On the Complexity of Solving Propositional Formulas

Florian Wörz

Institute of Theoretical Computer Science, Universität Ulm

Doctoral Advisor: Prof. Dr. Jacobo Torán

December 13, 2022

The SAT Problem

Given:

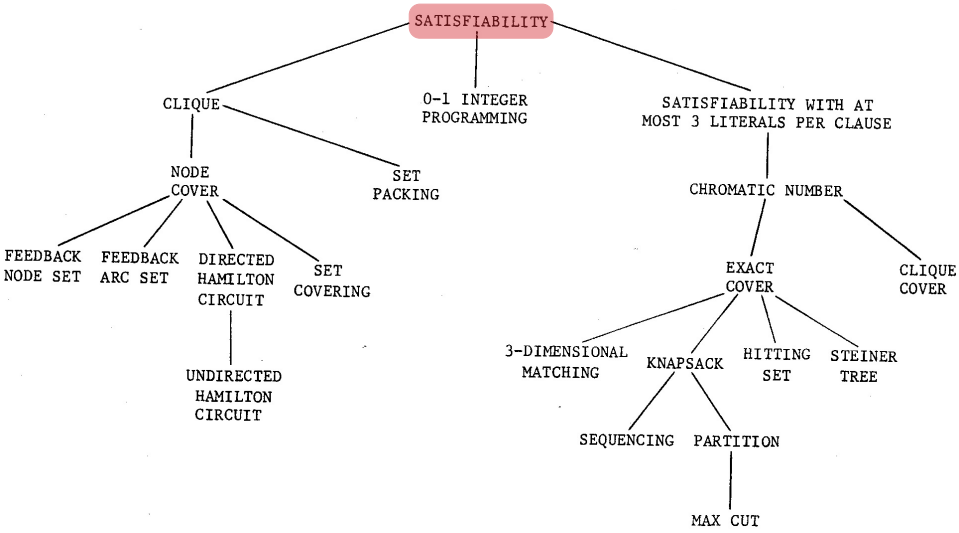
$$F = (x \vee y) \wedge (\overline{y} \vee z) \wedge (z) \wedge (\overline{x})$$

Question:

$$\exists\, \alpha = \left\{ \begin{matrix} x \mapsto ?, \\ y \mapsto ?, \\ z \mapsto ? \end{matrix} \right\} \text{ with } F\restriction_{\alpha} = 1\ ?$$

Karp's 21 NP-Complete Problems

[Kar72]



Two Approaches to Investigate SAT

The Bad News

Proof Complexity

- All: exponential runtime in worst-case
- Power and limits
- Upper & lower bounds

The Good News

Advances in Engineering

- Applied researchers don't care
- Solvers handle millions of variables
- Myth: "NP problems are hard"

Proof Complexity

Bad News in the Form of Lower Bounds

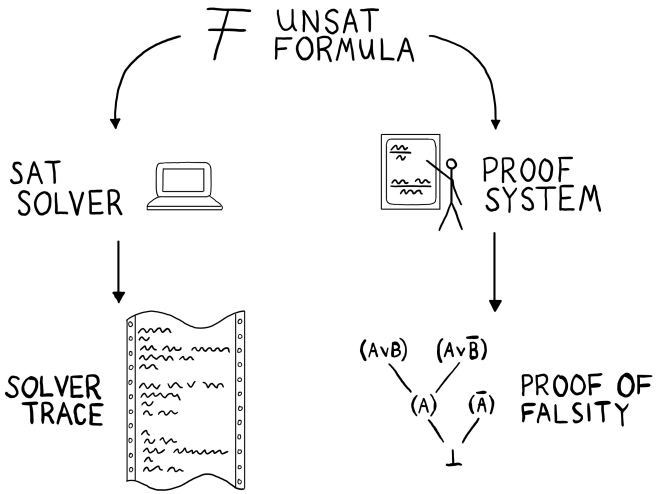
Abstract Models To Investigate SAT

Q: How to capture the **essence** of solvers
via a simple yet powerful mathematical abstraction?

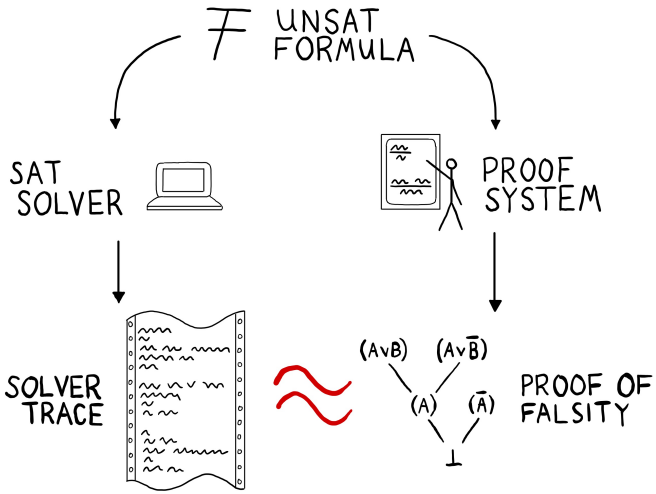
Abstraction can:

- enable **deeper understanding** of internals of solvers
- enable ease of **theoretical analysis**
- enable **better solver design**

Proof Systems



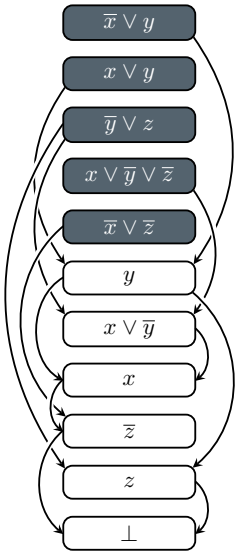
Proof Systems



The Proof System Resolution

Resolution Rule

$$\frac{A \vee x \qquad B \vee \overline{x}}{A \vee B}$$



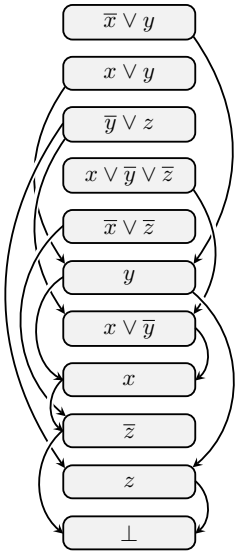
Complexity Measures for a Resolution Proof π

Size

clauses

Width

literals in largest clause



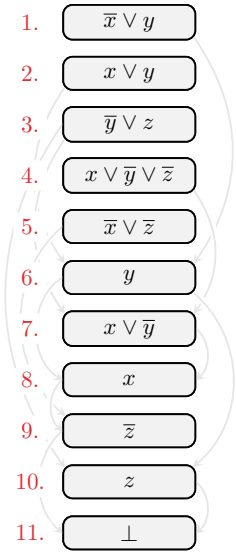
Complexity Measures for a Resolution Proof π

Size

clauses (here: 11)

Width

literals in largest clause



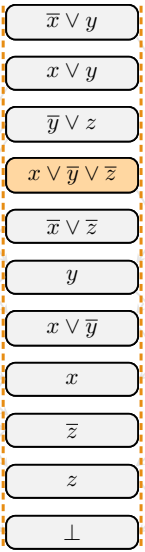
Complexity Measures for a Resolution Proof π

Size

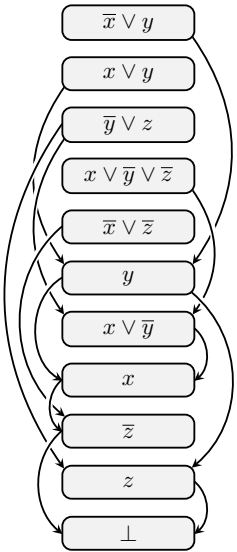
clauses (here: 11)

Width

literals in largest clause (here: 3)



Complexity Measures for Refuting a Formula F

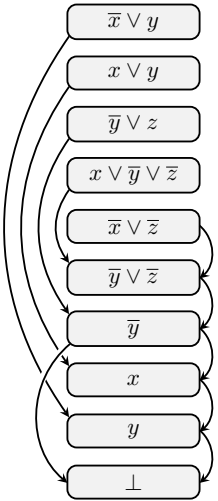


Fix a formula F.

For each complexity measure $\mathscr{C}$:

Take minimum over all refutations π of F

$$\mathscr{C}(F \vdash \bot) := \min_{\pi : F \vdash \bot} \mathscr{C}(\pi)$$



The [BW01] Result: A Connection Between Size & Width

Easy Observation

A **narrow** resolution refutation is necessarily **short**.

Proof. $\mathrm{Width}(\pi) \leq w \implies$ No. of possible clauses $\leq \sum_{i=0}^{w} \binom{|\mathrm{Vars}(F)|}{i} 2^i \leq n^{\mathrm{O}(k)}$. □

The [BW01] Result: A Connection Between Size & Width

Easy Observation

A **narrow** resolution refutation is necessarily **short**.

Proof. $\mathrm{Width}(\pi) \leq w \implies$ No. of possible clauses $\leq \sum_{i=0}^{w} \binom{|\mathrm{Vars}(F)|}{i} 2^i \leq n^{\mathrm{O}(k)}$. □

Sort of a Converse is True

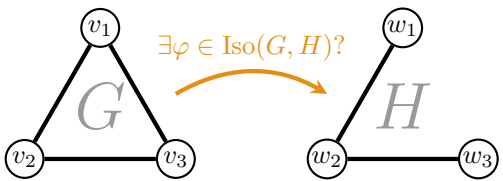
"**Short proofs are narrow**": $\mathrm{Size}\big(F \vdash \bot\big) \geq \exp\left(\Omega\left(\frac{\Big[\mathrm{Width}(F \vdash \bot) - \mathrm{Width}(F)\Big]^2}{|\mathrm{Vars}(F)|}\right)\right)$.

The Complexity of Graph Isomorphism

First Act: Resolution

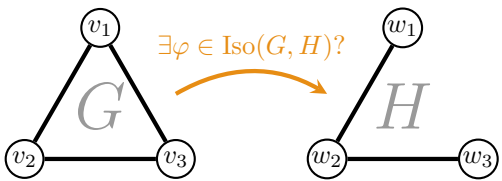
Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \iff v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

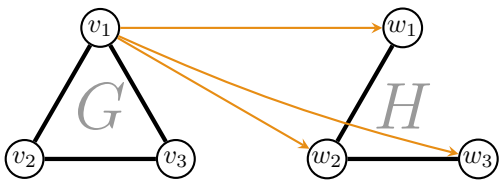
$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

- **Type 3 clauses:** adjacency relation

$$\forall i < j \text{ and } k \neq \ell \text{ with}$$
$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \iff v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

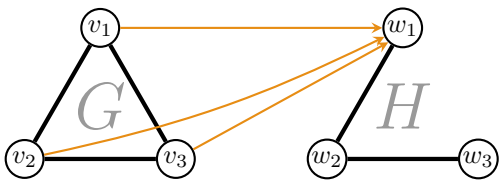
$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

- **Type 3 clauses:** adjacency relation

$$\forall i < j \text{ and } k \neq \ell \text{ with}$$
$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \iff v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

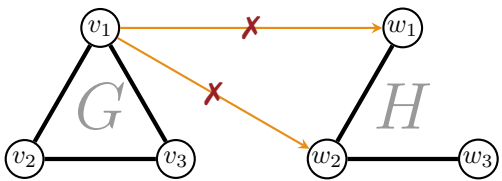
$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

- **Type 3 clauses:** adjacency relation

$$\forall i < j \text{ and } k \neq \ell \text{ with}$$
$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \iff v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

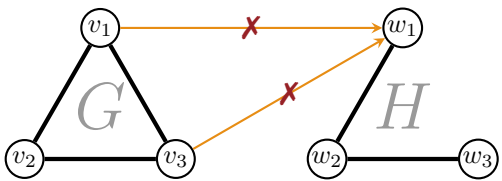
$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

- **Type 3 clauses:** adjacency relation

$$\forall i < j \text{ and } k \neq \ell \text{ with}$$
$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \iff v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

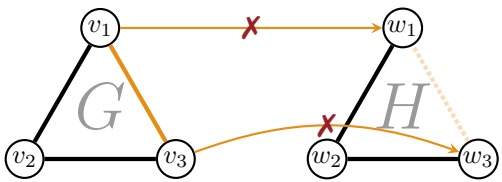
$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

- **Type 3 clauses:** adjacency relation

$$\forall i < j \text{ and } k \neq \ell \text{ with}$$
$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \iff v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

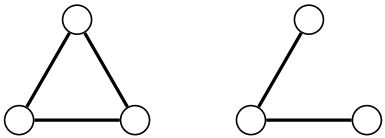
- **Type 2 clauses:** function + injective

$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

- **Type 3 clauses:** adjacency relation

$$\forall i < j \text{ and } k \neq \ell \text{ with}$$
$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

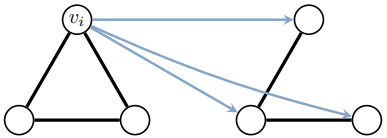
Problem When Using [BW01] for ISO-Formulas



First Lower Bound Attempt

Just use $\mathrm{Size}\big(F \vdash \bot\big) \geq \exp\left(\Omega\left(\frac{\left[\mathrm{Width}(F \vdash \bot) - \mathrm{Width}(F)\right]^2}{|\mathrm{Vars}(F)|}\right)\right)$?

Problem When Using [BW01] for ISO-Formulas



$$(x_{i,1} \vee \ldots \vee x_{i,n})$$

First Lower Bound Attempt

Just use $\mathrm{Size}(F \vdash \bot) \geq \exp\left(\Omega\left(\frac{\left[\mathrm{Width}(F \vdash \bot) - \mathrm{Width}(F)\right]^2}{|\mathrm{Vars}(F)|}\right)\right)$?

Problem! $\mathrm{Width}(F) = n$

$\mathrm{Width}(F \vdash \bot) = \mathrm{O}(n)$

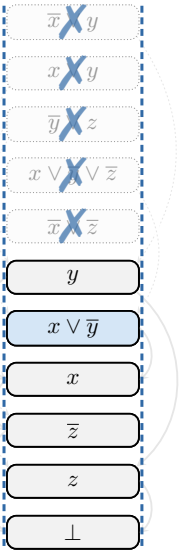
Idea: Use Narrow Resolution

Distinction by Cases Rule [GT05]

$$\frac{A_1 \vee \overline{\ell_1} \quad \ldots \quad A_m \vee \overline{\ell_m}}{A_1 \vee \cdots \vee A_m \vee B} \quad \text{if} \quad (B \vee \ell_1 \vee \cdots \vee \ell_m) \in F$$

Narrow Width

exclude all axioms in the count (here: 2)



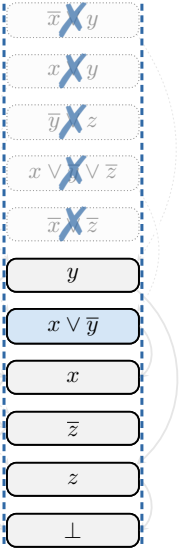
Idea: Use Narrow Resolution

Distinction by Cases Rule [GT05]

$$\frac{A_1 \vee \overline{\ell_1} \quad \ldots \quad A_m \vee \overline{\ell_m}}{A_1 \vee \cdots \vee A_m \vee B} \quad \text{if} \quad (B \vee \ell_1 \vee \cdots \vee \ell_m) \in F$$

① [BW01] for narrow width?
② How to get bounds?

Narrow Width
exclude all axioms in the count (here: 2)



Common Technique in Proof Complexity: Use Games!

Spoiler and Duplicator compete in the k-Witnessing Game on the formula $\mathrm{ISO}(G, H)$

- Game state is a partial assignment, initially $\alpha_0 = \varepsilon$
- In each round j

  Spoiler: Chooses a subset $\alpha' \subseteq \alpha_{j-1}$ of size at most $k - 1$
  Chooses a Type 1 clause C in $\mathrm{ISO}(G, H)$, say $(x_{i,1} \vee \cdots \vee x_{i,n})$

  Duplicator: Extends $\alpha_j := \alpha' \cup \{\ell = 1\}$ for some literal $\ell \in C$

- Game ends when Duplicator cannot extend such that
  - α_j satisfies C and
  - does not falsify any other clause in $\mathrm{ISO}(G, H)$

Main Result: Connection Between FO and PC

Narrow Width Refutation
of $\mathrm{ISO}(G, H)$

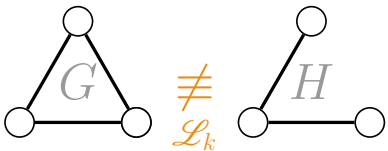
$$\text{N-Width}(\pi) \leq k - 1$$

k-Witnessing Game

Spoiler wins on $\mathrm{ISO}(G, H)$

Main Result: Connection Between FO and PC

Immerman's Pebble Game
on G and H



$\not\equiv_{\mathscr{L}_k}$

Narrow Width Refutation
of $\mathrm{ISO}(G, H)$

$$\text{N-Width}(\pi) \leq k - 1$$

$\Updownarrow$

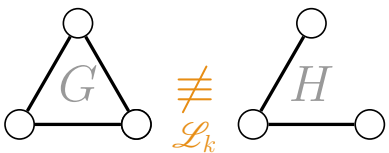
$\Leftrightarrow$

k-Witnessing Game

Spoiler wins on $\mathrm{ISO}(G, H)$

Main Result: Connection Between FO and PC

Immerman's Pebble Game
on G and H



Narrow Width Refutation
of $\mathrm{ISO}(G, H)$

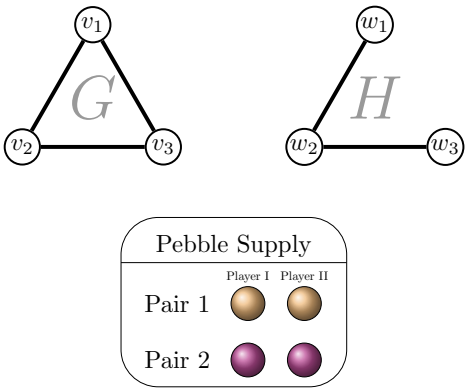
$$\text{N-Width}(\pi) \leq k - 1$$

k-Witnessing Game

Spoiler wins on $\mathrm{ISO}(G, H)$

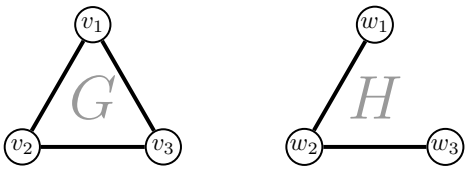
Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

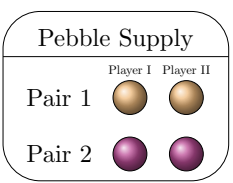
- Player I and Player II have k pebble pairs



Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

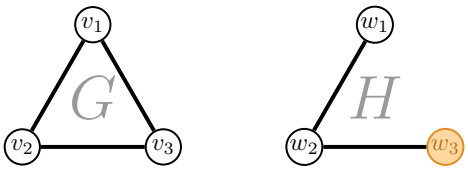
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph

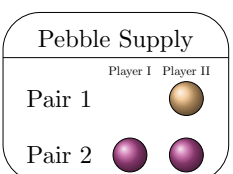




Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

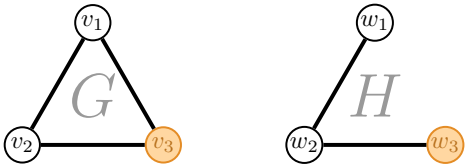
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph

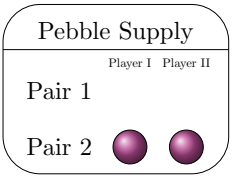




Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

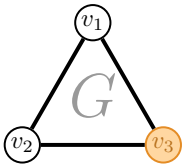
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**

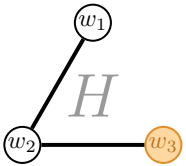


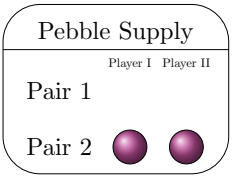


Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic

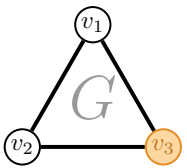


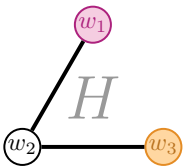


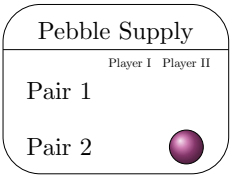


Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic

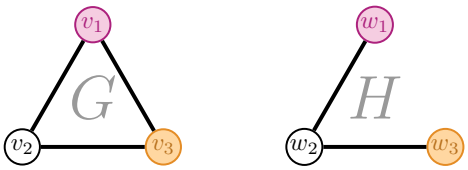


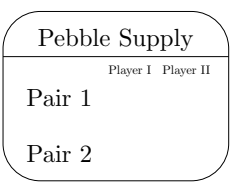




Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic ✗ Player I won!



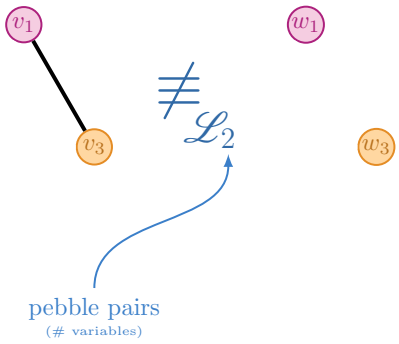


Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic ✗ Player I won!

Immerman's k-Pebble Game: Player I Wants to Show $G \ncong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic ✗ Player I won!



Resolution Is Strong Enough to Simulate Immerman's Game

$$G \not\equiv_{\mathscr{L}_k} H \implies \mathrm{Size}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \leq n^{\mathrm{O}(k)}$$

Basic Idea for Upper Bound

- $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \leq k-1$
  $\implies \text{N-Size}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \leq \sum_{i=0}^{k-1} \binom{n^2}{i} 2^i \leq n^{\mathrm{O}(k)}$
- Simulate a Narrow-step in at most n Res-steps

Application: Automated $\overline{\mathrm{GI}}$-Theorem Proving

Algorithm 1: Automated Graph Non-isomorphism Prover

Input: $\mathrm{ISO}(G, H)$, **Promise:** $G \ncong H$

$k \leftarrow 1$

Repeat

  Derive all resolvents derivable in narrow width k

  If $\perp$ *was derived* **then output** "non-isomorphic"

  $k \leftarrow k + 1$

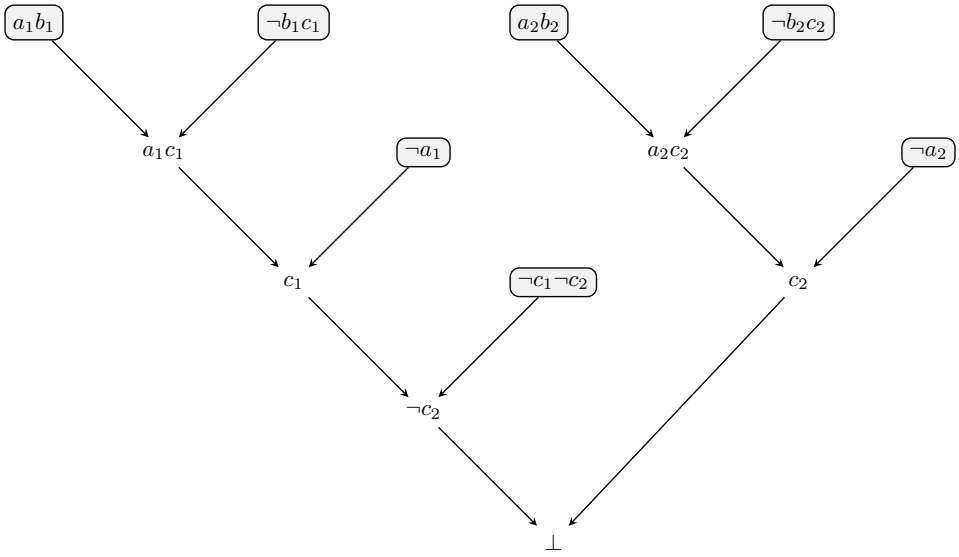
Running Time for Constant k:

$$n^{\mathrm{O}(k)} = \mathrm{poly}(n)$$

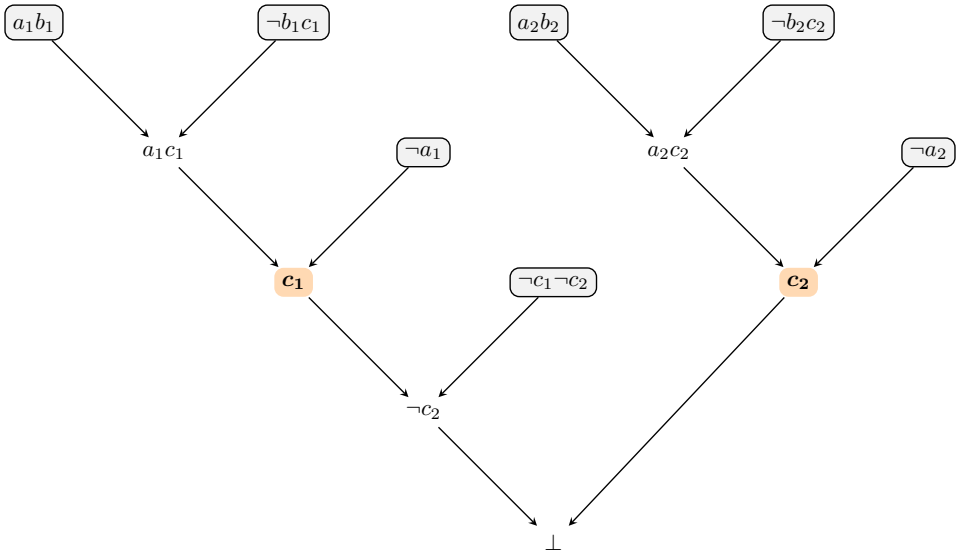
The Complexity of Graph Isomorphism

Second Act: Lower Bounds for Stronger Proof Systems

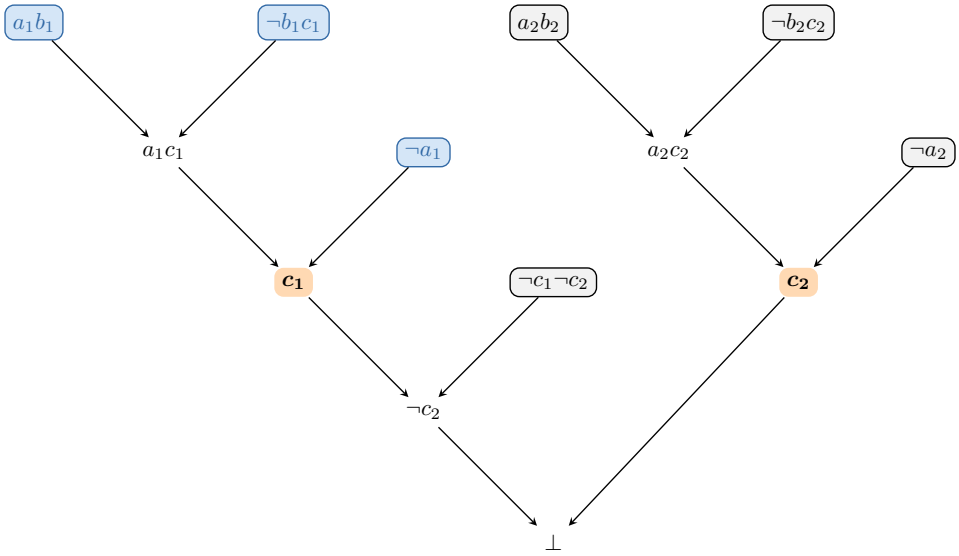
Extending Resolution: Krishnamurthy's Symmetry Rules



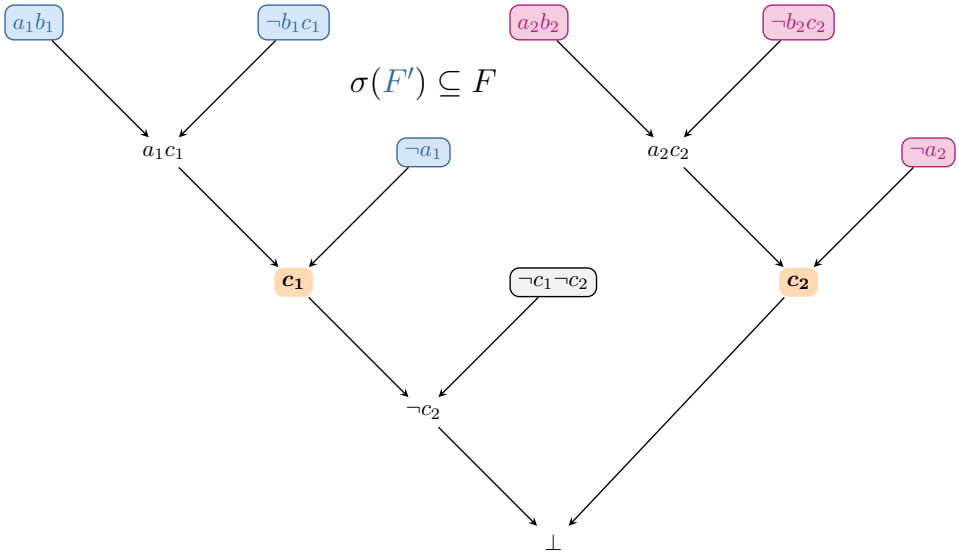
Extending Resolution: Krishnamurthy's Symmetry Rules



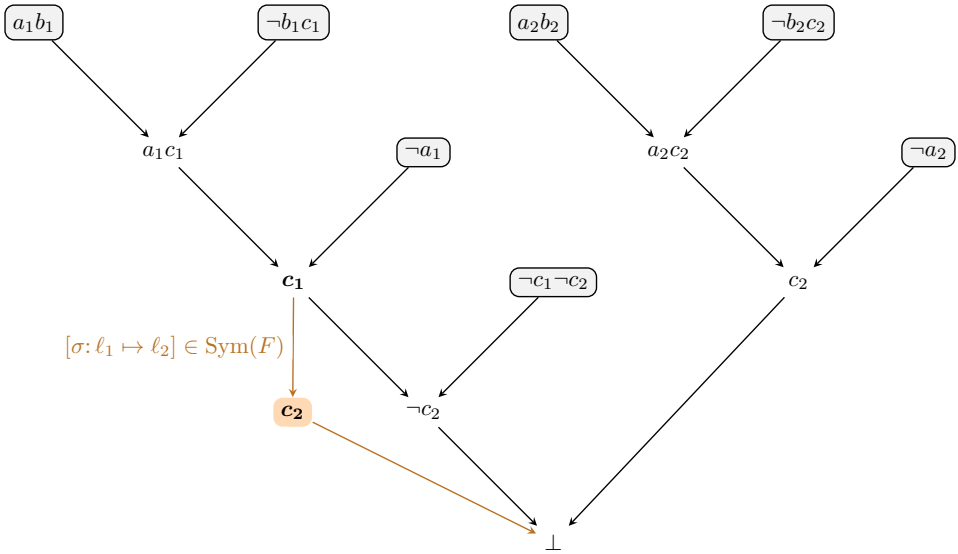
Extending Resolution: Krishnamurthy's Symmetry Rules



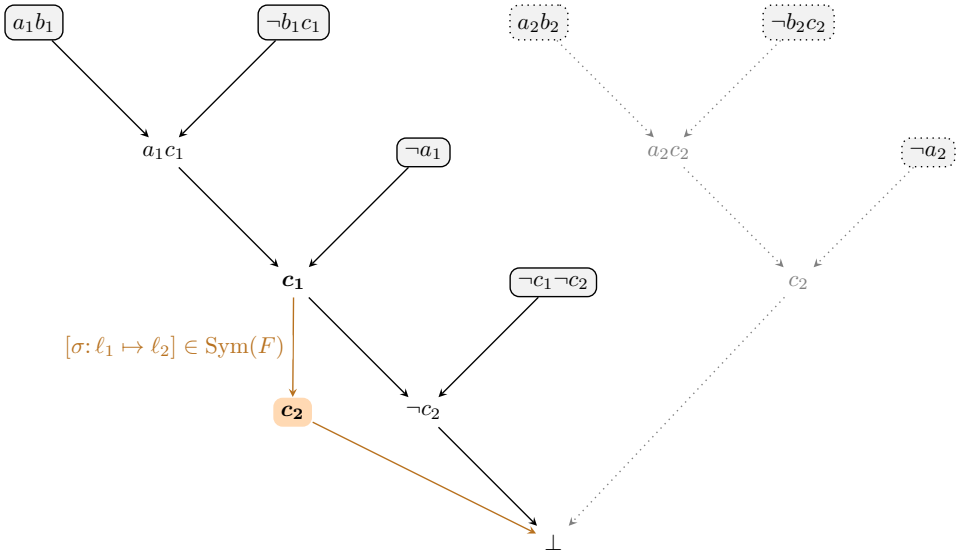
Extending Resolution: Krishnamurthy's Symmetry Rules



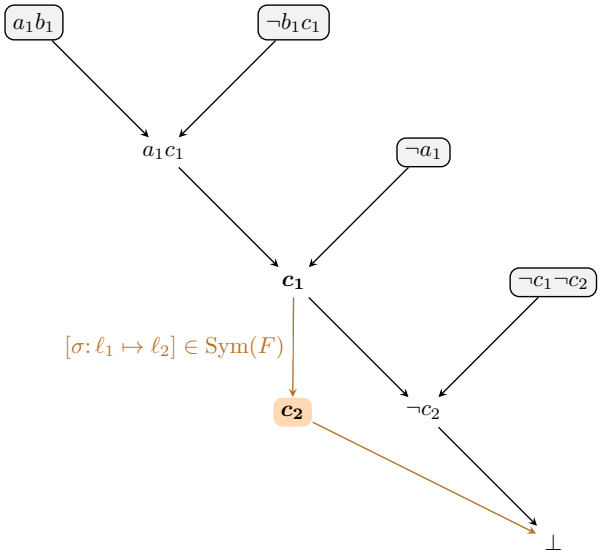
Extending Resolution: Krishnamurthy's Symmetry Rules



Extending Resolution: Krishnamurthy's Symmetry Rules



Extending Resolution: Krishnamurthy's Symmetry Rules



Is $\mathrm{GI} \in$ co-NP?

Cook & Reckhow '79

$$\mathrm{GI} \in \text{co-NP} \iff \exists \mathscr{S} : \mathscr{S}\text{-Size}\big(\mathrm{GI} \vdash \bot\big) \leq \mathrm{poly}(n)$$

Proof System	Symmetry Rule	Known Bounds	
SRC-2	local	only $\mathrm{O}\big(\mathrm{poly}(n)\big)$ known	[SS21]
SRC-1	global	?	
Res	none	$\exp\big(\Omega(n)\big)$	[Tor13]

Battle SRC-1 With Asymmetric Graphs

Asymmetric Graph G: $\mathrm{Aut}(G) = \{\mathrm{id}\}$

Battle SRC-1 With Asymmetric Graphs

Asymmetric Graph G: $\mathrm{Aut}(G) = \{\mathrm{id}\}$

Lemma: Asymmetric graphs $\implies$ Asymmetric ISO-formula

Battle SRC-1 With Asymmetric Graphs

Asymmetric Graph G:	$\mathrm{Aut}(G) = \{\mathrm{id}\}$
Lemma:	Asymmetric graphs $\implies$ Asymmetric ISO-formula
Lemma:	Asymmetric formula $\implies$ Res-Size $=$ SRC-1-Size [Szeider]

Asymmetric Graphs With Large Weisfeiler–Leman-Dimension

Without looking at ISO-formula:

$$(G,\lambda) \equiv_{\mathscr{L}_k} (H,\mu) \implies \operatorname{Size}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \geq \exp\left(\Omega\left(\frac{[\text{N-Width}(\mathrm{ISO}(G,H)\vdash\bot)]^2}{\text{sum of color class sizes}}\right)\right)$$

[Dawar and Khan] showed: There are pairs of non-isomorphic graphs that are

- asymmetric (unlike CFI-graphs)
- have small size $\mathrm{O}(k)$
- with large WL-dim k
- and color classes of size 4

Result: An Exponential GI Lower Bound for SRC-1

Our Result:

There is a family of non-isomorphic graph pairs (G_n, H_n)

- with $\mathrm{O}(n)$ vertices each,
- such that any SRC-1 refutation of $\mathrm{ISO}(G_n, H_n)$ requires

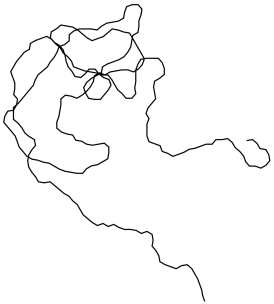
$$\text{size } \exp\big(\Omega(n)\big).$$

A Practitioner's View on SAT

Some Good News

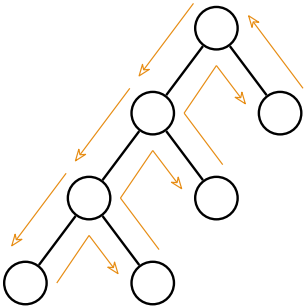
Two Solver Paradigms

SLS $\widehat{=}$ random exploration



+ Excels at random instances
− Can get stuck
− Incomplete

DPLL $\widehat{=}$ intelligent, systematic search



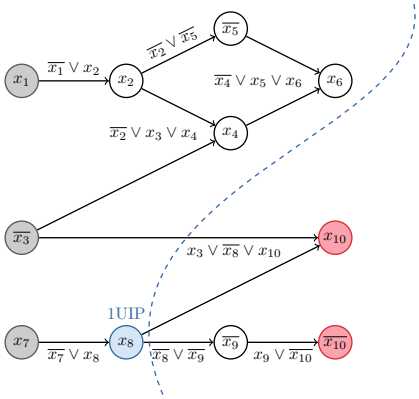
+ Well suited for application instances
+ Complete solver
− Complicated to analyze

Ten Challenges in Propositional Reasoning and Search

> *Demonstrate the successful combination of* **stochastic search** *and* **systematic search** *techniques, by the creation of a new algorithm that outperforms the best previous examples of both approaches.*

Selman, Kautz, McAllester; *Proc. IJCAI 1997*

Our Idea: Terraform the Landscape for SLS



Learn clause set L

```
p cnf 12 22
    1 2 3 0
    4 5 6 0
    7 8 9 0
 10 11 12 0
   -1 -4 0
   -1 -7 0
  -1 -10 0
   -4 -7 0
  -4 -10 0
  -7 -10 0
   -2 -5 0
   -2 -8 0
  -2 -11 0
   -5 -8 0
  -5 -11 0
  -8 -11 0
   -3 -6 0
   -3 -9 0
  -3 -12 0
   -6 -9 0
  -6 -12 0
  -9 -12 0
 10 11 1 0
  -6 7 1 0
    3 -8 0
```

Thought Experiment: Are all Implied Clauses Created Equal?

Backbone for satisfiable formula

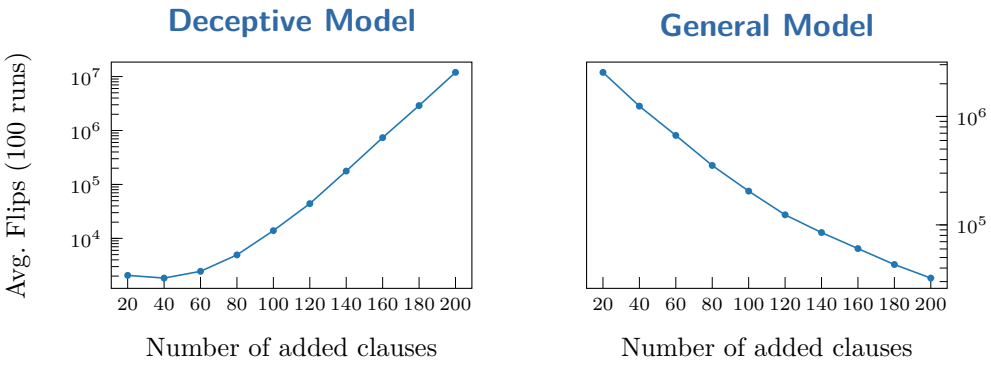
The **backbone** $\mathscr{B}(F)$ is the set of literals appearing in all satisfying assignments of F:

$$\mathscr{B}(F) \coloneqq \bigcap_{\alpha\colon F\restriction_\alpha = 1} \alpha.$$

Deceptive model: $(x \vee \overline{y} \vee \overline{z})$, where $x, y, z \in \mathscr{B}(F)$

General model: $(x \vee y \vee z)$, where $x \in \mathscr{B}(F)$ and $y, z \in \mathrm{Vars}(F)$

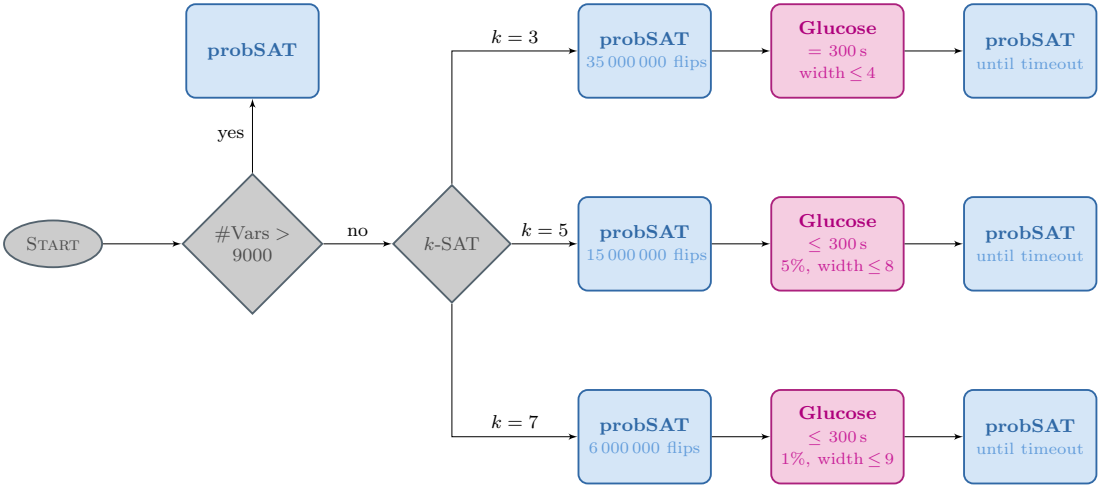
Effect of the Models



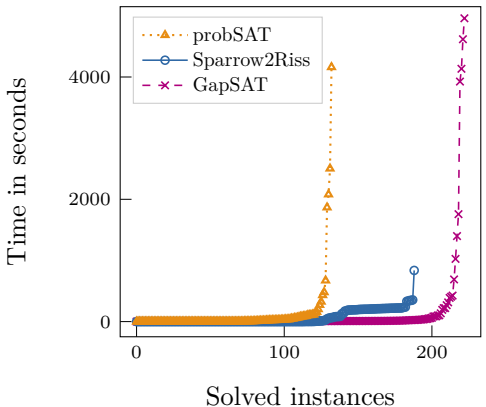
Effect of the Models

But: Both models are unrealistic.

GapSAT

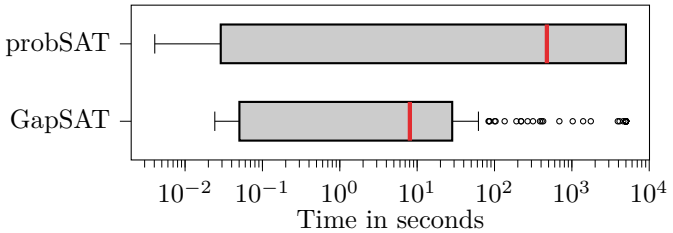

probSAT
yes
Start
#Vars > 9000
no
k-SAT
k = 3
k = 5
k = 7
probSAT
35 000 000 flips
Glucose
= 300 s
width ≤ 4
probSAT
until timeout
probSAT
15 000 000 flips
Glucose
≤ 300 s
5%, width ≤ 8
probSAT
until timeout
probSAT
6 000 000 flips
Glucose
≤ 300 s
1%, width ≤ 9
probSAT
until timeout

Our Very Own SAT Contest



	Instances solved	Score
probSAT	133	1 234 986
Sparrow2Riss	189	672 336
GapSAT	223	347 156

Improvement of Several Orders of Magnitude—but Outliers



Further Investigation:

Runtime Distributions and Restarts

Aim: Model modification process of GapSAT more generally
Analyze hardness of logically equivalent formulas

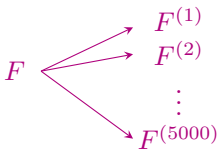
Abstraction: Adjusted Logical Formula Algorithm ALFA

Algorithm 2: Adjusted Logical Formula Algorithm (ALFA)

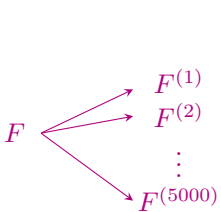
Input: Boolean formula F, **Promise:** $F \in \mathrm{SAT}$

Generate **randomly** a set L of clauses such that $F \models L$
Call $\mathrm{SLS}(F \cup L)$ for some SLS solver SLS

Experimental Setup



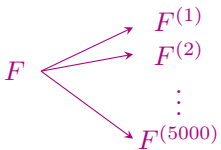
Experimental Setup



solve 100 times with each solver S

$\mathsf{flips}_S(F^{(1)}, s_1), \quad \ldots, \quad \mathsf{flips}_S(F^{(1)}, s_{100})$

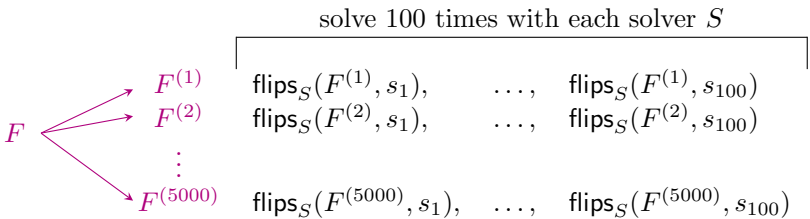
Experimental Setup



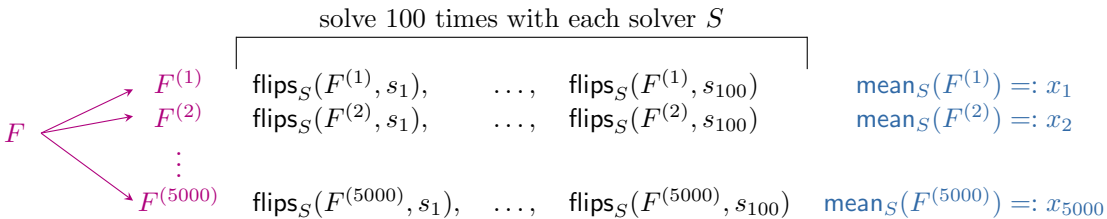
solve 100 times with each solver S

$$\begin{array}{llll} \mathsf{flips}_S(F^{(1)}, s_1), & \ldots, & \mathsf{flips}_S(F^{(1)}, s_{100}) \\ \mathsf{flips}_S(F^{(2)}, s_1), & \ldots, & \mathsf{flips}_S(F^{(2)}, s_{100}) \end{array}$$

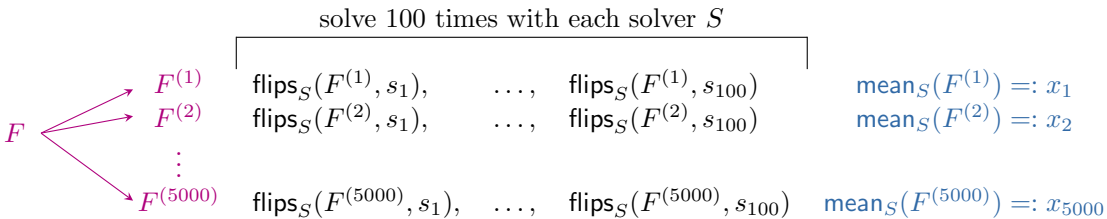
Experimental Setup



Experimental Setup



Experimental Setup



Given such a sample $(x_1, \ldots, x_{5000})$, plot the empirical distribution function

$$\widehat{F}_{5000}(t) := \frac{1}{5000} \sum_{i=1}^{5000} \mathbb{1}_{\{x_i \leq t\}}, \quad t \in \mathbb{R}.$$

Instance Types and Solvers Used

Instance Types:

1. Hidden Solution (different parameters)
2. Uniform Random
3. Factoring
4. Coloring

Used Solvers:

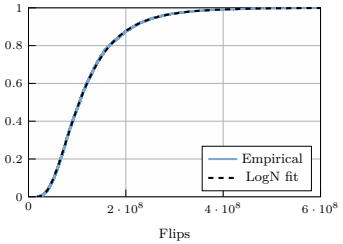
1. SRWA
2. probSAT solver family
3. YalSAT

Total CPU time:

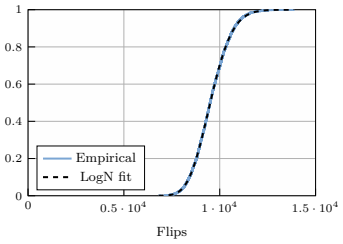
80 years!

Experimental Results and Statistical Evaluation

Type A Instances

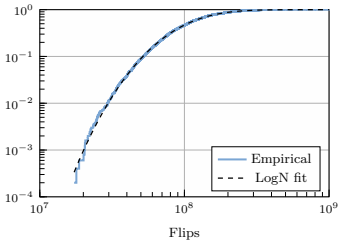


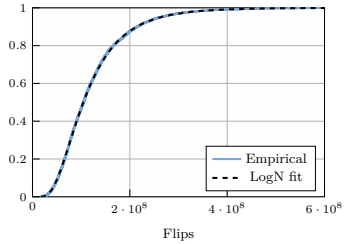
Type B Instances



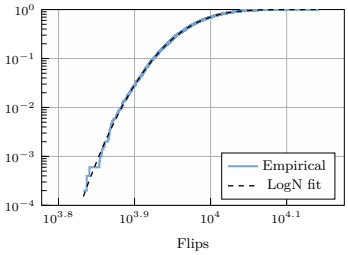
Experimental Results and Statistical Evaluation

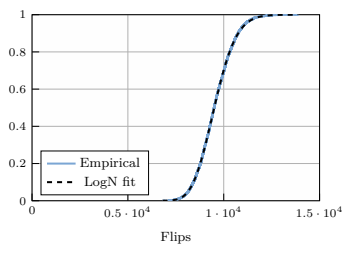
Type A Instances





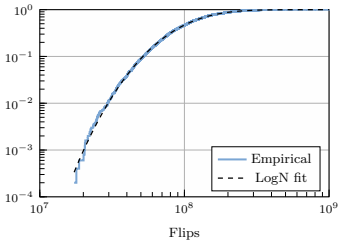
Type B Instances

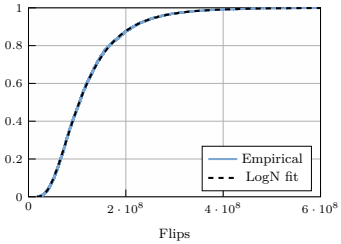


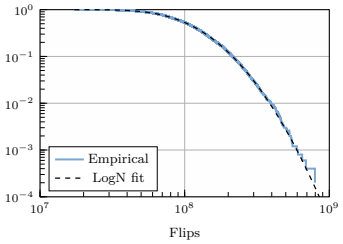


Experimental Results and Statistical Evaluation

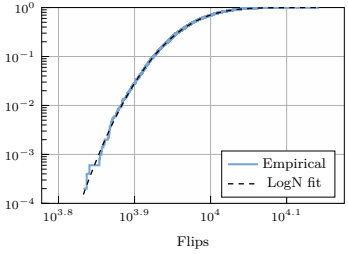
Type A Instances

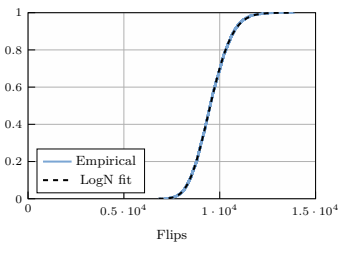


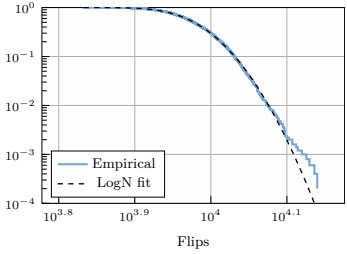




Type B Instances







Conjectures

Strong Conjecture

The runtimes of ALFA-algorithms follow **lognormal** distributions.

Conjectures

Strong Conjecture

The runtimes of ALFA-algorithms follow **lognormal** distributions.

Definition ([FKZ11])

A positive, real-valued random variable X is **long-tailed**, if and only if

$$\forall x \in \mathbb{R}^+ : \mathbb{P}[X > x] > 0 \quad \text{and} \quad \forall y \in \mathbb{R}^+ : \lim_{x \to \infty} \frac{\mathbb{P}[X > x + y]}{\mathbb{P}[X > x]} = 1.$$

Weak Conjecture

The runtimes of ALFA-algorithms follow **long-tailed** distributions.

Usefulness of Restarts

Consequence of the Strong Conjecture

Strong Conjecture (runtimes are lognormally distributed) $\overset{[\text{Lor18}]}{\Longrightarrow}$ **Restarts are useful**

Usefulness of Restarts

Consequence of the Strong Conjecture

Strong Conjecture (runtimes are lognormally distributed) $\overset{[\text{Lor18}]}{\Longrightarrow}$ **Restarts are useful**

New Mathematical Result:

Consequence of the Weak Conjecture

Weak Conjecture (runtimes are long-tailed distributed) $\Longrightarrow$ **Restarts are useful**

Contributions

List of Own Publications — Proof Complexity

Number of Variables for Graph Identification and the Resolution of GI Formulas
J. Torán and F. Wörz
– Journal version: Submitted to *ACM Transactions on Computational Logic (ACM TOCL)*
– Conference version: *EACSL Conference on Computer Science Logic (CSL)*, 2022

Reversible Pebble Games and the Relation Between Tree-Like and General Resolution Space
J. Torán and F. Wörz
– Journal version: *Computational Complexity* (2021)
– Conference version: *Int. Symposium on Theor. Aspects of Computer Science (STACS)*, 2020

List of Own Publications — Experiments

Towards an Understanding of Long-Tailed Runtimes
J.-H. Lorenz and F. Wörz
– Journal version: Accepted in *ACM Journal of Experimental Algorithmics (ACM JEA)*, 2022

Too Much Information: Why CDCL Solvers Need to Forget Learned Clauses
T. Krüger, J.-H. Lorenz, and F. Wörz
– Journal version: *PLOS ONE* (2022)

Evidence for Long-Tails in SLS Algorithms
F. Wörz and J.-H. Lorenz
– Conference version: *European Symposium on Algorithms (ESA)*, 2021
– Best Student Paper awarded by European Association for Theor. Comp. Science (EATCS)

On the Effect of Learned Clauses on Stochastic Local Search
J.-H. Lorenz and F. Wörz
– Conference version: *Theory and Applications of Satisfiability Testing (SAT)*, 2020

Some of Our Contributions

Proof Complexity:

- Systematic study of CS vs. Tree-CS
- Complexity of Graph Isomorphism in different proof systems

Contributions to Applied SAT Solving:

- Construction of a novel hybrid solver GapSAT
- Study of long-tails and proof of usefulness of restarts
- Theoretical runtime distribution analysis of Schöning's random walk
- Clause deletion and multimodality in CDCL solvers