

# AI FOR THEOREM PROVING

---

Josef Urban

Czech Technical University in Prague  
ERC project AI4REASON

OECD Workshop on AI and the productivity of science  
November 5, 2021



# Leibniz's/Hilbert's/Russell's Dream: Let Us Calculate!

Solve all (math, physics, law, economics, society, ...) problems by  
**reduction to logic/computation**



[Adapted from: *Logicomix: An Epic Search for Truth* by A. Doxiadis]

# What is Formal Mathematics and Theorem Proving?

- 1900s: Mathematics put on formal logic foundations – **symbolic logic**
- Culmination of a program by Leibniz/Frege/Russell/Hilbert/Church/...
- ... led also to the rise of **computers** (Turing/Church, 1930s)
- ... and rise of AI - Turing's 1950 paper: Learning Machines, Chess, etc.
- 1950s: First AI program: **Logic Theorist** by Newell & Simon
- Formalization of math (60s): combine formal foundations and computers
- **Proof assistants/Interactive theorem provers** and their large libraries:
- Automath (1967), LCF, Mizar, NQTHM, HOL, Coq, Isabelle, ACL2, Lean
- **Automated theorem provers** - search for proofs automatically:
- Otter, Vampire, E, SPASS, Prover9, CVC4, Z3, Satallax, ...
- **more limited logics**: SAT, QBF, SMT, UEQ, ... (DPLL, CDCL, ...)

# Why Do This Today?

## 1 Practically Useful for Verification of Complex HW/SW and Math

- Formal Proof of the Kepler Conjecture (2014 – Hales – 20k lemmas)
- Formal Proof of the Feit-Thompson Theorem (2 books, 2012 – Gonthier)
- Verification of several math textbooks and CS algorithms
- Verification of compilers (CompCert)
- Verification of OS microkernels (seL4), HW chips (Intel), transport, finance,
- Verification of cryptographic protocols, the OpenSSL stack (Amazon), etc.

## 2 Blue Sky AI Visions:

- Get **strong AI** by learning/reasoning over large KBs of **human thought**?
- Big formal theories: good **semantic** approximation of such thinking KBs?
- Deep non-contradictory semantics – better than scanning books?
- Gradually try **learning math/science**
- automate/verify them, include law, etc. (Leibniz, McCarthy, ..)
  - What are the components (inductive/deductive thinking)?
  - How to combine them together?

## Example: Irrationality of $\sqrt{2}$ (informal text)

small proof from Hardy & Wright:

**Theorem 43 (Pythagoras' theorem).**  $\sqrt{2}$  is irrational.

The traditional proof ascribed to Pythagoras runs as follows. If  $\sqrt{2}$  is rational, then the equation

$$a^2 = 2b^2 \tag{4.3.1}$$

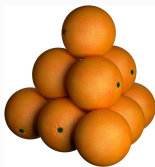
is soluble in integers  $a, b$  with  $(a, b) = 1$ . Hence  $a^2$  is even, and therefore  $a$  is even. If  $a = 2c$ , then  $4c^2 = 2b^2$ ,  $2c^2 = b^2$ , and  $b$  is also even, contrary to the hypothesis that  $(a, b) = 1$ .  $\square$

# Irrationality of $\sqrt{2}$ in Isabelle/HOL

```
theorem sqrt2_not_rational:
  "sqrt (real 2)  $\notin$   $\mathbb{Q}$ "
proof
  assume "sqrt (real 2)  $\in$   $\mathbb{Q}$ "
  then obtain m n :: nat where
    n_nonzero: "n  $\neq$  0" and sqrt_rat: "|sqrt (real 2)| = real m / real n"
    and lowest_terms: "gcd m n = 1" ..
  from n_nonzero and sqrt_rat have "real m = |sqrt (real 2)| * real n" by simp
  then have "real (m2) = (sqrt (real 2))2 * real (n2)"
    by (auto simp add: power2_eq_square)
  also have "(sqrt (real 2))2 = real 2" by simp
  also have "... * real (m2) = real (2 * n2)" by simp
  finally have eq: "m2 = 2 * n2" ..
  hence "2 dvd m2" ..
  with two_is_prime have dvd_m: "2 dvd m" by (rule prime_dvd_power_two)
  then obtain k where "m = 2 * k" ..
  with eq have "2 * n2 = 22 * k2" by (auto simp add: power2_eq_square mult_ac)
  hence "n2 = 2 * k2" by simp
  hence "2 dvd n2" ..
  with two_is_prime have "2 dvd n" by (rule prime_dvd_power_two)
  with dvd_m have "2 dvd gcd m n" by (rule gcd_greatest)
  with lowest_terms have "2 dvd 1" by simp
  thus False by arith
qed
```

# Big Example: The Flyspeck project

- Kepler conjecture (1611): The most compact way of stacking balls of the same size in space is a pyramid.



$$V = \frac{\pi}{\sqrt{18}} \approx 74\%$$

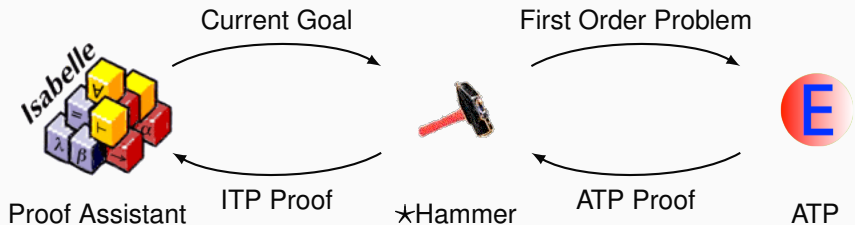
- Proved by Hales in 1998, 300-page proof + computations
- Big: Annals of Mathematics gave up reviewing after 4 years
- Formal proof finished in 2014
- 20000 lemmas in geometry, analysis, graph theory
- All of it at <https://code.google.com/p/flyspeck/>
- All of it **computer-understandable and verified** in HOL Light:  
`polyhedron s /\ c face_of s ==> polyhedron c`
- However, this took **20 – 30 person-years!**
- our 2014 work: AI/TP combinations can *hammer* **40%** of the 20k lemmas

# AI and ML Combinations with Theorem Proving

- **high-level**: pre-select lemmas from a large library, give them to ATPs
- **high-level**: pre-select a good ATP strategy/portfolio for a problem
- **high-level**: pre-select good *hints* for a problem, use them to guide ATPs
- **low-level**: guide every inference step of ATPs (tableau, superposition)
- **low-level**: guide every kernel step of LCF-style ITPs
- **mid-level**: guide application of tactics in ITPs
- **mid-level**: invent suitable ATP strategies for classes of problems
- **mid-level**: invent suitable conjectures for a problem
- **mid-level**: invent suitable concepts/models for problems/theories
- **proof sketches**: explore stronger/related theories to get proof ideas
- **theory exploration**: develop interesting theories by conjecturing/proving
- **feedback loops**: (dis)prove, learn from it, (dis)prove more, learn more, ...
- **autoformalization**: (semi-)automate translation from  $\text{\LaTeX}$  to formal
-



# Today's AI-ATP systems (★-Hammers)



How much can it do?

- Mizar / MML – MizAR
- Isabelle (Auth, Jinja) – Sledgehammer
- Flyspeck (including core HOL Light and Multivariate) – HOL(y)Hammer
- HOL4 (Gauthier and Kaliszyk)
- CoqHammer (Czajka and Kaliszyk) - about 40% on Coq standard library

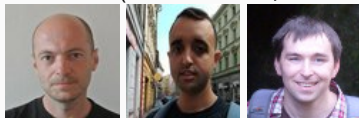
≈ 40-45% success by 2016, 60% on Mizar as of 2021

# AI/TP Examples and Demos

- **ENIGMA/hammer proofs of Pythagoras** : <https://bit.ly/2MVPAn7>  
(more at <http://grid01.ciirc.cvut.cz/~mptp/enigma-ex.pdf>) and  
simplified Carmichael <https://bit.ly/3oGBdRz>,
- **3-phase ENIGMA**: <https://bit.ly/3C0Lwa8>,<https://bit.ly/3BWqR6K>
- **Long trig proof from 1k axioms**: <https://bit.ly/2YZ0OgX>
- **Hammering demo**: <http://grid01.ciirc.cvut.cz/~mptp/out4.ogv>
- **TacticToe on HOL4**:  
[http://grid01.ciirc.cvut.cz/~mptp/tactictoe\\_demo.ogv](http://grid01.ciirc.cvut.cz/~mptp/tactictoe_demo.ogv)
- **Tactician for Coq**:  
<https://blaaubroek.eu/papers/cicm2020/demo.mp4>,  
<https://coq-tactician.github.io/demo.html>
- **Inf2formal over HOL Light**:  
<http://grid01.ciirc.cvut.cz/~mptp/demo.ogv>

# ENIGMA (2017): Guiding the Best ATPs like E Prover

- ENIGMA (Jan Jakubuv, Zar Goertzel, Karel Chvalovsky, others)



- The proof state are two large heaps of clauses *processed/unprocessed*
- learn on E's proof search traces, put classifier in E
- positive examples: clauses (lemmas) used in the proof
- negative examples: clauses (lemmas) not used in the proof
- 2021 **multi-phase architecture** (combination of different methods):
  - fast gradient-boosted decision trees (GBDTs)
  - logic-aware graph neural network (GNN) run on a GPU server
  - logic-based subsumption using fast indexing (discrimination trees)
- 2021: leapfrogging and Split&Merge:
- aiming at learning **reasoning/algo components**

# Feedback prove/learn loop for ENIGMA on Mizar data

- Done on 57880 Mizar problems recently
- Serious ML-guidance breakthrough applied to the best ATPs
- Ultimately a **70% improvement** over the original strategy in 2019
- From 14933 proofs to 25397 proofs (all 10s CPU - no cheating)
- Went up to 40k in more iterations and 60s time in 2020
- 75% of the Mizar corpus reached in July 2021 - higher times and many runs

	$S$	$S \odot M_9^0$	$S \oplus M_9^0$	$S \odot M_9^1$	$S \oplus M_9^1$	$S \odot M_9^2$	$S \oplus M_9^2$	$S \odot M_9^3$	$S \oplus M_9^3$
solved	<b>14933</b>	16574	20366	21564	22839	22413	23467	22910	23753
$S\%$	+0%	+10.5%	+35.8%	+43.8%	+52.3%	+49.4%	+56.5%	+52.8%	+58.4
$S+$	+0	+4364	+6215	+7774	+8414	+8407	+8964	+8822	+9274
$S-$	-0	-2723	-782	-1143	-508	-927	-430	-845	-454

	$S \odot M_{12}^3$	$S \oplus M_{12}^3$	$S \odot M_{16}^3$	$S \oplus M_{16}^3$
solved	24159	24701	25100	<b>25397</b>
$S\%$	+61.1%	+64.8%	+68.0%	<b>+70.0%</b>
$S+$	+9761	+10063	+10476	+10647
$S-$	-535	-295	-309	-183

# TacticToe: mid-level ITP Guidance (Gauthier'17,18)



- TTT learns from human and its own tactical HOL4 proofs
- No translation or reconstruction needed - native tactical proofs
- Fully integrated with HOL4 and easy to use
- Similar to rICoP: policy/value learning for applying tactics in a state
- However much more technically challenging - a real breakthrough:
  - tactic and goal state recording
  - tactic argument abstraction
  - absolutization of tactic names
  - nontrivial evaluation issues
  - these issues have often more impact than adding better learners
- policy: which tactic/parameters to choose for a current goal?
- value: how likely is this proof state succeed?
- 66% of HOL4 toplevel proofs in 60s (**better than a hammer!**)
- similar recent work for Isabelle (Nagashima 2018), HOL Light (Google)

# Conjecturing/Proof Synthesis by Neural Models

- Karpathy'15 - RNN experiments with generating fake Math over Stacks
- I have tried to use that for formal math in 2016 but it looked weak
- GPT (-2,3) looks stronger
- Renewed experiments in 2020 on formal libraries and ATP proofs
- The models show lack of semantic understanding, but still quite interesting
- Typically they can learn various syntactic tasks quite well

# Can you find the flaw(s) in this fake GPT-2 proof?

```
Applications Places emacs@dell Wed 15:02 Wed 15:02
File Edit Options Buffers Tools Index Mizar Hide/Show Help
Save Undo
:: generated theorem with "proof"
theorem Th23: :: STIRL2_1:23
for X, Y being finite set st not X is empty & X c= Y
& card X = card Y holds X = Y
proof
  let X, Y be finite set ;
  :: thesis: not X is empty & X c= Y & card X = card Y implies X = Y
  assume that
  A1: not X is empty and A2: X c= Y and A3: card X = card Y ;
  :: thesis: X = Y
  card (Y \ X) = (card Y) - (card X) by A1, A3, CARD_2:44;
  then A4: card (Y \ X) = ((card Y) - 1) - (card X) by CARD_1:30;
  X = Y \ X by A2, A3, Th22;
  hence X = Y by A4, XBOOLE_0:def_10;
  :: thesis: verum
end;
-:--- card_tst.miz 99% L2131 (Mizar Errors:13 hs Undo-Tree)
```

Figure: Fake full declarative GPT-2 “proof” - typechecks!

# A correct GPT conjecture that was too hard to prove

Original Mizar theorem stated for finite groups:

```
theorem Th10: :: GROUPE_1:10
  for G being finite Group for N being normal Subgroup of G
  st N is Subgroup of center G & G ./ N is cyclic holds
  G is commutative
```

Kinyon and Stanovsky (algebraists) confirmed that this GPT generalization that **avoids finiteness** is valid:

```
for G being Group for N being normal Subgroup of G
  st N is Subgroup of center G & G ./ N is cyclic holds
  G is commutative
```



# Prover9 - Research-Level Open Conjectures

- Michal Kinyon, Bob Veroff and Prover9: quasigroup and loop theory
- the **Abelian Inner Mappings** (AIM) Conjecture (>10 year program)
- The Weak AIM Conjecture **positively resolved in August 2021**
- 20-200k long proofs by Prover9 assisting the humans
- Prover9 hints strategy (Bob Veroff): extract hints from easier proofs to guide more difficult proofs
- Human-guided exploration to get good hints (not really automated yet)
- Millions of hints collected, various algorithms for their selection for a particular conjecture

# Neural Autoformalization (Wang et al., 2018)



- generate ca 1M Latex/Mizar (**informal/formal**) pairs
- train neural **seq-to-seq translation models** (Luong – NMT)
- evaluate on about 100k examples
- many architectures tested, some work much better than others
- very important latest invention: *attention* in the seq-to-seq models
- more data very important for neural training – our biggest bottleneck
- Recent addition: **unsupervised methods** (Lample et al 2018) – no need for aligned data!

# Neural Autoformalization data

---

Rendered  $\LaTeX$

Mizar

If  $X \subseteq Y \subseteq Z$ , then  $X \subseteq Z$ .

`X c= Y & Y c= Z implies X c= Z;`

Tokenized Mizar

`X c= Y & Y c= Z implies X c= Z ;`

$\LaTeX$

If  $\$X \subseteq Y \subseteq Z\$,$  then  $\$X \subseteq Z\$.$

Tokenized  $\LaTeX$

If  $\$ X \subseteq Y \subseteq Z \$ ,$  then  $\$ X \subseteq Z \$ .$

---

# Neural Fun – Performance after Some Training

Rendered  
L<sup>A</sup>T<sub>E</sub>X

Input L<sup>A</sup>T<sub>E</sub>X

Correct

Snapshot-  
1000

Snapshot-  
2000

Snapshot-  
3000

Snapshot-  
4000

Snapshot-  
5000

Snapshot-  
6000

Snapshot-  
7000

Suppose  $s_8$  is convergent and  $s_7$  is convergent . Then  $\lim(s_8+s_7) = \lim s_8 + \lim s_7$

```
Suppose $ { s _ { 8 } } $ is convergent and $ { s _ { 7 } } $  
$ is convergent . Then $ \mathop { \rm lim } ( { s _ { 8 } }  
{ + } { s _ { 7 } } ) \mathrel { = } \mathop { \rm lim }  
{ s _ { 8 } } { + } \mathop { \rm lim } { s _ { 7 } } $ .
```

```
seq1 is convergent & seq2 is convergent implies lim ( seq1  
+ seq2 ) = ( lim seq1 ) + ( lim seq2 ) ;
```

```
x in dom f implies ( x * y ) * ( f | ( x | ( y | ( y | y )  
 ) ) ) = ( x | ( y | ( y | ( y | y ) ) ) ) ;
```

```
seq is summable implies seq is summable ;
```

```
seq is convergent & lim seq = 0c implies seq = seq ;
```

```
seq is convergent & lim seq = lim seq implies seq1 + seq2  
is convergent ;
```

```
seq1 is convergent & lim seq2 = lim seq2 implies lim_inf  
seq1 = lim_inf seq2 ;
```

```
seq is convergent & lim seq = lim seq implies seq1 + seq2  
is convergent ;
```

```
seq is convergent & seq9 is convergent implies  
lim ( seq + seq9 ) = ( lim seq ) + ( lim seq9 ) ;
```

# Future: AITP Challenges/Bets

- Big challenge: Learn complicated **symbolic algorithms** (not black box)
- 3 AITP bets from my 2014 talk at Institut Henri Poincare
  - In 20 years, 80% of Mizar and Flyspeck toplevel theorems will be provable automatically (same hardware, same libraries as in 2014 - about 40% then)
  - In 10 years: 60% (**DONE** already in 2021)
  - In 25 years, 50% of the toplevel statements in LaTeX-written Msc-level math curriculum textbooks will be **parsed automatically** and with correct formal semantics (this may be **faster** than I expected)
- My (conservative?) estimate when we will do **Fermat**:
  - Human-assisted formalization: by 2050
  - Fully automated proof (hard to define precisely): by 2070
  - See the Foundation of Math thread: <https://bit.ly/300k9Pm>