ENiGMA's proof of Pythagoras theorem

Josef Urban

Czech Technical University in Prague
Czech Institute for Informatics, Robotics and Cybernetics

European Research Council
Established by the European Commission

Can You (or Your Machine) Prove the Pythagoras Theorem?

- What are those nerds on 6th floor doing all nights there?
- Formal statement and 15-line human-written verified proof:

```
:: The Pythagorean theorem
theorem :: BHSP_5:6
for X being RealUnitarySpace
for x, y being Point of X st x,y are_orthogonal holds
||.(x + y).|| ^2 = (||.x.|| ^2) + (||.y.|| ^2)
proof
let X be RealUnitarySpace; :: thesis:
let x, y be Point of X; :: thesis:
assume x,y are_orthogonal ; :: thesis:
then A1: x .|. y = 0 by BHSP_1:def 3;
A2: (x + y) .|. (x + y) >= 0 by BHSP_1:def 2;
A3: x .|. x >= 0 by BHSP_1:def 2;
A4: y .|. y >= 0 by BHSP_1:def 2;
||.(x + y).|| ^2 = (sqrt ((x + y) .|. (x + y))) ^2 by BHSP_1:def 4
.= (x + y) .|. (x + y) by A2, SQUARE_1:def 2
.= ((x .|. x) + (2 * (x .|. y))) + (y .|. y) by BHSP_1:16
.= ((sqrt (x .|. x)) ^2) + (y .|. y) by A1, A3, SQUARE_1:def 2
.= ((sqrt (x .|. x)) ^2) + ((sqrt (y .|. y)) ^2) by A4, SQUARE_1:def 2
.= (||.x.|| ^2) + ((sqrt (y .|. y)) ^2) by BHSP_1:def 4
.= (||.x.|| ^2) + (||.y.|| ^2) by BHSP_1:def 4 ;
hence ||.(x + y).|| ^2 = (||.x.|| ^2) + (||.y.|| ^2) ; :: thesis:
end;
```

Can You Prove the Pythagoras Theorem? - Details

- Our automatically found proof - the ENIGMA system: `http://grid01.ciirc.cvut.cz/~mptp/enigma_prf/t6_bhsp_5.out`
- `http://grid01.ciirc.cvut.cz/~mptp/7.13.01_4.181.1147/html/bhsp_5.html#T6`

```
# Proof object clause steps                    : 181
# Proof object initial clauses used            : 51
# Proof object initial formulas used           : 34
# Proof object simplifying inferences          : 210
# Parsed axioms                                : 342
# Initial clauses in saturation                : 440
# Processed clauses                            : 8804
# ...remaining for further processing          : 4119
# Generated clauses                            : 65094
# ...of the previous two non-trivial           : 60807
# User time                        : 28.629 s
```

Can You Prove the Pythagoras Theorem? - Details

- Jan Jakubuv (the main developer of ENIGMA) has proved it automatically on May 6 2020 in 30s (are you that fast?)
- ENIGMA was guided by 150 decision trees each of max depth 80 and having maximum 8000 leaves
- Trained by the LightGBM gradient boosted tree toolkit (state of the art ML, fast).
- On a corpus of ca 500k previous automatically found proofs.
- We could not prove the theorem automatically before that even with much higher time limits.

Can You Prove the Pythagoras Theorem? - Details

- The proof attempt started with 342 mathematical facts.
- You can think of the facts as flat language sentences.
- In fact they are first-order logic parse trees that can interact in many ways.
- Preselected from a knowledge base (math library) of ca 150k math facts
- The initial facts logically interact, producing more and more facts (inferences)
- Their number quickly explodes (millions) unless some control is introduced
- In 30s this proof attempt has generated ”only” 60k more nontrivial facts

Can You Prove the Pythagoras Theorem? - Details

- Each of the 60k facts was scored by the 150 LightGBM decision trees
- I.e. on average we scored 2k facts per second (on a single commodity CPU)
- This is also thanks to our fast (but accurate and memory efficient) characterization of the facts by syntactic features
- The scoring was gradually choosing the best of the generated facts
- These are the ones used to perform inferences with the previously chosen (processed) ones
- This is called the given clause loop and it is the basis of today's strongest theorem provers

Can You Prove the Pythagoras Theorem? - Details

- ENIGMA gradually chose 4k facts from the 60k and did all possible inferences (modus-ponens style) among them
- When the last one was chosen, it interacted with the previous facts in such a way that the proof was finished
- In the end, only 34 of the 342 initial facts were needed for the proof
- And the proof needed only 181 steps, not 4k (and could generate much fewer than 60k facts)
- So if we were smarter, we could do it even faster!
- We could learn from this proof - saying which facts/inferences were good/bad
- And in the next proving attempt, we would probably do the proof better
- So we could interleave proving and learning from proofs
- And thus train better and better automated mathematicians!

Feedback loop for ENIGMA on Mizar data

- Interleave proving and learning of ENIGMA guidance
- Done on 57880 Mizar problems very recently
- Ultimately a 70% improvement over the original strategy
- From 14933 proofs to 25397 proofs (all 10s - no cheating)
- As of 2021 we have 42519 proofs

	$\mathcal{S}$	$\mathcal{S} \odot \mathcal{M}_9^0$	$\mathcal{S} \oplus \mathcal{M}_9^0$	$\mathcal{S} \odot \mathcal{M}_9^1$	$\mathcal{S} \oplus \mathcal{M}_9^1$	$\mathcal{S} \odot \mathcal{M}_9^2$	$\mathcal{S} \oplus \mathcal{M}_9^2$	$\mathcal{S} \odot \mathcal{M}_9^3$
solved	14933	16574	20366	21564	22839	22413	23467	22910
$\mathcal{S}$%	+0%	+10.5%	+35.8%	+43.8%	+52.3%	+49.4%	+56.5%	+52.8%
$\mathcal{S}+$	+0	+4364	+6215	+7774	+8414	+8407	+8964	+8822
$\mathcal{S}-$	-0	-2723	-782	-1143	-508	-927	-430	-845

	$\mathcal{S} \odot \mathcal{M}_{12}^3$	$\mathcal{S} \oplus \mathcal{M}_{12}^3$	$\mathcal{S} \odot \mathcal{M}_{16}^3$	$\mathcal{S} \oplus \mathcal{M}_{16}^3$
solved	24159	24701	25100	25397
$\mathcal{S}$%	+61.1%	+64.8%	+68.0%	+70.0%
$\mathcal{S}+$	+9761	+10063	+10476	+10647
$\mathcal{S}-$	-535	-295	-309	-183