polynomial Time https://doi.org/10.1137/17M1145288
- [dR, Göös, Nordström, Pitassi, Robere, Sokolov '21] Automating Algebraic Proof Systems Is NP-Hard https://doi.org/10.1145/3406325.3451080
- [dR '21] Automating Tree-Like Resolution in Time n^o(log n) Is ETH-Hard https://doi.org/10.1016/j.procs.2021.11.021
- [Eickmeyer, Grohe, Grüber '08] Approximation of Natural W[P]-Complete Minimisation Problems Is Hard https://doi.org/10.1109/CCC.2008.24
- [Galesi, Lauria '10] On the Automatizability of Polynomial Calculus https://doi.org/10.1007/s00224-009-9195-5
- [Garlik '20] Failure of feasible disjunction property for k-DNF Resolution and NP-hardness of automating https://doi.org/10.48550/arXiv.2003.10230
- [Göös, Koroth, Mertz, Pitassi '20] Automating cutting planes is NP-hard https://doi.org/10.1145/3357713.3384248
- [Hakoniemi '21] Monomial size vs. bit-complexity in sums-of-squares and polynomial calculus https://doi.org/10.1109/LICS52264.2021.9470545
- [IPS'99] Impagliazzo, Pudlák, Sgall Lower bounds for the polynomial calculus and the Gröbner basis algorithm https://doi.org/10.1007/s000370050024
- [Itsykson, Riazanov '22] Automating OBDD proofs is NP-hard https://doi.org/10.4230/LIPIcs.MFCS.2022.59
- [Iwama '97] Complexity of finding short resolution proofs https://doi.org/10.1007/BFb0029974
- [Krajíček, Pudlák '98] Some consequences of cryptographical conjectures for S21 and EF https://www2.karlin.mff.cuni.cz/~krajicek/j-crypto.ps
- [Mertz, Pitassi, Wei '19] Short proofs are hard to find https://doi.org/10.4230/LIPIcs.ICALP.2019.84
- [Papamakarios '24] Depth-d Frege Systems Are Not Automatable Unless P = NP doi.org/10.4230/LIPIcs.CCC.2024.22
- [Pudlák '01] On reducibility and symmetry of disjoint NP pairs https://doi.org/10.1016/S0304-3975(02)00411-5
- [Pudlák '20] Reflection principles, propositional proof systems, and theories https://doi.org/10.48550/arXiv.2007.14835