When Should We Add Theory Axioms And Which Ones?

Giles Reger[1], Martin Suda[1→2]

[1]School of Computer Science, University of Manchester, UK

[2]Institute for Information Systems, TU Vienna, Austria

AITP, April 4, 2016

Outline

Outline

Vampire

- Automated theorem prover for first-order logic (+)
- Regular winner of various divisions in the CACS competition
- Notoriously hard to obtain

Outline

Vampire

- Automated theorem prover for first-order logic (+)
- Regular winner of various divisions in the CACS competition
- Notoriously hard to obtain

Machine learning

- How to select theory axioms
- Our current machine learning playground
- Work in progress report

Vampire and the CASC competition

CASC 2015 results[1]

Higher-order Theorems	Satallax 2.8	LEO-II 1.6.2	*Satallax-l 1.3*	Isabelle 2015
Solved/400	271/400	195/400	285/400	267/400
Av. CPU Time	14.96	12.25	21.67	61.02
Solutions	268/400	191/400	0/400	0/400

Higher-order Non-theorems	Nitpick 2015	Refute 2015	Satallax 2.8
Solved/200	200/200	74/200	49/200
Av. CPU Time	7.92	24.34	0.05

Typed First-order Theorems +*-/	VampireZ 1.0	CVC4 TFA-1.5	Vampire 4.0	Beagle 0.9.22	SPASS+T 2.2.22	ZenonAri 0.1.0	Princess 20150706	*CVC4 1.4-TFF*
Solved/200	172/200	163/200	160/200	131/200	108/200	60/200	143/200	131/200
Av. CPU Time	11.85	17.27	10.75	21.76	10.04	2.86	17.38	10.67
Solutions	172/200	163/200	160/200	131/200	108/200	60/200	0/200	0/200

Typed First-order Non-theorems +*-/	CVC4 TFN-1.5	Princess 20150706	Beagle 0.9.22
Solved/20	10/20	6/20	6/20
Av. CPU Time	0.00	0.97	1.33

First-order Theorems	Vampire 4.0	*Vampire 2.6*	E 1.9.1	ET 0.2	CVC4 FOF-1.5	iProver 2.0	leanCoP 2.2	iProverM 0.7-0.3	Prover9 1109a	ePrincess 1.0	Muscadet 4.5	Geo-III 2015E
Solved/400	380/400	371/400	316/400	303/400	257/400	222/400	159/400	127/400	111/400	113/400	37/400	37/400
Av. CPU Time	12.20	14.86	20.18	20.96	33.40	21.12	46.76	30.15	28.01	48.39	7.32	38.47
Solutions	374/400	368/400	316/400	303/400	256/400	217/400	159/400	127/400	111/400	88/400	37/400	37/400

First-order Non-theorems	Vampire SAT-4.0	iProver SAT-2.0	*iProver SAT-1.0*	CVC4 FNT-1.5	E FNT-1.9.1	Geo-III 2015E
Solved/200	195/200	163/200	134/200	71/200	51/200	38/200
Av. CPU Time	38.95	44.11	79.93	57.78	9.62	21.89
Solutions	195/200	163/200	134/200	71/200	51/200	38/200

Effectively Propositional CNF	Vampire 4.0	*iProver 0.9*	iProver 2.0	E 1.9.1	Geo-III 2015E
Solved/200	192/200	161/200	153/200	101/200	9/200
Av. CPU Time	27.61	27.91	36.57	11.09	86.71

Large Theory Batch Problems	Vampire 4.0-LTB	MaLARea 0.5	E 1.9.1-LTB	iProver 2.0-LTB
Solved/1600	1208/1600	837/1600	799/1600	352/1600
Av. WC Time	6.51	8.30	7.40	13.04
Solutions	1208/1600	837/1600	799/1600	352/1600

[1] http://www.cs.miami.edu/~tptp/CASC/25/WWWFiles/DivisionSummary1.html

Why I think Vampire is good

- State of the art calculi / techniques
  - superposition [BG94,NR01]
  - AVATAR [V14]
  - InstGen [GK03]
  - finite model finding [McC94,CS04]
  - SInE [HV11]

Why I think Vampire is good

- State of the art calculi / techniques
  - superposition [BG94,NR01]
  - AVATAR [V14]
  - InstGen [GK03]
  - finite model finding [McC94,CS04]
  - SInE [HV11]
- Careful engineering
  - indexing is essential [V95,V01]

Why I think Vampire is good

- State of the art calculi / techniques
  - superposition [BG94,NR01]
  - AVATAR [V14]
  - InstGen [GK03]
  - finite model finding [McC94,CS04]
  - SInE [HV11]
- Careful engineering
  - indexing is essential [V95,V01]
- Heavy (optional) use of incomplete but useful procedures
  - Limited Resource Strategy [RV03]
  - Literal selection [HRSV16]
  - Set of Support
  - ...

Why I think Vampire is good

- State of the art calculi / techniques
  - superposition [BG94,NR01]
  - AVATAR [V14]
  - InstGen [GK03]
  - finite model finding [McC94,CS04]
  - SInE [HV11]
- Careful engineering
  - indexing is essential [V95,V01]
- Heavy (optional) use of incomplete but useful procedures
  - Limited Resource Strategy [RV03]
  - Literal selection [HRSV16]
  - Set of Support
  - ...
- Decades of experience about the right design decisions

  [Andrei Voronkov]

Why I think Vampire is good

- State of the art calculi / techniques
  - superposition [BG94,NR01]
  - AVATAR [V14]
  - InstGen [GK03]
  - finite model finding [McC94,CS04]
  - SInE [HV11]
- Careful engineering
  - indexing is essential [V95,V01]
- Heavy (optional) use of incomplete but useful procedures
  - Limited Resource Strategy [RV03]
  - Literal selection [HRSV16]
  - Set of Support
  - ...
- Decades of experience about the right design decisions

  [Andrei Voronkov]
- Database of problems and proofs and strategy scheduling based on it

The need for many strategies

- Theorem proving is hard
- Chaos reigns (butterfly effect)
- If a strategy solves, it usually does so very fast!
- We need to combine strategies
  - not only good ones overall
  - but also complementary / exotic ones

The need for many strategies

- Theorem proving is hard
- Chaos reigns (butterfly effect)
- If a strategy solves, it usually does so very fast!
- We need to combine strategies
  - not only good ones overall
  - but also complementary / exotic ones

CASC-mode

- Conditional schedule of strategies
- Optimized for a good coverage over the TPTP

A CASC-mode code excerpt

```
case Property::FNE:
  if (atoms > 2000) {
    quick.push("dis+1011_40_bs=on:cond=on:gs=on:gsaa=from_current:nwc=1:sfr
    quick.push("lrs+1011_3_nwc=1:stl=90:sos=on:spl=off:sp=reverse_arity_133
    quick.push("dis-10_5_cond=fast:gsp=input_only:gs=on:gsem=off:nwc=1:sas=
    quick.push("lrs+1011_5_cond=fast:gs=on:nwc=2.5:stl=30:sd=3:ss=axioms:sd
    quick.push("lrs-3_5:4_bs=on:bsr=on:cond=on:fsr=off:gsp=input_only:gs=on
  }
  else if (atoms > 1200) {
    quick.push("lrs+1011_5_cond=fast:gs=on:nwc=2.5:stl=30:sd=3:ss=axioms:sd
    quick.push("dis+1011_8_bsr=unit_only:cond=fast:fsr=off:gs=on:gsaa=full_
    quick.push("dis+11_7_gs=on:gsaa=full_model:lcm=predicate:nwc=1.1:sas=mi
    quick.push("ins+11_5_br=off:gs=on:gsem=off:igbrr=0.9:igrr=1/64:igrp=140
  }
  else {
    quick.push("dis+11_7_16");
    quick.push("dis+1011_5:4_gs=on:gsssp=full:nwc=1.5:sas=minisat:ssac=none
    quick.push("dis+1011_40_bs=on:cond=on:gs=on:gsaa=from_current:nwc=1:sfr
    ...
```

Vampire and arithmetic

The big next challenge

- Reasoning with quantifiers and theories

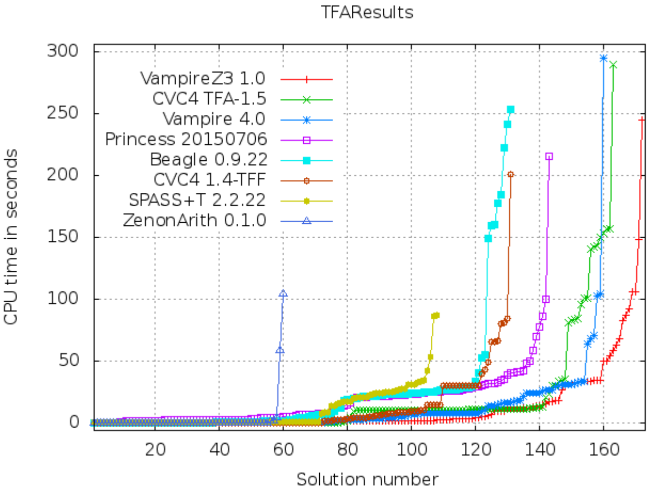
Vampire and arithmetic

The big next challenge

- Reasoning with quantifiers and theories

- Evaluation of ground interpreted terms ($1 + 1 \longrightarrow 2$)
- Interpreted operations treated specially by ordering
- Normalization of interpreted operations, i.e. only use $\leq$
- Theory axioms
  - hand-crafted set
  - either all added or none added (based on option)
- AVATAR with an SMT solver
  - current implementation for Z3
  - Idea: Vampire only explores theory-consistent ground sub-problems

Results for TFA (Typed First-order Theorems +*-/)[2]



[2] http://www.cs.miami.edu/~tptp/CASC/25/WWWFiles/ResultsPlots.html

Axiom selection experiment

```
tff(mix_quant_ineq_sys_solvable_2,conjecture,(
    ! [X: $int] : ( $less(5,X)
     => ? [Y: $int] : ( $less(Y,3)
          & $less(7,$sum(X,Y)) ) ) )).
```

Motivation

ARI581=1.p is a small problem which the default strategy solves instantly if we add all axioms except the commutativity of $+$, but does not solve in 60 seconds with commutativity.

Axiom selection experiment

```
tff(mix_quant_ineq_sys_solvable_2,conjecture,(
    ! [X: $int] : ( $less(5,X)
     => ? [Y: $int] : ( $less(Y,3)
         & $less(7,$sum(X,Y)) ) ) )).
```

Motivation

ARI581=1.p is a small problem which the default strategy solves instantly if we add all axioms except the commutativity of $+$, but does not solve in 60 seconds with commutativity.

The experiment

Take the 15 pre-selected axioms for reasoning about linear integers, consider all 2^{15} strategies corresponding to each subset, evaluate them on a set of problems and see what can be (machine-) learned from that.

The 15 hand-crafted axioms (for linear integers)

1	$X + 0 = X$	2	$0 + X = X$
3	$X + Y = Y + X$	4	$X + (Y + Z) = (X + Y) + Z$
5	$0 = X + (-X)$	6	$(-X) + (-Y) = -(X + Y)$
7	$(X + (-Y)) + Y = X$		
8	$X \leq X$	9	$X \leq Y \vee Y \leq X$
10	$X \not\leq Y \vee Y \not\leq X \vee X = Y$	11	$X \not\leq Y \vee Y \not\leq Z \vee X \leq Z$
12	$X \leq Y \vee Y + 1 \leq X$	13	$X \not\leq Y \vee Y + 1 \not\leq X$
14	$X + 1 \not\leq X$	15	$X \not\leq Y \vee X + Z \leq Y + Z$

Preparation

Test problems selection

- Start with all TFA problems in TPTP (1128 problems)
- Focus on pure integer arithmetic with linear operators (+,-) (giving 515 problems)
- Drop those solvable by Vampire using the default strategy without theory axioms (and no Z3) in 30 seconds
- Giving us 282 problems in total

Preparation

Test problems selection

- Start with all TFA problems in TPTP (1128 problems)
- Focus on pure integer arithmetic with linear operators (+,-) (giving 515 problems)
- Drop those solvable by Vampire using the default strategy without theory axioms (and no Z3) in 30 seconds
- Giving us 282 problems in total

Obtaining the data

- There are 15 theory axioms relevant to our set of problems
- This gives 32,768 combinations of theory axioms
- Given 282 problems this gives 9,273,344 experiments
- We ran each experiment for 5 seconds
- Almost 1.4 years of computation time... Thank you, StarExec!

"The cube" – basic info

Strategies

- min: 0 at (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
- med: 63 at (0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1)
- max: 115 at (0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1)
- avg: 60.9
- $2^{15} - 4$ such that there exists a problem solved by it

“The cube” – basic info

Strategies

- min: 0 at (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
- med: 63 at (0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1)
- max: 115 at (0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1)
- avg: 60.9
- $2^{15} - 4$ such that there exists a problem solved by it

Problems

- min: 9 at `ARI182=1.p`
- med: 11869 at `DAT026=1.p`
- max: 32460 at `NUM893=1.p`
- avg: 14054.0
- 142 such that there exists a strategy solving it

Reducing the complexity without losing solutions

$*$-notations

$S(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) = 0$
$S(0,1,1,0,0,0,1,1,0,1,1,1,0,1,1) = 115$
$S(*,*,*,*,*,*,*,*,*,*,*,*,*,*,*) = 142$
$C(*,*,*,*,*,*,*,*,*,*,*,*,*,*,*) = 15$

Reducing the complexity without losing solutions

$*$-notations

$S(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) = 0$
$S(0,1,1,0,0,0,1,1,0,1,1,1,0,1,1) = 115$
$S(*,*,*,*,*,*,*,*,*,*,*,*,*,*,*) = 142$
$C(*,*,*,*,*,*,*,*,*,*,*,*,*,*,*) = 15$

Hardcoding choices about particular axioms

- Start from $a = (*,*,*,*,*,*,*,*,*,*,*,*,*,*,*)$
- If there is an index $i = 1,\ldots,15$ s.t. $a[i] = *$ and a value $v \in \{0,1\}$ s.t. $S(a) = S(a[i \to v])$ then recurse on $a[i \to v]$
- otherwise report a and $C(a)$

Reducing the complexity – results

Four winners

$a_1 = (0, 0, *, *, 0, 0, *, 1, *, *, *, *, 0, *, *)$
$a_2 = (1, *, *, *, 0, 0, 1, 0, *, *, *, *, 0, *, *)$
$a_3 = (1, 0, *, *, 0, 0, *, 0, *, *, *, *, 0, *, *)$
$a_4 = (1, 0, *, 1, *, 0, *, 1, *, *, *, *, *, 0, *)$
$C(a_i) = 9, S(a_i) = 142$

Reducing the complexity – results

Four winners

$a_1 = (0, 0, *, *, 0, 0, *, 1, *, *, *, *, 0, *, *)$
$a_2 = (1, *, *, *, 0, 0, 1, 0, *, *, *, *, 0, *, *)$
$a_3 = (1, 0, *, *, 0, 0, *, 0, *, *, *, *, 0, *, *)$
$a_4 = (1, 0, *, 1, *, 0, *, 1, *, *, *, *, *, 0, *)$
$C(a_i) = 9, S(a_i) = 142$

Other "leaf" nodes

- $S(\_) = 142$, but $C(\_) > 9$ and cannot be minimized further
- 31 more with $C(\_) = 10$
  20 more with $C(\_) = 11$
  6 more with $C(\_) = 12$

“Greedy” CASC mode creation

Finding a good schedule

- pose as the set cover problem
- employ the obvious greedy algorithm

"Greedy" CASC mode creation

Finding a good schedule

- pose as the set cover problem
- employ the obvious greedy algorithm

ord	contrib	choices	best	strategy
1	115	1	115	(0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1)
2	12	5	93	(0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1)
3	6	5	87	(0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1)
4	3	38	90	(1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1)
5	2	17	49	(1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0)
6	1	459	100	(1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1)
7	1	450	88	(0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1)
8	1	229	85	(0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1)
9	1	166	67	(1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0)
	142	$\ll 2^{15}$		

Machine Learning Experiments

- Tried to apply some out-of-the-box techniques
- More specifically
  - Split problems into training and testing
  - Extracted some features from problems
  - Used these to prepare some
  - Downloaded WEKA and tried running some of the algorithms
- Details next...
- Summary of lessons learned
  - Nothing truly 'out-of-the-box' as need to understand parameters
  - WEKA struggled with amount of data
  - Still not clear how best to harness machine learning

Problem Features

- Just considered static features initially. For example,
  - Standard syntactic features not related to theory reasoning
  - Frequency of each interpreted operation (generally and in goal)
  - Frequency of sorted variables and equalities (generally and in goal)
  - Usage of special numbers 0 and 1
- Ideas for dynamic features (i.e. after short run)

  Inspect descendants of each theory axiom and look for
  - Involvement with goal
  - Reductions (of and with)
  - Interaction with other theory axioms (pure descendants)
  - Groundness

Idea 1: Classification

- Want: function from problem feature vector to set of theory axioms
- Issue: 2^{15} different 'classes'
- Idea: train classifier per axiom, with other axioms as extra features
- i.e. *given problem and other axioms should I use this one?*
- New Issue: Unclear how to combine classifiers (search problem?)

- Tried a few algorithms on slightly different problem
  - Given problem features, axioms used and class (whether solved)
  - Build model for predicting class
  - Linear regression had 0.72 accuracy
  - Naive Bayes had 0.829 precision, 0.593 recall
  - SVM methods never finished

Idea 2: Association Rule Mining

- Idea: Mine rules that indicate associations between axioms
- Hopefully of the form *If adding A then I should probably (not) add B*
- Could be used to suggest which axiom sets are sensible

- Input is just the set of axioms used for each experiment
- Currently treat positive and negative data separately

- Use association rule mining
- Initial experiment failed to find rules with good confidence

Tentative conclusion

What have been done?

- No blood, sweat, nor tears, yet!
- Simplified “small” setup

Tentative conclusion

What have been done?

- No blood, sweat, nor tears, yet!
- Simplified “small” setup

In real life . . .

- only limited number of samples from the strategy space
  - but can get as many as we want
- how to sample adaptively?

Tentative conclusion

What have been done?

- No blood, sweat, nor tears, yet!
- Simplified "small" setup

In real life ...

- only limited number of samples from the strategy space
  - but can get as many as we want
- how to sample adaptively?

Other things to try

- mining proofs to see which axioms were used together in proofs, or more complex relations
- ...

More questions

How do we evaluate what we (will) have done?

- It is too easy to win against a single best strategy!
- With time reduced to 2.5s the best strategy still solves 112 problems and the largest union of two strategies has size 125.

More questions

How do we evaluate what we (will) have done?

- It is too easy to win against a single best strategy!
- With time reduced to 2.5s the best strategy still solves 112 problems and the largest union of two strategies has size 125.

For theory axioms; what is important?

- Is it more important to be conservative, i.e., knowing what not to add to avoid explosion?
- Is there actually a problem to be solved via machine learning here, or can we just develop some hand-built heuristics that are good enough?

Thank you for attention!

Any answers?

Thank you for attention!

Any questions?

Thank you for attention!

Let's go skiing!