Some news from the semantic AI paradise

Josef Urban

Czech Technical University in Prague

Mikefest, May 12, 2022, IHES

European Research Council
Established by the European Commission

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

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

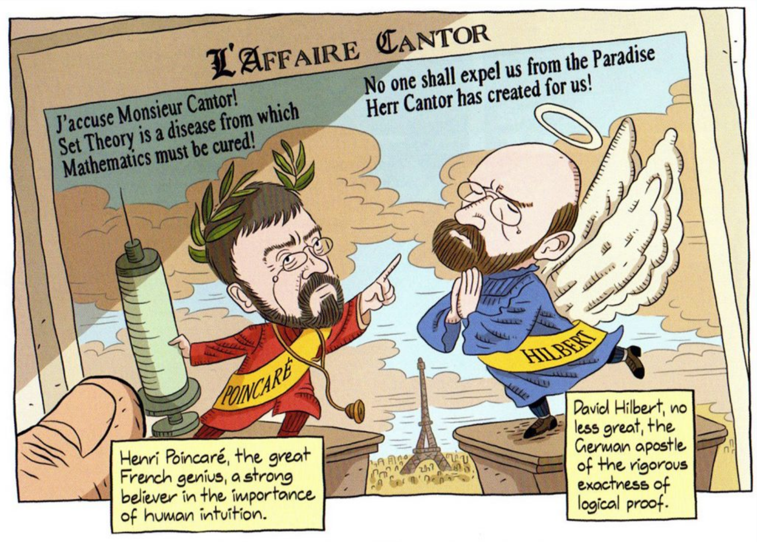
[Adapted from: *Logicomix: An Epic Search for Truth* by A. Doxiadis]

How Do We Automate Math, Science, Programming?

- What is mathematical and scientific thinking?
- Pattern-matching, analogy, induction from examples
- Deductive reasoning
- Complicated feedback loops between induction and deduction
- Using a lot of previous knowledge - both for induction and deduction

- We need to develop such methods on computers
- Are there any large corpora suitable for nontrivial deduction?
- Yes! Large libraries of formal proofs and theories
- So let's develop strong AI on them!

Intuition vs Formal Reasoning – Poincaré vs Hilbert

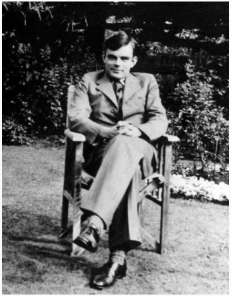


[Adapted from: *Logicomix: An Epic Search for Truth* by A. Doxiadis]

Induction/Learning vs Reasoning – Henri Poincaré

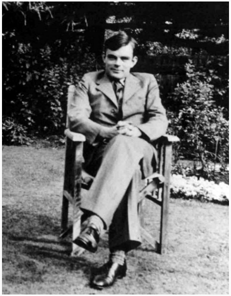
- Science and Method: Ideas about the interplay between correct deduction and induction/intuition
- *"And in demonstration itself logic is not all. The true mathematical reasoning is a real induction [...]"*
- I believe he was right: strong general reasoning engines have to combine deduction and induction (learning patterns from data, making conjectures, etc.)

Learning vs Reasoning – Alan Turing 1950 – AI



- 1950: *Computing machinery and intelligence* – AI, Turing test
- *“We may hope that machines will eventually compete with men in all purely intellectual fields.”* (regardless of his 1936 undecidability result!)
- last section on Learning Machines:
- *“But which are the best ones [fields] to start [learning on] with?”*
- *“... Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best.”*
- Why not try with math? It is much more (universally?) expressive ...

Induction/Learning vs Reasoning – Turing 1950 – AI



- 1950: *Computing machinery and intelligence* – AI, Turing test
- On pure deduction: *"For at each stage when one is using a logical system, there is a very large number of alternative steps, any of which one is permitted to apply, so far as obedience to the rules of the logical system is concerned. These choices make the difference between a brilliant and a footling reasoner, not the difference between a sound and a fallacious one."*

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, ...)
- TP-motivated PLs: ML, Prolog, (*logic programming* - Hayes, Kowalski)
- My MSc (1998): Try ILP to learn explainable rules/heuristics from Mizar
- Since: Do AI/TP over (in)formal math corpora: Mizar, Isabelle, HOL, ...

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 (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 $\sqrt{2}$ (Formal Proof Sketch)

exactly the same text in Mizar syntax:

```
theorem Th43: :: Pythagoras' theorem
  sqrt 2 is irrational
proof
  assume sqrt 2 is rational;
  consider a,b such that
4_3_1: a^2 = 2*b^2 and
    a,b are relative prime;
  a^2 is even;
  a is even;
  consider c such that a = 2*c;
  4*c^2 = 2*b^2;
  2*c^2 = b^2;
  b is even;
  thus contradiction;
end;
```

Irrationality of $\sqrt{2}$ in HOL Light

```
let SQRT_2_IRRATIONAL = prove
 (`~rational(sqrt(&2))`,
  SIMP_TAC[rational; real_abs; SQRT_POS_LE; REAL_POS] THEN
  REWRITE_TAC[NOT_EXISTS_THM] THEN REPEAT GEN_TAC THEN
  DISCH_THEN(CONJUNCTS_THEN2 ASSUME_TAC MP_TAC) THEN
  SUBGOAL_THEN `~((&p / &q) pow 2 = sqrt(&2) pow 2)`
    (fun th -> MESON_TAC[th]) THEN
  SIMP_TAC[SQRT_POW_2; REAL_POS; REAL_POW_DIV] THEN
  ASM_SIMP_TAC[REAL_EQ_LDIV_EQ; REAL_OF_NUM_LT; REAL_POW_LT;
               ARITH_RULE `0 < q <=> ~(q = 0)`] THEN
  ASM_MESON_TAC[NSQRT_2; REAL_OF_NUM_POW; REAL_OF_NUM_MUL; REAL_OF_NUM_EQ]);;
```

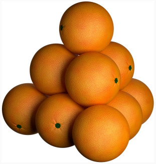
Irrationality of $\sqrt{2}$ in Isabelle/HOL

```
theorem sqrt2_not_rational:
  "sqrt (real 2) ∉ ℚ"
proof
  assume "sqrt (real 2) ∈ ℚ"
  then obtain m n :: nat where
    n_nonzero: "n ≠ 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.

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

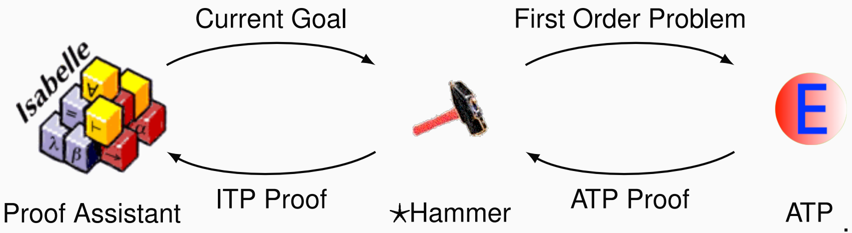


- Proved by Hales in 1998, 300-page proof + computations
- Big: Annals of Mathematics gave up reviewing after 4 years
- Formal proof finished in 2014
- 20000 lemmas in geometry, analysis, graph theory
- All of it at `https://code.google.com/p/flyspeck/`
- All of it computer-understandable and verified 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 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

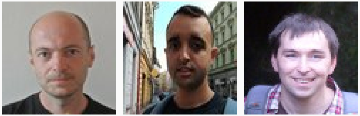
$\approx$ 40-45% success by 2016, 60% on Mizar as of 2021

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`
- Extreme Deepire/AVATAR proof of $\epsilon_0 = \omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}$ `https://bit.ly/3Ne4WNX`
- Hammering demo: `http://grid01.ciirc.cvut.cz/~mptp/out4.ogv`
- TacticToe on HOL4: `http://grid01.ciirc.cvut.cz/~mptp/tactictoe_demo.ogv`
- Tactician for Coq: `https://blaauwbroek.eu/papers/cicm2020/demo.mp4`, `https://coq-tactician.github.io/demo.html`
- Inf2formal over HOL Light: `http://grid01.ciirc.cvut.cz/~mptp/demo.ogv`
- QSynt: AI rediscovers the Fermat primality test: `https://www.youtube.com/watch?v=24oejR9wsXs`

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) used in 2 ways
  - fast logic-aware graph neural network (GNN - Olsak) run on a GPU server
  - logic-based subsumption using fast indexing (discrimination trees - Schulz)
- The GNN scores many clauses (context/query) together in a large graph
- Sparse - vastly more efficient than transformers ($\sim$100k symbols)
- 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: `https://github.com/ai4reason/ATP_Proofs`

	$\mathcal{S}$	$\mathcal{S}\odot\mathcal{M}_9^0$	$\mathcal{S}\oplus\mathcal{M}_9^0$	$\mathcal{S}\odot\mathcal{M}_9^1$	$\mathcal{S}\oplus\mathcal{M}_9^1$	$\mathcal{S}\odot\mathcal{M}_9^2$	$\mathcal{S}\oplus\mathcal{M}_9^2$	$\mathcal{S}\odot\mathcal{M}_9^3$	$\mathcal{S}\oplus\mathcal{M}_9^3$
solved	**14933**	16574	20366	21564	22839	22413	23467	22910	23753
$\mathcal{S}\%$	+0%	+10.5%	+35.8%	+43.8%	+52.3%	+49.4%	+56.5%	+52.8%	+58.4
$\mathcal{S}+$	+0	+4364	+6215	+7774	+8414	+8407	+8964	+8822	+9274
$\mathcal{S}-$	-0	-2723	-782	-1143	-508	-927	-430	-845	-454

	$\mathcal{S}\odot\mathcal{M}_{12}^3$	$\mathcal{S}\oplus\mathcal{M}_{12}^3$	$\mathcal{S}\odot\mathcal{M}_{16}^3$	$\mathcal{S}\oplus\mathcal{M}_{16}^3$
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

ENIGMA Anonymous: Learning from patterns only

- The GNN and GBDTs only learn from formula structure, not symbols
- Not from symbols like $+$ and $*$ as Transformer & Co.
- E.g., learning on additive groups thus transfers to multiplicative groups
- Evaluation of old-Mizar ENIGMA on 242 new Mizar articles
- 13370 new theorems, $>$ 50% of them with new terminology:
- The 3-phase ENIGMA is **58%** better on them than unguided E
- While **53.5%** on the old Mizar (where this ENIGMA was trained)
- Generalizing, analogizing and transfer abilities unusual in the large transformer models

Neural Clause Selection in Vampire (M. Suda)

Deepire: Similar to ENIGMA:

- build a *classifier* for recognizing *good* clauses
- *good* are those that appeared in past proofs

Deepire's contributions:

- Learn from clause *derivation trees only*
  Not looking at what it says, just who its ancestors were.
- Integrate using *layered clause queues*
  A smooth improvement of the base clause selection strategy.
- Tree Neural Networks: constant work per derived clause
- A signature agnostic approach
- Delayed evaluation trick (not all derived need to be evaluated)

Preliminary Evaluation on Mizar "57880"

- Learn from 63595 proofs of 23071 problems (three 30s runs)
- Deepire solves 26217 (i.e. +4054) problems in a *single 10s run*

ENIGMA Proof Example – Knaster

```
theorem Th21:
  ex a st a is_a_fixpoint_of f
proof
  set H = {h where h is Element of L: h [= f.h};
  set fH = {f.h where h is Element of L: h [= f.h};
  set uH = "\/"(H, L);
  set fuH = "\/"(fH, L);
  take uH;
  now
    let fh be Element of L;
    assume fh in fH;
    then consider h being Element of L such that
A1: fh = f.h and
A2: h [= f.h;
    h in H by A2;
    then h [= uH by LATTICE3:38;
    hence fh [= f.uH by A1,QUANTAL1:def 12;
  end;
  then fH is_less_than f.uH by LATTICE3:def 17;
  then
A3: fuH [= f.uH by LATTICE3:def 21;
  now
    let a be Element of L;
    assume a in H;
    then consider h being Element of L such that
A4: a = h & h [= f.h;
    reconsider fh = f.h as Element of L;
    take fh;
    thus a [= fh & fh in fH by A4;
  end;
  then uH [= fuH by LATTICE3:47;
  then
A5: uH [= f.uH by A3,LATTICES:7;
  then f.uH [= f.(f.uH) by QUANTAL1:def 12;
  then f.uH in H;
  then f.uH [= uH by LATTICE3:38;
  hence uH = f.uH by A5,LATTICES:8;
end;
```

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 rlCoP: 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!)
- also for Isabelle (Nagashima), HOL Light (Google), Coq (Blaauwbroek)

Tactician: Tactical Guidance for Coq (Blaauwbroek'20)

- Tactical guidance of Coq proofs
- Technically very challenging to do right - the Coq internals again nontrivial
- 39.3% on the Coq standard library, 56.7% in a union with CoqHammer (orthogonal)
- Fast approximate hashing for k-NN makes a lot of difference
- Speed more important than better learners
- Fully integrated with Coq, should work for any development
- User friendly, installation friendly, integration friendly and maintenance friendly
- Took several years, but could become a very common tool for Coq formalizers

More on Conjecturing and Synthesis in Math

- Targeted: generate intermediate lemmas (cuts) for a harder conjecture
- Unrestricted (theory exploration):
- Creation of interesting conjectures/concepts based on the previous theory
- One of the most interesting activities mathematicians do (how?)
- Higher-level AI/reasoning task - can we learn it?
- If so, we have solved math:
- ... just (recursively) divide Fermat into many subtasks ...
- ... and conquer (I mean: hammer) them away

A bit of history

- The topic goes back at least to Lenat (AM) and Fajtlowicz (Graffiti)
- Combined with automated theorem proving by Colton et al. in early 2000s (HR)
- Theory exploration for Isabelle by Johansson et al (Hipster)
- Several learning-based/neural approaches by our groups since 2015:
- Based mainly on learning analogies and informalization followed by probabilistic/neural disambiguation
- Gauthier's deep RL-based synthesis toolkit in HOL:
  - Guiding synthesis of combinators for a given lambda expression
  - Guiding synthesis of a diophantine equation characterizing a given set
  - Guiding synthesis of programs describing integer sequences (OEIS)

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

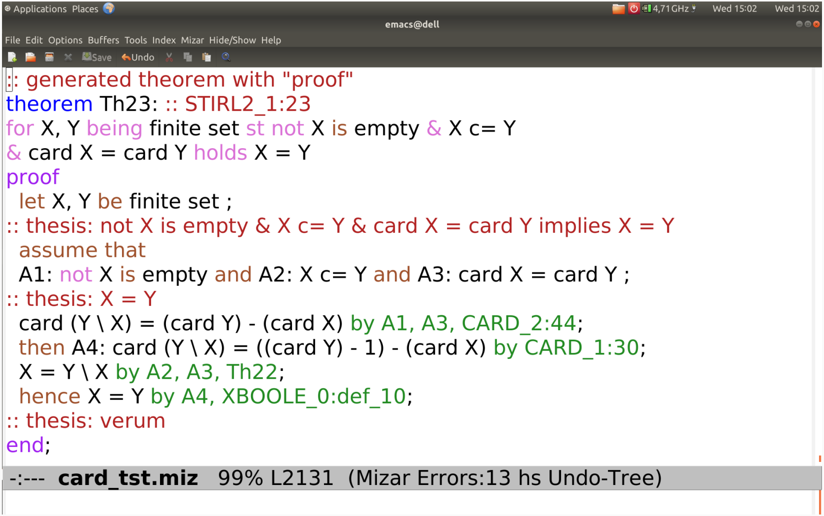


Figure: Fake full declarative GPT-2 “proof” - typechecks!

A correct GPT conjecture that was too hard to prove

Original Mizar theorem stated for finite groups:

```
theorem Th10: :: GROUPP_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
```

Gibberish Generator Provoking Algebraists

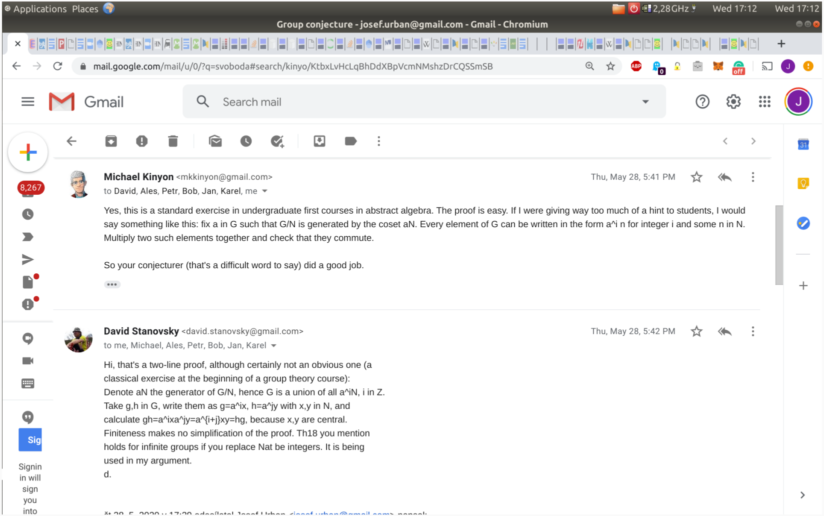


Figure: First successes in making mathematicians comment on AI.

More cuts

- In total 33100 in this experiment
- Ca 9k proved by trained ENIGMA
- Some are clearly false, yet quite natural to ask:

```
theorem :: SINCOS10:17

sec is increasing on [0, pi/2)
```

leads to conjecturing the following:

```
Every differentiable function is increasing.
```

RL for Semantics-Aware Synthesis of Math Objects

- Gauthier'19,20:
- Tree Neural Nets and RL (MCTS, policy/value) for:
- Guiding normalization in Robinson arithmetic
- Guiding synthesis of combinators for a given lambda expression
- Guiding synthesis of a diophantine equation characterizing a given set
- Quite encouraging results with a good curriculum (LPAR, CICM)
- Motivated by his TacticToe: argument synthesis and conjecturing is the big missing piece
- Gauthier's deep RL framework verifies the whole series (proof) in HOL4
- 2022: OEIS invention from scratch - 50k sequences discovered: `https://www.youtube.com/watch?v=24oejR9wsXs`, `http://grid01.ciirc.cvut.cz/~thibault/qsynt.html`
- Non-neural (Turing complete) computing and semantics collaborates with the statistical learning

Prover9 - Research-Level Open Conjectures

- Michal Kinyon, Bob Veroff and Prover9: quasigroup and loop theory
- the Abelian Inner Mappinngs (AIM) Conjecture (>10 year program)
- Strong AIM: Q is AIM implies Q/Nuc(Q) is abelian and Q/Z(Q) is a group
- The Weak AIM Conjecture positively resolved in August 2021
- Q is AIM implies Q is nilpotent of class at most 3.
- 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
- Symbolic machine learning?

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 all 2018) – no need for aligned data!

Neural Autoformalization data

Rendered LaTeX	If $X \subseteq Y \subseteq Z$, then $X \subseteq Z$.
Mizar	`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 LaTeX	Suppose s_8 is convergent and s_7 is convergent . Then $\lim(s_8+s_7) = \lim s_8 + \lim s_7$
Input LaTeX	`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 } } $ .`
Correct	`seq1 is convergent & seq2 is convergent implies lim ( seq1 + seq2 ) = ( lim seq1 ) + ( lim seq2 ) ;`
Snapshot-1000	`x in dom f implies ( x * y ) * ( f | ( x | ( y | ( y | y ) ) ) ) = ( x | ( y | ( y | ( y | y ) ) ) ) ) ;`
Snapshot-2000	`seq is summable implies seq is summable ;`
Snapshot-3000	`seq is convergent & lim seq = 0c implies seq = seq ;`
Snapshot-4000	`seq is convergent & lim seq = lim seq implies seq1 + seq2 is convergent ;`
Snapshot-5000	`seq1 is convergent & lim seq2 = lim seq2 implies lim_inf seq1 = lim_inf seq2 ;`
Snapshot-6000	`seq is convergent & lim seq = lim seq implies seq1 + seq2 is convergent ;`
Snapshot-7000	`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 - 3 years ahead of schedule)
  - 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`

Science, Physics, ..., AI – How did we get here?

- What were Galileo, Kepler & Co trying to do in 1600s?
- What are we trying to do today?
- Kepler's Conjecture in Strena in 1611 (among many other conjectures)
- Kepler's laws, Galileo, Newton, ..., age of science, math, machines
- ..., Poincare, Russell, Von Neumann, Turing, ... age of computing machines?
- 1998 computing machine helps to find a proof of Kepler's Conjecture
- 2014 computing machine verifies a proof of Kepler's Conjecture
- ... 2050? computing machine finds a proof of Kepler's Conjecture?

1600s Prague like today's AI/Singularity Hype

- Rudolf II inviting all sorts artists, alchemist, astrologists (eventually astronomers, geometers, chemists, scientists?)
- Brahe, Dee, Kelly, Kepler, Sendivogius, Rabbi Loew ben Bezalel, ...
- About 40 alchemists at the top - working hard on:
  - searching for the Philosopher's Stone (gold, immortality, and all that)
  - cosmic harmony through numbers and combinations of symbols
  - mysticism, Kabbalah and "Harmony of spheres"
- Dee's claims that world will soon go through a miracle reform
- Kepler's claim that Earth has a soul – subjected to astrological harmony
- Conjecturing about the shape of snowflakes and stacking of spheres?
- What gives the shell of the snail its spiral form?
- Why do most flowers have five petals?
- Explain the Universe through math and geometry? Scientific Revolution?

Scientific Revolution of 1600s: Data, Math, Induction

- Brahe, Kepler, Galileo, Newton, Bacon, Descartes, ..
- Experience Not Only Doctrine! Data-driven science! (*Natural Philosophy*)
- Bacon: not just Aristotelian deduction/syllogisms, but *induction* from data
- *Men have sought to make a world from their own conception and to draw from their own minds all the material which they employed, but if, instead of doing so, they had consulted experience and observation, they would have the facts and not opinions to reason about, and might have ultimately arrived at the knowledge of the laws which govern the material world.*
- Mathematization of philosophy: rebellion by the mathematicians against philosophers
- apparent in Newton's title: Mathematical Principles of Natural Philosophy

Their Science Revolution vs AI Revolution of Today?

- They were obsessed with the nature, its data and conjectures over them
- Produced lots of *observation/experimental data*
- And we are obsessed with our thinking process (AI)
- Pat Langley (PhD at CMU under H. Simon) – the Bacon system (1978):
- *learn the laws*: Kepler's third law , Coulomb's law, Ohm's law, ...
- ... from the observation data
- But they also produced *thinking data* (science and math)
- Today the thinking data is made understandable to our *thinking machines*
- Today's proof libraries are our data, and we are trying to come up with algorithms that *learn our thinking*
- Shall we teach machines to discover the proof of Kepler's conjecture?

Where does this lead us?

- Acceleration of science? - feedback loops between:
- Better observation/exploration, more data, better theories, better thinking tools, even better theories, even better observations, ...
- Is this already some "singularity"?
- Not clear - e.g. in AlphaGo/Zero the feedback loop plateaus at some point
- In a similar way our learning/proving loops currently plateau
- There might be physical limits to our observation and thinking tools limiting how far the feedback loops can get
- But physics as we know it will likely be quickly outdated when automated science takes off
- ... and likely also our current thinking tools
- It’s very hard to believe that our (or our thinking successors’) discovery process of the universe could stop somehow
- At least math seems at the moment unfinishable

AITP and Mike

- To push AI methods in math and theorem proving, in 2022 we organize:
- AITP – Artificial Intelligence and Theorem Proving
- `aitp-conference.org`
- ATP/ITP/Math vs AI/Machine-Learning people, Computational linguists
- and AI for Physics with **Mike Douglas** as a co-chair!
- Happy Birthday and lots of AI/TP-for-Physics fun Mike!