Verified substitution redundancy checking

Cayden Codel
With Marijn Heule and Jeremy Avigad

FMCAD 2024 in Prague, Czechia
Thursday, October 17

[citation needed]

SAT solving is a crucial (research) tool

Source: half of this conference!

SAT solving is trustworthy

D: deletion; L: linear

RAT: reverse asymmetric tautology

PR: propagation redundancy

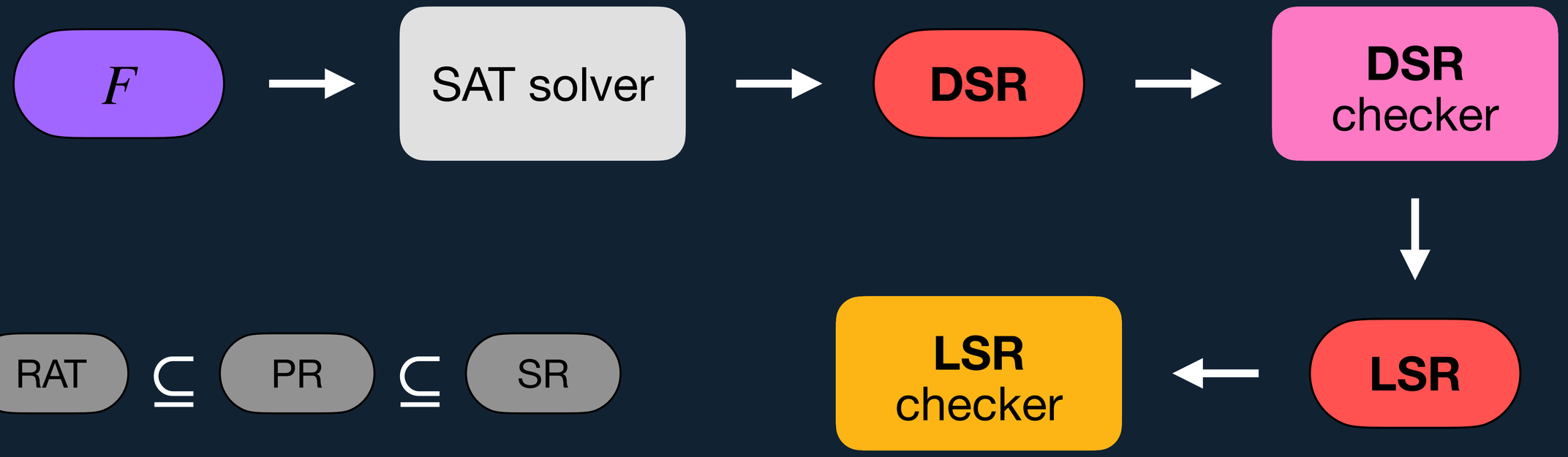
SR: substitution redundancy

In the 2023 SAT Competition

2nd place

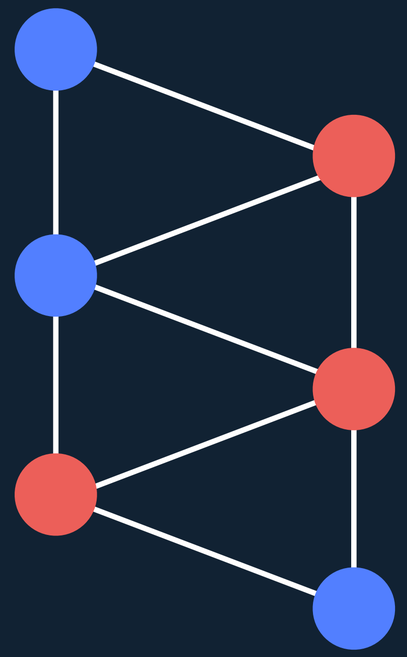
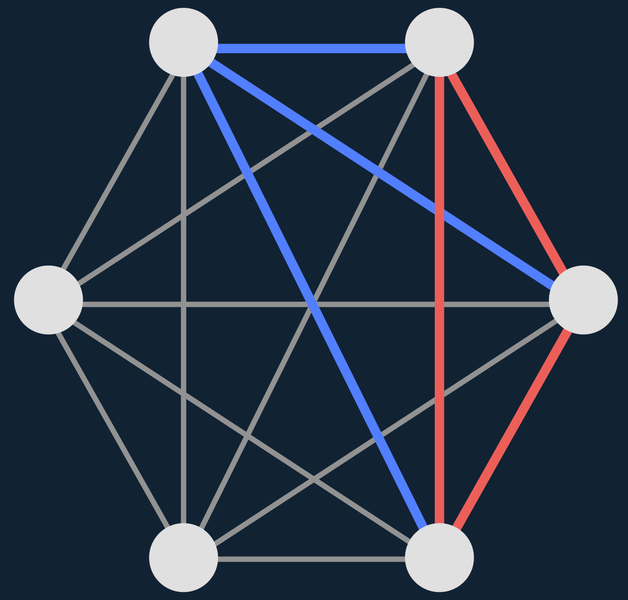
8th place

Used PR reasoning



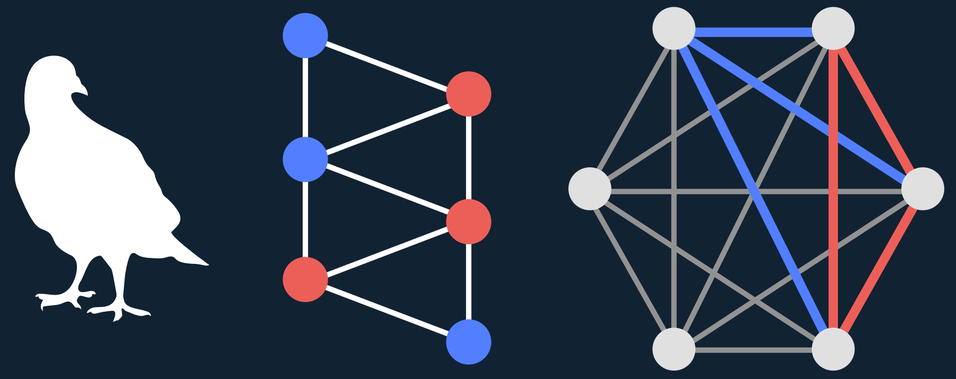
RAT $\subseteq$ PR $\subseteq$ SR

SR admits short proofs

	Pigeonhole	Tseitin formulas	Ramsey graphs
			
Resolution:	$O(2^n)$*	Exponential	1 billion steps
PR:	$O(n^3)$	Polynomial	R(4, 4): 400,000 steps
SR:	$O(n^2)$	$O(\|E\|)$	38 steps

Our contributions

Handcrafted **Proofs** for:



sr2drat

dsr-trim

lsr-check

Lean LSR checker

Verified!

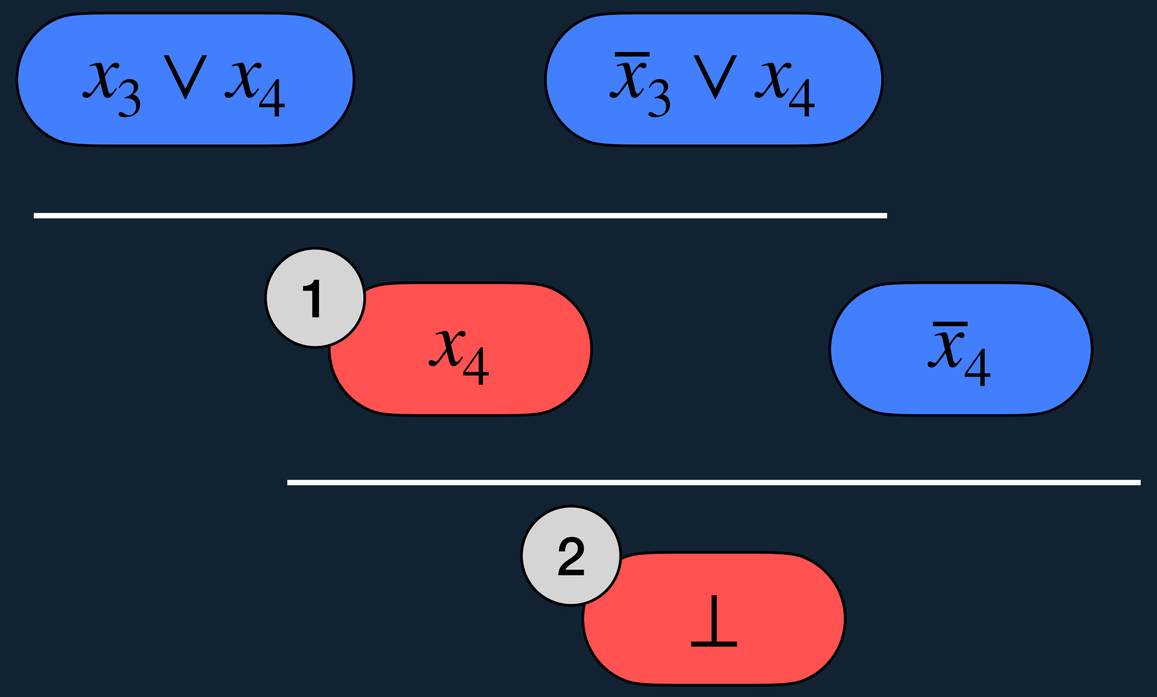
Competitive!

How to write an UNSAT proof?

Write the clauses in some order

Derive the empty clause $\bot$

Resolution proofs



Clausal proofs

Redundant clauses

$$F \equiv F \wedge \overset{1}{x_1 \vee \bar{x}_2 \vee x_3} \quad \overset{2}{x_1 \vee \bar{x}_2} \quad \dots \quad \overset{n}{\bot}$$

$$\equiv F \wedge x_1 \vee \bar{x}_2 \vee x_3 \wedge x_1 \vee \bar{x}_2$$

$$\dots$$

$$\equiv F \wedge x_1 \vee \bar{x}_2 \vee x_3 \wedge x_1 \vee \bar{x}_2 \wedge \dots \wedge \bot$$

A redundancy proof?

$$\exists\, \tau \models F \quad \Leftrightarrow \quad \exists\, \tau \models F \wedge C$$

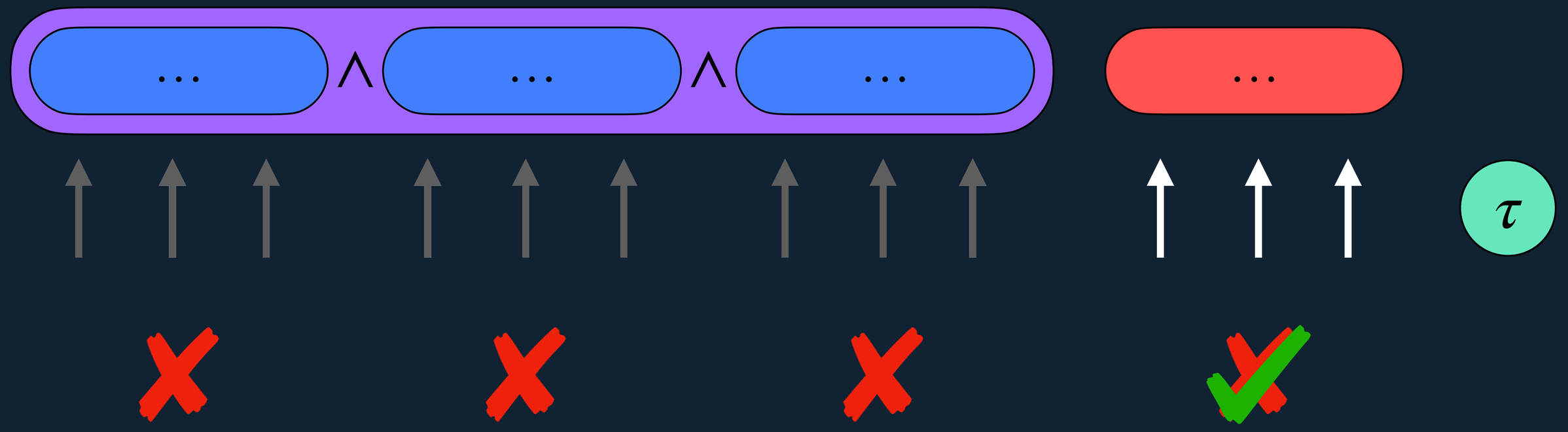
Proof. Assume that $\tau \models F$

Case 1: $\tau \models C$ ✓

Case 2: $\tau \not\models C$

Question: can we repair the assignment?

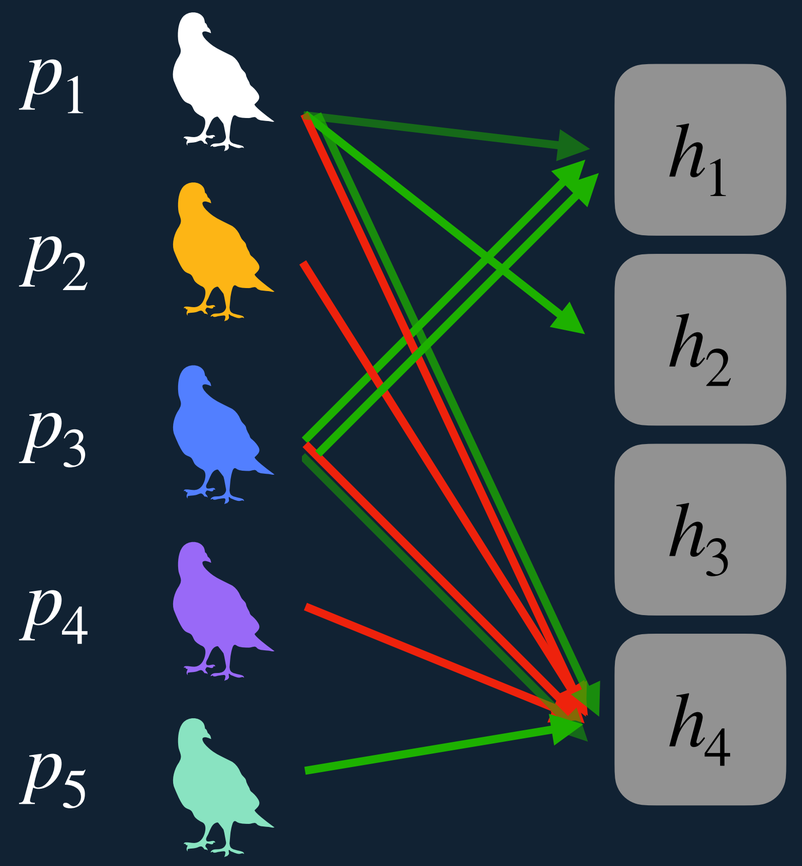
A redundancy proof?



Question: which repairs are allowed?

Question: how to specify the repair?

Example: pigeonhole



$\overline{p}_{1,4}$

$\overline{p}_{2,4}$

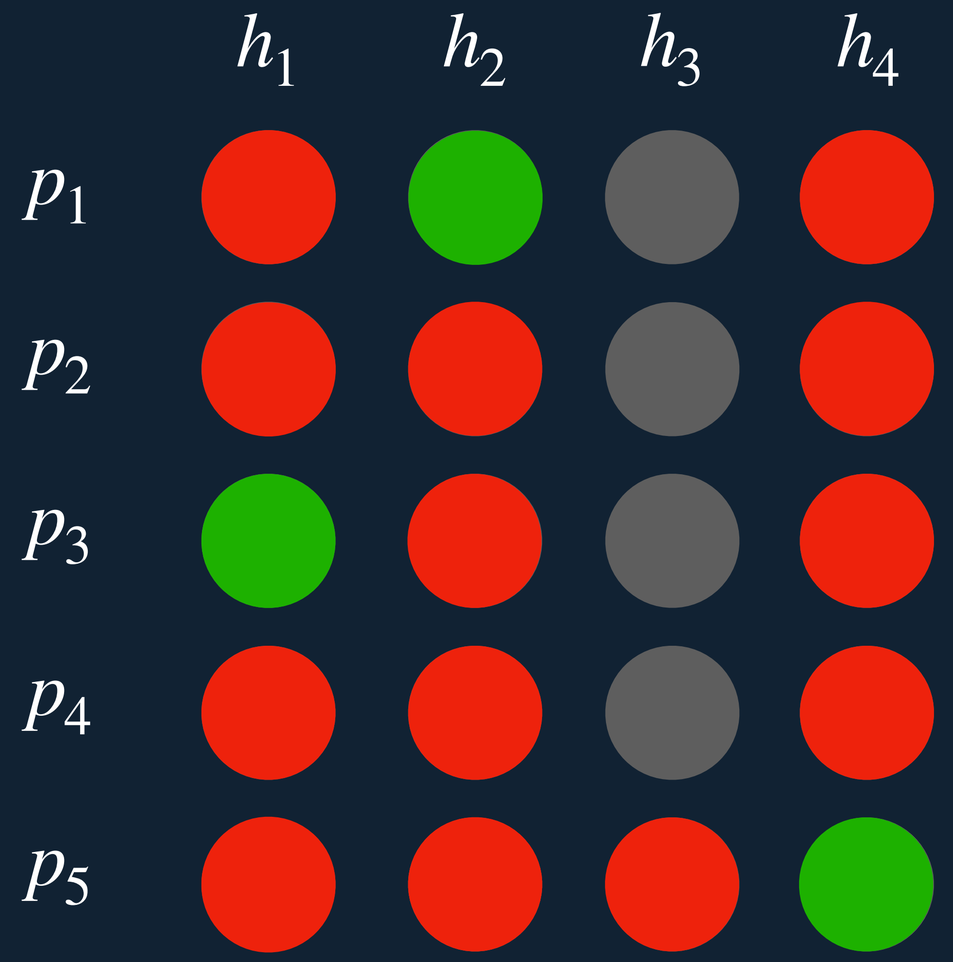
$\overline{p}_{3,4}$

$\overline{p}_{4,4}$

$p_{5,4}$

$\overline{p}_{5,[1-3]}$

Recurse!

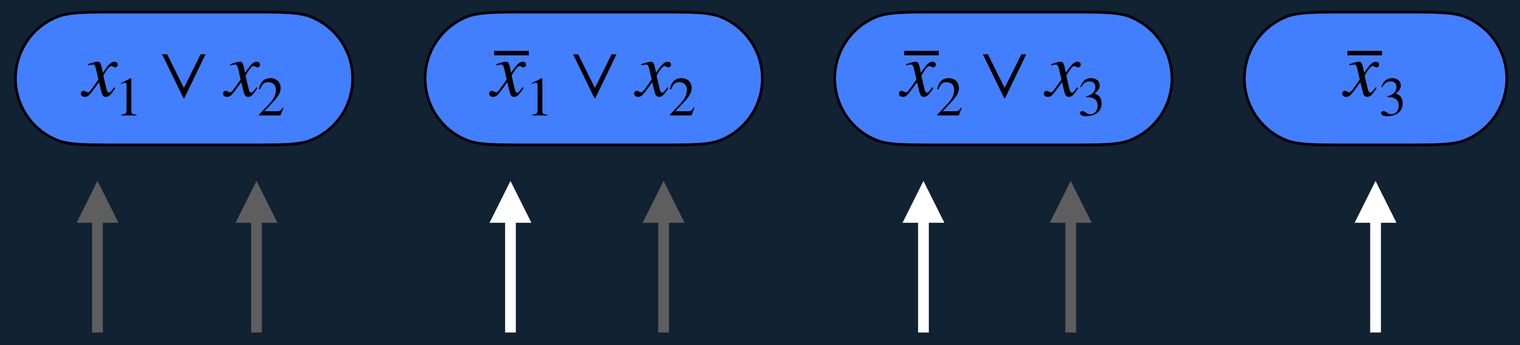


σ $p_{1,4} \mapsto \bot$, $p_{3,4} \mapsto \top$, $p_{1,x} \leftrightarrow p_{3,x}$

Substitution redundancy (SR)

$$F \wedge \neg C \vdash_1 (F \wedge C)|_\sigma$$

Aside: unit propagation



τ

$x_3 \mapsto \bot$

$x_2 \mapsto \bot$

$x_1 \mapsto \bot$

Substitution redundancy (SR)

$$F \wedge \neg C \vdash_1 (F \wedge C)|_\sigma$$

SR proof checking

(Assume that $\sigma \models C$)

1. Determine if $F \wedge \neg C \vdash_1 \bot$

Perform unit propagation until conflict

2. Determine if $F \wedge \neg C \vdash_1 F|_\sigma$

For each reduced clause $D \in F|_\sigma$

Determine if $F \wedge \neg C \wedge \neg D \vdash_1 \bot$

Perform unit propagation until conflict

Our verified Lean LSR checker

- 8k lines of code + proof code
- Took 4 person months to complete
- Implements common SAT tricks
  - Which made the formalization more complicated

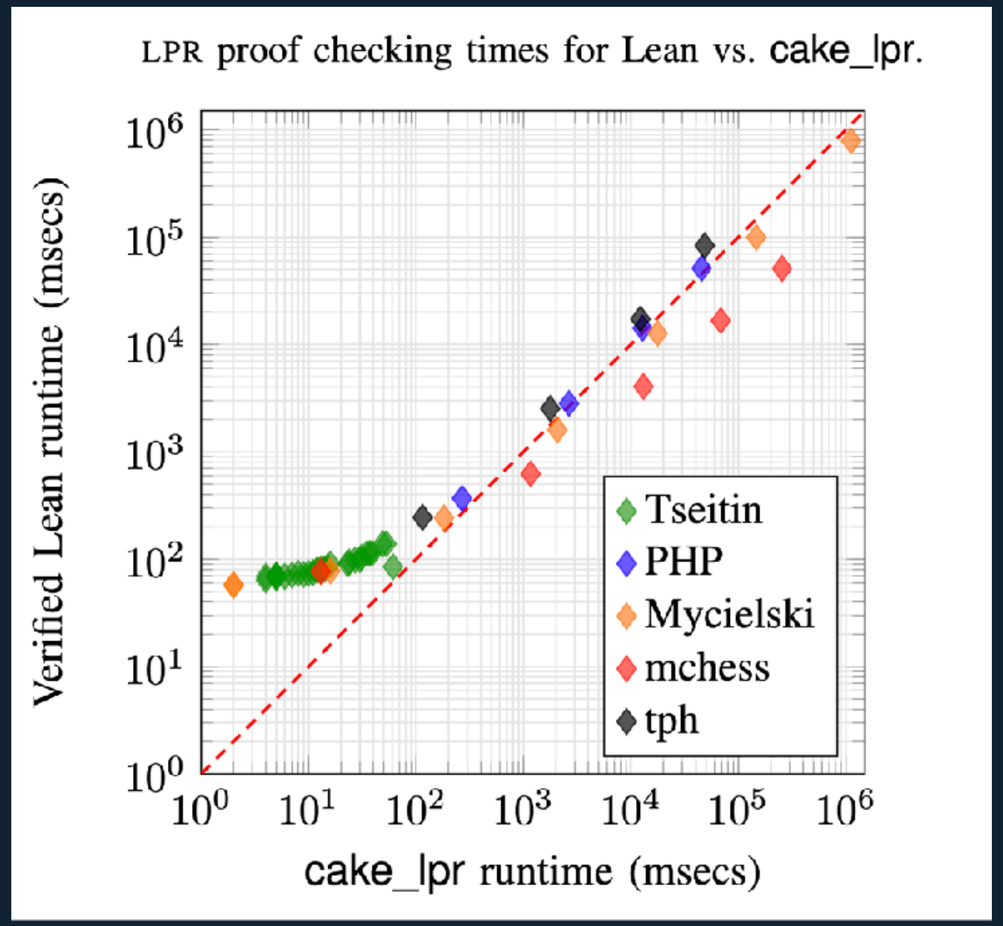
Our verified Lean LSR checker

But is it fast?

Pretty much…

Experimental results

Experiment 1: The Lean checker is comparable to other verified checkers

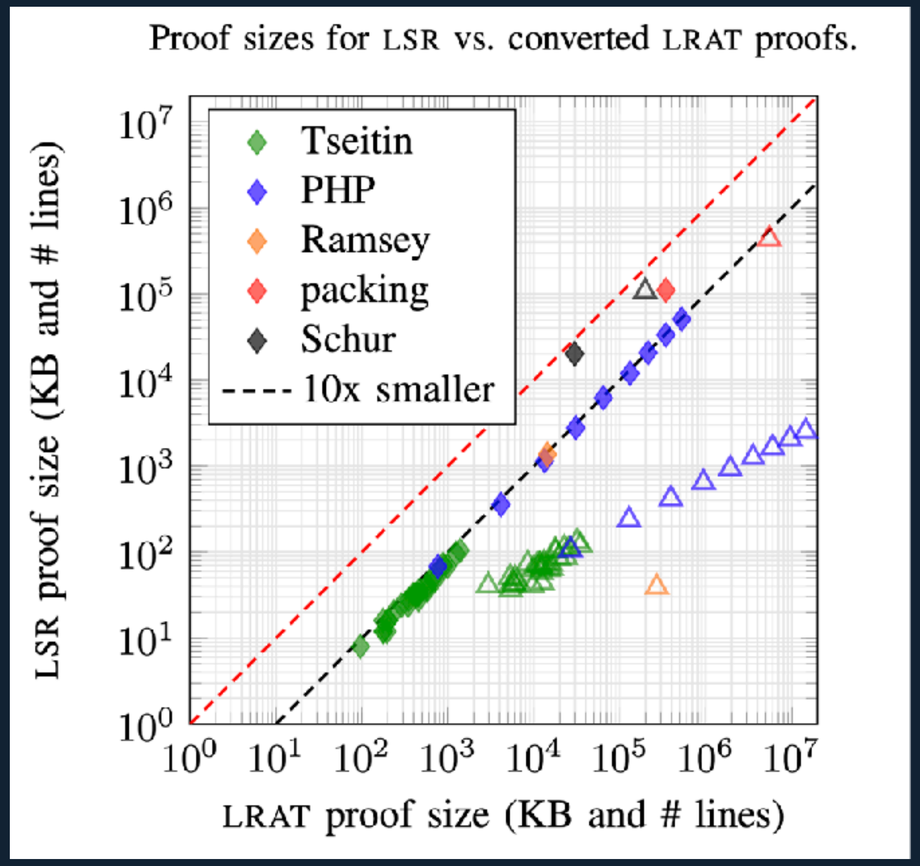


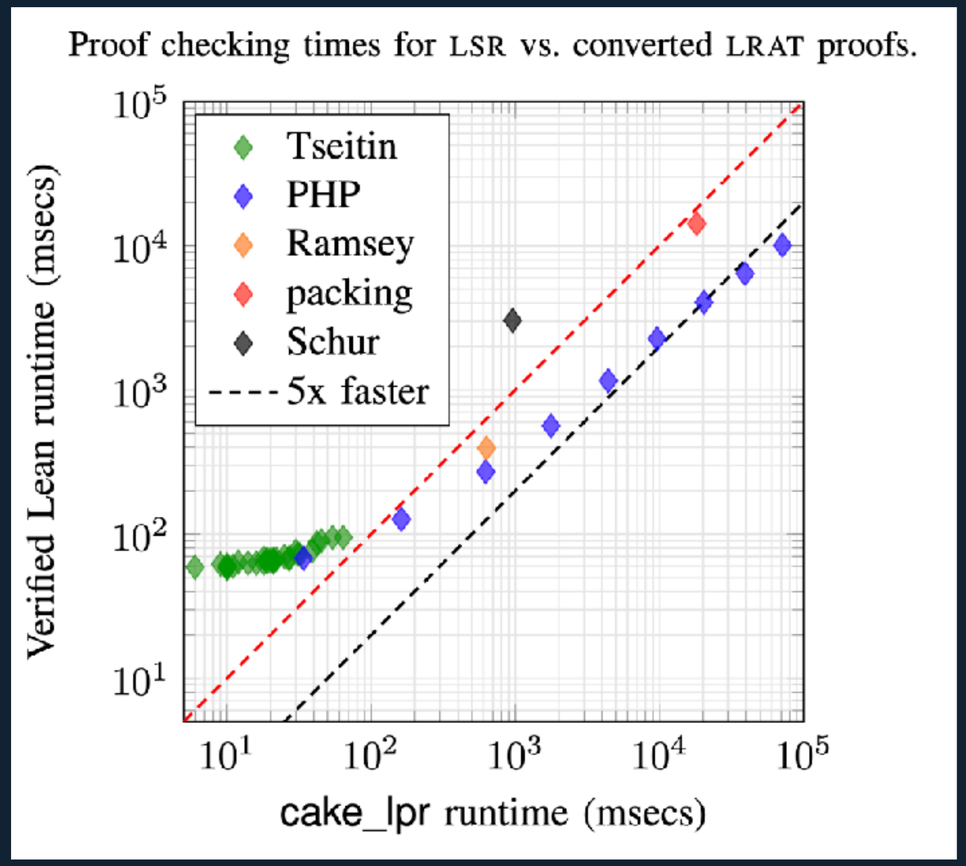
Points below the red line indicate that the verified Lean code was faster.

This is a log-log plot.

Experimental results

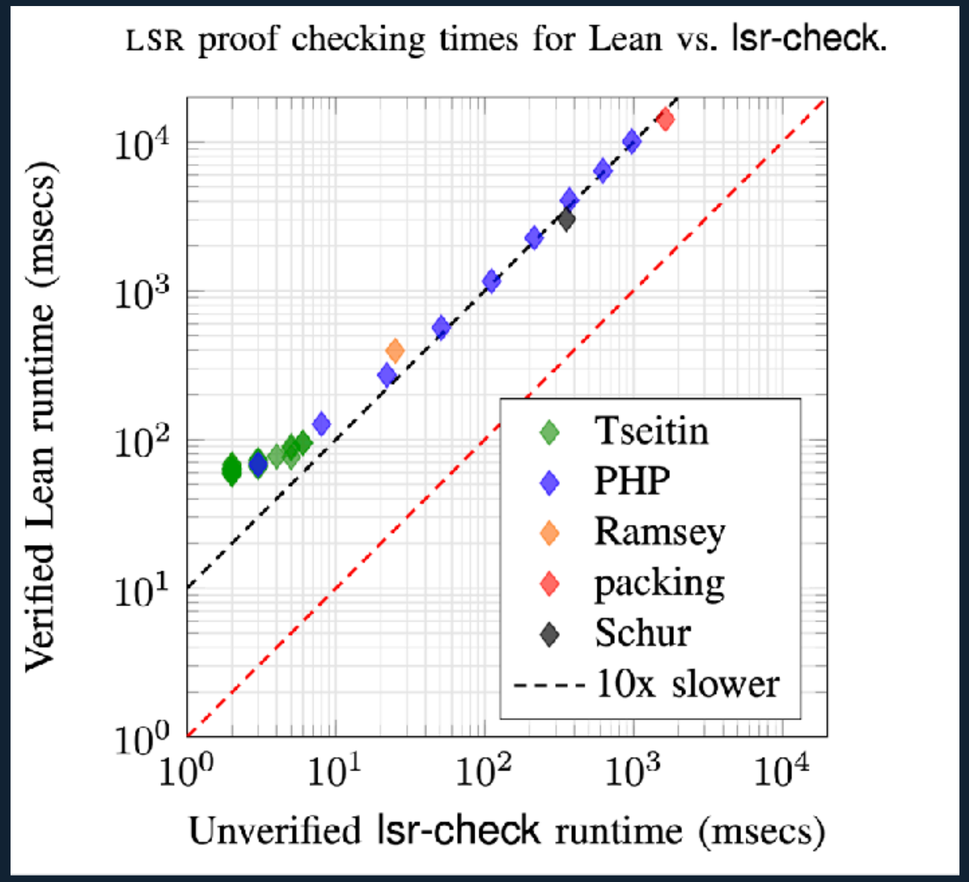
Experiment 2: LSR proofs are smaller and faster to check than LRAT counterparts





Experimental results

Experiment 3: The Lean checker doesn't have too much overhead, when compared against unverified code



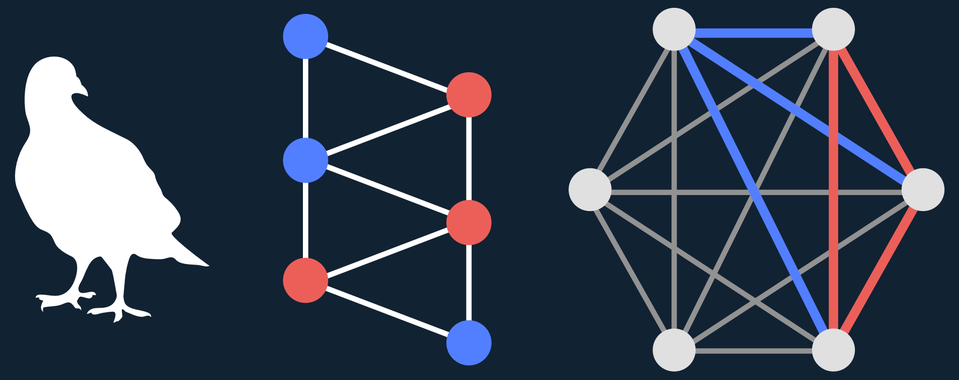
Future work

- Automatic SR symmetry breaking
- Faster proof checking tools
- Proof streaming

Questions?

Redundant clauses

$$F \equiv F \wedge \overset{1}{x_1 \vee \bar{x}_2 \vee x_3} \quad \overset{2}{x_1 \vee \bar{x}_2} \quad \ldots \quad \overset{n}{\bot}$$



dsr-trim

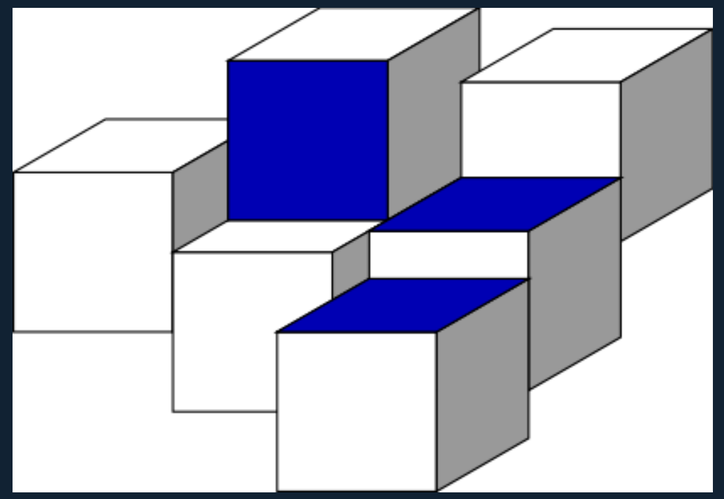
lsr-check

Lean LSR checker

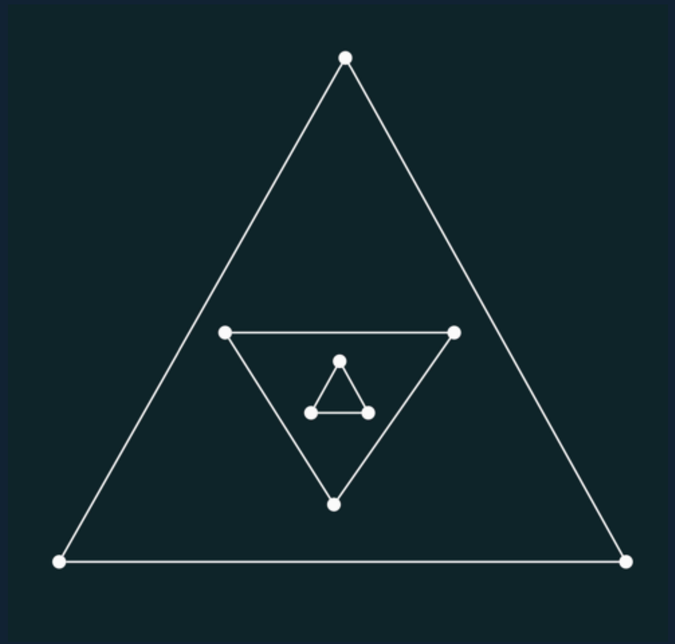
$$F \wedge \neg C \vdash_1 (F \wedge C)|_\sigma$$

Extra content

SAT solving is a crucial (research) tool



Keller's Conjecture



Happy Ending
problem variant

aws
Zelkova

Industry SAT
applications

SR variants

$$F \wedge \neg C \vdash_1 (F \wedge C)|_\sigma$$

$$F \wedge \neg C \vdash_1 F|_\sigma \qquad \sigma \models C$$

Quick SAT primer

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

Can we find a τ satisfying F

(written as: $\tau \models F$)

Quick SAT solving primer

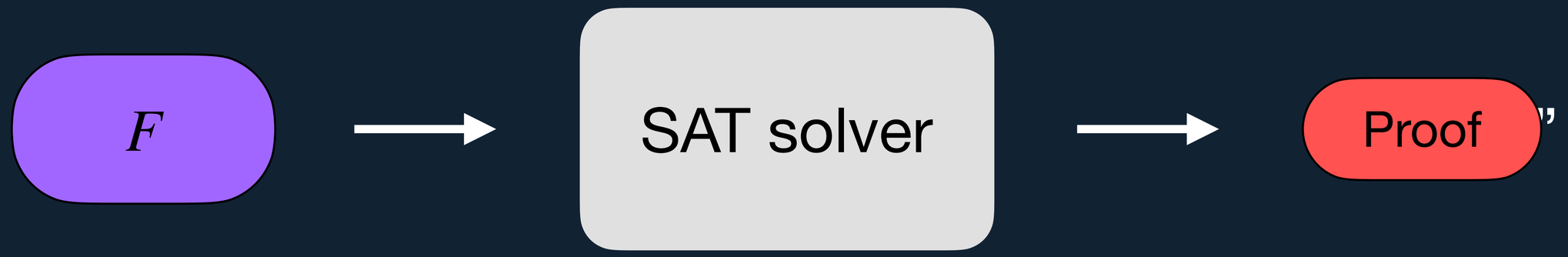
Case 1: Satisfiable

$$x_1 \lor x_2 \lor \bar{x}_3 \quad \land \quad x_1 \lor \bar{x}_2 \lor x_3 \quad \land \quad \bar{x}_1 \lor \bar{x}_2 \lor \bar{x}_3$$

$$\longrightarrow \quad \begin{aligned} x_1 &\mapsto \top \\ x_2 &\mapsto \top \\ x_3 &\mapsto \bot \end{aligned}$$

Quick SAT solving primer

Case 2: Unsatisfiable

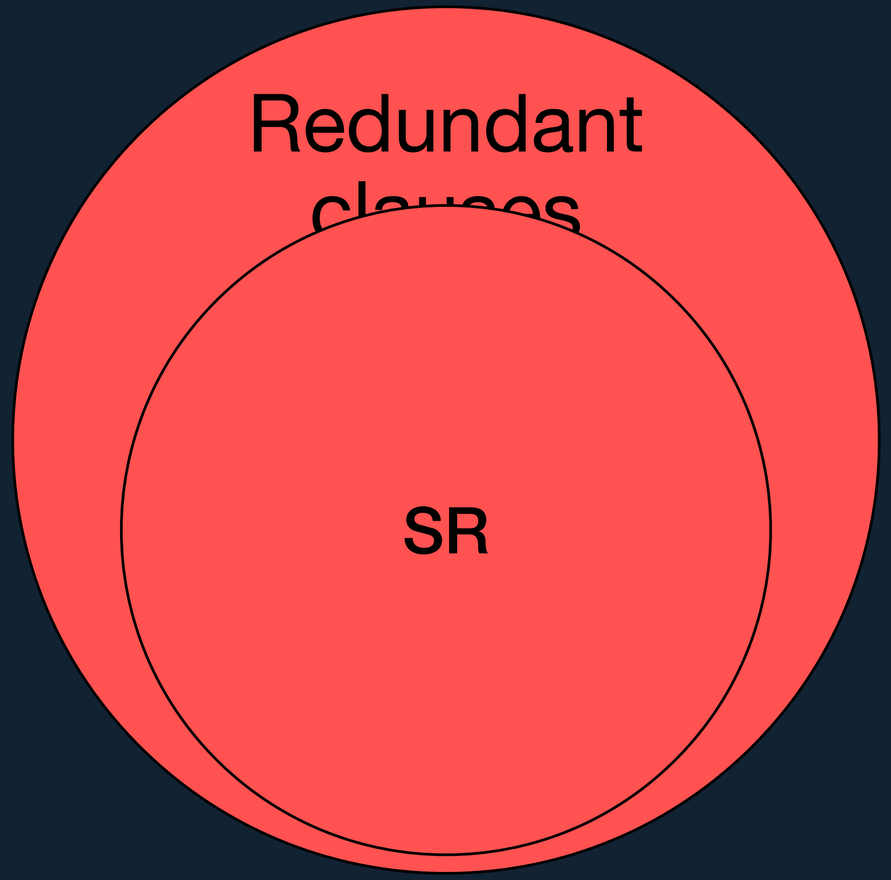



```
int main() {
  printf("UNSAT\n");
  return 0;
}
```

About redundancy

σ = a substitution of lits

$$\frac{x \vee C \qquad \bar{x} \vee D}{C \vee D}$$



LSR proof format

```
P cnf 12 22
 1   2   3   0
 4   5   6   0
 7   8   9   0
    ...
```

Clause ID

Substitution witness

Reduced clause UP hints

Redundant clause

UP hints

```
23 -10 -10 7 -10 8 11 11 8 9 12 12 9 0  7 9 10 -4 3 -6 -8 -12 13 ... 0
24  -7  -7 4  -7 5  8  8 5 6  9  9 6 0 23 6  8 -3 2 -5 -9 -11 12 ... 0
24 d 5 7 12 0       // deletion line
...
27 0 23 24 25 26 4 21 22 2 3 14 0
```

Implementing truth assignments

$x_1 \mapsto \top$
$x_2 \mapsto \top$
$x_3 \mapsto \bot$

g	3
$\max\lvert g_i \rvert$	3
g_1	3
g_2	3
g_3	-3

Set to:

$x_1 \wedge \bar{x}_2$

$x_1 \mapsto \top$
$x_2 \mapsto \bot$
$x_3 \mapsto ?$

g	4
$\max\lvert g_i \rvert$	8
g_1	8
g_2	-8
g_3	-3

Assume

x_3

$x_1 \mapsto \top$
$x_2 \mapsto \bot$
$x_3 \mapsto \top$

g	4
$\max\lvert g_i \rvert$	8
g_1	8
g_2	-8
g_3	4

```
theorem uniform_bump {τ : PPA} {offset : Nat} :
  uniform τ offset -> uniform τ.bump (offset - 1) := ...
```