Towards the Dream of Self-Improving Universal Reasoning AI

Josef Urban

Czech Technical University in Prague

AGI-21
October 18, 2021, Palo Alto

European Research Council
Established by the European Commission

Outline

- Philosophical ramblings about where we are and how we should do things
- Some recent combinations of ML/RL with AR/TP and what they can do
- Possibly some more open ideas
- Questions?
- (If I run out of time, use my AGI'18 talk `https://bit.ly/3qifhg4` is an intro)

Kepler vs Vinge, Prague vs California

- Kepler (1600s): the beginning of scientific revolution
- Understand and predict the universe, explore and exploit it
- Driven by crazy geometric, alchymistic, God (Singularity?) theories
- Mathematization of philosophy: rebellion by the mathematicians against philosophers
- Leading to a lot of observation and unhinged conjecturing about how the natural world works
- Experiments, tools, physics, chemistry, science, ...
- ... machines, logic, ..., Turing machines, AI, ...

Kepler vs Vinge, Prague vs California

- Vinge: Zones of Thought, Singularity, Fire Upon the Deep, Coldsleep
- Encouraging technological progress on a backward planet where you accidentaly crashed to invent space travel and escape
- How do WE do this on our backward planet?
- How do we invent ...
- ... wormholing technology, safe fusion, infinite batteries, ...
- ... immortality, self upload, coldsleep, ...
- ... infinite empathy and compassion, ...
- ... all in our lifetime?
- We Automate Scientific Discovery!
- Art Quaife (1993, Berkeley): *Automate math to accelerate science*

Automation of math/science vs human-level AI

- Martin Davis: logicists vs heuristists
- Logic Theorist (Newell, Simon) vs DPLL (Davis etc)
- Modern versions of the DPLL algorithm are superhuman
- (a bit like modern neural nets for image recognition are superhuman)
- AR is a constant battle between logic-designed/algorithmic and more heuristic/AI methods
- DPLL, CDCL, Resolution, AVATAR vs my agenda: combine ML and TP

Automation of math/science vs human-level AI

- Robinson's resolution in Herbrand universe as an abstraction of the "embodied" Gelernter's Geometry Machine in Euclidean plane
- Why should we care if something is human/nature inspired if we have a better/faster inhuman algo/hw for it?
- And vice-versa: it is stupid to try to "fully design a solution" when it seems impossibly hard and we are not yet as good as the learning humans
- And frankly, the System 1 vs System 2 idea is just one human-like option.
- There are many plausible inhuman AR ideas and some work great.
- E.g. Voronkov's recent AVATAR:
- SAT solver used for high-level decomposition of the search space, chasing FOL prover on the components.

Automation of math/science vs human-level AI

- My conclusion: AR is an experimental science that should not fall just for ideologies
- I have gradually become suspicious of both the logicists and the humanoids (and their little voices in me)
- We should be Turings/hackers, not Wittgensteins/philosophers
- So I said around 2000: Let the approaches battle it all out.
- And found the largest formal math corpus, translated it for ATPs and ML systems, and started to measure performance.
- Eventually done for Mizar, Isabelle, HOL, Coq and other formal math corpora.
- So you can take our set of 58k toplevel Mizar problems and try to prove as many of them as you can by various AR/TP, ML, RL, AI methods
- Leibniz: Calculemus!

Automation of math/science vs human-level AI

- And we are still only at the beginning.
- You need to do quite little to beat the state of the art.
- Basic feedback loops can take us quite a bit higher than we were before.
- In some sense, it's not really yet the time of very complex AI architectures.
- But you need to do the implementation, tuning and experiments right.
- It's hard (but possible) to beat a good ATP by more AI-ish methods.
- But instead of very naive ideas like "GPT will do it all", we need much more serious cross-fertilization of the learning and inferencing algos.
- At this point we need much more work on real AR architectures and their cautious evaluation on non-fake corpora. Ideas are cheap.

The coming of logic-ization and computerization of math/science

- Most of the talks here were natural language (mine too).
- One exception: the talk by Alexander & Hutter.
- The (old) news: All of math can be very safely checked
- The invention is called symbolic logic and it’s over 100 years old.
- And it led quite directly to the creation of computers, which are very much symbolic logic machines.
- For 50-60 years, people have been trying to embed/translate math (and scientific) arguments into the precise logic

The coming of logic-ization and computerization of math/science

- Mike Beeson (nearby in San Jose) has once called it the QED Singularity.
- And this QED Singularity seems to be finally coming.
- And even if we don’t immediatelly get superhuman AIs for math and science.
- But the general embedding of our discourse in logic will be a big deal.
- It may be the one cure for the current world of disinformation and hacking.
- Things like attention hacking and hacking of credit assignment in science.
- So I could almost have something that was fact-checking Sam’s talk in real time yesterday.
- His proof was sufficiently verbose and not too ambiguous.
- The recent advances in AI/NLP/TP and formal proof technology may allow such fact-checking and assistance of a lot of math/science discourse quite soon.

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

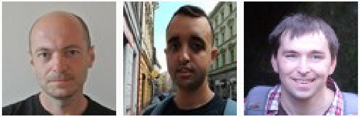
- ENIGMA/hammer proofs of Pythagoras : `https://bit.ly/2MVPAn7` (more at `http://grid01.ciirc.cvut.cz/~mptp/enigma-ex.pdf`) and simplified Carmichael `https://bit.ly/3oGBdRz`,
- 3-phase ENIGMA: `https://bit.ly/3C0Lwa8`, `https://bit.ly/3BWqR6K`
- Long trig proof from 1k axioms: `https://bit.ly/2YZ0OgX`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

ENIGMA: Guiding the Best ATPs like 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: Guiding the Best ATPs like E Prover

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



- Fast/hashed feature extraction followed by fast/sparse linear classifier
- about 80% improvement on the AIM benchmark
- Deep guidance: convolutional nets - too slow to be competitive
- ENIGMA-NG: better features and ML, gradient-boosted trees, tree NNs
- NNs made competitive in real-time, boosted trees still best
- 2020: fast GNN added (Olsak, Jakubuv), now competitive with GBDTs
- However very different: the GNN scores many clauses (context and query) simultaneously in a large graph
- 2021: 3-phase architecture with GPU server - 17.4% better
- 2021: leapfrogging and Split&Merge:
- aimed at learning reasoning/algo components

Feedback loop for ENIGMA on Mizar data

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

	$\mathcal{S}$	$\mathcal{S} \odot \mathcal{M}_9^0$	$\mathcal{S} \oplus \mathcal{M}_9^0$	$\mathcal{S} \odot \mathcal{M}_9^1$	$\mathcal{S} \oplus \mathcal{M}_9^1$	$\mathcal{S} \odot \mathcal{M}_9^2$	$\mathcal{S} \oplus \mathcal{M}_9^2$	$\mathcal{S} \odot \mathcal{M}_9^3$	$\mathcal{S} \oplus \mathcal{M}_9^3$
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 Clause Selection in Vampire (M. Suda)

Deepire: Similar to ENIGMA:

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

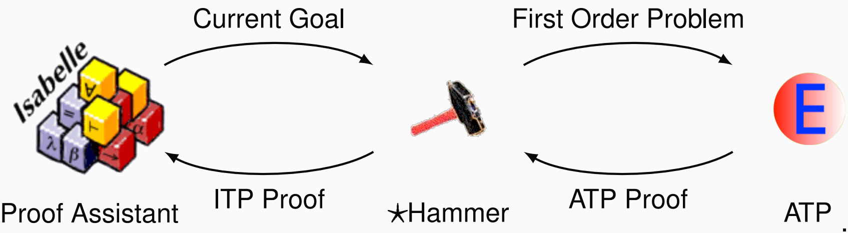
Deepire's contributions:

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

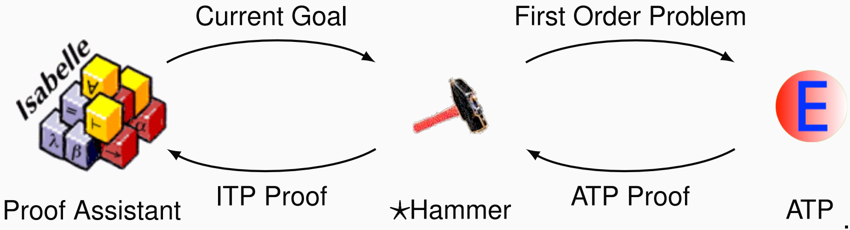
Preliminary Evaluation on Mizar "57880"

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

Today's AI-ATP systems (⋆-Hammers)

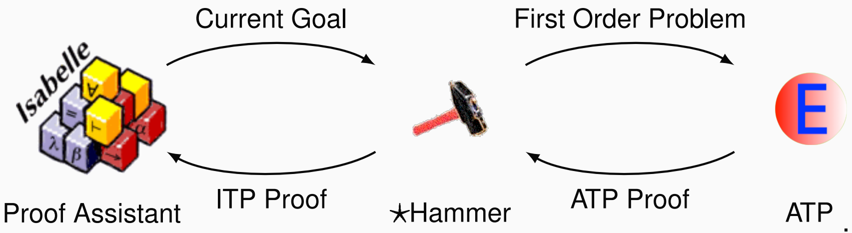


Today's AI-ATP systems (⋆-Hammers)



How much can it do?

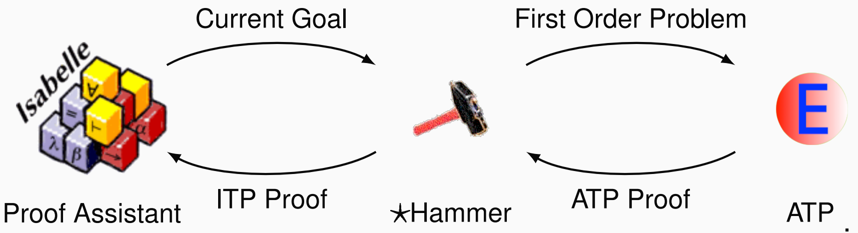
Today's AI-ATP systems (⋆-Hammers)



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

$\approx$ 40-45% success rate (close to 60% on Mizar as of 2021)

Premise Selection and Hammer Methods

- Many syntactic features (symbols, walks in the parse trees)
- More semantic features encoding
- term matching/unification, validity in models, latent semantics (LSI)
- Distance-weighted k-nearest neighbor, SVMs, Naive Bayes
- Gradient boosted decision trees (GBDTs - XGBoost, LightGBM)
- Neural models: CNNs, RNNs/Attention/Transformers/GPT, GNNs
- As of 2020, tough competition between GBDTs, GNNs and RNNs/Transformers (and relatives)
- K-NN still very good, Olsak's logic-aware GNN probably best
- RNNs/Transformers good at stateful premise selection (Piotrowski 2019,2020)
- Ensemble methods combining the different predictors help a lot

Premise Selection and Hammer Methods

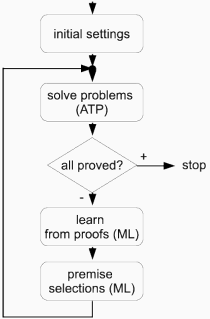
- Learning in a binary setting from many alternative proofs
- Interleaving many learning and proving runs (*MaLARea loop*) to get positives/negatives (ATPBoost - Piotrowski 2018)
- Matching and transferring concepts and theorems between libraries (Gauthier & Kaliszyk) – allows “superhammers”, conjecturing, and more
- Lemmatization – extracting and considering millions of low-level lemmas and learning from their proofs
- Hammers combined with guided tactical search: TacticToe (Gauthier - HOL4) and its later relatives

ENIGMA Proof Example – Knaster

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

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
- strategy evolution and ENIGMA learning added later
- winning AI/ATP benchmarks (MPTPChallenge, CASC 2008/12/13/18/20)



MaLARea improvement over E in CASC’20

Large Theory Batch Problems	MaLARea 0.9	E LTB-2.5	iProver LTB-3.3	Zipperpin LTB-2.0	Leo-III LTB-1.5	ATPBoost 1.0	GKC LTB-0.5.1	*Leo-III* *LTB-1.4*
Solved/10000	7054/10000	3393/10000	3164/10000	1699/10000	1413/10000	1237/10000	493/10000	134/10000
Solutions	7054 70%	3393 33%	3163 31%	1699 16%	1413 14%	1237 12%	493 4%	134 1%

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

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

Tactician: Tactical Guidance for Coq (Blaauwbroek'20)

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

More on Conjecturing in Mathematics

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

A bit of conjecturing history

- The topic goes back at least to Lenat (AM) and Fajtlowicz (Graffiti)
- Combined with automated theorem proving by Colton et al. in early 2000s (HR)
- Theory exploration for Isabelle by Johansson et al (Hipster)
- Several learning-based/neural approaches by our groups since 2015:
- Based mainly on learning analogies and informalization followed by probabilistic/neural disambiguation ...
- ... Gauthier, Kaliszyk, Chvalovsky, Piotrowski, Goertzel, Wang, Brown, JU

Conjecturing and Proof Synthesis by Neural Language models

- Karpathy’15 - RNN experiments with generating fake Math over Stacks
- I have tried to use that for formal math in 2016 but it looked weak
- GPT (-2,3) looks stronger
- Renewed experiments in 2020 on:
- All Mizar articles, stripped of comments and concatenated together (78M)
- Articles with added context/disambiguation (156M) (types, names, thesis)
- TPTP proofs of 28271 Mizar/MPTP theorems by E/ENIGMA (658M)
- Just the conjecture and premises needed for the 28271 proofs printed in prefix notation
- Quite interesting results, server for Mizar authors
- Quickly taken up by others on HOL, Isabelle, MetaMath ...

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

```
:: generated theorem with "proof"
theorem Th23: :: STIRL2_1:23
for X, Y being finite set st not X is empty & X c= Y
& card X = card Y holds X = Y
proof
  let X, Y be finite set ;
:: thesis: not X is empty & X c= Y & card X = card Y implies X = Y
  assume that
  A1: not X is empty and A2: X c= Y and A3: card X = card Y ;
:: thesis: X = Y
  card (Y \ X) = (card Y) - (card X) by A1, A3, CARD_2:44;
  then A4: card (Y \ X) = ((card Y) - 1) - (card X) by CARD_1:30;
  X = Y \ X by A2, A3, Th22;
  hence X = Y by A4, XBOOLE_0:def_10;
:: thesis: verum
end;
```

-:--- **card_tst.miz** 99% L2131 (Mizar Errors:13 hs Undo-Tree)

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

Mizar autocompletion server in action

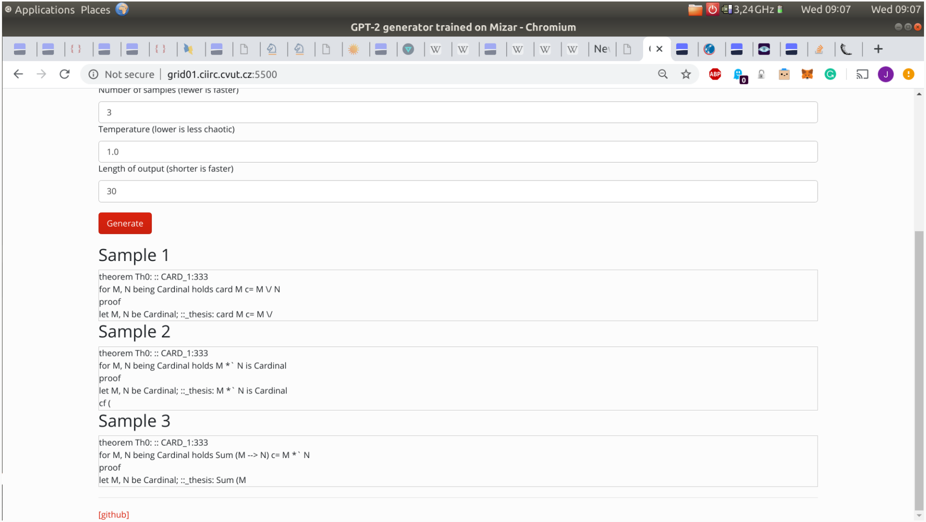


Figure: MGG - Mizar Gibberish Generator.

Proving the conditioned completions - MizAR hammer

```
Applications Places                                     4,81 GHz   Wed 14:42   Wed 14:42
emacs@dell
File Edit Options Buffers Tools Index Mizar Hide/Show Help
Save Undo

begin


for M, N being Cardinal holds card M c= M \/ N by XBOOLE_1:7,CARD_3:44,CARD_1:7,CARD_1:3; :: [ATP details]

for X, Y being finite set st not X is empty & X c= Y & card X = card Y holds X = Y by CARD_FIN:1; :: [ATP details]

for M, N being Cardinal holds
( M in N iff card M c= N ) by Unsolved; :: [ATP details]

for M, N being Cardinal holds
( M in N iff card M in N ) by CARD_3:44,CARD_1:9; :: [ATP details]

for M, N being Cardinal holds Sum (M --> N) = M *` N by CARD_2:65; :: [ATP details]


for M, N being Cardinal holds M /\ (union N) in N by Unsolved; :: [ATP details]


for M, N being Cardinal holds M *` N = N *` M by ATP-Unsolved; :: [ATP details]

-:---  card_tst.miz    3% L47    (Mizar Errors:2 hs Undo-Tree)
Wrote /home/urban/mizwrk/7.13.01_4.181.1147/tst8/card_tst.miz
```

A correct conjecture that was too hard to prove

- Kinyon and Stanovsky (algebraists) confirmed that this cut is valid:

```
theorem Th10: :: GROUPP_1:10
for G being finite Group for N being normal Subgroup of G st
N is Subgroup of center G & G ./. N is cyclic holds G is commutative

The generalization that avoids finiteness:

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

Gibberish Generator Provoking Algebraists

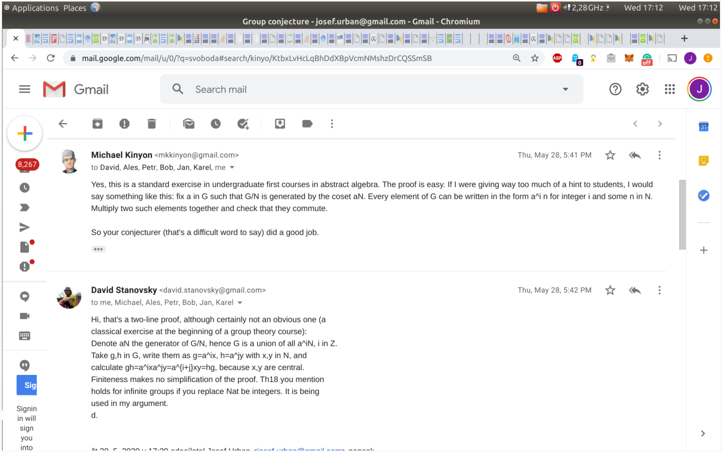


Figure: First successes in making mathematicians comment on AI.

More cuts

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

```
theorem :: SINCOS10:17
sec is increasing on [0, pi/2)

leads to conjecturing the following:

Every differentiable function is increasing.
```

Neural Autoformalization (Wang et al., 2018)

- generate ca 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 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 ) ;`

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 ;
```

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 General and Hammer/Tactical References

- J. C. Blanchette, C. Kaliszyk, L. C. Paulson, J. Urban: Hammering towards QED. J. Formalized Reasoning 9(1): 101-148 (2016)
- Cezary Kaliszyk, Josef Urban: Learning-Assisted Automated Reasoning with Flyspeck. J. Autom. Reason. 53(2): 173-213 (2014)
- Cezary Kaliszyk, Josef Urban: MizAR 40 for Mizar 40. J. Autom. Reason. 55(3): 245-256 (2015)
- Cezary Kaliszyk, Josef Urban: Learning-assisted theorem proving with millions of lemmas. J. Symb. Comput. 69: 109-128 (2015)
- Jasmin Christian Blanchette, David Greenaway, Cezary Kaliszyk, Daniel Kühlwein, Josef Urban: A Learning-Based Fact Selector for Isabelle/HOL. J. Autom. Reason. 57(3): 219-244 (2016)
- Bartosz Piotrowski, Josef Urban: ATPboost: Learning Premise Selection in Binary Setting with ATP Feedback. IJCAR 2018: 566-574
- T. Gauthier, C. Kaliszyk, J. Urban, R. Kumar, M. Norrish: Learning to Prove with Tactics. CoRR abs/1804.00596 (2018).
- Lasse Blaauwbroek, Josef Urban, Herman Geuvers: Tactic Learning and Proving for the Coq Proof Assistant. LPAR 2020: 138-150
- Lasse Blaauwbroek, Josef Urban, Herman Geuvers: The Tactician (extended version): A Seamless, Interactive Tactic Learner and Prover for Coq. CoRR abs/2008.00120 (2020)
- L. Czajka, C. Kaliszyk: Hammer for Coq: Automation for Dependent Type Theory. J. Autom. Reasoning 61(1-4): 423-453 (2018)
- 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
- J. Urban, J. Vyskocil: Theorem Proving in Large Formal Mathematics as an Emerging AI Field. LNCS 7788, 240-257, 2013.

Some References on E/ENIGMA, CoPs and Related

- Stephan Schulz: System Description: E 1.8. LPAR 2013: 735-743
- S. Schulz, Simon Cruanes, Petar Vukmirovic: Faster, Higher, Stronger: E 2.3. CADE 2019: 495-507
- J. Jakubuv, J. Urban: Extending E Prover with Similarity Based Clause Selection Strategies. CICM 2016: 151-156
- J. Jakubuv,J. Urban: ENIGMA: Efficient Learning-Based Inference Guiding Machine.CICM 2017:292-302
- Cezary Kaliszyk, Josef Urban, Henryk Michalewski, Miroslav Olsák: Reinforcement Learning of Theorem Proving. NeurIPS 2018: 8836-8847
- Zarathustra Goertzel, Jan Jakubuv, Stephan Schulz, Josef Urban: ProofWatch: Watchlist Guidance for Large Theories in E. ITP 2018: 270-288
- S. M. Loos, G. Irving, C. Szegedy, C. Kaliszyk: Deep Network Guided Proof Search. LPAR 2017: 85-105
- Karel Chvalovský, Jan Jakubuv, Martin Suda, Josef Urban: ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E. CADE 2019: 197-215
- Jan Jakubuv, Josef Urban: Hammering Mizar by Learning Clause Guidance. ITP 2019: 34:1-34:8
- Zarathustra Goertzel, Jan Jakubuv, Josef Urban: ENIGMAWatch: ProofWatch Meets ENIGMA. TABLEAUX 2019: 374-388
- Zarathustra Amadeus Goertzel: Make E Smart Again (Short Paper). IJCAR (2) 2020: 408-415
- Jan Jakubuv, Karel Chvalovský, Miroslav Olsák, Bartosz Piotrowski, Martin Suda, Josef Urban: ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine. IJCAR (2) 2020: 448-463
- Zsolt Zombori, Adrián Csiszárik, Henryk Michalewski, Cezary Kaliszyk, Josef Urban: Towards Finding Longer Proofs. CoRR abs/1905.13100 (2019)
- Zsolt Zombori, Josef Urban, Chad E. Brown: Prolog Technology Reinforcement Learning Prover - (System Description). IJCAR (2) 2020: 489-507
- Miroslav Olsák, Cezary Kaliszyk, Josef Urban: Property Invariant Embedding for Automated Reasoning. ECAI 2020: 1395-1402

Some Conjecturing References

- Douglas Bruce Lenat. *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search*. PhD thesis, Stanford, 1976.
- Siemion Fajtlowicz. On conjectures of Graffiti. *Annals of Discrete Mathematics*, 72(1–3):113–118, 1988.
- Simon Colton. *Automated Theory Formation in Pure Mathematics*. Distinguished Dissertations. Springer London, 2012.
- Moa Johansson, Dan Rosén, Nicholas Smallbone, and Koen Claessen. Hipster: Integrating theory exploration in a proof assistant. In *CICM 2014*, pages 108–122, 2014.
- Thibault Gauthier, Cezary Kaliszyk, and Josef Urban. Initial experiments with statistical conjecturing over large formal corpora. In *CICM'16 WiP Proceedings*, pages 219–228, 2016.
- Thibault Gauthier, Cezary Kaliszyk: Sharing HOL4 and HOL Light Proof Knowledge. LPAR 2015: 372-386
- Thibault Gauthier. Deep reinforcement learning in HOL4. *CoRR*, abs/1910.11797, 2019.
- Chad E. Brown and Thibault Gauthier. Self-learned formula synthesis in set theory. *CoRR*, abs/1912.01525, 2019.
- Bartosz Piotrowski, Josef Urban, Chad E. Brown, Cezary Kaliszyk: Can Neural Networks Learn Symbolic Rewriting? AITP 2019, CoRR abs/1911.04873 (2019)
- Zarathustra Goertzel and Josef Urban. Usefulness of Lemmas via Graph Neural Networks (Extende Abstract). AITP 2019.
- Karel Chvalovský, Thibault Gauthier and Josef Urban: First Experiments with Data Driven Conjecturing (Extended Abstract). AITP 2019.
- Thibault Gauthier: Deep Reinforcement Learning for Synthesizing Functions in Higher-Order Logic. LPAR 2020: 230-248
- Bartosz Piotrowski, Josef Urban: Guiding Inferences in Connection Tableau by Recurrent Neural Networks. CICM 2020: 309-314
- Josef Urban, Jan Jakubuv: First Neural Conjecturing Datasets and Experiments. CICM 2020: 315-323

References on PCFG and Neural Autoformalization

- Cezary Kaliszyk, Josef Urban, Jirí Vyskocil: Learning to Parse on Aligned Corpora (Rough Diamond). ITP 2015: 227-233
- Cezary Kaliszyk, Josef Urban, Jirí Vyskocil, Herman Geuvers: Developing Corpus-Based Translation Methods between Informal and Formal Mathematics: Project Description. CICM 2014: 435-439
- C. Kaliszyk, J. Urban, J. Vyskocil: Automating Formalization by Statistical and Semantic Parsing of Mathematics. ITP 2017: 12-27
- Cezary Kaliszyk, Josef Urban, Jirí Vyskocil: System Description: Statistical Parsing of Informalized Mizar Formulas. SYNASC 2017: 169-172
- Q. Wang, C. Kaliszyk, J. Urban: First Experiments with Neural Translation of Informal to Formal Mathematics. CICM 2018: 255-270
- Qingxiang Wang, Chad E. Brown, Cezary Kaliszyk, Josef Urban: Exploration of neural machine translation in autoformalization of mathematics in Mizar. CPP 2020: 85-98

Thanks and Advertisement

- Thanks for your attention!
- AITP – Artificial Intelligence and Theorem Proving
- September 4–9, 2022, 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