Machine Learning and Automated Reasoning - Introduction

Josef Urban

Czech Technical University in Prague

March 8, 2019

European Research Council
Established by the European Commission

Course Overview

- Connections between two AI fields: Machine Learning (ML) and Automated Reasoning (AR)
- ML: apply various forms of *inductive reasoning* to large datasets to obtain the most plausible explanations, models and conjectures
- AR: apply various forms of *deductive reasoning* to prove that particular explanations and conjectures are correct.
- Humans combine induction and deduction - let's teach computers too!
- We will mostly explore ML/AR combinations in a *formal proof* setting
- *Typical problem:* How can learning help with logical reasoning?

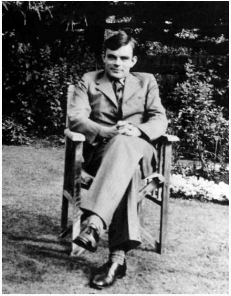
Course Overview - Particular settings and topics

- ML and first-order logic (FOL), saturation-style theorem provers (ATPs)
- Higher-order logic (HOL), Set theory, formal proof asistants (ITPs)
- ML and reasoning in large theories, hammers for ITP, premise selection
- Symbolic vs statistical learning for theorem proving
- ML in tableau-style and tactical reasoning systems
- Learning in prpositional logic (SAT), QBF, SMT, instantiation-based methods and model finding.
- Representations and conjecturing - how do we characterize reasoning data for learning?
- Feeback loops for proving and learning, reinforcement learning of ATP, positive/negative proof mining
- Alignment and translation between informal and formal corpora, automated formalization
- Exam: do a small project in combining ML and AR

Induction/Learning vs Reasoning – Henri Poincaré

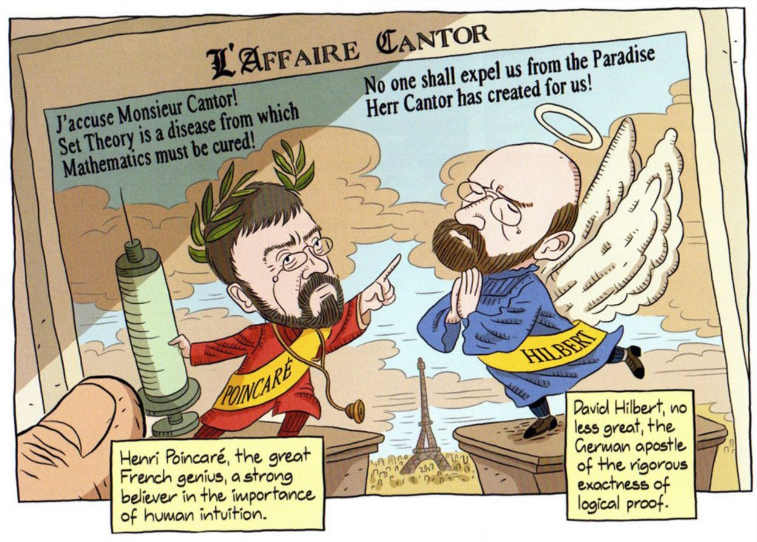
- Science and Method: Ideas about the interplay between correct deduction and induction/intuition
- *"And in demonstration itself logic is not all. The true mathematical reasoning is a real induction [...]"*
- I believe he was right: strong general reasoning engines have to combine deduction and induction (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 large computer-understandable math corpora?

Intuition vs Formal Reasoning – Poincaré vs Hilbert



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

What is Formal Mathematics?

- 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
- 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

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

tiny 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 $(a, b) = 1$. □

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 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 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 2 in Metamath Proof Explorer

sqr2irr - Metamath Proof Explorer - Chromium

us.metamath.org/mpegif/sqr2irr.html

Proof of Theorem sqr2irr

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

Colors of variables: wff set class

Syntax hints:

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*.

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

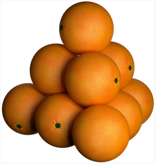
Hubble Sees Lensing Galaxy Cluster,

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

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!

What Has Been Formalized?

top 100 of interesting theorems/proofs
(Paul & Jack Abad, 1999, tracked by Freek Wiedijk)

1. $\sqrt{2} \notin \mathbb{Q}$
2. fundamental theorem of algebra
3. $|\mathbb{Q}| = \aleph_0$
4. 

$\Rightarrow a^2 + b^2 = c^2$
5. $\pi(x) \sim \frac{x}{\ln x}$
6. Gödel's incompleteness theorem
7. $\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$
8. impossibility of trisecting the angle and doubling the cube

   ⋮

32. four color theorem
33. Fermat's last theorem

   ⋮

99. Buffon needle problem
100. Descartes rule of signs

What Has Been Formalized?

top 100 of interesting theorems/proofs
(Paul & Jack Abad, 1999, tracked by Freek Wiedijk)

1. $\sqrt{2} \notin \mathbb{Q}$
2. fundamental theorem of algebra
3. $|\mathbb{Q}| = \aleph_0$
4. 

$\Rightarrow a^2 + b^2 = c^2$
5. $\pi(x) \sim \frac{x}{\ln x}$
6. Gödel's incompleteness theorem
7. $\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$
8. impossibility of trisecting the angle and doubling the cube

$\vdots$

32. four color theorem
33. Fermat's last theorem

$\vdots$

99. Buffon needle problem
100. Descartes rule of signs

all together	88%
HOL Light	86%
Mizar	57%
Isabelle	52%
Coq	49%
ProofPower	42%
Metamath	24%
ACL2	18%
PVS	16%

What Has Been Formalized?

top 100 of interesting theorems/proofs
(Paul & Jack Abad, 1999, tracked by Freek Wiedijk)

1. $\sqrt{2} \notin \mathbb{Q}$
2. fundamental theorem of algebra
3. $|\mathbb{Q}| = \aleph_0$
4. 

$\Rightarrow a^2 + b^2 = c^2$
5. $\pi(x) \sim \frac{x}{\ln x}$
6. Gödel's incompleteness theorem
7. $\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$
8. impossibility of trisecting the angle and doubling the cube

$\vdots$

32. four color theorem
33. Fermat's last theorem

$\vdots$

99. Buffon needle problem
100. Descartes rule of signs

all together	88%
HOL Light	86%
Mizar	57%
Isabelle	52%
Coq	49%
ProofPower	42%
Metamath	24%
ACL2	18%
PVS	16%

What Has Been Formalized?

top 100 of interesting theorems/proofs
(Paul & Jack Abad, 1999, tracked by Freek Wiedijk)

1. $\sqrt{2} \notin \mathbb{Q}$
2. fundamental theorem of algebra
3. $|\mathbb{Q}| = \aleph_0$
4. 

$\Rightarrow a^2 + b^2 = c^2$
5. $\pi(x) \sim \frac{x}{\ln x}$
6. Gödel's incompleteness theorem
7. $\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$
8. impossibility of trisecting the angle and doubling the cube

⋮

32. four color theorem
33. Fermat's last theorem

⋮

99. Buffon needle problem
100. Descartes rule of signs

all together	88%
HOL Light	86%
Mizar	57%
Isabelle	52%
Coq	49%
ProofPower	42%
Metamath	24%
ACL2	18%
PVS	16%

Named Theorems in the Mizar Library

FM - Chromium

FM

fm.uwb.edu.pl/mmlquery/fillin.php?filledfilename=mml-facts.mqt&argument=number+102

Mizar home, download
files: abstr, articles, bin, doc, emacs gabs, fmbibs, gabs (more)
semantic MML

MML 情報

MML Query (beta)

Template maker
Environment explanation

Mizar TWiki
MML Query server
Megrez services
Journals:
FM: MetaPRESS, server, proof-read, regeneration
MM&A (preparation)

Syntax: xml, html
Downloads

Mizar syntax, xml, txt

MML 5.25.1220
- most important facts (other collection)

- Birkhoff

The most important facts in MML (decode)

add description

See also Name carrying facts/theorems/definitions in MML

1	"Alexander\'s Lemma"	=> WAYBEL_7:31	VOTE
2	"All Primes (1 mod 4) Equal the Sum of Two Squares"	=> NAT_5:23	VOTE
3	"Axiom of Choice"	=> WELLORD2:18	VOTE
4	"Baire Category Theorem (Banach spaces)"	=> LOPBAN_5:3	VOTE
5	"Baire Category Theorem (Hausdorff spaces)"	=> NORMSP_2:10	VOTE
6	"Baire Category Theorem for Continuous Lattices"	=> WAYBEL12:39	VOTE
7	"Banach Fix Point Theorem for Compact Spaces"	=> ALI2:1	VOTE
8	"Banach-Steinhaus theorem (uniform boundedness)"	=> LOPBAN_5:7	VOTE
9	"Bertrand\'s Ballot Theorem"	=> BALLOT_1:28	VOTE
10	"Bertrand\'s postulate"	=> NAT_4:56	VOTE
11	"Bezout\'s Theorem"	=> NEWTON:67	VOTE
12	"Bing Theorem"	=> NAGATA_2:22	VOTE
13	"Binomial Theorem"	=> BINOM:25	VOTE
14	"Birkhoff Variety Theorem"	=> BIRKHOFF:sch 12	VOTE
15	"Bolzano theorem (intermediate value)"	=> TOPREAL5:8	VOTE
16	"Bolzano-Weierstrass Theorem (1 dimension)"	=> SEQ_4:40	VOTE
17	"Borsuk Theorem on Decomposition of Strong Deformation Retracts"	=> BORSUK_1:42	VOTE
18	"Borsuk-Ulam Theorem"	=> BORSUK_7:condreg 3	VOTE
19	"Boundary Points of Locally Euclidean Spaces"	=> MFOLD_0:2	VOTE
20	"Brouwer Fixed Point Theorem"	=> BROUWER:14	VOTE
21	"Brouwer Fixed Point Theorem for Disks on the Plane"	=> BROUWER:15	VOTE
22	"Brouwer Fixed Point Theorem for Intervals"	=> TREAL_1:24	VOTE
23	"Brown Theorem"	=> GCD_1:40	VOTE
24	"Cantor Theorem"	=> CARD_1:14	VOTE
25	"Cantor-Bernstein Theorem"	=> CARD_1:10	VOTE

Big Formalizations

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

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)

Large Software Verifications

- seL4 – operating system microkernel
  - Gerwin Klein and his group at NICTA, Isabelle/HOL
- CompCert – a formaly 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, 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

The Top 100 The

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

Unhackable kernel could keep all computers safe from cyberattack

POPULAR

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

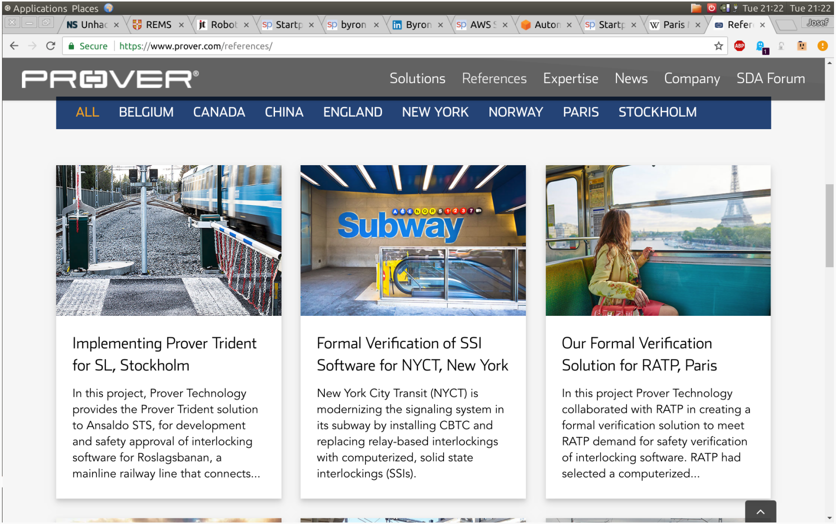
End of days: Is Western civilisation on the brink of collapse?

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

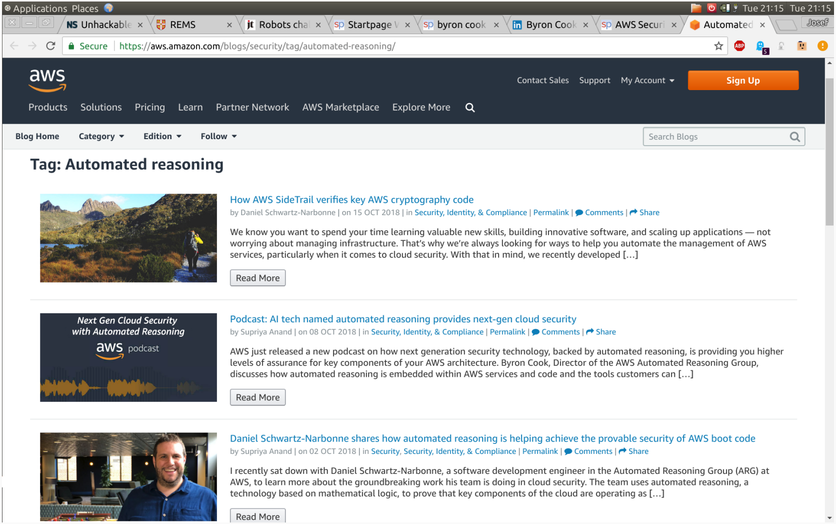
The brain’s 7D sandcastles could be

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

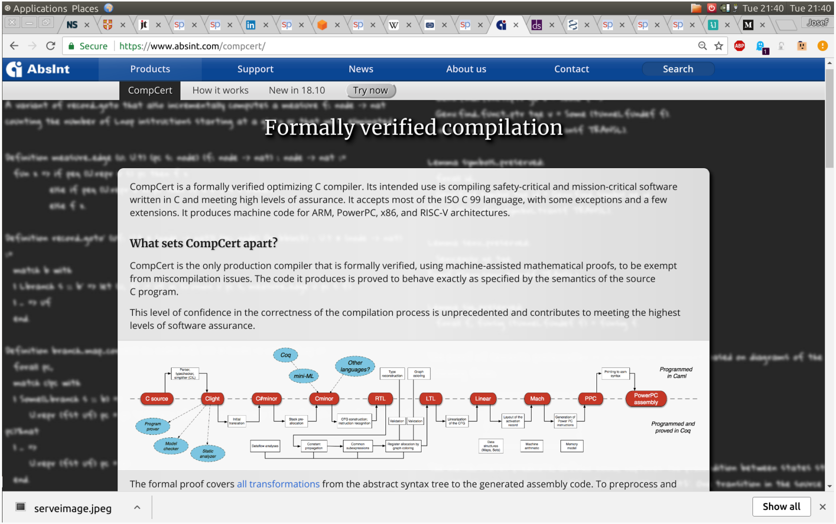
Today's Applications



Today’s Applications



Today's Applications



Today's Applications

Overview About Research Education Industry People Projects Events Publications Institutions Jobs Visitors Student Positions Login

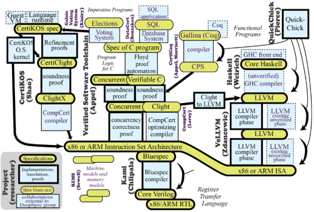
deep spec **the science of deep specification**

DeepSpec is an Expedition in Computing funded by the National Science Foundation.

We focus on the specification and verification of full functional correctness of software and hardware.

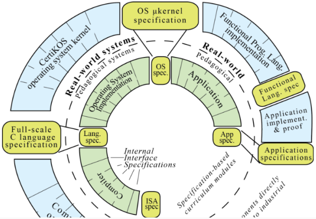
Research

We have several major research projects, and our ambitious goal is to connect them at specification interfaces to prove end-to-end correctness of whole systems.



Education

To deliver secure and reliable products, the software industry of the future needs engineers trained in specification and verification. We'll produce that curriculum.



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

What Are Automated Theorem Provers?

- Computer programs that (try to) determine if
  - A conjecture C is a logical consequence of a set of axioms Ax
  - The derivation of conclusions that follow inevitably from facts.
- Systems: Vampire, E, SPASS, Prover9, Z3, CVC4, Satallax, iProver, ...
- Brute-force search calculi (resolution, superposition, tableaux, SMT, ...)
- Human-designed heuristics for pruning of the search space
- Fast combinatorial explosion on large knowledge bases like Flyspeck and Mizar
- Need to be equipped with good domain-specific inference guidance ...
- ... this what we will try to do here ...
- ... by learning from the knowledge bases and reasoning feedback ...
- Details on particular ATP systems and ML settings later

Mizar demo

http://grid01.ciirc.cvut.cz/~mptp/out4.ogv

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, ...
- ...

Sample of Learning Approaches We Have Been Using

- **neural networks** (statistical ML) – backpropagation, deep learning, convolutional, recurrent, etc.
- **decision trees, random forests, gradient tree boosting** – 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
- combinations of statistical and symbolic approaches (probabilistic grammars, semantic features, ...)
- supervised, unsupervised, reinforcement learning (actions, explore/exploit, cumulative reward)

Learning – Features and Data Preprocessing

- Extremely important - if irrelevant, there is no use to learn the function from input to output ("garbage in garbage out")
- Feature discovery – a big field
- Deep Learning – design neural architectures that automatically find important high-level features for a task
- Latent Semantics, dimensionality reduction: use linear algebra (eigenvector decomposition) to discover the most similar features, make approximate equivalence classes from them
- word2vec and related methods: represent words/sentences by *embeddings* (in a high-dimensional real vector space) learned by predicting the next word on a large corpus like Wikipedia
- math and theorem proving: syntactic/semantic patterns/abstractions
- how do we represent math objects (formulas, proofs, ideas) in our mind?

Reasoning Datasets - Large ITP Libraries and Projects

- Mizar / MML / MPTP – since 2003
- MPTP Challenge (2006), MPTP2078 (2011), Mizar40 (2013)
- Isabelle (and AFP) – since 2005
- Flyspeck (including core HOL Light and Multivariate) – since 2012
- HOLStep – 2016, kernel inferences
- Coq – since 2013/2016
- HOL4 – since 2014
- ACL2 – 2014?
- Lean? – 2017?
- Stacks?, ProofWiki?, Arxiv?

High-level ATP guidance: Premise Selection

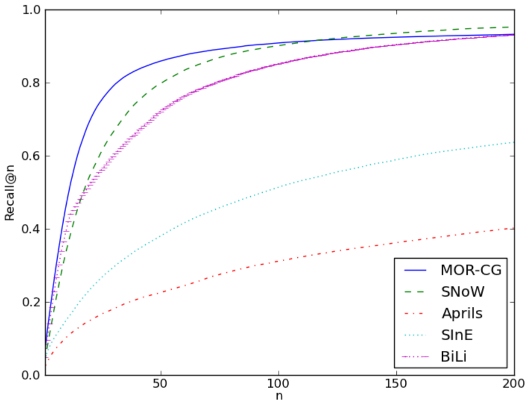
- Early 2003: Can existing ATPs be used over the freshly translated Mizar library?
- About 80000 nontrivial math facts at that time – impossible to use them all
- Is good premise selection for proving a new conjecture possible at all?
- Or is it a mysterious power of mathematicians? (Penrose)
- Today: Premise selection is not a mysterious property of mathematicians!
- Reasonably good algorithms started to appear (more below).
- Will extensive human (math) knowledge get obsolete?? (cf. Watson, Debater, etc)

Example system: Mizar Proof Advisor (2003)

- train naive-Bayes fact selection on all previous Mizar/MML proofs (50k)
- input features: conjecture symbols; output labels: names of facts
- recommend relevant facts when proving new conjectures
- give them to unmodified FOL ATPs
- possibly reconstruct inside the ITP afterwards (lots of work)
- 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)
- Today's methods: about 45-50% (and we are still just beginning!)

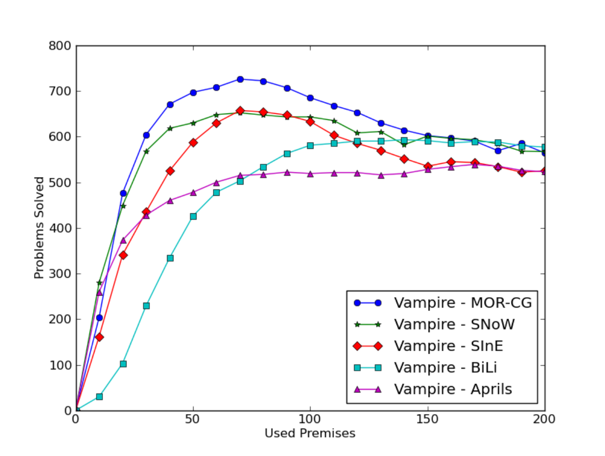
ML Evaluation of methods on MPTP2078 – recall

- Coverage (recall) of facts needed for the Mizar proof in first *n* predictions
- MOR-CG – kernel-based, SNoW - naive Bayes, BiLi - bilinear ranker
- SINe, Aprils - heuristic (non-learning) fact selectors



ATP Evaluation of methods on MPTP2078

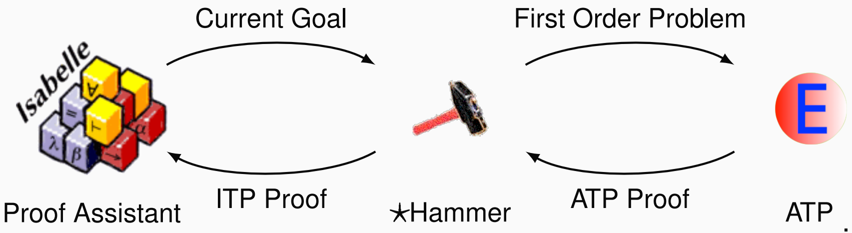
- Number of the problems proved by ATP when given *n* best-ranked facts
- Good machine learning on previous proofs really matters for ATP!



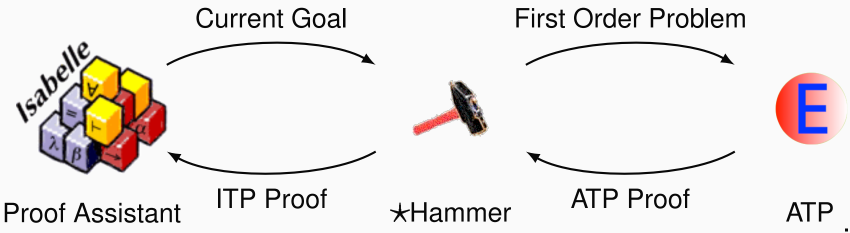
High-level ATP guidance: Premise Selection/Hammers

- 2003: Can existing ATPs 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):
- train naive-Bayes fact selection on previous Mizar/MML
- recommend relevant premises when proving new conjectures
- give them to unmodified FOL ATPs
- possibly reconstruct inside the ITP afterwards (lots of work)
- 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)

Today’s AI-ATP systems (⋆-Hammers)

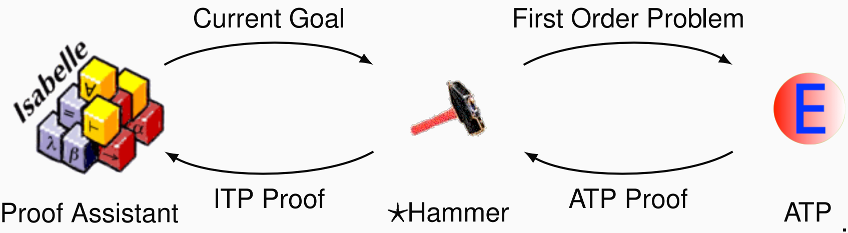


Today's AI-ATP systems (⋆-Hammers)



How much can it do?

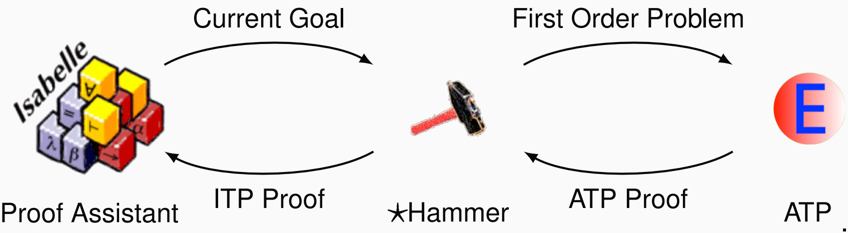
Today's AI-ATP systems (⋆-Hammers)



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 ($\star$-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

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:

$c_1 : P(x)$

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

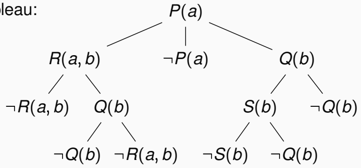
$c_3 : S(x) \lor \neg Q(b)$

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

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

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

Closed Connection Tableau:



Statistical Guidance of Connection Tableau

- **MaLeCoP** (2011): first prototype Machine Learning Connection Prover
- extension rules chosen by naive Bayes trained on good decisions
- training examples: tableau features plus the name of the chosen clause
- initially slow: off-the-shelf learner 1000 times slower than raw leanCoP
- 20-time search shortening on the MPTP Challenge
- second version: 2015, with C. Kaliszyk
- both prover and naive Bayes in OCAML, fast indexing
- Fairly Efficient MaLeCoP = **FEMaLeCoP**
- 15% improvement over untrained leanCoP on the MPTP2078 problems
- using iterative deepening - enumerate shorter proofs before longer ones

Statistical Guidance of Connection Tableau – rlCoP

- 2018: stronger learners via C interface to OCAML (boosted trees)
- remove iterative deepening, the prover can go arbitrarily deep
- added Monte-Carlo Tree Search (MCTS)
- MCTS search nodes are sequences of clause application
- a good heuristic to explore new vs exploit good nodes:

$$\frac{w_i}{n_i} + c \cdot p_i \cdot \sqrt{\frac{\ln N}{n_i}}$$ (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

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 (features engineering), Deep guidance (neural nets)
- 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

ProofWatch: Statistical/Semantic Guidance of E

- Bob Veroff's *hints* method used for Prover9/AIM
- solve many easier/related problems
- load their useful lemmas on the *watchlist*
- boost inferences on clauses that subsume a watchlist clause
- watchlist parts are fast thinking, bridged by standard search
- ProofWatch (2018): load many proofs separately
- dynamically boost those that have been covered more
- needed for heterogeneous ITP libraries
- statistical: watchlists chosen using similarity and usefulness
- semantic/deductive: dynamic guidance based on exact proof matching
- results in better vectorial characterization of saturation proof searches

ProofWatch: Statistical/Symbolic Guidance of E

```
theorem Th36: :: YELLOW_5:36
for L being non empty Boolean RelStr for a, b being Element of L
holds ( 'not' (a "\/" b) = ('not' a) "/\" ('not' b)
          & 'not' (a "/\" b) = ('not' a) "\/" ('not' b) )
```

- De Morgan's laws for Boolean lattices
- guided by 32 related proofs resulting in 2220 watchlist clauses
- 5218 given clause loops, resulting ATP proof is 436 clauses
- 194 given clauses match the watchlist and 120 (61.8%) used in the proof
- most helped by the proof of WAYBEL_1:85 – done for lower-bounded Heyting

```
theorem :: WAYBEL_1:85
for H being non empty lower-bounded RelStr st H is Heyting holds
for a, b being Element of H holds
  'not' (a "/\" b) >= ('not' a) "\/" ('not' b)
```

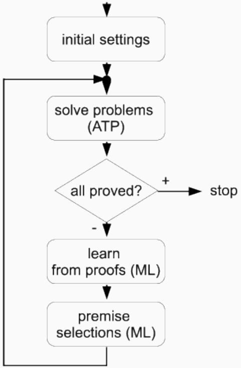
ProofWatch: Vectorial Proof State

Final state of the proof progress for the 32 proofs guiding `YELLOW_5:36`

0	0.438	42/96	1	0.727	56/77	2	0.865	45/52	3	0.360	9/25
4	0.750	51/68	5	0.259	7/27	6	0.805	62/77	7	0.302	73/242
8	0.652	15/23	9	0.286	8/28	10	0.259	7/27	11	0.338	24/71
12	0.680	17/25	13	0.509	27/53	14	0.357	10/28	15	0.568	25/44
16	0.703	52/74	17	0.029	8/272	18	0.379	33/87	19	0.424	14/33
20	0.471	16/34	21	0.323	20/62	22	0.333	7/21	23	0.520	26/50
24	0.524	22/42	25	0.523	45/86	26	0.462	6/13	27	0.370	20/54
28	0.411	30/73	29	0.364	20/55	30	0.571	16/28	31	0.357	10/28

Machine Learner for Automated Reasoning

- MaLARea (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



Recent Improvements and Additions

- Semantic features encoding term matching/unification [IJCAI'15]
- Distance-weighted k-nearest neighbor, LSI, boosted trees (XGBoost)
- Matching and transferring concepts and theorems between libraries (Gauthier & Kaliszyk) – allows "superhammers", conjecturing, and more
- Lemmatization – extracting and considering millions of low-level lemmas
- First useful CoqHammer (Czajka & Kaliszyk 2016), 40%–50% reconstruction/ATP success on the Coq standard library
- Neural sequence models, definitional embeddings (Google Research)
- Hammers combined with statistical tactical search: TacticToe (HOL4)
- Learning in binary setting from many alternative proofs
- Negative/positive mining (ATPBoost)

Summary of Features Used

- 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
- Evaluation in a large set of (finite) models
- LSI/PCA combinations of above
- Neural embeddings of above

TacticToe: mid-level ITP Guidance (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?
- 66% of HOL4 toplevel proofs in 60s (better than a hammer!)
- work in progress for Coq
- earlier Coq work: SEPIA (Gransden et al, 2015) - inferred automata

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!)

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 Autoformalization results

Parameter	Final Test Perplexity	Final Test BLEU	Identical Statements (%)	Identical No-overlap (%)
128 Units	3.06	41.1	40121 (38.12%)	6458 (13.43%)
256 Units	1.59	64.2	63433 (60.27%)	19685 (40.92%)
512 Units	1.6	**67.9**	66361 (63.05%)	21506 (44.71%)
1024 Units	**1.51**	61.6	**69179 (65.73%)**	**22978 (47.77%)**
2048 Units	2.02	60	59637 (56.66%)	16284 (33.85%)

Neural Fun – Performance after Some Training

Rendered LaTeX	Suppose s_8 is convergent and s_7 is convergent . Then $\mathop{\rm lim}(s_8+s_7) = \mathop{\rm lim} s_8 + \mathop{\rm lim} s_7$
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 ) ;`

Some References

- 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
- April 8–12, 2019, Obergurgl, Austria, `aitp-conference.org`
- ATP/ITP/Math vs AI/Machine-Learning people, Computational linguists
- Discussion-oriented and experimental
- Grown to 60 people in 2018