SAT with Theories for System Security Verification

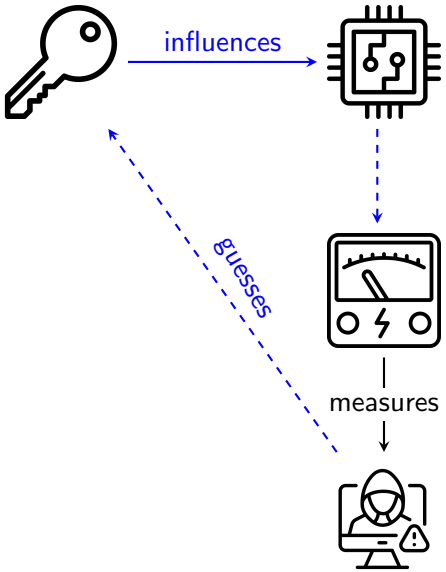
PhD Proficiency Presentation

Robin Coutelier

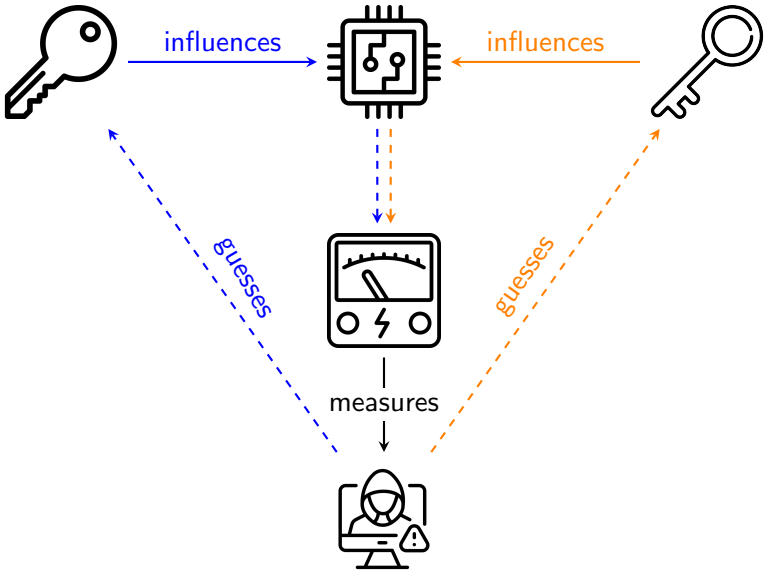
TU Wien, Vienna, Austria
robin.coutelier@tuwien.ac.at

June 17th 2025

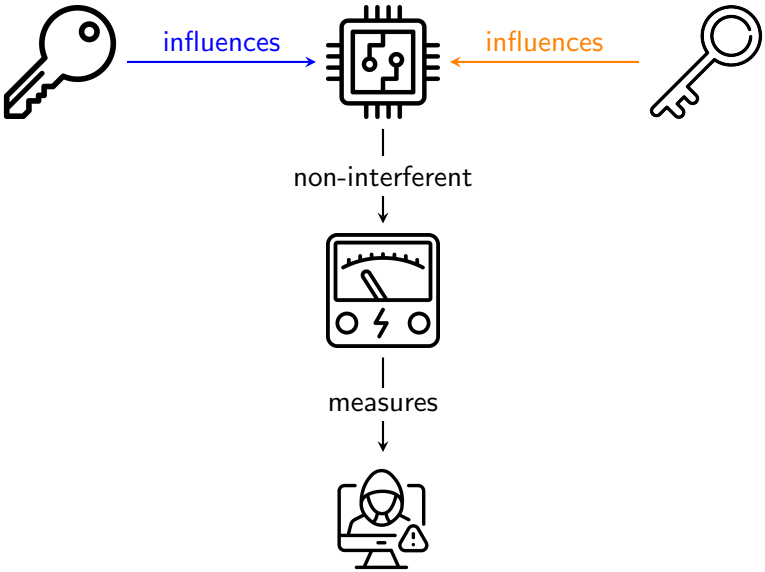
Side-Channel Attacks



Side-Channel Attacks



Side-Channel Attacks



Shares for Secret Masking

Secret Masking

Let s be a secret, we split the secret into n shares $s_1, s_2, \ldots, s_n$ such that:

$$s = s_1 \oplus s_2 \oplus \ldots \oplus s_n$$

Shares for Secret Masking

Secret Masking

Let s be a secret, we split the secret into n shares $s_1, s_2, \ldots, s_n$ such that:

$$s = s_1 \oplus s_2 \oplus \ldots \oplus s_n$$

$\mathbf{s}$	$\mathbf{s_1}$	$\mu\mathbf{P}$
0	0	$0W$
1	1	$1W$

Shares for Secret Masking

Secret Masking

Let s be a secret, we split the secret into n shares $s_1, s_2, \ldots, s_n$ such that:

$$s = s_1 \oplus s_2 \oplus \ldots \oplus s_n$$

$\mathbf{s}$	$\mathbf{s_1}$	$\mathbf{s_2}$	$\mu\mathbf{P}$	$\mu^\mathbf{2}\mathbf{P}$
0	0	0	$1W$	$1W^2$
0	1	1		
1	0	1	$1W$	$0W^2$
1	1	0		

Shares for Secret Masking

Secret Masking

Let s be a secret, we split the secret into n shares $s_1, s_2, \ldots, s_n$ such that:

$$s = s_1 \oplus s_2 \oplus \ldots \oplus s_n$$

$\mathbf{s}$	$\mathbf{s_1}$	$\mathbf{s_2}$	$\mathbf{s_3}$	$\mu\mathbf{P}$	$\mu^\mathbf{2}\mathbf{P}$	$\mu^\mathbf{3}\mathbf{P}$
0	0	0	0	$1.5W$	$0.75W^2$	$-0.75W^3$
0	1	1	0			
0	1	0	1			
0	0	1	1			
1	0	0	1	$1.5W$	$0.75W^2$	$0.75W^3$
1	1	1	1			
1	0	1	0			
1	1	0	0			

Probe-Isolating Non-Interference

Abstract Goal

$$\forall s, s'.\ P(power|s) \sim P(power|s')$$

Probe-Isolating Non-Interference

Abstract Goal

$$\forall s, s'.\ P(power|s) \sim P(power|s')$$

Probe isolation - Idealized attacker model

The attacker can sample t wires in the circuit.
Can the attacker obtain information about the secret s?

Probe-Isolating Non-Interference

Abstract Goal

$$\forall s, s'.\ P(power|s) \sim P(power|s')$$

Probe isolation - Idealized attacker model

The attacker can sample t wires in the circuit.
Can the attacker obtain information about the secret s?

Practical Goal

$$\bigwedge_{W \in \mathcal{C}} \forall w, s, s'.\ probe(W) \Rightarrow P(W = w|S = s) = P(W = w|S = s')$$

Probe-Isolating Non-Interference

Abstract Goal

$$\forall s, s'.\ P(power|s) \sim P(power|s')$$

Probe isolation - Idealized attacker model

The attacker can sample t wires in the circuit.
Can the attacker obtain information about the secret s?

Practical Goal

$$\bigwedge_{W \in \mathcal{C}} \forall w, s.\ probe(W) \Rightarrow P(W = w|S = s) = P(W = w)$$

Probe-Isolating Non-Interference

Abstract Goal

$$\forall s, s'.\ P(power|s) \sim P(power|s')$$

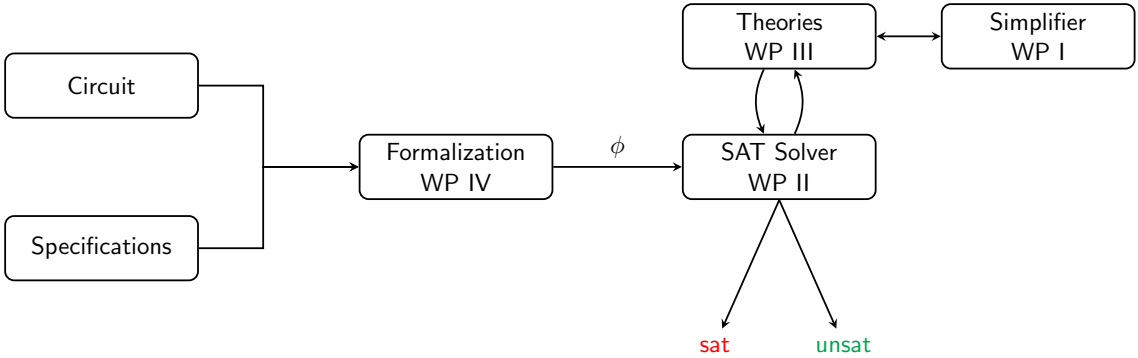
Probe isolation - Idealized attacker model

The attacker can sample t wires in the circuit.
Can the attacker obtain information about the secret s?

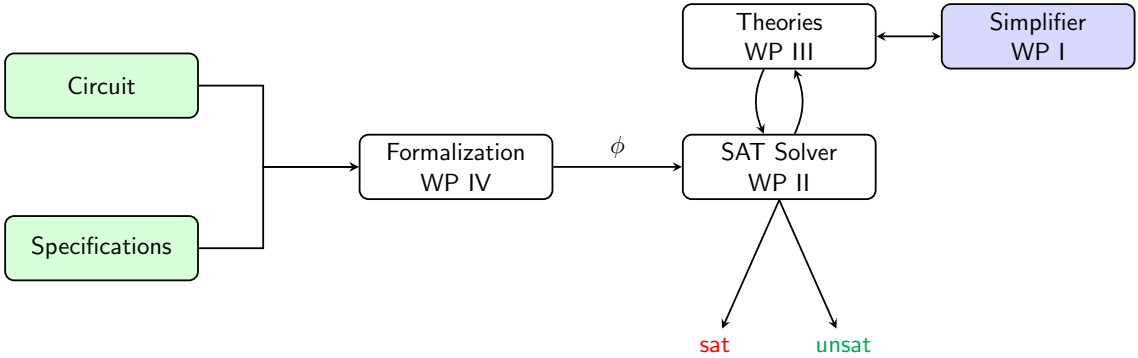
Practical Goal (simplified)

$$\bigwedge_{W \in \mathcal{C}} \forall w, s_1, \ldots, s_n.\ probe(W) \Rightarrow \bigvee_{i=1}^{n} P(W = w|S_i = s_i) = P(W = w)$$

Verification Pipeline for PINI



WP I: Simplifier



WP I: SAT Solving for Variants of Subsumption

published at FMSD 2024 [CRR+24]

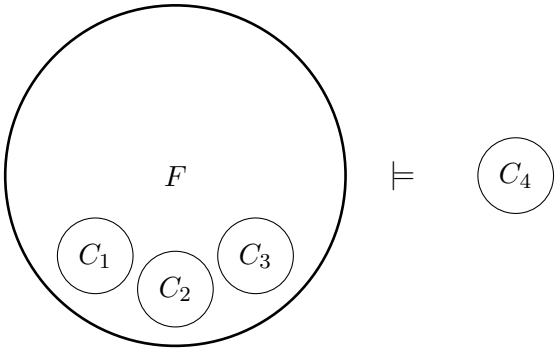
Research Question

How to simplify the search space efficiently?

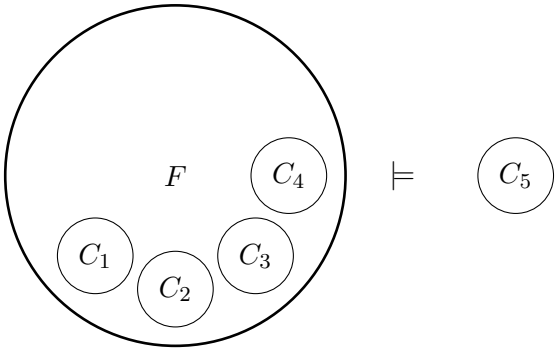
Subsumption and Subsumption Resolution

Subsumption removes redundant clauses from the search space.
Subsumption Resolution (SR) removes a redundant literal from a clause.

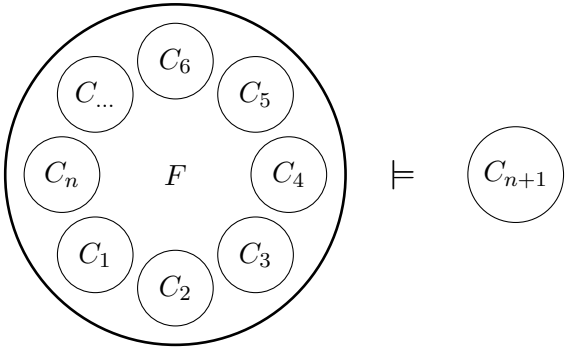
WP I: Saturation in FOL Theorem Proving



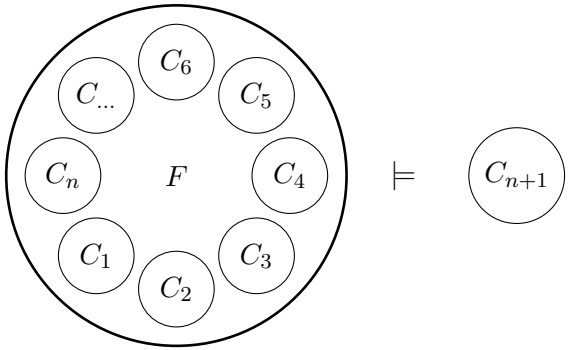
WP I: Saturation in FOL Theorem Proving



WP I: Saturation in FOL Theorem Proving

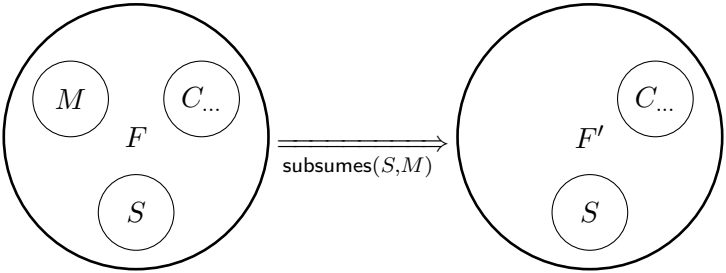


WP I: Saturation in FOL Theorem Proving

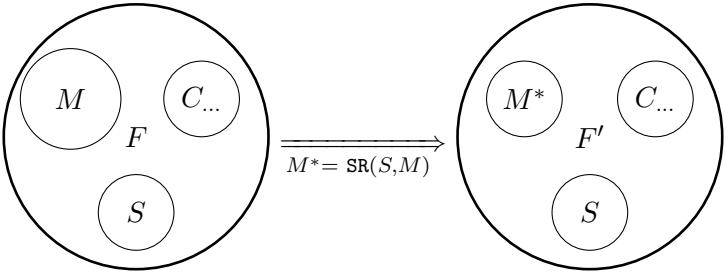


Out of memory!

WP I: Subsumption - Intuition



WP I: Subsumption Resolution - Intuition



WP I: Two Encodings

Direct Encoding $\mathcal{E}^d_{\text{SR}}(\Pi)$

positive compatibility $\bigwedge_i \bigwedge_j \left(b^+_{i,j} \Rightarrow \Sigma^+_{i,j} \subseteq \sigma\right)$

negative compatibility $\bigwedge_i \bigwedge_j \left(b^-_{i,j} \Rightarrow \Sigma^-_{i,j} \subseteq \sigma\right)$

existence $\bigvee_i \bigvee_j b^-_{i,j}$

uniqueness $\bigwedge_j \bigwedge_i \bigwedge_{i' \geq i} \bigwedge_{j' > j} \neg b^-_{i,j} \vee \neg b^-_{i',j'}$

completeness $\bigwedge_i \bigvee_j b^+_{i,j} \vee b^-_{i,j}$

coherence $\bigwedge_j \bigwedge_i \bigwedge_{i'} \neg b^+_{i,j} \vee \neg b^-_{i',j}$

Complexities

$\mathcal{E}^d_{\text{SR}}(\Pi)$ has $O(|\Pi|)$ variables and $O(|\Pi|^2)$ clauses.

Indirect Encoding $\mathcal{E}^i_{\text{SR}}(\Pi)$

positive compatibility $\bigwedge_i \bigwedge_j \left(b^+_{i,j} \Rightarrow \Sigma^+_{i,j} \subseteq \sigma\right)$

negative compatibility $\bigwedge_i \bigwedge_j \left(b^-_{i,j} \Rightarrow \Sigma^-_{i,j} \subseteq \sigma\right)$

structurality $\bigwedge_j \left[\neg c_j \vee \bigvee_i b^-_{i,j}\right] \wedge \bigwedge_j \bigwedge_i \left(c_j \vee \neg b^-_{i,j}\right)$

revised existence $\bigvee_j c_j$

revised uniqueness $AMO(\{c_j, j = 1, ..., |M|\})$

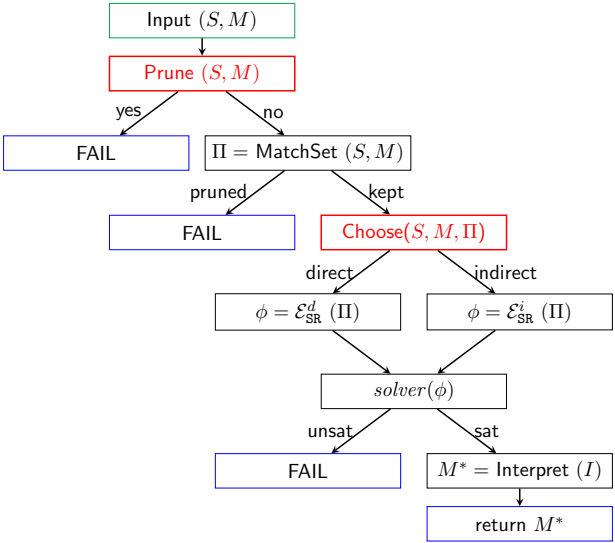
completeness $\bigwedge_i \bigvee_j b^+_{i,j} \vee b^-_{i,j}$

revised coherence $\bigwedge_j \bigwedge_i \left(\neg c_j \vee \neg b^+_{i,j}\right)$

Complexities

$\mathcal{E}^i_{\text{SR}}(\Pi)$ has $O(|\Pi| + |M|)$ variables and $O(|\Pi|)$ clauses.

WP I: SAT-Based Subsumption Resolution



WP I: Improvements since CADE 2023

Prover	Average	Std. Dev.	Boost
VAMPIRE_M	$42.63\,\mu s$	$1609.06\,\mu s$	1.00
VAMPIRE_I^*	$34.55\,\mu s$	$250.25\,\mu s$	1.23

Table: Without Optimization (CADE 2023 [CKRR23])

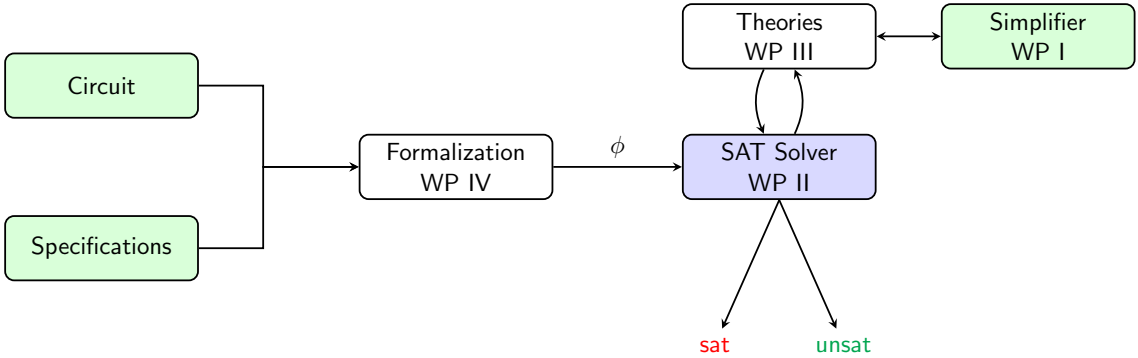
Prover	Average	Std. Dev.	Boost
VAMPIRE_M	33.63 μs	1839.25 μs	1.00
VAMPIRE_H^*	24.73 μs	190.69 μs	1.36

Table: With Optimization (FMSD 2024 [CRR+24])

WP I: Performance on TPTP

Prover	Total Solved	Gain/Loss
$\textsc{Vampire}_M$	10 728	baseline
$\textsc{Vampire}^*_D$	10 791	$(+94, -31)$
$\textsc{Vampire}^*_I$	10 785	$(+92, -35)$
$\textsc{Vampire}^*_H$	10 794	$(+97, -31)$

WP II: SAT Solving



WP II: Lazy Reimplication in Chronological Backtracking

published at SAT 2024 [CFK24]

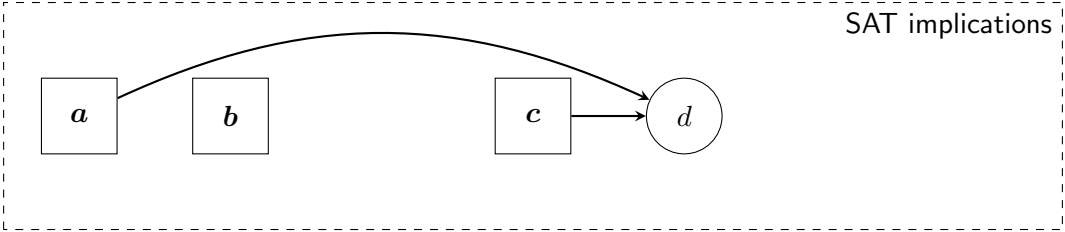
Research Question

How to reduce the number of theory propagations?

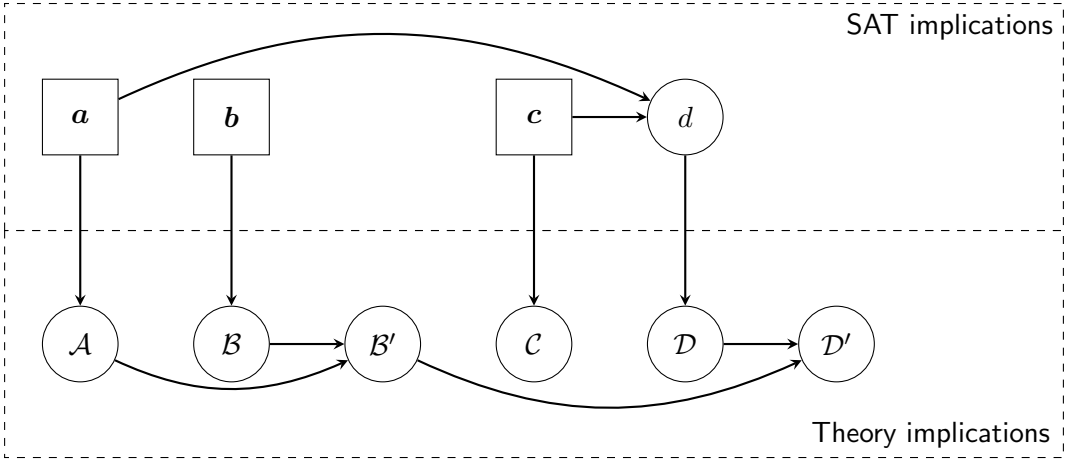
Motivation

- Chronological Backtracking is complicated and not well understood.
- Reimplication is a necessary but expensive operation.
- Theory propagations can be expensive.

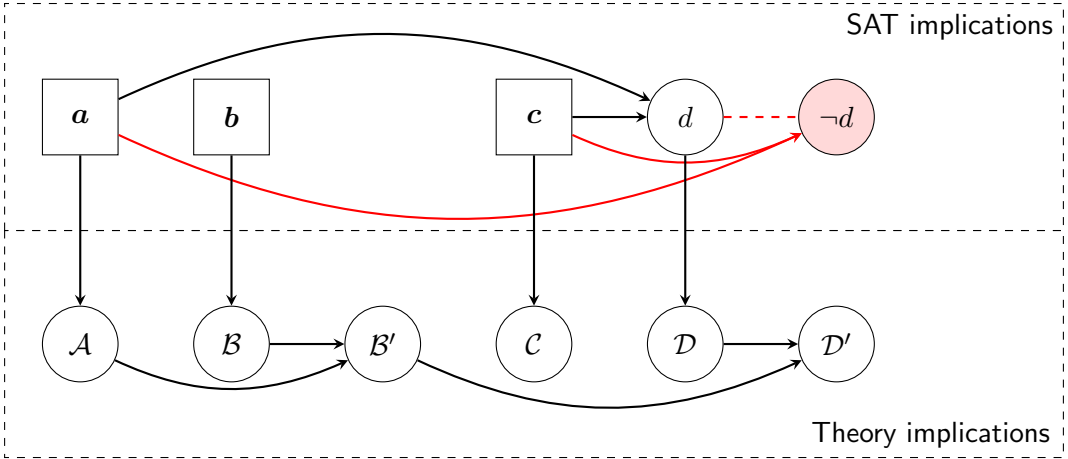
WP II: SAT and Theories (NCB)



WP II: SAT and Theories (NCB)



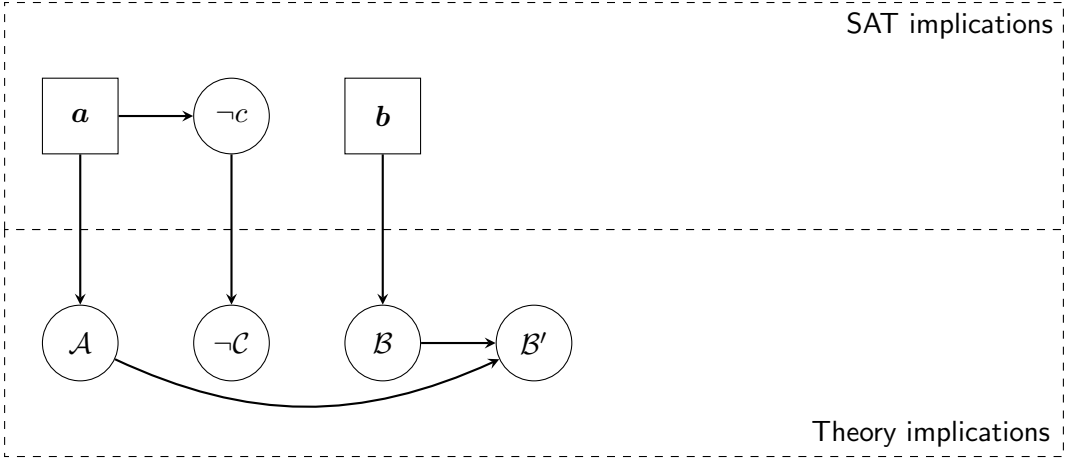
WP II: SAT and Theories (NCB)



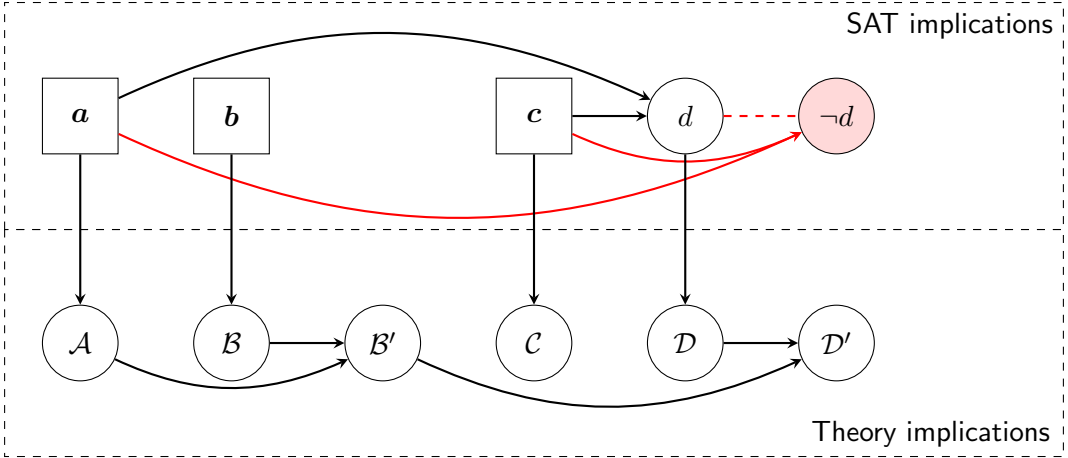
WP II: SAT and Theories (NCB)

WP II: SAT and Theories (NCB)

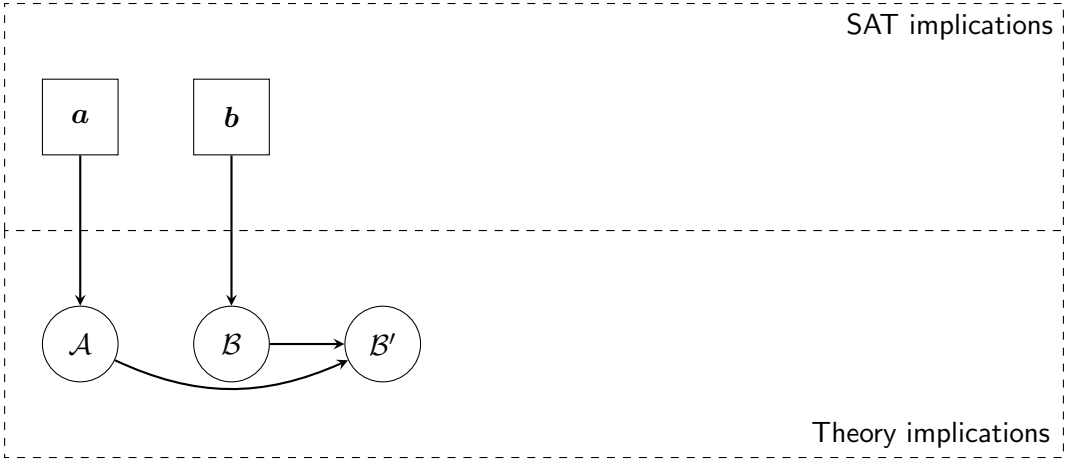
WP II: SAT and Theories (NCB)



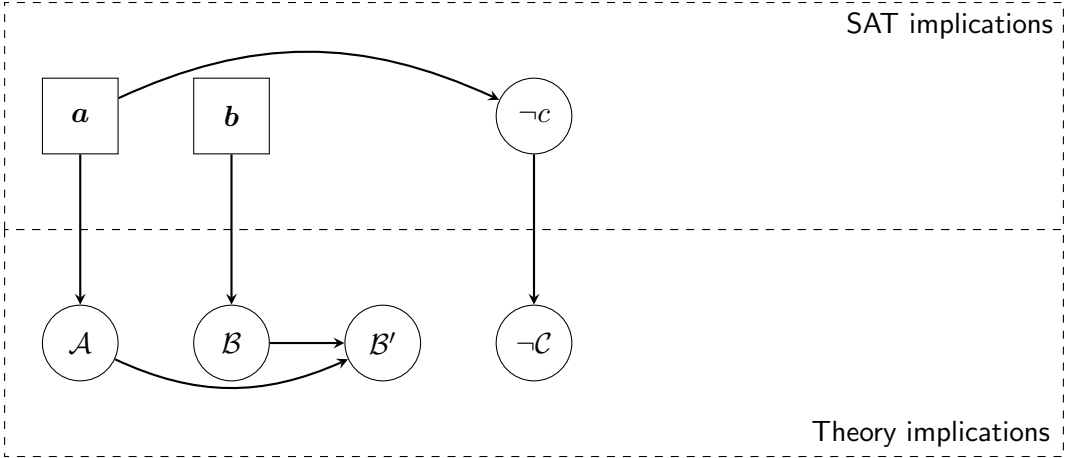
WP II: SAT and Theories (CB)



WP II: SAT and Theories (CB)

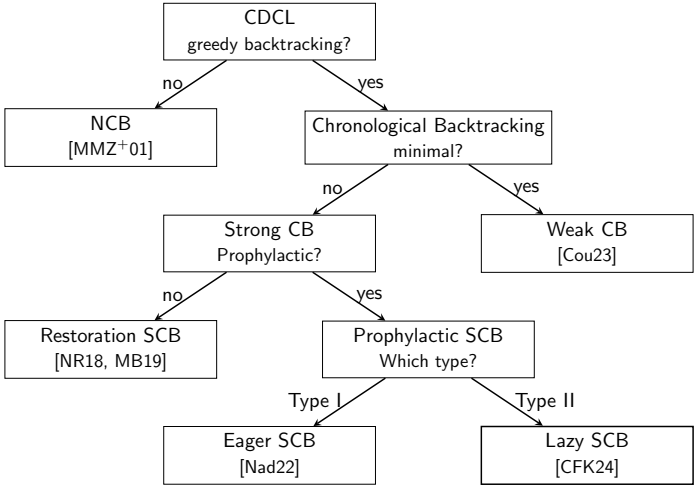


WP II: SAT and Theories (CB)



WP II: CDCL Backtracking Variants

published at SAT 2024 [CFK24]



WP II: New SAT Solver: NapSAT

Table: Implemented variants in solvers

	NCB	WCB	RSCB	ESCB	LSCB
NapSAT	✓	✓	✓	✗	✓
CaDiCaL	✓	✗	✓	✓	✓
Glucose	✓	✗	✗	✗	✓

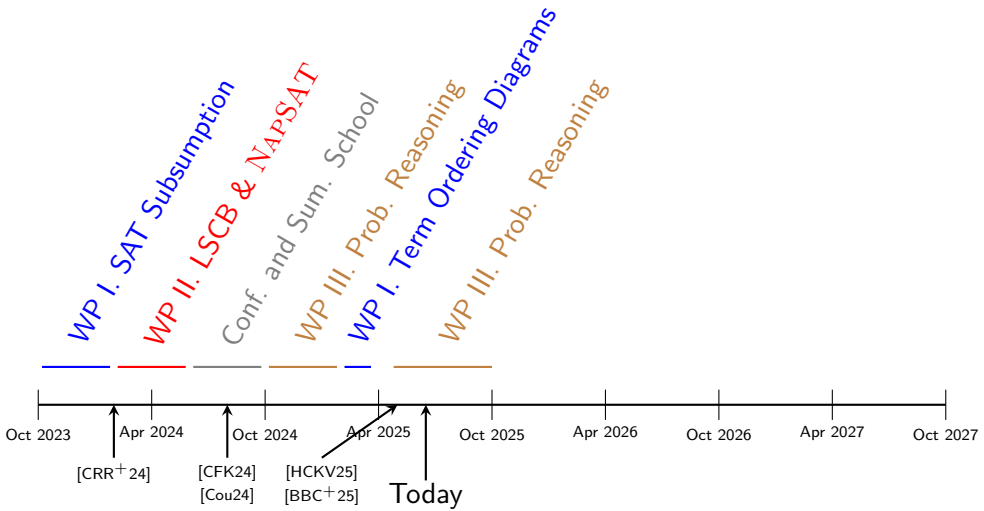
WP II: NapSAT Infrastructure

WP II: Results in CaDiCaL

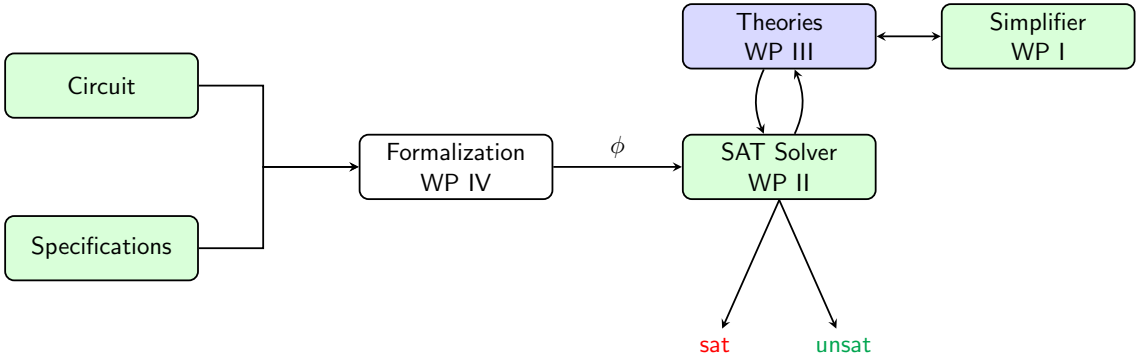
Table: Number of solved instances by different variants of strong backtracking on the SC2023 competition, using a $5000\,\mathrm{s}$ timeout

CaDiCaL version	solved	PAR-2 ($\times 10^3$)
base-CaDiCaL= RSCB	**248**	**4.09**
LSCB, Analyze-2 and minimization	246	4.16
ESCB	245	4.16
LSCB and Analyze-2	246	4.19
NCB	247	4.19
LSCB and Analyze-1	242	4.24

Research Timeline



WP III: Probability Theory



WP III: Probabilistic Reasoning

Research Question

How to reason about probabilistic statements?

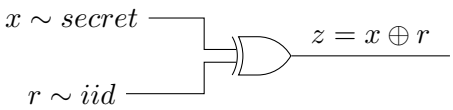
Language

A probabilistic statement is a polynomial expression over a probability distributions.

Goal

Given a set of equalities and disequalities of probabilistic statements, prove that they are consistent, or not.

WP III: Motivating Example



Goal

Prove that the output z is independent of the secret x.

$$P(z) = P(x) + P(r) - 2P(x, r) \quad \text{(circuit)}$$

$$z = x \oplus r \quad \text{(circuit)}$$

$$P(z, x) \neq P(z)P(x) \quad \text{(query)}$$

WP III: Motivating Example

Goal

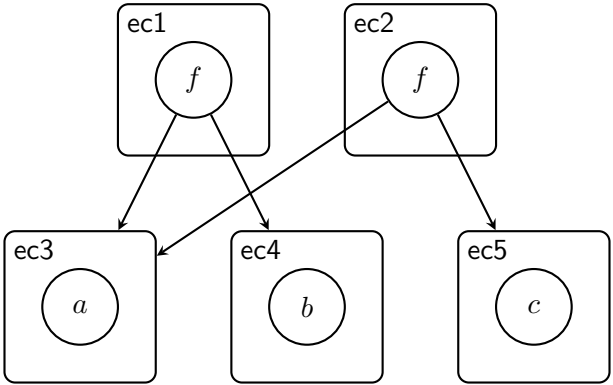
Prove that the output z is independent of the secret x.

$$P(z) = P(x) + P(r) - 2P(x, r) \quad \text{(circuit)}$$
$$z = x \oplus r \quad \text{(circuit)}$$
$$P(z, x) \neq P(z)P(x) \quad \text{(query)}$$

What do we need?

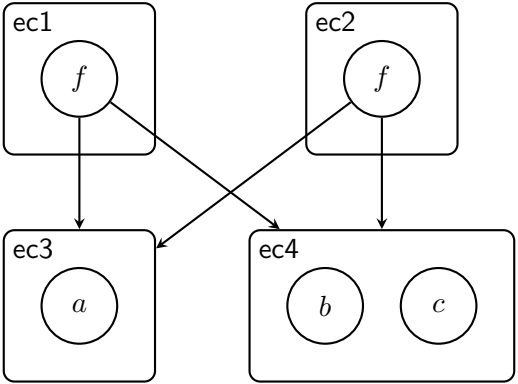
- Probability reasoning
- Equality reasoning
- Non-linear real arithmetic
- Algebraic boolean reasoning

WP III: Egraphs



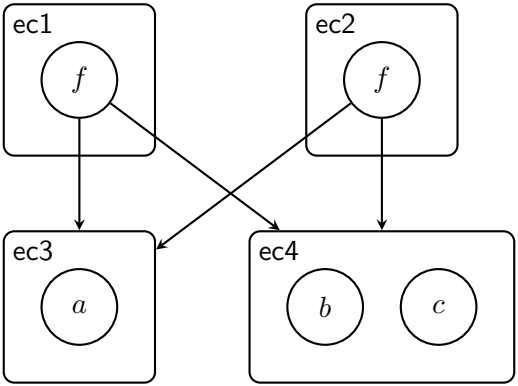
Assert : $b = c$

WP III: Egraphs



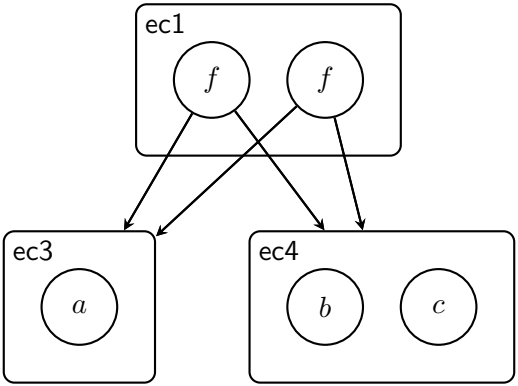
Assert : $b = c$

WP III: Egraphs



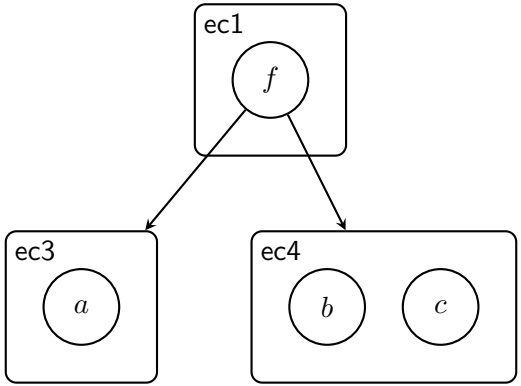
Congruence : $f(ec3, ec4)$

WP III: Egraphs



Congruence : $f(ec3, ec4)$

WP III: Egraphs



Congruence : $f(ec3, ec4)$

WP III: Why Egraphs?

Advantages

- Compact set of equalities.
- Studied by SMT and rewriting communities.
- Efficient for congruence closure.

WP III: Why Egraphs?

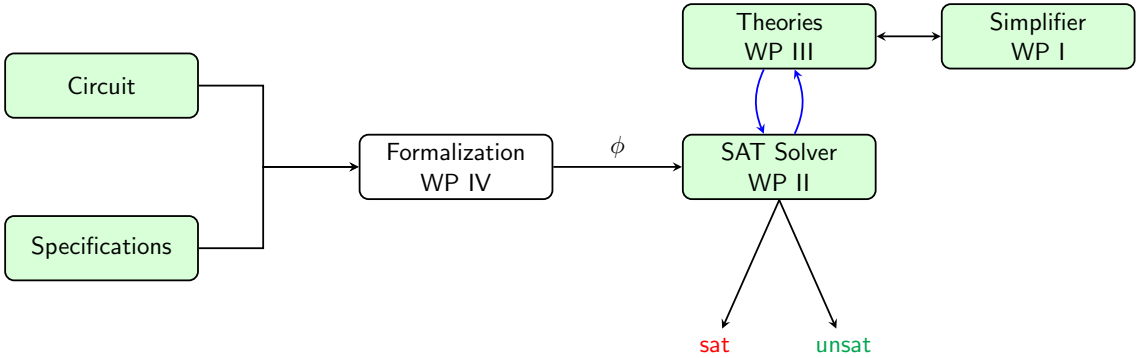
Advantages

- Compact set of equalities.
- Studied by SMT and rewriting communities.
- Efficient for congruence closure.

Challenges

- Probability calculus is not well understood.
- Combination of theories.
- Polynomials are known to be difficult.
- Complex term introduction.

WP II & III: Combination of SAT and Probability Theory



WP II & III: Combination of SAT and Probability Theory

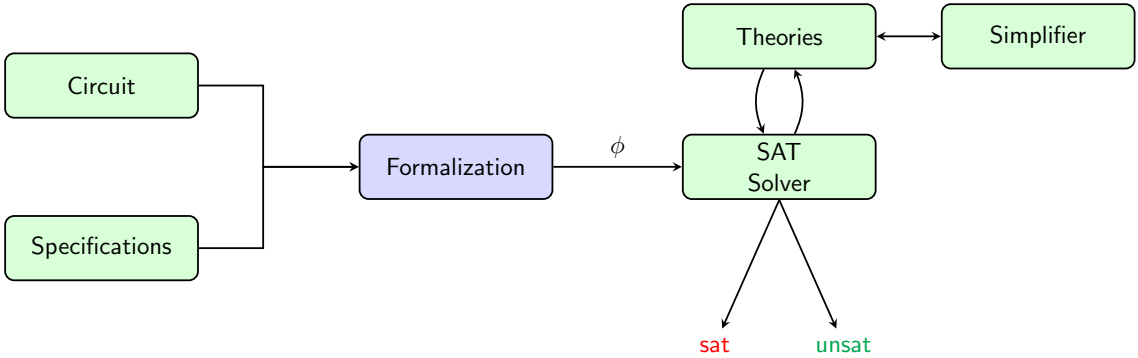
What the theory should do.

- Provide conflict clauses to the SAT solver (adapted from SMT).
- Backtrack to a previous decision level (more complicated than SMT).
- Provide unsatisfiability proofs (adapted from SMT).

Challenges

- Many rules in the theory.
- Backtracking is not well understood for rewrite systems.
- Chronological backtracking.

WP IV: Security Property Verification



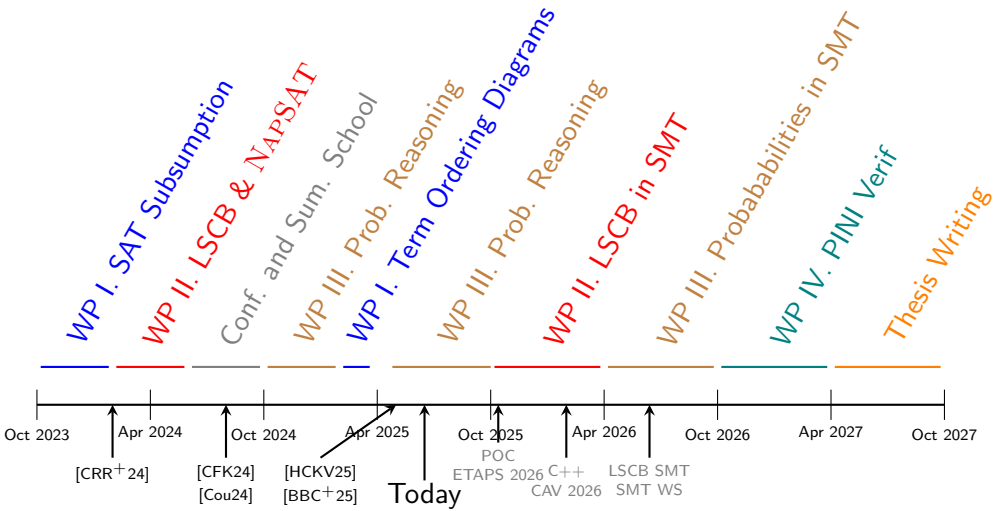
WP IV: Security Property Verification

Example Constraint

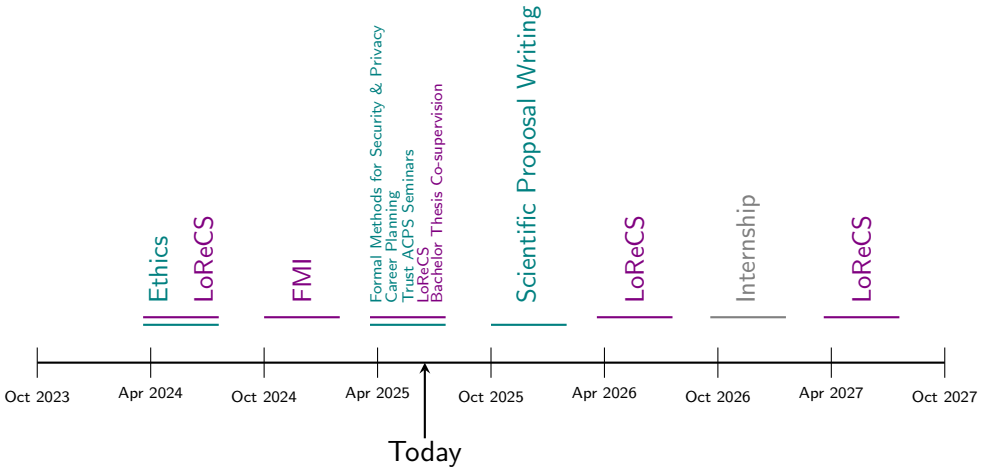
$$\bigwedge_{W \in \mathcal{C}} \forall w, s_1, \ldots, s_n.\ probe(W) \Rightarrow \bigvee_{i=1}^{n} P(W = w | S_i = s_i) = P(W = w)$$

$$AtMost_{W \in \mathcal{C}}(probed(W), t)$$

Research Timeline



Course Timeline



References I

Filip Bártek, Ahmed Bhayat, Robin Coutelier, Márton Hajdu, Matthias Hetzenberger, Petra Hozzová, Laura Kovács, Jakob Rath, Michael Rawson, Giles Reger, Martin Suda, Johannes Schoisswohl, and Andrei Voronkov.

The vampire diary (to appear).

2025.

Robin Coutelier, Mathias Fleury, and Laura Kovács.

Lazy reimplication in chronological backtracking.

In *SAT*, volume 305 of *LIPIcs*, pages 9:1–9:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

References II

Robin Coutelier, Laura Kovács, Michael Rawson, and Jakob Rath.

SAT-based subsumption resolution.

In *CADE*, volume 14132 of *Lecture Notes in Computer Science*, pages 190–206. Springer, 2023.

Robin Coutelier.

Chronological vs. Non-Chronological Backtracking in Satisfiability Modulo Theories, 2023.

Robin Coutelier.

To link or not to link? that is a watch list.

2024.

References III

Robin Coutelier, Jakob Rath, Michael Rawson, Armin Biere, and Laura Kovács.

SAT solving for variants of first-order subsumption.

Formal Methods in System Design, pages 1–44, 2024.

Flaticon.

Free icons https://www.flaticon.com/free-icons/.

Accessed: 2025-06-15.

Márton Hajdu, Robin Coutelier, Laura Kovács, and Andrei Voronkov.

Term ordering diagrams (to appear).

In *CADE*, 2025.

References IV

Sibylle Möhle and Armin Biere.

Backing backtracking.

In *SAT*, volume 11628 of *Lecture Notes in Computer Science*, pages 250–266. Springer, 2019.

Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik.

Chaff: Engineering an efficient SAT solver.

In *DAC*, pages 530–535. ACM, 2001.

References V

Alexander Nadel.

Introducing intel(r) SAT solver.

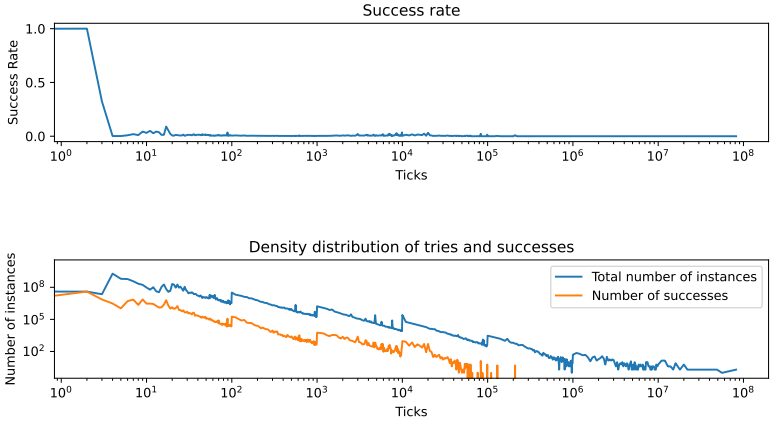
In *SAT*, volume 236 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

Alexander Nadel and Vadim Ryvchin.

Chronological backtracking.

In *SAT*, volume 10929 of *Lecture Notes in Computer Science*, pages 111–121. Springer, 2018.

Cutting off Difficult Instances?



Success rate of direct SR with respect to difficulty

Computation Saved