Hammering Mizar by Learning Clause Guidance

Jan Jakubův[1] Josef Urban[1]

[1]Czech Technical University in Prague, Czech Republic

ITP'19, Portland, USA, 9th September 2019

Outline

Overview: Hammers, Main Result, ATPs & Given Clauses

Enigma: The story so far. . .

Enigma: What's new?

Hammering Mizar

Outline

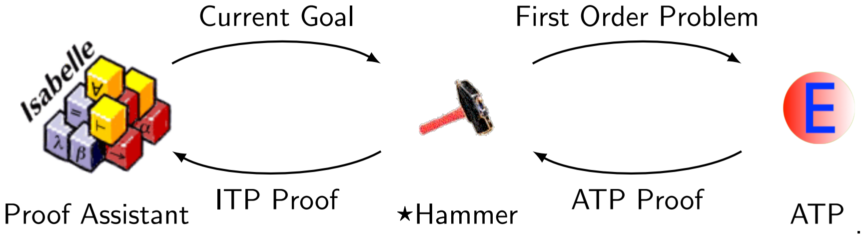
Overview: Hammers, Main Result, ATPs & Given Clauses

Enigma: The story so far...

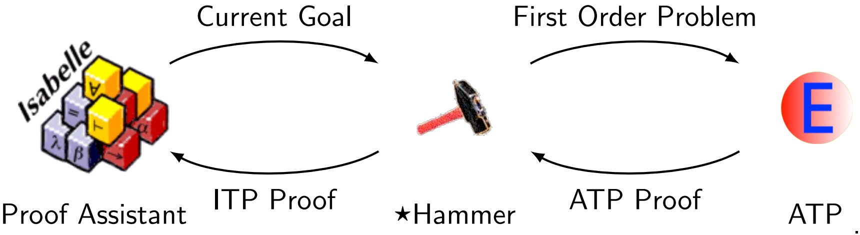
Enigma: What's new?

Hammering Mizar

Today's AI-ATP systems (⋆-Hammers)

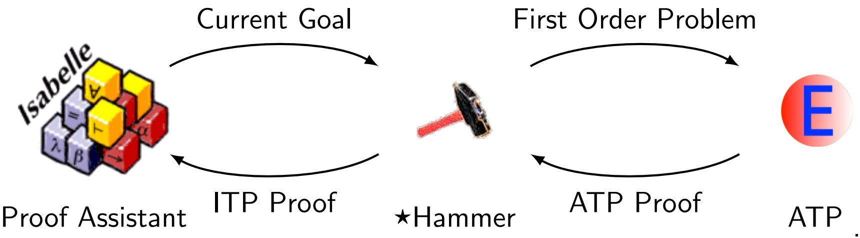


Today's AI-ATP systems (⋆-Hammers)



How much can it do?

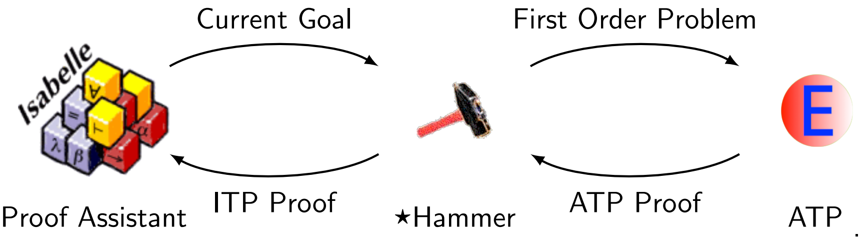
Today's AI-ATP systems (⋆-Hammers)



How much can it do?

- Isabelle (Auth, Jinja) – Sledgehammer
- **Mizar / MML – MPTP, MizAR** – this talk
- Flyspeck (including core HOL Light and Multivariate) – HOL(y)Hammer
- HOL4 (Gauthier and Kaliszyk) , TacTicToe (Gauthier et al.)
- CoqHammer (Czajka and Kaliszyk) – 40% on Coq st. lib.

Today's AI-ATP systems (⋆-Hammers)



How much can it do?

- Isabelle (Auth, Jinja) – Sledgehammer
- **Mizar / MML – MPTP, MizAR** – this talk
- Flyspeck (including core HOL Light and Multivariate) – HOL(y)Hammer
- HOL4 (Gauthier and Kaliszyk) , TacTicToe (Gauthier et al.)
- CoqHammer (Czajka and Kaliszyk) – 40% on Coq st. lib.

$$\approx 45\% \text{ success rate}$$

Our Main Result

- Strengthening the E prover on the Mizar library by 70%
- Done by several iterations of proving and learning over many math problems
- The learning and guidance is done directly in E prover
- This requires strong and fast learning systems
- ... and good engineering choices
- The good news is: it works! Machine learning helps a lot!
- We can gradually learn better and better mathematical tricks by proving and learning over a large math library!
- But it took us some time to get there

Basic Saturation Style ATP Loop – E Prover

```
Proc = {}
Unproc = all available clauses
while (no proof found)
{
   select a given clause C from Unproc
   move C from Unproc to Proc
   apply inference rules to C and Proc
   put inferred clauses to Unproc
}
```

The main non-determinism point:
Which clauses to select as given for further inferences?

E Prover Strategies

- Collections of parameters influencing the proof search
- Weight functions select the "good" clauses
- Can be arbitrarily complicated
- Can be combined in a round-robin way

```
(10  *  ClauseWeight1(10,0.1,...),
  1  *  ClauseWeight2(...),
 20  *  ClauseWeight3(...))
```

Outline

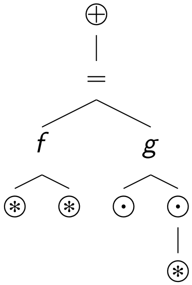
ENIGMA

- **Idea:** Train classifiers by machine learning from a large number of proofs to do good inferences!
- The idea works in other TP contexts and is 20 years old – e.g. premise selection
- The problem is to make it work – and *efficiently*
- **ENIGMA** – since 2017 – stands for...

Efficient learNing-based Inference Guiding MAchine

Clauses as Feature Vectors

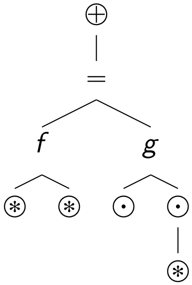
Features are descending paths of length 3 (triples of symbols).
Collect and enumerate all the features. Count the clause features.
Take the counts as a feature vector.



#	feature	count
1	(⊕,=,a)	0
⋮	⋮	⋮
11	(⊕,=,f)	1
12	(⊕,=,g)	1
13	(=,f,⊛)	2
14	(=,g,⊙)	2
15	(g,⊙,⊛)	1
⋮	⋮	⋮

Clauses as Feature Vectors

Features are descending paths of length 3 (triples of symbols).
Collect and enumerate all the features. Count the clause features.
Take the counts as a feature vector.



#	feature	count
1	$(\oplus,=,a)$	0
⋮	⋮	⋮
11	$(\oplus,=,f)$	1
12	$(\oplus,=,g)$	1
13	$(=,f,\circledast)$	2
14	$(=,g,\odot)$	2
15	$(g,\odot,\circledast)$	1
⋮	⋮	⋮

Enigma Model Construction

1. Collect training examples from E runs (useful/useless clauses).
2. Translate clauses to feature vectors.
3. Translate conjectures to feature vectors.
4. Train a classifier on good/bad vector pairs (clause,conjecture)

Outline

Overview: Hammers, Main Result, ATPs & Given Clauses

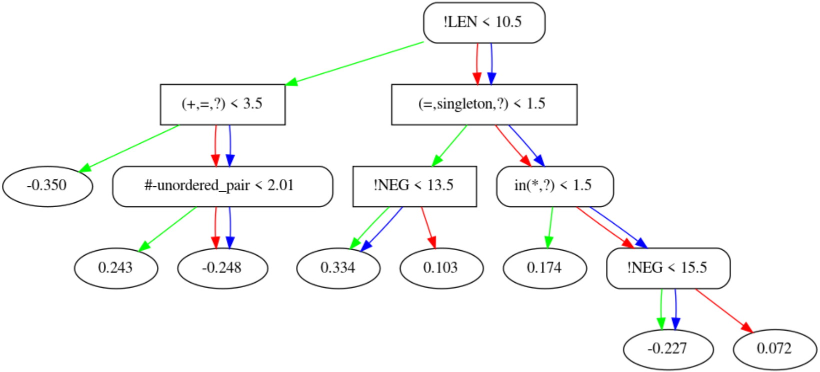
Enigma: The story so far...

Enigma: What's new?

Hammering Mizar

Tree Boosting Classifiers – XGBoost

- State of the art in Machine Learning (before linear/neural ML)
- Much more efficient than deep neural nets
- Stronger than linear classifiers and comparably fast
- An XGBoost model consists of a set of decision trees.
- Leaf scores are summed and translated into probabilities.



Fast Feature Hashing

- In large ITP libraries there are milions of features.
- Handling too long vectors ($> 10^5$) is ineffiecient.
- Solution: Reduce vector dimension with feature hashing.
- Encode features by strings and ...
- ...use a general purpose string hashing function.
- The string hash is reduced to a small integer (e.g. $0..2^{15}$)
- Values are summed in the case of a collision.

Outline

Overview: Hammers, Main Result, ATPs & Given Clauses

Enigma: The story so far. . .

Enigma: What's new?

Hammering Mizar

Evaluation: Hammering Mizar

- 57880 problems extracted from the Mizar Library (MML).
- Good E strategy $\mathcal{S}$ fixed – solves 14933 problems
- We train an XGBoost classifier $\mathcal{M}$ on the proofs
- $\mathcal{S}$ is combined in two ways with the trained classifier $\mathcal{M}$: $\mathcal{S} \odot \mathcal{M}$ and $\mathcal{S} \oplus \mathcal{M}$
- All strategies evaluated with time limit of 10 seconds.

Solved problems: one looping iteration

- Decision trees depth $= 9$
- $\mathcal{M}^0$ is trained on problems solved by $\mathcal{S}$
- $\mathcal{M}^n$ $(n > 0)$ is trained on problems solved by $\mathcal{S}$ and $\mathcal{S} \odot \mathcal{M}^i$ (for all $i < n$) and $\mathcal{S} \oplus \mathcal{M}^i$ (for all $i < n$)

	$\mathcal{S}$	$\mathcal{S} \odot \mathcal{M}^0$	$\mathcal{S} \oplus \mathcal{M}^0$	$\mathcal{S} \odot \mathcal{M}^1$	$\mathcal{S} \oplus \mathcal{M}^1$
solved	14933	16574	20366	21564	22839
$\mathcal{S}\%$	+0%	+10.5%	+35.8%	+43.8%	+52.3%
$\mathcal{S}+$	+0	+4364	+6215	+7774	+8414
$\mathcal{S}-$	-0	-2723	-782	-1143	-508

Solved problems: more loops

	$\mathcal{S}$	$\mathcal{S} \oplus \mathcal{M}^0$	$\mathcal{S} \oplus \mathcal{M}^1$	$\mathcal{S} \oplus \mathcal{M}^2$	$\mathcal{S} \oplus \mathcal{M}^3$
solved	14933	20366	22839	23467	23753
$\mathcal{S}$%	+0%	+35.8%	+52.3%	+56.5%	+58.4
$\mathcal{S}+$	+0	+6215	+8414	+8964	+9274
$\mathcal{S}-$	-0	-782	-508	-430	-454

Solved problems: deeper trees

- Increase tree depth to 12 and 16
- Train the model on the same data as $\mathcal{M}^3$
- Our ultimate strategy solves 70% more than the original in the same real time!

	$\mathcal{S} \odot \mathcal{M}^3_{12}$	$\mathcal{S} \oplus \mathcal{M}^3_{12}$	$\mathcal{S} \odot \mathcal{M}^3_{16}$	$\mathcal{S} \oplus \mathcal{M}^3_{16}$
solved	24159	24701	25100	25397
$\mathcal{S}\%$	+61.1%	+64.8%	+68.0%	**+70.0%**
$\mathcal{S}+$	+9761	+10063	+10476	+10647
$\mathcal{S}-$	-535	-295	-309	-183

ENIGMA Proof Example – Knaster

- 135-long E proof, using 1k given clauses, generating 6k clauses
- solved in the last iteration in 5 seconds: `http://grid01.ciirc.cvut.cz/~mptp/t21_knaster`
- 60-line original proof in MML: `http://grid01.ciirc.cvut.cz/~mptp/7.13.01_4.181.1147/html/knaster#T21`

```
for L being complete Lattice for f being monotone UnOp of L
ex a being Element of L st a is_a_fixpoint_of f
proof
  let L be complete Lattice;
  let f be monotone UnOp of L;
  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
     [... code skipped ]
  end;
  then fH is_less_than f.uH by LATTICE3:def 17;
  then
A3: fuH [= f.uH by LATTICE3:def 21;
  now
    [... code skipped ]
  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;
```

Statistics: data, tree depths, training times, models, speed

- 1.8 M features (hashed to 2^{15})
- vector dimension is 2^{16}
- input trains file is 38 GB
- ... and contains 63 M training samples (4.2M pos x 59M neg)
- ... with 5000 M non-zero values (density 0.1%)

depth	error	real time	CPU time	size (MB)	speed
9	0.201	**2h41m**	**4d20h**	5.0	**5665.6**
12	0.161	4h12m	8d10h	17.4	4676.9
16	**0.123**	6h28m	11d18h	**54.7**	3936.4

Future work

- Do it on other large ITP libraries - AFP, Flyspeck, HOL4, ...
- Dynamic and semantic proof state characterization (ENIGMAWatch)
- Name-independent features (seem to work well)
- Joint training on all ITP libraries (harder)
- Even more iterations and data (now possible)
- *Efficient* Tree and Graph neural nets? (our CADE'19 paper)
- Other ML methods
- ...

Thank you.

Questions?