The boolean Pythagorean Triples Problem

Oliver Kullmann

Computer Science Department
Swansea University
http://cs.swan.ac.uk/~csoliver/papers.html#PYTHAGOREAN2016C

Internal Seminar
June 9, 2016

The theorem

https://en.wikipedia.org/wiki/Boolean_Pythagorean_triples_problem

We showed in our SAT 2016 paper ([1, 2]):

Theorem

For all sets A_1, A_2 with $A_1 \cup A_2 = \mathbb{N} = \{1, 2, \ldots\}$ there is $i \in \{1, 2\}$ and $a, b, c \in A_i$ with $a^2 + b^2 = c^2$. More precisely:

For all $n \geq 7825$ it is true, that
for all A_1, A_2 with $A_1 \cup A_2 = \{1, \ldots, n\}$
there is $i \in \{1, 2\}$ and $a, b, c \in A_i$ with $a^2 + b^2 = c^2$.
And for all $n \leq 7824$ there exist A_1, A_2 with $A_1 \cup A_2 = \{1, \ldots, n\}$, none of A_1, A_2 containing such a Pythagorean triple.

This solved a problem open for 30 years, with the longest proof yet.

Different from all previous contributions to Ramsey theory via SAT solving, here also the existence problem for n was open.

Propositional formulation

Consider boolean variables $v_1, \ldots, v_{7825}$, and let F be

the disjunction of

$$(v_a \wedge v_b \wedge v_c) \vee (\neg v_a \wedge \neg v_b \wedge \neg v_c)$$

for all $1 \leq a < b < c \leq 7825$ with $a^2 + b^2 = c^2$
(i.e., for all Pythagorean triples in $\{1, \ldots, 7825\}$).

(There are 9472 such triples.)

The Theorem is true iff F (a DNF) is a tautology (and we can find a falsifying assignment after removing all clauses mentioning v_{7825}, which is relatively easy).

SAT solving

The previous DNF is negated, yielding a CNF, and the task is to show

unsatisfiability (i.e., inconsistency).

- The hybrid SAT-solving method Cube-and-Conquer, whose idea we developed in the context of applications to Ramsey theory, was adopted to the task (various heuristics optimised), and solved the task in about $30,000$ hours.
- Due to the nature of “C&C”, whether performed on a single computer or on a cluster doesn’t matter.
- $30000/24 = 1250$ (so “super-computer” wasn’t really needed).

Proof extraction

From the computation a propositional proof in the DRAT format was extracted,

whose size is 200 TB, as made the headlines.

(That for the paper we emphasised the compression to 68 GB (via providing only the main points of the computation, the rest must be re-computed), is to be ignored here.)

- The idea of the DRAT format goes back to a generalisation of Extended Resolution I found when proving new worst-case SAT solving upper bounds ([3]).
- It further generalises this, and combines it with unit-clause propagation, to connect with the most fundamental technique of SAT solving.
- Compared to ordinary propositional proof systems, this achieves already a compression by, say, a factor of 1000.

Useful?!

A typical argument, as articulated in the Nature article ([4]):

> *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.*

One can see here a widespread missing understanding of computer science: computers simply “tick off possibilities”.

- The *systematic background* and motivation for e.g., ATP, SAT solving, formal methods, is not known.
- The *complexity issues* touched here might be far more interesting/relevant than the concrete result in Ramsey theory.
- Last, but not least: the “possibilities” are non-trivial, and simple algorithms might take forever.

Perhaps “meaningless” is the true meaning?!

- 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 problem of computer science, the question what makes a problem hard (e.g. the P vs NP problem).

It might thus well be, that at the direct level, the number 7,825 is a kind of “pure fact”, but the methods for determining this are highly relevant – perhaps it is precisely that the "7,825" has no meaning, which makes these computational(!) problems meaningful – the bugs in the designs of complicated artificial systems also have “no meaning”!

“Alien” versus “human” truth

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.
- 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).)
- Moreover, typically here the social-economic situation(!) actually inhibits any form of understanding (even if possible) – this needs to go quickly, in a fully automated way, and likely also in a secretive fashion. (Thanks to Markus Roggenbach and his forthcoming book for pointing this out.)

More on “alien truths”

Here a concrete example of an alien truth: link
More formally:

- Short statement.
- Only very very long proof (best inherently!).
- But all kind of combinatorial counts (the number of so-and-so of size 1234 etc.) are only “weakly alien”; they are not truly awe-inspiring.
- Also a real proof is needed, not just a computation (like the minimum number of givens is 17 in general Sudoku; these examples typically are also only “weakly alien”).
- It must also not just involve mathematical reasoning, plus a derived list of possibly many, but simple cases. Our proof is “truly alien”: **no insights**! And **mysteriously** avoiding an enormous exponential effort.

Not “sweetness”, but “hardness”

Proof mining (here in a concrete, direct sense – mining the 200 TB) can become very interesting here:

- 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?!

Summary and outlook

I Currently, it is hard to plant real computer-science questions into media discussions.

II The conceptual breakthroughs of *computational* complexity (NOT butterflies) are basically not understood at all.

III The “story of SAT” especially, the “science of brute-force”, is untold.

IV Hopefully you can help us with the task of changing this.

End

(references on the remaining slides).

For my papers see
http://cs.swan.ac.uk/~csoliver/papers.html.

Bibliography I

[1] Marijn J.H. Heule, Oliver Kullmann, and Victor W. Marek. Solving and verifying the boolean Pythagorean Triples problem via Cube-and-Conquer. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016*, volume 9710 of *Lecture Notes in Computer Science*, pages 228–245. Springer, 2016. ISBN 978-3-319-40969-6. doi:10.1007/978-3-319-40970-2_15.

[2] Marijn J.H. Heule, Oliver Kullmann, and Victor W. Marek. Solving and verifying the boolean Pythagorean Triples problem via Cube-and-Conquer. Technical Report arXiv:1605.00723v1 [cs.DM], arXiv, May 2016. URL http://arxiv.org/abs/1605.00723.

[3] Oliver Kullmann. New methods for 3-SAT decision and worst-case analysis. *Theoretical Computer Science*, 223(1-2):1–72, July 1999. doi:10.1016/S0304-3975(98)00017-6.

Bibliography II

[4] Evelyn Lamb. Maths proof smashes size record: Supercomputer produces a 200-terabyte proof – but is it really mathematics? *Nature*, 534:17–18, June 2016. doi:10.1038/nature.2016.19990.