Everything's Bigger in Texas
The Largest Math Proof Ever

Solving and Verifying the boolean Pythagorean Triples problem via Cube-and-Conquer

Marijn J.H. Heule

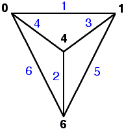
THE UNIVERSITY OF TEXAS AT AUSTIN

Joint work with Oliver Kullmann and Victor W. Marek

SAT 2016 Conference July 8, 2016

Satisfiability (SAT) solving has many applications

formal verification

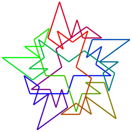


graph theory

bioinformatics

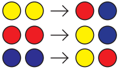
train safety

planning



combinatorics

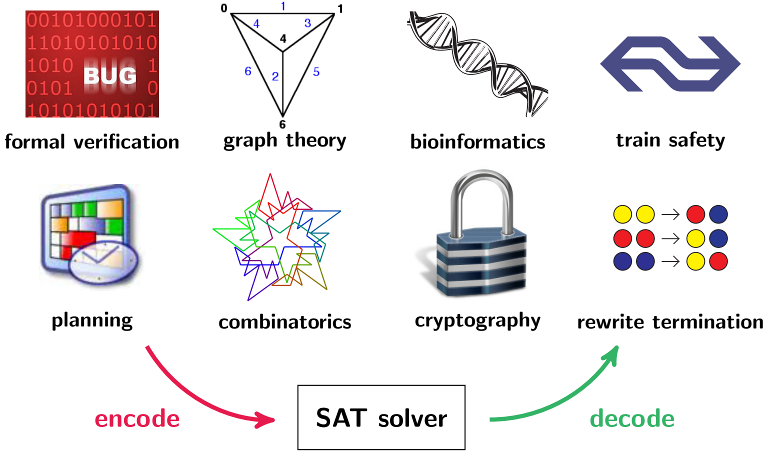
cryptography



rewrite termination

encode → **SAT solver** → decode

Satisfiability (SAT) solving has many applications



Main challenges to solve hard problems and trust the results:

- Can we achieve linear speedups on multi-core systems?
- Can we produce proofs to gain confidence in the results?

Pythagorean Triples Problem

Linear Speedups using Cube-and-Conquer

Producing & Verifying a Proof

Media, Meaning, and Truth

Conclusions and Future Work

Pythagorean Triples Problem

Schur's Theorem [Schur 1917] (1)

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be k-colored such that there is no monochromatic solution of $a+b=c$? Otherwise, what is the smallest finite counter-example?

Schur's Theorem [Schur 1917] (1)

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be k-colored such that there is no monochromatic solution of $a+b=c$? Otherwise, what is the smallest finite counter-example?

Consider the case $k = 2$ with the colors named red and blue:

1	$\rightarrow$	1 2	$\rightarrow$	1 2 4	$\rightarrow$	1 2 3 4	$\rightarrow$	$\times$
init		$1+1=2$		$2+2=4$		$1+3=4$		$1+4=5$ $2+3=5$

Schur's Theorem [Schur 1917] (1)

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be k-colored such that there is no monochromatic solution of $a+b=c$? Otherwise, what is the smallest finite counter-example?

Consider the case $k = 2$ with the colors named red and blue:

1	$\to$	1 2	$\to$	1 2 4	$\to$	1 2 3 4	$\to$	$\times$
init		$1+1=2$		$2+2=4$		$1+3=4$		$1+4=5$ $2+3=5$

Let S_n^2 denote the inference rules for $k = 2$ with $a, b, c \leq n$.

S_5^2 : $1+1=2$ $1+2=3$ $1+3=4$ $1+4=5$ $2+2=4$ $2+3=5$

The above shows: $S_5^2 + 1 \rightsquigarrow \times$. Now we can make a proof:

$$\frac{S_5^2 + 1 \rightsquigarrow \times \qquad S_5^2 + 1 \rightsquigarrow \times}{S_5^2 \rightsquigarrow \times}$$

Schur's Theorem [Schur 1917] (2)

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be k-colored such that there is no monochromatic solution of $a+b=c$? Otherwise, what is the smallest finite counter-example?

Consider the case $k = 2$ with the colors named red and blue:

1	$\rightarrow$	1 2	$\rightarrow$	1 2 4	$\rightarrow$	1 2 3 4	$\rightarrow$	$\times$
init		$1+1=2$		$2+2=4$		$1+3=4$		$1+4=5$ $2+3=5$

Schur's Theorem [Schur 1917] (2)

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be k-colored such that there is no monochromatic solution of $a+b=c$? Otherwise, what is the smallest finite counter-example?

Consider the case $k = 2$ with the colors named red and blue:

1	→	1 2	→	1 2 4	→	1 2 3 4	→	×
init		$1+1=2$		$2+2=4$		$1+3=4$		$1+4=5$ $2+3=5$

Theorem (Schur's Theorem)

For each $k > 0$, there exists a number $S(k)$, known as Schur number, such that there exists a k-coloring of $[1, S(k)]$ without a monochromatic solution of $a + b = c$ with $a, b, c \leq S(k)$, while this is impossible for $[1, S(k)+1]$.

$S(1) = 1, S(2) = 4, S(3) = 13, S(4) = 44$ [Baumert 1965], $160 \leq S(5) \leq 315$ [Exoo 1994, Fredricksen 1979].

Pythagorean Triples Problem [Graham]

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be colored with red and blue such that there is no monochromatic Pythagorean triple ($a, b, c \in \mathbb{N}$ with $a^2 + b^2 = c^2$)? Otherwise, what is the smallest finite counter-example?

Best lower bound: a bi-coloring of $[1, 7664]$ s.t. there is no monochromatic Pythagorean triple [Cooper & Overstreet 2015].

Pythagorean Triples Problem [Graham]

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be colored with red and blue such that there is no monochromatic Pythagorean triple ($a, b, c \in \mathbb{N}$ with $a^2 + b^2 = c^2$)? Otherwise, what is the smallest finite counter-example?

Best lower bound: a bi-coloring of $[1, 7664]$ s.t. there is no monochromatic Pythagorean triple [Cooper & Overstreet 2015].

A bi-coloring of $[1, n]$ is encoded using Boolean variables x_i with $i \in \{1, 2, \ldots, n\}$ such that $x_i = 1$ ($= 0$) means that i is colored red (blue). For each Pythagorean triple (a, b, c) two clauses are added: $(x_a \vee x_b \vee x_c) \wedge (\bar{x}_a \vee \bar{x}_b \vee \bar{x}_c)$.

Pythagorean Triples Problem [Graham]

Can the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$ be colored with red and blue such that there is no monochromatic Pythagorean triple ($a, b, c \in \mathbb{N}$ with $a^2 + b^2 = c^2$)? Otherwise, what is the smallest finite counter-example?

Best lower bound: a bi-coloring of $[1, 7664]$ s.t. there is no monochromatic Pythagorean triple [Cooper & Overstreet 2015].

A bi-coloring of $[1, n]$ is encoded using Boolean variables x_i with $i \in \{1, 2, \ldots, n\}$ such that $x_i = 1$ $(= 0)$ means that i is colored red (blue). For each Pythagorean triple (a, b, c) two clauses are added: $(x_a \vee x_b \vee x_c) \wedge (\bar{x}_a \vee \bar{x}_b \vee \bar{x}_c)$.

Theorem (Main result via parallel SAT solving + proof logging)

$[1, 7824]$ *can be bi-colored s.t. there is no monochromatic Pythagorean triple. This is impossible for* $[1, 7825]$.

A Small SAT Problem: Pythagorean Triples up to $n = 55$

$$
\begin{array}{l}
(x_3 \vee x_4 \vee x_5) \wedge (\bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_5 \vee x_{12} \vee x_{13}) \wedge (\bar{x}_5 \vee \bar{x}_{12} \vee \bar{x}_{13}) \wedge \\
(x_7 \vee x_{24} \vee x_{25}) \wedge (\bar{x}_7 \vee \bar{x}_{24} \vee \bar{x}_{25}) \wedge (x_9 \vee x_{40} \vee x_{41}) \wedge (\bar{x}_9 \vee \bar{x}_{40} \vee \bar{x}_{41}) \wedge \\
(x_6 \vee x_8 \vee x_{10}) \wedge (\bar{x}_6 \vee \bar{x}_8 \vee \bar{x}_{10}) \wedge (x_8 \vee x_{15} \vee x_{17}) \wedge (\bar{x}_8 \vee \bar{x}_{15} \vee \bar{x}_{17}) \wedge \\
(x_{10} \vee x_{24} \vee x_{26}) \wedge (\bar{x}_{10} \vee \bar{x}_{24} \vee \bar{x}_{26}) \wedge (x_{12} \vee x_{35} \vee x_{37}) \wedge (\bar{x}_{12} \vee \bar{x}_{35} \vee \bar{x}_{37}) \wedge \\
(x_{14} \vee x_{48} \vee x_{50}) \wedge (\bar{x}_{14} \vee \bar{x}_{48} \vee \bar{x}_{50}) \wedge (x_9 \vee x_{12} \vee x_{15}) \wedge (\bar{x}_9 \vee \bar{x}_{12} \vee \bar{x}_{15}) \wedge \\
(x_{15} \vee x_{36} \vee x_{39}) \wedge (\bar{x}_{15} \vee \bar{x}_{36} \vee \bar{x}_{39}) \wedge (x_{12} \vee x_{16} \vee x_{20}) \wedge (\bar{x}_{12} \vee \bar{x}_{16} \vee \bar{x}_{20}) \wedge \\
(x_{16} \vee x_{30} \vee x_{34}) \wedge (\bar{x}_{16} \vee \bar{x}_{30} \vee \bar{x}_{34}) \wedge (x_{20} \vee x_{48} \vee x_{52}) \wedge (\bar{x}_{20} \vee \bar{x}_{48} \vee \bar{x}_{52}) \wedge \\
(x_{15} \vee x_{20} \vee x_{25}) \wedge (\bar{x}_{15} \vee \bar{x}_{20} \vee \bar{x}_{25}) \wedge (x_{18} \vee x_{24} \vee x_{30}) \wedge (\bar{x}_{18} \vee \bar{x}_{24} \vee \bar{x}_{30}) \wedge \\
(x_{24} \vee x_{45} \vee x_{51}) \wedge (\bar{x}_{24} \vee \bar{x}_{45} \vee \bar{x}_{51}) \wedge (x_{21} \vee x_{28} \vee x_{35}) \wedge (\bar{x}_{21} \vee \bar{x}_{28} \vee \bar{x}_{35}) \wedge \\
(x_{20} \vee x_{21} \vee x_{29}) \wedge (\bar{x}_{20} \vee \bar{x}_{21} \vee \bar{x}_{29}) \wedge (x_{24} \vee x_{32} \vee x_{40}) \wedge (\bar{x}_{24} \vee \bar{x}_{32} \vee \bar{x}_{40}) \wedge \\
(x_{28} \vee x_{45} \vee x_{53}) \wedge (\bar{x}_{28} \vee \bar{x}_{45} \vee \bar{x}_{53}) \wedge (x_{27} \vee x_{36} \vee x_{45}) \wedge (\bar{x}_{27} \vee \bar{x}_{36} \vee \bar{x}_{45}) \wedge \\
(x_{30} \vee x_{40} \vee x_{50}) \wedge (\bar{x}_{30} \vee \bar{x}_{40} \vee \bar{x}_{50}) \wedge (x_{33} \vee x_{44} \vee x_{55}) \wedge (\bar{x}_{33} \vee \bar{x}_{44} \vee \bar{x}_{55})
\end{array}
$$

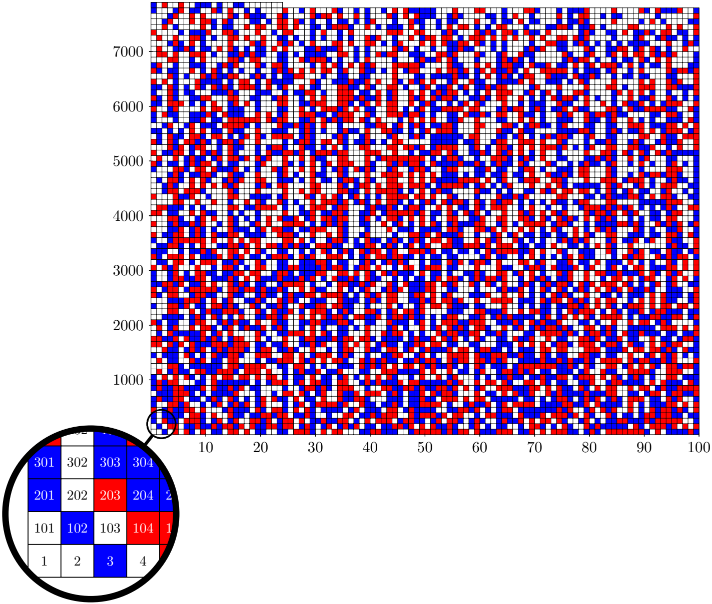
Does there exist an assignment satisfying all clauses of F_{55}?

Search for a satisfying assignment (or proof none exists)

$$(x_3 \vee x_4 \vee x_5) \wedge (\bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_5 \vee x_{12} \vee x_{13}) \wedge (\bar{x}_5 \vee \bar{x}_{12} \vee \bar{x}_{13}) \wedge$$
$$(x_7 \vee x_{24} \vee x_{25}) \wedge (\bar{x}_7 \vee \bar{x}_{24} \vee \bar{x}_{25}) \wedge (x_9 \vee x_{40} \vee x_{41}) \wedge (\bar{x}_9 \vee \bar{x}_{40} \vee \bar{x}_{41}) \wedge$$
$$(x_6 \vee x_8 \vee x_{10}) \wedge (\bar{x}_6 \vee \bar{x}_8 \vee \bar{x}_{10}) \wedge (x_8 \vee x_{15} \vee x_{17}) \wedge (\bar{x}_8 \vee \bar{x}_{15} \vee \bar{x}_{17}) \wedge$$
$$(x_{10} \vee x_{24} \vee x_{26}) \wedge (\bar{x}_{10} \vee \bar{x}_{24} \vee \bar{x}_{26}) \wedge (x_{12} \vee x_{35} \vee x_{37}) \wedge (\bar{x}_{12} \vee \bar{x}_{35} \vee \bar{x}_{37}) \wedge$$
$$(x_{14} \vee x_{48} \vee x_{50}) \wedge (\bar{x}_{14} \vee \bar{x}_{48} \vee \bar{x}_{50}) \wedge (x_9 \vee x_{12} \vee x_{15}) \wedge (\bar{x}_9 \vee \bar{x}_{12} \vee \bar{x}_{15}) \wedge$$
$$(x_{15} \vee x_{36} \vee x_{39}) \wedge (\bar{x}_{15} \vee \bar{x}_{36} \vee \bar{x}_{39}) \wedge (x_{12} \vee x_{16} \vee x_{20}) \wedge (\bar{x}_{12} \vee \bar{x}_{16} \vee \bar{x}_{20}) \wedge$$
$$(x_{16} \vee x_{30} \vee x_{34}) \wedge (\bar{x}_{16} \vee \bar{x}_{30} \vee \bar{x}_{34}) \wedge (x_{20} \vee x_{48} \vee x_{52}) \wedge (\bar{x}_{20} \vee \bar{x}_{48} \vee \bar{x}_{52}) \wedge$$
$$(x_{15} \vee x_{20} \vee x_{25}) \wedge (\bar{x}_{15} \vee \bar{x}_{20} \vee \bar{x}_{25}) \wedge (x_{18} \vee x_{24} \vee x_{30}) \wedge (\bar{x}_{18} \vee \bar{x}_{24} \vee \bar{x}_{30}) \wedge$$
$$(x_{24} \vee x_{45} \vee x_{51}) \wedge (\bar{x}_{24} \vee \bar{x}_{45} \vee \bar{x}_{51}) \wedge (x_{21} \vee x_{28} \vee x_{35}) \wedge (\bar{x}_{21} \vee \bar{x}_{28} \vee \bar{x}_{35}) \wedge$$
$$(x_{20} \vee x_{21} \vee x_{29}) \wedge (\bar{x}_{20} \vee \bar{x}_{21} \vee \bar{x}_{29}) \wedge (x_{24} \vee x_{32} \vee x_{40}) \wedge (\bar{x}_{24} \vee \bar{x}_{32} \vee \bar{x}_{40}) \wedge$$
$$(x_{28} \vee x_{45} \vee x_{53}) \wedge (\bar{x}_{28} \vee \bar{x}_{45} \vee \bar{x}_{53}) \wedge (x_{27} \vee x_{36} \vee x_{45}) \wedge (\bar{x}_{27} \vee \bar{x}_{36} \vee \bar{x}_{45}) \wedge$$
$$(x_{30} \vee x_{40} \vee x_{50}) \wedge (\bar{x}_{30} \vee \bar{x}_{40} \vee \bar{x}_{50}) \wedge (x_{33} \vee x_{44} \vee x_{55}) \wedge (\bar{x}_{33} \vee \bar{x}_{44} \vee \bar{x}_{55})$$

Solving F_{55} is easy. How to solve hard problems: F_{7824} or F_{7825}?

An Extreme Solution (a valid partition of $[1, 7824]$) I



Main Contribution

We present a framework that combines, for the first time, all pieces to produce verifiable SAT results for very hard problems.

The status quo of using combinatorial solvers and years of computation is arguably intolerable for mathematicians:

- Kouril and Paul [2008] computed the sixth van der Waerden number ($W(2,6) = 1132$) using dedicated hardware without producing a proof.
- McKay's and Radziszowski's big result [1995] in Ramsey Theory ($R(4,5) = 25$) still cannot be reproduced.

We demonstrate our framework on the Pythagorean triples problem, potentially the hardest problem solved with SAT yet.

Linear Speedups using Cube-and-Conquer

Cube-and-Conquer [Heule, Kullmann, Wieringa, and Biere 2011]

There exists two main SAT solving paradigms:

- Conflict-driven clause-learning (CDCL) aims to find a short refutation using (cheap) local heuristics.
- Look-ahead aims to construct a small binary search-tree using (expensive) global heuristics.

Cube-and-Conquer [Heule, Kullmann, Wieringa, and Biere 2011]

There exists two main SAT solving paradigms:

- Conflict-driven clause-learning (CDCL) aims to find a short refutation using (cheap) local heuristics.
- Look-ahead aims to construct a small binary search-tree using (expensive) global heuristics.

The combination: Create a tautological DNF using look-ahead techniques and solve the problem using CDCL each time under the assumption that a given cube is true.

Example (Partitioning using a tautological DNF)

Given a formula F and a DNF $D = (a \wedge b) \vee (a \wedge \bar{b}) \vee (\bar{a})$. Instead of solving CDCL (F), we solve CDCL $(F \wedge (a \wedge b))$, CDCL $(F \wedge (a \wedge \bar{b}))$, and CDCL $(F \wedge (\bar{a}))$.

The approaches are equivalent if and only if D is a tautology.

Cube-and-Conquer [Heule, Kullmann, Wieringa, and Biere 2011]

There exists two main SAT solving paradigms:

- Conflict-driven clause-learning (CDCL) aims to find a short refutation using (cheap) local heuristics.
- Look-ahead aims to construct a small binary search-tree using (expensive) global heuristics.

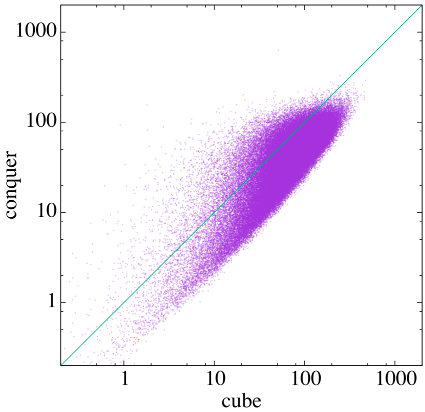
Combining look-ahead and CDCL, called cube-and-conquer, does not work out of the box. Crucial details are:

- Partition a given formula into many (millions) of subproblems. When just a few subproblems are created, say only 32, the performance could actually decrease.
- Use heuristics to create equally hard subproblems, i.e., not simply using the depth of the search-tree.

Cube-and-conquer solves many hard-combinatorial problems significantly faster than both pure CDCL and pure look-ahead.

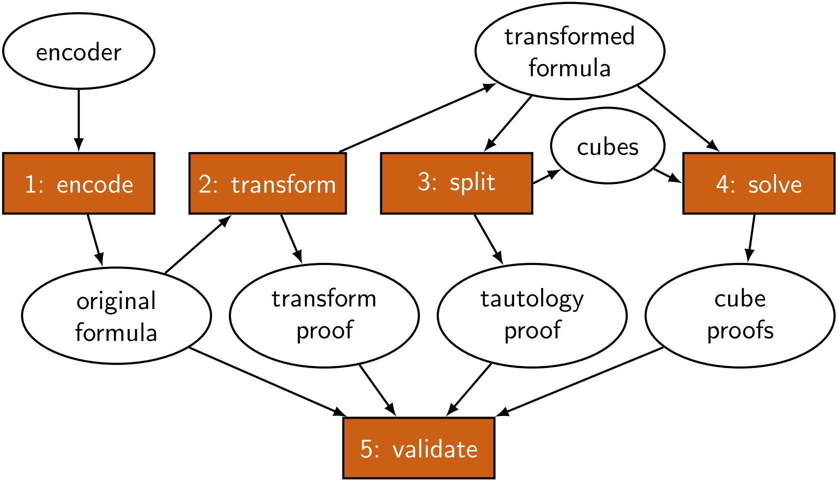
Results Summary

- After splitting —into a million subproblems— there were no hard subproblems: each could be solved within 1000 seconds;
- We used 800 cores on the TACC Stampede cluster;
- The total computation was about 4 CPU years, but less than 2 days in wallclock time;
- If we could use all 110 000 cores, then the problem could be solved in less than an hour;
- Almost linear speed-ups even when using 1000's of cores.

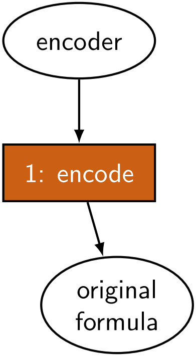


Producing & Verifying a Proof

Overview of Solving Framework



Phase 1: Encode



Input: encoder program

Output: the "original" CNF formula

Goal: make the translation to SAT as simple as possible

```
for (int a = 1; a <= n; a++)
  for (int b = a; b <= n; b++) {
     int c = sqrt (a*a + b*b);
     if ((c <= n) && ((a*a + b*b) == (c*c))) {
       addClause ( a,  b,  c);
       addClause (-a, -b, -c); } }
```

F_{7824} has 6492 (occurring) variables and 18930 clauses, and
F_{7825} has 6494 (occurring) variables and 18944 clauses.

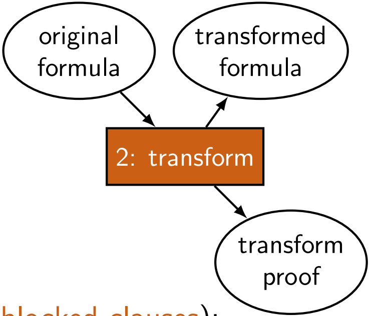
Notice $F_{7825} = F_{7824} + 14$ clauses. These 14 make it UNSAT.

Phase 2: Transform

Input: original CNF formula

Output: transformed formula and transformation proof

Goal: optimize the formula for the later (solving) phases



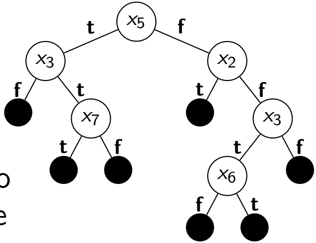
We applied two transformations (via blocked clauses):

- Pythagorean Triple Elimination removes Pythagorean Triples that contain an element that does not occur in any other Pythagorean Triple, e.g. $3^2 + 4^2 = 5^2$ (till fixpoint).
- Symmetry breaking colors the number most frequently occurring in Pythagorean triples (2520) red.

All transformation (pre-processing) techniques can be expressed using RAT steps [Järvisalo, Heule, and Biere 2012].

Phase 3: Split

Input:	transformed formula
Output:	cubes and tautology proof
Goal:	partition the given formula to minimize total wallclock time



Two layers of splitting F_{7824}:

- The top level split partitions the transformed formula into exactly a million subproblems;
- Each subproblem is partitioned into tens of thousands of subsubproblems. Total time: 25,000 CPU hours

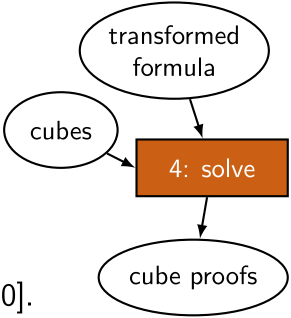
$$D = (x_5 \wedge \bar{x}_3) \vee (x_5 \wedge x_3 \wedge x_7) \vee (x_5 \wedge x_3 \wedge \bar{x}_7) \vee (\bar{x}_5 \wedge x_2) \vee (\bar{x}_5 \wedge \bar{x}_2 \wedge x_3 \wedge \bar{x}_6) \vee (\bar{x}_5 \wedge \bar{x}_2 \wedge x_3 \wedge x_6) \vee (\bar{x}_5 \wedge \bar{x}_2 \wedge \bar{x}_3)$$

Phase 4: Solve

Input: transformed formula and cubes

Output: cube proofs (or a solution)

Goal: solve —with proof logging— all cubes as fast as possible



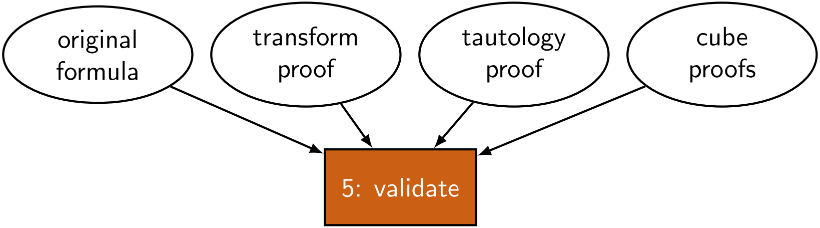
Let φ_i be the i^{th} cube with $i \in [1, 1\,000\,000]$.

We first solved all $F_{7824} \wedge \varphi_i$, total runtime was $13,000$ CPU hours (less than a day on the cluster). One cube is satisfiable.

The backbone of a formula is the set of literals that are assigned to true in all solutions. The backbone of F_{7824} after symmetry breaking (2520) consists of 2304 literals, including

- x_{5180} and x_{5865}, while $5180^2 + 5865^2 = 7825^2 \rightarrow 7825$
- $\bar{x}_{625}$ and $\bar{x}_{7800}$, while $625^2 + 7800^2 = 7825^2 \rightarrow 7825$

Phase 5: Validate Pythagorean Triples Proofs.

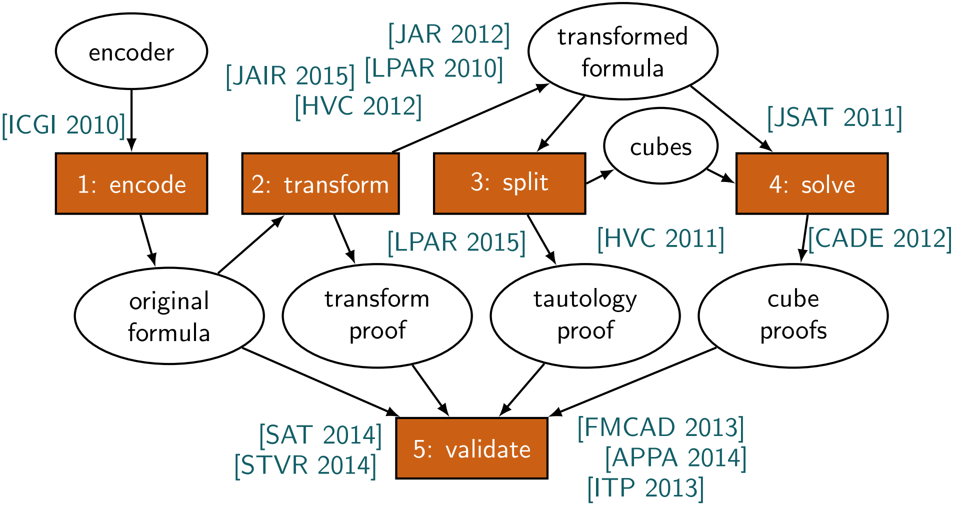


We check the proofs with the DRAT-trim checker, which has been used to validate the UNSAT results of the international SAT Competitions since 2013.

Recently it was shown how to validate DRAT proofs in parallel [Heule and Biere 2015].

The size of the merged proof is almost 200 terabyte and has been validated in 16,000 CPU hours.

Overview of Solving Framework: Contributions



Joint work with: Armin Biere, Warren Hunt, Matti Järvisalo, Oliver Kullmann, Florian Lonsing, Victor Marek, Martina Seidl, Antonio Ramos, Peter van der Tak, Sicco Verwer, Nathan Wetzler and Siert Wieringa.

Media, Meaning, and Truth

Media: The Largest Math Proof Ever

Mathematics versus Computer Science

A typical argument, as articulated in the Nature 543, pp 17–18:

> *If mathematicians' work is understood to be a quest to increase human understanding of mathematics, rather than to accumulate an ever-larger collection of facts, a solution that rests on theory seems superior to a computer ticking off possibilities.*

Widespread missing understanding of computer science:

- Computers do not simply "tick off possibilities";
- The "possibilities" are non-trivial, and simple algorithms might take forever;
- The complexity issues touched here might be far more interesting/relevant than the concrete result in Ramsey theory.

Perhaps meaningless is the true meaning?

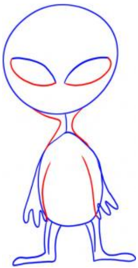
Facts may be meaningless, but...

- The "computer ticking off possibilities" is actually quite a sophisticated thing here, and is absolutely crucial for the analysis for example of the correctness of microprocessors.
- For some not yet understood reasons it seems that these benchmarks from the field of Ramsey theory are relevant for the perhaps most fundamental question in computer science: what makes a problem hard (P vs NP)?

Perhaps it is precisely that the fact 7825 has no meaning, which makes these computational problems meaningful – the bugs in the designs of complicated artificial systems also have no meaning!

Alien Truths

Let's call alien a true statement (best rather short) with only a very long proof.



- Already the question, whether we can show something (like our case) to be alien, is of highest relevance. There may be a short proof for the Pythagorean Triples problem, but probably not for exact bound of 7825.
- But independently, such "alien truths" or "alien questions" arise in formal contexts, where large propositional formulas come out from engineering systems, which in its complexity, especially what concerns "small" bugs, is perhaps beyond "understanding". Mathematicians dislike "nitty-gritty details", but prefer "the big picture" (handwaving).

Human and Alien Truth Hierarchy

Human Classical math proofs, e.g. Schur's Theorem.

Weakly Human Proofs with a large human component and some computer effort, e.g. Four Color Theorem.

Weakly Alien A giant humanly generated case-split, e.g. minimum number of givens is 17 in Sudoku.

Alien A giant case-split that mysteriously avoids an enormous exponential effort, e.g. the sixth van der Waerden number, vdW(6,6), is 1132.

Strongly Alien An alien truth regarding a high-level statement, e.g. any two-coloring of the natural numbers yields a monochromatic Pythagorean triple.

The traditional interest is to search for a short proof. But perhaps the question, why there isn't one, or what makes the problem hard, is the real question here?

Conclusions and Future Work

Conclusions

Theorem (Main result)

$[1, 7824]$ *can be bi-colored s.t. there is no monochromatic Pythagorean triple. This is impossible for* $[1, 7825]$.

We solved and verified the theorem via SAT solving:

- Cube-and-conquer facilitated massive parallel solving.
- A new heuristic was developed to substantially reduce the search space. Moreover the heuristic facilitated almost linear speed-ups while using 800 cores.
- The proof is huge (200 terabyte), but can be compressed to 68 gigabyte (13,000 CPU hours to decompress) and be validated in 16,000 CPU hours.

Future Directions

Apply our solving framework to other challenges in Ramsey Theory and elsewhere:

- Existing results for which no proof was produced, for example $W(2,6) = 1132$ [Kouril and Paul 2008].
- Century-old open problems appear solvable now, such as Schur number 5.

Look-ahead heuristics are crucial and we had to develop dedicated heuristics to solve the Pythagorean triples problem.

- Develop powerful heuristics that work out of the box.
- Alternatively, add heuristic-tuning techniques to the tool chain [Hoos 2012].

Develop a mechanically-verified, fast clausal proof checker.

Everything's Bigger in Texas
The Largest Math Proof Ever

Solving and Verifying the boolean Pythagorean Triples problem via Cube-and-Conquer

Marijn J.H. Heule

THE UNIVERSITY OF TEXAS AT AUSTIN

Joint work with Oliver Kullmann and Victor W. Marek

SAT 2016 Conference July 8, 2016