AIAI4AITP: Adventures in Artificial Intelligence for Automated and Interactive Theorem Proving

Josef Urban

Czech Technical University in Prague

Deduction Mentoring Workshop, August 25th, 2019

European Research Council
Established by the European Commission

Outline

Motivation: Learning vs. Reasoning

"C'est par la logique qu'on démontre, c'est par l'intuition qu'on invente."
(It is by logic that we prove, but by intuition that we discover.)
Henri Poincaré, Mathematical Definitions and Education.

"Hypothesen sind Netze; nur der fängt, wer auswirft."
(Hypotheses are nets: only he who casts will catch.)
Novalis, quoted by Popper – The Logic of Scientific Discovery

How Do We Automate Math and Science?

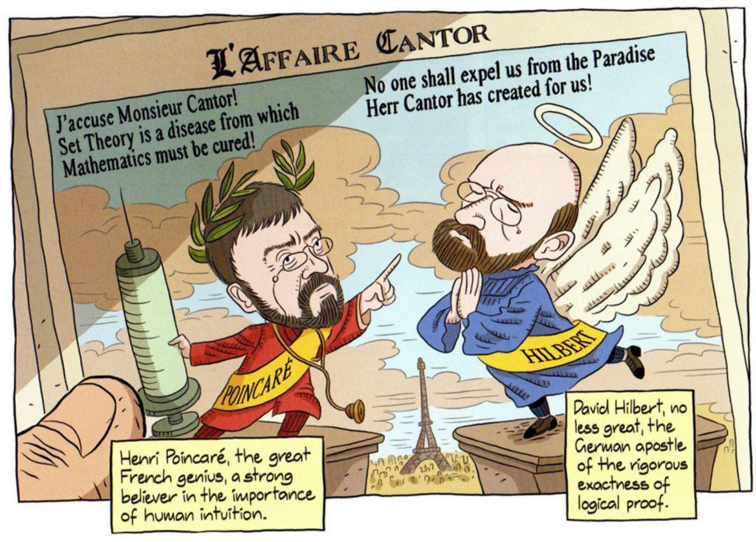
- What is mathematical and scientific thinking?
- Pattern-matching, analogy, induction from examples
- Deductive reasoning
- Complicated feedback loops between induction and deduction
- Using a lot of previous knowledge - both for induction and deduction

- 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!

History, Motivation, AI/TP/ML/DL

- Intuition vs Formal Reasoning – Poincaré vs Hilbert, Science & Method
- Turing's 1950 paper: Learning Machines, learn Chess?, undecidability??
- Lenat, Langley, etc: manually-written heuristics, learn Kepler laws,...
- Denzinger, Schulz, Goller, Fuchs – late 90's, ATP-focused:
- *Learning from Previous Proof Experience*
- My MSc (1998): Try ILP to learn rules and heuristics from IMPS/Mizar
- Since: Use large formal math (Big Proof) corpora: Mizar, Isabelle, HOL
- ... to combine/develop symbolic/statistical deductive/inductive ML/TP/AI
- ... hammer-style methods, feedback loops, internal guidance, ...
- AI vs ML vs DL?: Ben Goertzel's 2018 Prague talk: `https://youtu.be/Zt2HSTuGBn8`

Intuition vs Formal Reasoning – Poincaré vs Hilbert

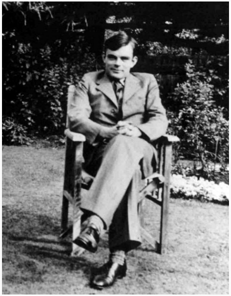


[Adapted from: *Logicomix: An Epic Search for Truth* by A. Doxiadis]

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 ...

Why Combine Learning and Reasoning Today?

1. It practically helps!
    - Automated theorem proving for large formal verification is useful:
        - Formal Proof of the Kepler Conjecture (2014 – Hales – 20k lemmas)
        - Formal Proof of the Feit-Thompson Theorem (2012 – Gonthier)
        - Verification of compilers (CompCert) and microkernels (seL4)
        - ...
    - But good learning/AI methods needed to cope with large theories!
    - Learning is already very useful in guiding longer proof searches.
2. Blue Sky AI Visions:
    - General AI for science must include also Reasoning and Deduction
    - 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:
        - What are the components (inductive/deductive thinking)?
        - How to combine them together?

The Plan

Followed by me for $>$ 20 years

1. Make large “formal thought” accessible to strong reasoning and learning AI tools – DONE or well under way
    - Mizar/MML
    - Isabelle/HOL/AFP
    - HOL Light/Flyspeck
    - HOL4/CakeML
    - Coq, etc.
2. Test/Use/Evolve existing ATP/ML/AI systems on such large corpora
3. Build custom/combined inductive/deductive tools/metasystems
4. Continuously test performance, define harder AI tasks as the performance grows

What is Formal Mathematics and ITP? Why Do It?

- Developed thanks to the Leibniz/Russell/Frege/Hilbert/... program
- Mathematics put on formal logic foundations (*symbolic computation*)
- ... which btw. led also to the rise of computers (Turing/Church, 1930s)
- Formal math (1950/60s): combine formal foundations and the newly available computers
- Today large ITP systems used for verifying nontrivial math, SW, HW ...
- Conceptually very simple:
- Write all your axioms and theorems so that computer understands them
- Write all your inference rules so that computer understands them
- Use the computer to check that your proofs follow the rules
- But in practice, it turns out not to be so simple
- Many approaches, still not mainstream, but big breakthroughs recently

ITP Systems in One Slide by T. Hales

HOL Light

HOL Light has an exquisite minimal design. It has the smallest kernel of any system. John Harrison is the sole

Mizar

Once the clear front-runner, it now shows signs of age. Do not expect
to understand the inner workings of this system unless you have been

Coq

Coq is built of modular components on a foundation of dependent type theory. This system has grown one PhD thesis at a time.

Isabelle

Designed for use with multiple foundational architectures, Isabelle's early
development featured classical constructions in set theory. However,

Metamath

Does this really work? Defying expectations, Metamath seems to function
shockingly well for those who are happy to live without plumbing.

Lean

Lean is ambitious, and it will be massive. Do not be fooled by the name.
"*Construction area keep out*" signs are prominently posted on the perimeter fencing.

The QED Manifesto – 1994

- *QED is the very tentative title of a project to build a computer system that effectively represents all important mathematical knowledge and techniques.*
- *The QED system will conform to the highest standards of mathematical rigor, including the use of strict formality in the internal representation of knowledge and the use of mechanical methods to check proofs of the correctness of all entries in the system.*
- *The QED project will be a major scientific undertaking requiring the cooperation and effort of hundreds of deep mathematical minds, considerable ingenuity by many computer scientists, and broad support and leadership from research agencies.*
- ....
- Never happened, but a lot of inspiration/motivation.

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

tiny proof from Hardy & Wright, collected by F. Wiedijk:

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

$$a^2 = 2b^2 \qquad (4.3.1)$$

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 $(a, b) = 1$. □

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
```

Irrationality of $\sqrt{2}$ in Coq

```
Theorem irrational_sqrt_2: irrational (sqrt 2%nat).
intros p q H H0; case H.
apply (main_thm (Zabs_nat p)).
replace (Div2.double (q * q)) with (2 * (q * q));
 [idtac | unfold Div2.double; ring].
case (eq_nat_dec (Zabs_nat p * Zabs_nat p) (2 * (q * q))); auto; intros H1.
case (not_nm_INR _ _ H1); (repeat rewrite mult_INR).
rewrite <- (sqrt_def (INR 2)); auto with real.
rewrite H0; auto with real.
assert (q <> 0%R :> R); auto with real.
field; auto with real; case p; simpl; intros; ring.
Qed.
```

Irrationality of $\sqrt{2}$ in Metamath

```
${
    $d x y $.
    $( The square root of 2 is irrational. $)
    sqr2irr $p |- ( sqr ` 2 ) e/ QQ $=
      ( vx vy c2 csqr cfv cq wnel wcel wn cv cdiv co wceq cn wrex cz cexp
      cmulc sqr2irrlem3 sqr2irrlem5 bi2rexa mtbir cc0 clt wbr wa wi wb nngt0t
      adantr cr ax0re ltmuldivt mp3an1 nnret zret syl2an mpd ancoms 2re 2pos
      sqrgt0i breq2 mpbii syl5bir cc nncnt mulzer2t syl breq1d adantl sylibd
      exp r19.23adv anc2li elnnz syl6ibr impac r19.22i2 mto elq df-nel mpbir )
      CDEZFGWDFHZIWEWDAJZBJZKLZMZBNOZAPOZWKWJANOZWLWFCQLCWGCQLRLMZBNOANOABSWIWM
      ABNNWFWGTUAUBWJWJAPNWFPHZWJWFNHZWNWJWNUCWFUDUEZUFWOWNWJWPWNWIWPBNWNWGNHZW
      IWPUGWNWQUFZWIUCWGRLZWFUDUEZWPWRWTUCWHUDUEZWIWQWNWTXAUHZWQWNUFUCWGUDUEZXB
      WQXCWNWGUIUJWGUKHZWFUKHZXCXBUGZWQWNUCUKHXDXEXFULUCWGWFUMUNWGUOWFUPUQURUSW
      IUCWDUDUEXACUTVAVBWDWHUCUDVCVDVEWQWTWPUHWNWQWSUCWFUDWQWGVFHWSUCMWGVGWGVHV
      IVJVKVLVMVNVOWFVPVQVRVSVTABWDWAUBWDFWBWC $.
      $( [8-Jan-02] $)
  $}
```

Irrationality of $\sqrt{2}$ in Metamath Proof Explorer

sqr2irr - Metamath Proof Explorer - Chromium

sqr2irr - Metamat ×

us.metamath.org/mpegif/sqr2irr.html

Proof of Theorem sqr2irr

Step	Hyp	Ref	Expression
1		sqr2irrlem3 10838	$\vdash \neg \exists x \in \mathbb{N}\, \exists y \in \mathbb{N}\, (x \uparrow 2) = (2 \cdot (y \uparrow 2))$
2		sqr2irrlem5 10840	$\vdash ((x \in \mathbb{N} \wedge y \in \mathbb{N}) \to ((\surd\text{‘}2) = (x / y) \leftrightarrow (x \uparrow 2) = (2 \cdot (y \uparrow 2))))$
3	2	2rexbiia 2329	$\vdash (\exists x \in \mathbb{N}\, \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y) \leftrightarrow \exists x \in \mathbb{N}\, \exists y \in \mathbb{N}\, (x \uparrow 2) = (2 \cdot (y \uparrow 2)))$
4	1, 3	mtbir 288	$\vdash \neg \exists x \in \mathbb{N}\, \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y)$
5		2re 8838	$\vdash 2 \in \mathbb{R}$
6		2pos 8849	$\vdash 0 < 2$
7	5, 6	sqrgt0ii 10213	$\vdash 0 < (\surd\text{‘}2)$
8		breq2 3595	$\vdash ((\surd\text{‘}2) = (x / y) \to (0 < (\surd\text{‘}2) \leftrightarrow 0 < (x / y)))$
9	7, 8	mpbii 200	$\vdash ((\surd\text{‘}2) = (x / y) \to 0 < (x / y))$
10		zre 9029	$\vdash (x \in \mathbb{Z} \to x \in \mathbb{R})$
11	10	adantr 444	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to x \in \mathbb{R})$
12		nnre 8788	$\vdash (y \in \mathbb{N} \to y \in \mathbb{R})$
13	12	adantl 445	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to y \in \mathbb{R})$
14		nngt0 8807	$\vdash (y \in \mathbb{N} \to 0 < y)$
15	14	adantl 445	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to 0 < y)$
16		gt0div 8683	$\vdash ((x \in \mathbb{R} \wedge y \in \mathbb{R} \wedge 0 < y) \to (0 < x \leftrightarrow 0 < (x / y)))$
17	11, 13, 15, 16	syl3anc 1145	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to (0 < x \leftrightarrow 0 < (x / y)))$
18	9, 17	syl5ibr 210	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to ((\surd\text{‘}2) = (x / y) \to 0 < x))$
19		simpl 436	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to x \in \mathbb{Z})$
20	18, 19	jctild 522	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to ((\surd\text{‘}2) = (x / y) \to (x \in \mathbb{Z} \wedge 0 < x)))$
21		elnnz 9035	$\vdash (x \in \mathbb{N} \leftrightarrow (x \in \mathbb{Z} \wedge 0 < x))$
22	20, 21	syl6ibr 216	$\vdash ((x \in \mathbb{Z} \wedge y \in \mathbb{N}) \to ((\surd\text{‘}2) = (x / y) \to x \in \mathbb{N}))$
23	22	rexlimdva 2414	$\vdash (x \in \mathbb{Z} \to (\exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y) \to x \in \mathbb{N}))$
24	23	impac 598	$\vdash ((x \in \mathbb{Z} \wedge \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y)) \to (x \in \mathbb{N} \wedge \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y)))$
25	24	reximi2 2396	$\vdash (\exists x \in \mathbb{Z}\, \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y) \to \exists x \in \mathbb{N}\, \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y))$
26	4, 25	mto 165	$\vdash \neg \exists x \in \mathbb{Z}\, \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y)$
27		elq 9308	$\vdash ((\surd\text{‘}2) \in \mathbb{Q} \leftrightarrow \exists x \in \mathbb{Z}\, \exists y \in \mathbb{N}\, (\surd\text{‘}2) = (x / y))$
28	26, 27	mtbir 288	$\vdash \neg (\surd\text{‘}2) \in \mathbb{Q}$
29		df-nel 2210	$\vdash ((\surd\text{‘}2) \notin \mathbb{Q} \leftrightarrow \neg (\surd\text{‘}2) \in \mathbb{Q})$
30	28, 29	mpbir 198	$\vdash (\surd\text{‘}2) \notin \mathbb{Q}$

Colors of variables: wff set class

Irrationality of $\sqrt{2}$ in Otter

Problem

```
set(auto).
set(ur_res).
assign(max_distinct_vars, 1).
list(usable).
x = x.
m(1,x) = x.  %identity
m(x,1) = x.
m(x,m(y,z)) = m(m(x,y),z).   %assoc
m(x,y) = m(y,x).             %comm
m(x,y) != m(x,z) | y = z.   %cancel
-d(x,y) | m(x,f(x,y)) = y.  %divides
m(x,z) != y | d(x,y).
-d(2,m(x,y)) | d(2,x) | d(2,y). %2 prime
m(a,a) = m(2,m(b,b)).  % a/b=sqrt(2)
-d(x,a) | -d(x,b) | x = 1. % a/b lowest
2 != 1.
end_of_list.
```

Proof

```
1 [] m(x,y)!=m(x,z)|y=z.
2 [] -d(x,y)|m(x,f(x,y))=y.
3 [] m(x,y)!=z|d(x,z).
4 [] -d(2,m(x,y))|d(2,x)|d(2,y).
5 [] -d(x,a)| -d(x,b)|x=1.
6 [] 2!=1.
7 [factor,4.2.3] -d(2,m(x,x))|d(2,x).
13 [] m(x,m(y,z))=m(m(x,y),z).
14 [copy,13,flip.1] m(m(x,y),z)=m(x,m(y,
16 [] m(x,y)=m(y,x).
17 [] m(a,a)=m(2,m(b,b)).
18 [copy,17,flip.1] m(2,m(b,b))=m(a,a).
30 [hyper,18,3] d(2,m(a,a)).
39 [para_from,18.1.1,1.1.1] m(a,a)!=m(2,
42 [hyper,30,7] d(2,a).
46 [hyper,42,2] m(2,f(2,a))=a.
48 [ur,42,5,6] -d(2,b).
50 [ur,48,7] -d(2,m(b,b)).
59 [ur,50,3] m(2,x)!=m(b,b).
60 [copy,59,flip.1] m(b,b)!=m(2,x).
145 [para_from,46.1.1,14.1.1.1,flip.1] m
189 [ur,60,39] m(a,a)!=m(2,m(2,x)).
190 [copy,189,flip.1] m(2,m(2,x))!=m(a,a
1261 [para_into,145.1.1.2,16.1.1] m(2,m(
1272 [para_from,145.1.1,190.1.1.2] m(2,m
1273 [binary,1272.1,1261.1] $F.
```

Today: Computers Checking Large Math Proofs

SCI NEWS

Top Headlines: Researchers Find 40,000-Year-Old Figurative Paintings in Bornean Cave

HOME | ASTRONOMY | SPACE EXPLORATION | ARCHAEOLOGY | PALEONTOLOGY | BIOLOGY | PHYSICS | MEDICINE | GENETICS | GEOLOGY | MORE

Scientists Deliver Formal Proof of Famous Kepler Conjecture

Jun 16, 2017 by News Staff / Source « Previous | Next »

Published in
Mathematics

Tagged as
Johannes Kepler
Kepler conjecture

Follow
You Might Like

Researchers Develop First-Ever 3D Numerical Model of Melting Snowflake

Researchers Develop Mathematical Model for How Innovations Emerge

An international team of mathematicians led by University of Pittsburgh Professor Thomas Hales has delivered a formal proof of the Kepler conjecture, a famous problem in discrete geometry. The team's paper is published in the journal *Forum of Mathematics, Pi*.

cdn.sci-news.com/images/enlarge3/image_4960e-Kepler-Conjecture.jpg

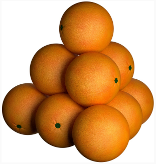
LATEST NEWS

- SPHERE Captures Young Exoplanet Beta Pictoris b Orbiting around Its Star — Nov 13, 2018 | Astronomy
- Mirarce eatoni: Newly-Discovered Cretaceous Bird Lived Among Dinosaurs, Was Strong Flier — Nov 13, 2018 | Paleontology
- Juno Takes Closer Look at Jupiter's Magnificent, Swirling Clouds — Nov 13, 2018 | Space Exploration
- Physicists Solve Structure of Unusually Complex Form of Nitrogen — Nov 13, 2018 | Physical Chemistry
- Natural Compound Protects Hypertensive Rats against Heart Disease — Nov 13, 2018 | Medicine
- Inventive Orangutans Make Hook Tools to Retrieve Food — Nov 12, 2018 | Biology
- Researchers Find 40,000-Year-Old Figurative Paintings in Bornean Cave — Nov 12, 2018 | Archaeology
- Hubble Sees Lensing Galaxy Cluster,

Big Example: The Flyspeck project

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

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



- 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!

Big Math Formalizations

- Kepler Conjecture (Hales et all, 2014, HOL Light, Isabelle)
- Feit-Thompson (odd-order) theorem
  - Two graduate books
  - Gonthier et all, 2012, Coq
- The Four Color Theorem (Gonthier and Werner, 2005, Coq)
- Compendium of Continuous Lattices (CCL)
  - 60% of the book formalized in Mizar
  - Bancerek, Trybulec et all, 2003

Mid-size Formalizations

- Gödel's First Incompleteness Theorem — Natarajan Shankar (NQTHM), Russell O'Connor (Coq)
- Brouwer Fixed Point Theorem — Karol Pak (Mizar), John Harrison (HOL Light)
- Jordan Curve Theorem — Tom Hales (HOL Light), Artur Kornilowicz et al. (Mizar)
- Prime Number Theorem — Jeremy Avigad et al (Isabelle/HOL), John Harrison (HOL Light)
- Gödel's Second incompleteness Theorem — Larry Paulson (Isabelle/HOL)
- Central Limit Theorem – Jeremy Avigad (Isabelle/HOL)
- Consistency of the Negation of CH – Jesse Han and Floris van Doorn (Lean, 2019)

Large Software Verifications

- seL4 – operating system microkernel
  - Gerwin Klein and his group at NICTA, Isabelle/HOL
- CompCert – a formally verified C compiler
  - Xavier Leroy and his group at INRIA, Coq
- EURO-MILS – verified virtualization platform
  - ongoing 6M EUR FP7 project, Isabelle
- CakeML – verified implementation of ML
  - Magnus Myreen, Ramana Kumar and others, HOL4

Central Limit Theorem in Isabelle/HOL

The Top 100 Theorems in Isabelle - Chromium

www.cse.unsw.edu.au/~kleing/top100/#47

```
theorem (in prob_space) central_limit_theorem:
  fixes
    X :: "nat ⇒ 'a ⇒ real" and
    μ :: "real measure" and
    σ :: real and
    S :: "nat ⇒ 'a ⇒ real"
  assumes
    X_indep: "indep_vars (λi. borel) X UNIV" and
    X_integrable: "⋀n. integrable M (X n)" and
    X_mean_0: "⋀n. expectation (X n) = 0" and
    σ_pos: "σ > 0" and
    X_square_integrable: "⋀n. integrable M (λx. (X n x)²)" and
    X_variance: "⋀n. variance (X n) = σ²" and
    X_distrib: "⋀n. distr M borel (X n) = μ"
  defines
    "S n ≡ λx. ∑i<n. X i x"
  shows
    "weak_conv_m (λn. distr M borel (λx. S n x / sqrt (n * σ²)))
        (density lborel std_normal_density)"
```

Sylow's Theorems in Mizar

```
theorem :: GROUP_10:12
  for G being finite Group, p being prime (natural number)
  holds ex P being Subgroup of G st P is_Sylow_p-subgroup_of_prime p;

theorem :: GROUP_10:14
  for G being finite Group, p being prime (natural number) holds
    (for H being Subgroup of G st H is_p-group_of_prime p holds
      ex P being Subgroup of G st
      P is_Sylow_p-subgroup_of_prime p & H is Subgroup of P) &
    (for P1,P2 being Subgroup of G
      st P1 is_Sylow_p-subgroup_of_prime p & P2 is_Sylow_p-subgroup_of_prime p
      holds P1,P2 are_conjugated);

theorem :: GROUP_10:15
  for G being finite Group, p being prime (natural number) holds
    card the_sylow_p-subgroups_of_prime(p,G) mod p = 1 &
    card the_sylow_p-subgroups_of_prime(p,G) divides ord G;
```

Gödel Theorems in Isabelle

The Top 100 Theorems in Isabelle - Chromium

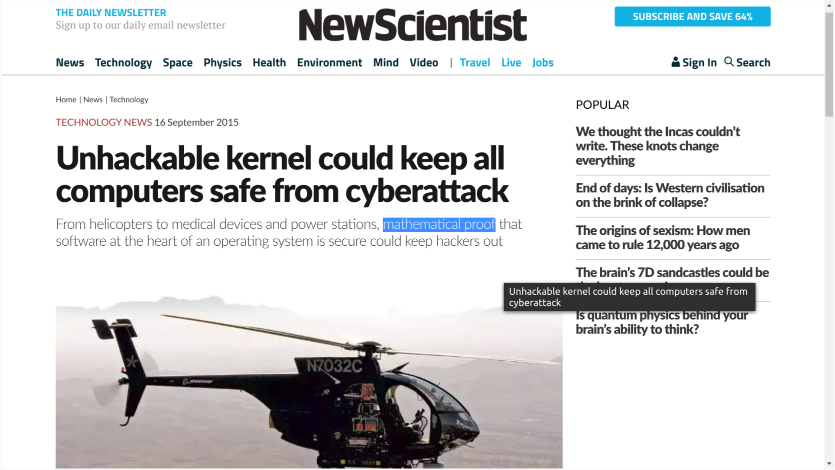
www.cse.unsw.edu.au/~kleing/top100/#6

```
theorem Goedel_I:
  assumes "¬ {} ⊢ Fls"
  obtains δ where
      "{} ⊢ δ IFF Neg (PfP ⌈δ⌉)"
      "¬ {} ⊢ δ"
      "¬ {} ⊢ Neg δ"
      "eval_fm e δ"
      "ground_fm δ"

theorem Goedel_II:
  assumes "¬ {} ⊢ Fls"
    shows "¬ {} ⊢ Neg (PfP ⌈Fls⌉)"
```

http://afp.sourceforge.net/entries/Incompleteness.shtml

Today’s Applications



THE DAILY NEWSLETTER
Sign up to our daily email newsletter

NewScientist

SUBSCRIBE AND SAVE 64%

News Technology Space Physics Health Environment Mind Video | Travel Live Jobs

Sign In Search

Home | News | Technology

TECHNOLOGY NEWS 16 September 2015

Unhackable kernel could keep all computers safe from cyberattack

From helicopters to medical devices and power stations, mathematical proof that software at the heart of an operating system is secure could keep hackers out

POPULAR

We thought the Incas couldn’t write. These knots change everything

End of days: Is Western civilisation on the brink of collapse?

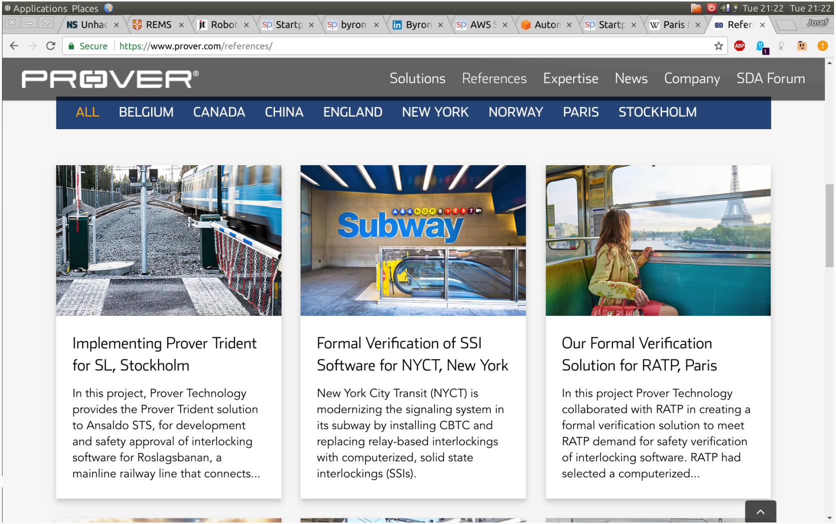
The origins of sexism: How men came to rule 12,000 years ago

The brain’s 7D sandcastles could be

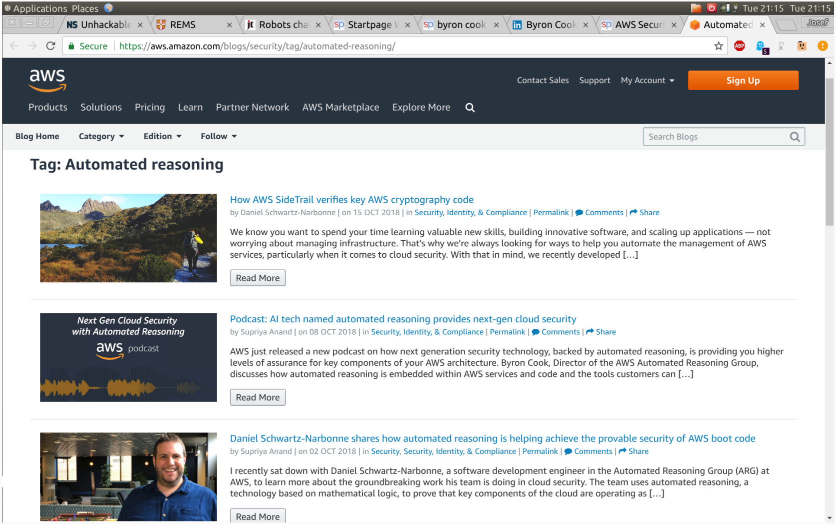
Unhackable kernel could keep all computers safe from cyberattack

Is quantum physics behind your brain’s ability to think?

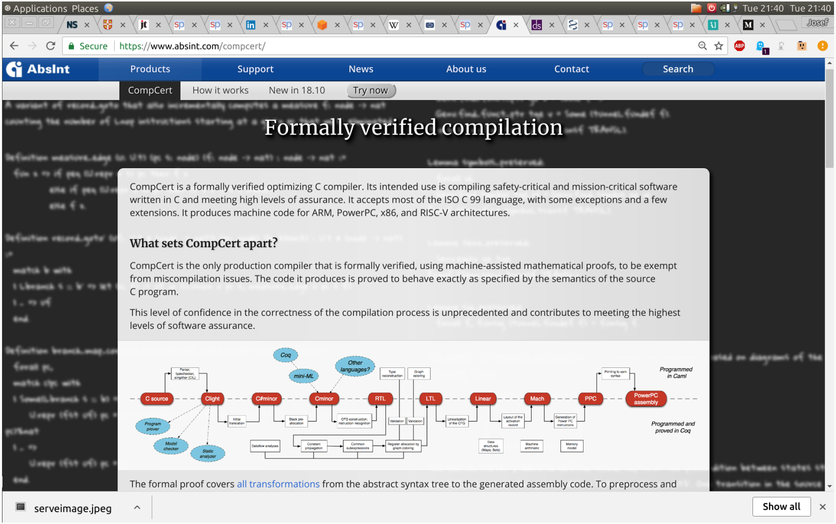
Today’s Applications



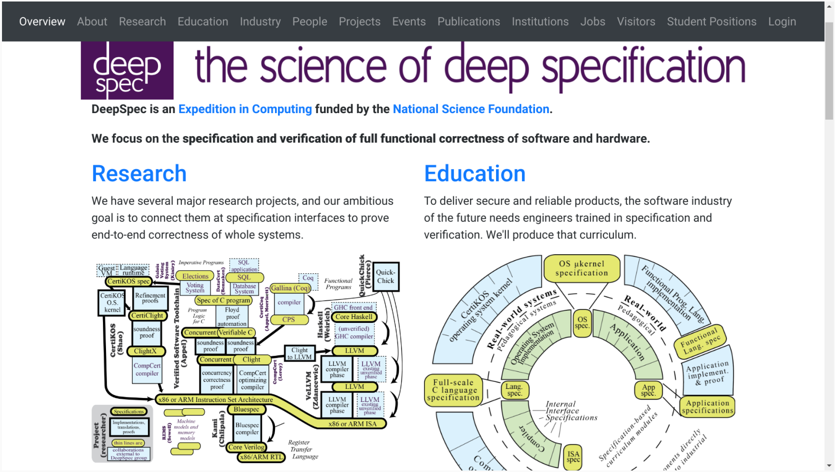
Today's Applications



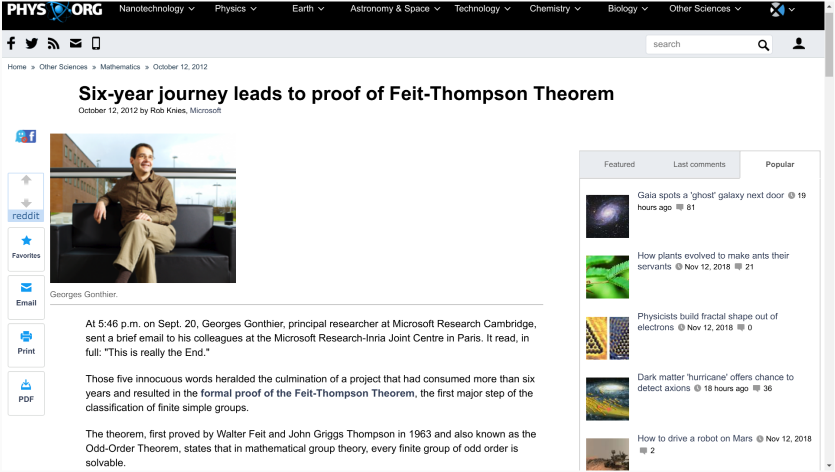
Today's Applications



Today’s Applications



Today’s Applications



PHYS ORG Nanotechnology Physics Earth Astronomy & Space Technology Chemistry Biology Other Sciences

search

Home » Other Sciences » Mathematics » October 12, 2012

Six-year journey leads to proof of Feit-Thompson Theorem

October 12, 2012 by Rob Knies, Microsoft

reddit Favorites Email Print PDF

Georges Gonthier.

At 5:46 p.m. on Sept. 20, Georges Gonthier, principal researcher at Microsoft Research Cambridge, sent a brief email to his colleagues at the Microsoft Research-Inria Joint Centre in Paris. It read, in full: "This is really the End."

Those five innocuous words heralded the culmination of a project that had consumed more than six years and resulted in the **formal proof of the Feit-Thompson Theorem**, the first major step of the classification of finite simple groups.

The theorem, first proved by Walter Feit and John Griggs Thompson in 1963 and also known as the Odd-Order Theorem, states that in mathematical group theory, every finite group of odd order is solvable.

Featured | Last comments | Popular

Gaia spots a 'ghost' galaxy next door 19 hours ago 81

How plants evolved to make ants their servants Nov 12, 2018 21

Physicists build fractal shape out of electrons Nov 12, 2018 0

Dark matter 'hurricane' offers chance to detect axions 18 hours ago 36

How to drive a robot on Mars Nov 12, 2018 2

The AI Part: Learning to Guide Theorem Proving

- How do we use all these corpora to learn doing math automatically?
- How can we combine AI methods with existing ATP systems?
- How do we practically assist formalization?

Using Learning to Guide 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
- ...

Demos

- Hammering Mizar: `http://grid01.ciirc.cvut.cz/~mptp/out4.ogv`
- TacticToe on HOL4: `http://grid01.ciirc.cvut.cz/~mptp/tactictoe_demo.ogv`
- Inf2formal over HOL Light: `http://grid01.ciirc.cvut.cz/~mptp/demo.ogv`
- TacticToe longer (Thibault's PxTP talk!): `https://www.youtube.com/watch?v=BO4Y8ynwT6Y`

Sample of Learning Approaches

- **neural networks** (statistical ML) – backpropagation, deep learning, convolutional, recurrent, graph neural nets, etc.
- **decision trees, random forests** – find good classifying attributes (and/or their values); more explainable
- **support vector machines** – find a good classifying hyperplane, possibly after non-linear transformation of the data (*kernel methods*)
- **k-nearest neighbor** – find the k nearest neighbors to the query, combine their solutions
- **naive Bayes** – compute probabilities of outcomes assuming complete (naive) independence of characterizing features (just multiplying probabilities)
- **inductive logic programming** (symbolic ML) – generate logical explanation (program) from a set of ground clauses by generalization
- **genetic algorithms** – evolve large population by crossover and mutation
- various combinations of statistical and symbolic approaches
- supervised, unsupervised, reinforcement learning (actions, explore/exploit, cumulative reward)

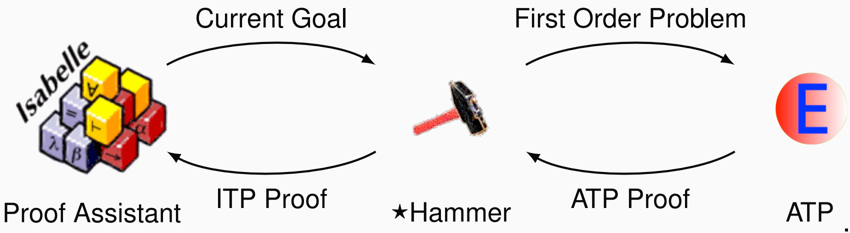
Learning – Features and Data Preprocessing

- Extremely important - *garbage in garbage out*
- Distributed repres., Deep Learning – design (neural) architectures that automatically find important high-level features for a task
- How do we represent math objects (formulas, proofs, ideas) in our mind?
  - From syntactic to more semantic:
  - Constant and function symbols
  - Walks in the term graph
  - Walks in clauses with polarity and variables/skolems unified
  - Subterms, de Bruijn normalized
  - Subterms, all variables unified
  - Matching terms, no generalizations
  - terms and (some of) their generalizations
  - Substitution tree nodes
  - All unifying terms
  - LSI/PCA, word2vec, fasttext, etc.
  - Neural embeddings: CNN, RNN, Tree NN, Graph CNN, ...
  - Evaluation in a large set of (finite) models
  - Vectors of proof similarities (proof search hidden states)
  - Vectors of problems solved (for ATP strategies)

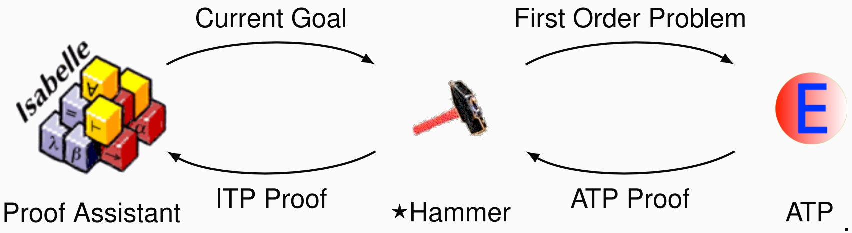
Early Machine Learning for Fact Selection over Mizar

- 2003: Can existing ATPs (E, SPASS, Vampire) be used on the freshly translated Mizar library?
- About 80000 nontrivial math facts at that time – impossible to use them all
- Mizar Proof Advisor (2003):
- Learn fact selection from previous proof!
- Recommend relevant premises when proving new conjectures
- Give them to existing ATPs
- First results over the whole Mizar library in 2003:
  - about 70% coverage in the first 100 recommended premises
  - chain the recommendations with strong ATPs to get full proofs
  - about 14% of the Mizar theorems were then automatically provable (SPASS)
  - sometimes we can find simpler proofs!
- Done with much more developed tools for Flyspeck in 2012, Mizar, HOL4, Coq, ...

Today's AI-ATP systems (⋆-Hammers)

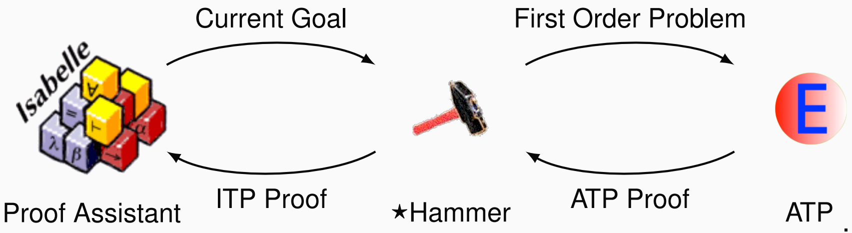


Today's AI-ATP systems (⋆-Hammers)



How much can it do?

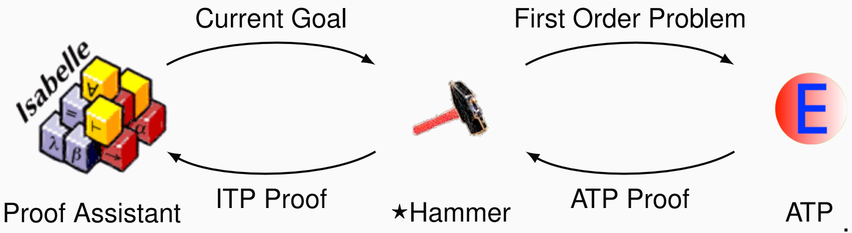
Today's AI-ATP systems (⋆-Hammers)



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

Today's AI-ATP systems (⋆-Hammers)



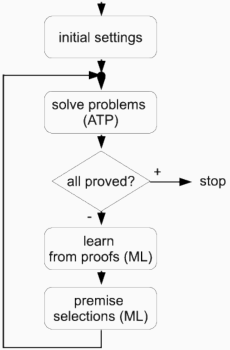
How much can it do?

- 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

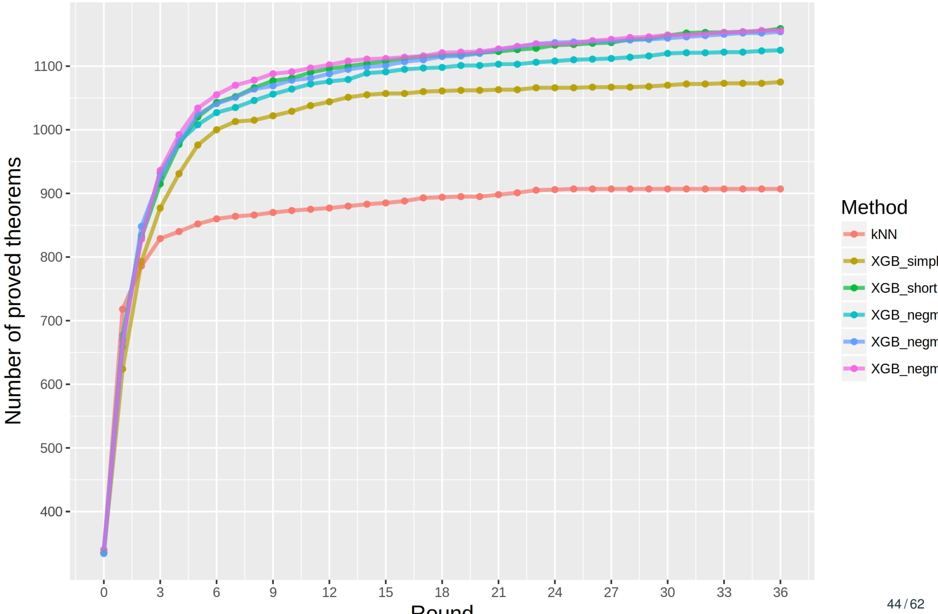
$\approx$ 45% success rate

High-level feedback loops – MALARea

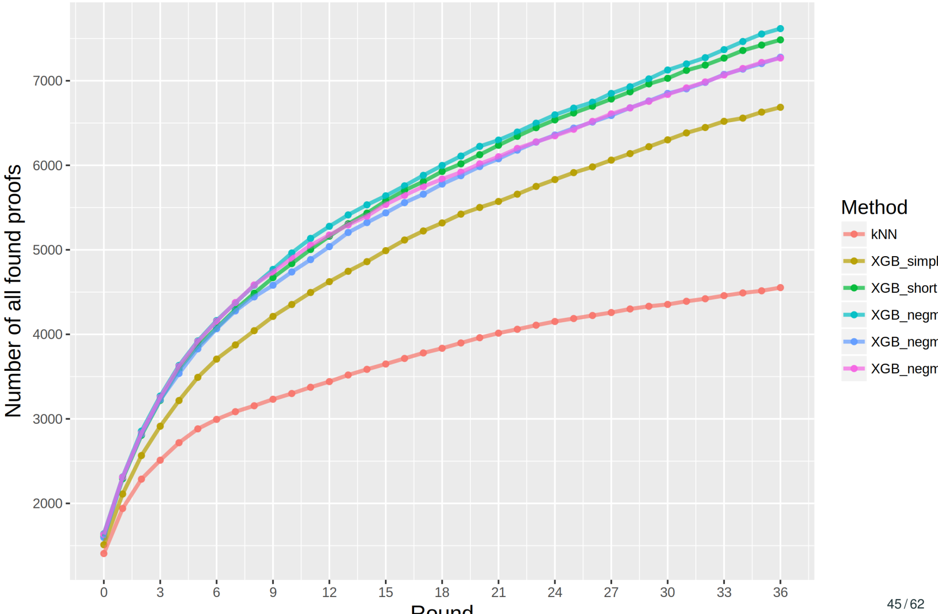
- Machine Learner for Autom. Reasoning (2006) – infinite hammering
- feedback loop interleaving ATP with learning premise selection
- both syntactic and semantic features for characterizing formulas:
- evolving set of finite (counter)models in which formulas evaluated
- ATPBoost (Piotrowski) - recent incarnation focusing on multiple proofs



Prove-and-learn loop on MPTP2078 data set



Prove-and-learn loop on MPTP2078 data set



Low-level: Statistical Guidance of Connection Tableau

- learn guidance of every clausal inference in connection tableau (leanCoP)
- set of first-order clauses, *extension* and *reduction* steps
- proof finished when all branches are closed
- a lot of nondeterminism, requires backtracking
- *Iterative deepening* used in leanCoP to ensure completeness
- good for learning – the tableau compactly represents the proof state

Clauses:

$c_1 : P(x)$

$c_2 : R(x, y) \lor \neg P(x) \lor Q(y)$

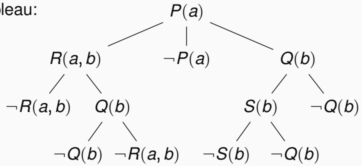
$c_3 : S(x) \lor \neg Q(b)$

$c_4 : \neg S(x) \lor \neg Q(x)$

$c_5 : \neg Q(x) \lor \neg R(a, x)$

$c_6 : \neg R(a, x) \lor Q(x)$

Closed Connection Tableau:



Statistical Guidance of Connection Tableau – rlCoP

- **MaLeCoP** (2011): first prototype Machine Learning Connection Prover
- Fairly Efficient MaLeCoP = **FEMaLeCoP** (15% better than leanCoP)
- 2018: stronger learners via C interface to OCAML (boosted trees)
- remove iterative deepening, the prover can go deep (completeness bad!)
- Monte-Carlo Tree Search (MCTS) governs the search (AlphaGo/Zero!)
- MCTS search nodes are sequences of clause application
- a good heuristic to explore new vs exploit good nodes:

$$\frac{w_i}{n_i} + c \cdot p_i \cdot \sqrt{\frac{\ln N}{n_i}} \qquad \text{(UCT - Kocsis, Szepesvari 2006)}$$

- learning both *policy* (clause selection) and *value* (state evaluation)
- clauses represented not by names but also by features (generalize!)
- binary learning setting used: | proof state | clause features |
- mostly term walks of length 3 (trigrams), hashed into small integers
- many iterations of proving and learning

Statistical Guidance of Connection Tableau – rlCoP

- On 32k Mizar40 problems using 200k inference limit
- nonlearning CoPs:

System	leanCoP	bare prover	rlCoP no policy/value (UCT only)
Training problems proved	10438	4184	7348
Testing problems proved	**1143**	431	804
Total problems proved	11581	4615	8152

- rlCoP with policy/value after 5 proving/learning iters on the training data
- $1624/1143 = 42.1$% improvement over leanCoP on the testing problems

Iteration	1	2	3	4	5	6	7	8
Training proved	12325	13749	14155	14363	14403	14431	14342	**14498**
Testing proved	1354	1519	1566	1595	**1624**	1586	1582	1591

TacticToe: Tactic Guidance of ITPs (Gauthier et al.)

- learns from human tactical HOL4 proofs to solve new goals
- no translation or reconstruction needed
- similar to rlCoP: policy/value learning
- however much more technically challenging:
  - tactic and goal state recording
  - tactic argument abstraction
  - absolutization of tactic names
  - nontrivial evaluation issues
- policy: which tactic/parameters to choose for a current goal?
- value: how likely is this proof state succeed?
- 2018: 66% of HOL4 toplevel proofs in 60s (better than a hammer!)
- work in progress for Coq

Side Note on Symbolic Learning with NNs

- Recurrent NNs with attention recently very good at the inf2formal task
- Experiments with using them for symbolic rewriting (Piotrowski et. al)
- We can learn rewrite rules from sufficiently many data
- 80-90% on algebra datasets, 70-99% on normalizing polynomials
- again, complements symbolic methods like ILP that suffer if too much data
- Similar use for conjecturing (Chvalovsky et al):
- Learn *consistent translations* between different math contexts:
- additive groups → multiplicative groups

Side Note on Symbolic Learning with NNs

Table: Examples in the AIM data set.

Rewrite rule:	Before rewriting:	After rewriting:
`b(s(e,v1),e)=v1`	`k(b(s(e,v1),e),v0)`	`k(v1,v0)`
`o(V0,e)=V0`	`t(v0,o(v1,o(v2,e)))`	`t(v0,o(v1,v2))`

Table: Examples in the polynomial data set.

Before rewriting:	After rewriting:
`(x * (x + 1)) + 1`	`x ^ 2 + x + 1`
`(2 * y) + 1 + (y * y)`	`y ^ 2 + 2 * y + 1`
`(x + 2) * ((2 * x) + 1) + (y + 1)`	`2 * x ^ 2 + 5 * x + y + 3`

Side Note on Conjecturing with RNNs

We can obtain a new valid automatically provable lemma

$(X \cap Y) \backslash Z = (X \backslash Z) \cap (Y \backslash Z)$

from

$(X \cup Y) \backslash Z = (X \backslash Z) \cup (Y \backslash Z)$

Examples of false but syntactically consistent conjectures:

```
for n, m being natural numbers holds n gcd m = n div m;

for R being Relation holds
with_suprema(A) <=> with_suprema(inverse_relation(A));
```

Statistical Guidance the Given Clause in E Prover

- harder for learning than tableau
- the proof state are two large heaps of clauses *processed*/*unprocessed*
- 2017: ENIGMA - manual feature engineering (Jakubuv & JU 2017)
- 2017: Deep guidance (neural nets) (Loos et al. 2017)
- both learn on E's proof search traces, put classifier in E
- positive examples: given clauses used in the proof
- negative examples: given clauses not used in the proof
- ENIGMA: fast feature extraction followed by fast/sparse linear classifier
- about 80% improvement on the AIM benchmark
- Deep guidance: convolutional nets - no feature engineering but slow
- ENIGMA-NG: better features and ML, gradient-boosted trees, tree NNs
- NNs made competitive in real-time, boosted trees still best

Feedback loop for ENIGMA on Mizar data

- Similar to rlCoP - interleave proving and learning of ENIGMA guidance
- Done on 57880 Mizar problems very recently
- Ultimately a 70% improvement over the original strategy
- Example Mizar proof found by ENIGMA: `http://grid01.ciirc.cvut.cz/~mptp/7.13.01_4.181.1147/html/knaster#T21`
- Its E-ENIGMA proof: `http://grid01.ciirc.cvut.cz/~mptp/t21_knaster`

	$\mathcal{S}$	$\mathcal{S} \odot \mathcal{M}_9^0$	$\mathcal{S} \oplus \mathcal{M}_9^0$	$\mathcal{S} \odot \mathcal{M}_9^1$	$\mathcal{S} \oplus \mathcal{M}_9^1$	$\mathcal{S} \odot \mathcal{M}_9^2$	$\mathcal{S} \oplus \mathcal{M}_9^2$	$\mathcal{S} \odot \mathcal{M}_9^3$	$\mathcal{S} \oplus \mathcal{M}_9^3$
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

Neural Autoformalization (Wang et al., 2018)

- generate about 1M Latex - Mizar pairs based on Bancerek's work
- 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 (you can help!)
- Recent addition: unsupervised methods (Lample et all 2018) – no need for aligned data!

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 ) ;`

Unsupervised NMT Fun on Short Formulas

```
len <* a *> = 1 ;                    len <* a *> = 1 ;
assume i < len q ;                   i < len q ;
len <* q *> = 1 ;                    len <* q *> = 1 ;
s = apply ( v2 , v1 ast t ) ;        s = apply ( v2 , v1 ) . t ;
s . ( i + 1 ) = tt . ( i + 1 )       s . ( i + 1 ) = tau1 . ( i + 1 )
1 + j <= len v2 ;                    1 + j <= len v2 ;
1 + j + 0 <= len v2 + 1 ;            1 + j + 0 <= len v2 + 1 ;
let i be Nat ;                       i is_at_least_length_of p ;
assume v is_applicable_to t ;        not v is applicable ;
let t be type of T ;                 t is_orientedpath_of v1 , v2 , T ;
a ast t in downarrow t ;             a *' in downarrow t ;
t9 in types a ;                      t '2 in types a ;
a ast t <= t ;                       a *' <= t ;
A is_applicable_to t ;               A is applicable ;
Carrier ( f ) c= B                   support ppf n c= B
u in B or u in { v } ;               u in B or u in { v } ;
F . w in w & F . w in I ;            F . w in F & F . w in I ;
GG . y in rng HH ;                   G0 . y in rng ( H1 ./. y ) ;
a * L = Z_ZeroLC ( V ) ;             a * L = ZeroLC ( V ) ;
not u in { v } ;                     u >> v ;
u <> v ;                             u <> v ;
v - w = v1 - w1 ;                    vw = v1 - w1 ;
v + w = v1 + w1 ;                    v + w = v1 + w1 ;
x in A & y in A ;                    assume [ x , y ] in A ;
```

More Personal Notes

- *Think globally, act locally.* Big dreams about AI, etc. But act by trying small steps/experiments.
- Sometimes need to commit a lot. The Mizar-to-ATP translation took years, but bore a lot of fruit. Today millions USD in Google HOL/RL.
- Follow your dream mercilessly - avoid distractions (stay focused - hard for many smart people). Find/do what you are convinced/passionate about.
- Avoid the "theorem envy". AI is not Math. We want to replace mathematicians, not be them. Always reflect and implement your thinking.
- Many AI improvements me from bringing ideas/systems together: "Automate, automate, automate!"
- Become a hacker. Learn rapid prototyping. Learn to gain maximum info from initial experiments, then iterate. "Experience, not only doctrine".
- Learn at least one high-level symbolic language - lisp, prolog, ml, haskell. At least one scripting language: perl, python, ruby, shell.
- Stay motivated by reading giants of science: Einstein, Poincare, Russel, Heisenberg, Turing, Deutsch, Dawkins ....
- Read good sci-fi: Heinlein, Stephenson, Stroth, ...

More Personal Notes – Conferences, Evaluation

- This is a constant search related to evaluation metrics.
- Good conferences in CS today count more than journals.
- Part of what we do should influence the metrics – value of a theorem?
- In my research several communities: ITP/Formalization, ATP, AI, ML, DL
- Citation counts wildly differ across the communities (ML vs AR vs Math).
- Reviewing wildly differs across the communities.
- I had mixed successes with ATP conferences, more with ITP, IJCAI/AAAI can be hard for new topics.
- The best reviewing processes and open-mindedness I have seen is now in the NIPS/ICLR community (ML).
- They should be focused to deep neural nets. But managed to attract non-neural and even reasoning topics when combined with ML. One of the reasons for their success.
- ERC: currently the best evaluation worldwide. Much deeper than just bean-counting. Inspiration in many ways.
- Today also high-paid research jobs in AI companies/startups (a bubble?).

Acknowledgments

- Prague Automated Reasoning Group http://arg.ciirc.cvut.cz/:
  - Jan Jakubuv, Chad Brown, Martin Suda, Karel Chvalovsky, Bob Veroff, Zar Goertzel, Bartosz Piotrowski, Lasse Blaauwbroek, Martin Smolik, Jiri Vyskocil, Petr Pudlak, David Stanovsky, Krystof Hoder, ...
- HOL(y)Hammer group in Innsbruck:
  - Cezary Kaliszyk, Thibault Gauthier, Michael Faerber, Yutaka Nagashima, Shawn Wang
- ATP and ITP people:
  - Stephan Schulz, Geoff Sutcliffe, Andrej Voronkov, Kostya Korovin, Larry Paulson, Jasmin Blanchette, John Harrison, Tom Hales, Tobias Nipkow, Andrzej Trybulec, Piotr Rudnicki, Adam Pease, ...
- Learning2Reason people at Radboud University Nijmegen:
  - Herman Geuvers, Tom Heskes, Daniel Kuehlwein, Evgeni Tsivtsivadze, ...
- Google Research: Christian Szegedy, Geoffrey Irving, Alex Alemi, Francois Chollet, Sarah Loos
- ... and many more ...
- Funding: Marie-Curie, NWO, ERC

Some References

- ARG ML&R course: `http://arg.ciirc.cvut.cz/teaching/mlr19/index.html`
- C. Kaliszyk: `http://cl-informatik.uibk.ac.at/teaching/ss18/mltp/content.php`
- C. Kaliszyk, J. Urban, H. Michalewski, M. Olsak: Reinforcement Learning of Theorem Proving. CoRR abs/1805.07563 (2018)
- Z. Goertzel, J. Jakubuv, S. Schulz, J. Urban: ProofWatch: Watchlist Guidance for Large Theories in E. CoRR abs/1802.04007 (2018)
- T. Gauthier, C. Kaliszyk, J. Urban, R. Kumar, M. Norrish: Learning to Prove with Tactics. CoRR abs/1804.00596 (2018).
- J. Jakubuv, J. Urban: ENIGMA: Efficient Learning-Based Inference Guiding Machine. CICM 2017: 292-302
- S. M. Loos, G. Irving, C. Szegedy, C. Kaliszyk: Deep Network Guided Proof Search. LPAR 2017: 85-105
- L. Czajka, C. Kaliszyk: Hammer for Coq: Automation for Dependent Type Theory. J. Autom. Reasoning 61(1-4): 423-453 (2018)
- J. C. Blanchette, C. Kaliszyk, L. C. Paulson, J. Urban: Hammering towards QED. J. Formalized Reasoning 9(1): 101-148 (2016)
- G. Irving, C. Szegedy, A. Alemi, N. Eén, F. Chollet, J. Urban: DeepMath - Deep Sequence Models for Premise Selection. NIPS 2016: 2235-2243
- C. Kaliszyk, J. Urban, J. Vyskocil: Efficient Semantic Features for Automated Reasoning over Large Theories. IJCAI 2015: 3084-3090
- J. Urban, G. Sutcliffe, P. Pudlák, J. Vyskocil: MaLARea SG1- Machine Learner for Automated Reasoning with Semantic Guidance. IJCAR 2008: 441-456
- C. Kaliszyk, J. Urban, J. Vyskocil: Automating Formalization by Statistical and Semantic Parsing of Mathematics. ITP 2017: 12-27
- Q. Wang, C. Kaliszyk, J. Urban: First Experiments with Neural Translation of Informal to Formal Mathematics. CoRR abs/1805.06502 (2018)
- J. Urban, J. Vyskocil: Theorem Proving in Large Formal Mathematics as an Emerging AI Field. LNCS 7788, 240-257, 2013.

Thanks and Advertisement

- Thanks for your attention!
- AITP – Artificial Intelligence and Theorem Proving
- March 22–27, 2020, Aussois, France, `aitp-conference.org`
- ATP/ITP/Math vs AI/Machine-Learning people, Computational linguists
- Discussion-oriented and experimental - submit a talk abstract!
- Grown to 80 people in 2019