When Should We Add Theory Axioms And Which Ones?

Giles Reger\textsuperscript{1}, Martin Suda\textsuperscript{1→2}

\textsuperscript{1}School of Computer Science, University of Manchester, UK

\textsuperscript{2}Institute for Information Systems, TU Vienna, Austria

AITP, April 4, 2016
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
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

<table>
<thead>
<tr>
<th>Higher-order Theorems</th>
<th>Satallax</th>
<th>LEO-II</th>
<th>Satallax - Isabelle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solved: 400</td>
<td>271/400</td>
<td>195/400</td>
<td>285/400 267/400</td>
</tr>
<tr>
<td>Av. CPU Time</td>
<td>14.96</td>
<td>12.25</td>
<td>21.67 61.02</td>
</tr>
<tr>
<td>Solutions</td>
<td>268/400</td>
<td>191/400</td>
<td>0/400   0/400</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Higher-order Non-theorems</th>
<th>Nitpick</th>
<th>Refute</th>
<th>Satallax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solved: 200</td>
<td>200/200</td>
<td>74/200</td>
<td>49/200</td>
</tr>
<tr>
<td>Av. CPU Time</td>
<td>7.92</td>
<td>24.34</td>
<td>0.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Typed First-order Theorems +<em>/</em>-</th>
<th>Vampire</th>
<th>CVC4</th>
<th>Vampire</th>
<th>Beagle</th>
<th>SPASS+T</th>
<th>ZenonArI</th>
<th>Princess</th>
<th>CYC4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Av. CPU Time</td>
<td>11.85</td>
<td>17.27</td>
<td>10.75</td>
<td>21.76</td>
<td>10.04</td>
<td>2.86</td>
<td>17.38</td>
<td>10.67</td>
</tr>
<tr>
<td>Solutions</td>
<td>172/200</td>
<td>163/200</td>
<td>160/200</td>
<td>131/200</td>
<td>108/200</td>
<td>60/200</td>
<td>0/200</td>
<td>0/200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Typed First-order Non-theorems +<em>/</em>-</th>
<th>CVC4</th>
<th>Princess</th>
<th>Beagle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Av. CPU Time</td>
<td>0.00</td>
<td>0.97</td>
<td>1.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First-order Theorems</th>
<th>Vampire</th>
<th>Vampire</th>
<th>E</th>
<th>ET</th>
<th>CVC4</th>
<th>iProver</th>
<th>leanCoP</th>
<th>iProverM</th>
<th>Prover9</th>
<th>ePrincess</th>
<th>Mace4</th>
<th>Geo-III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solved: 400</td>
<td>380/400</td>
<td>371/400</td>
<td>316/400</td>
<td>303/400</td>
<td>257/400</td>
<td>222/400</td>
<td>159/400</td>
<td>127/400</td>
<td>111/400</td>
<td>113/400</td>
<td>37/400</td>
<td></td>
</tr>
<tr>
<td>Av. CPU Time</td>
<td>12.20</td>
<td>14.86</td>
<td>20.18</td>
<td>20.96</td>
<td>33.40</td>
<td>21.12</td>
<td>46.76</td>
<td>30.15</td>
<td>28.01</td>
<td>48.39</td>
<td>7.32</td>
<td>38.47</td>
</tr>
<tr>
<td>Solutions</td>
<td>374/400</td>
<td>368/400</td>
<td>316/400</td>
<td>303/400</td>
<td>256/400</td>
<td>217/400</td>
<td>159/400</td>
<td>127/400</td>
<td>111/400</td>
<td>88/400</td>
<td>37/400</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First-order Non-theorems</th>
<th>Vampire</th>
<th>iProver</th>
<th>iProver</th>
<th>CVC4</th>
<th>E</th>
<th>Geo-III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solved: 200</td>
<td>195/200</td>
<td>163/200</td>
<td>134/200</td>
<td>71/200</td>
<td>51/200</td>
<td>38/200</td>
</tr>
<tr>
<td>Av. CPU Time</td>
<td>38.95</td>
<td>44.11</td>
<td>57.09</td>
<td>71/200</td>
<td>51/200</td>
<td>38/200</td>
</tr>
<tr>
<td>Solutions</td>
<td>195/200</td>
<td>163/200</td>
<td>134/200</td>
<td>71/200</td>
<td>51/200</td>
<td>38/200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Effectively Propositional CNF</th>
<th>Vampire</th>
<th>iProver</th>
<th>iProver</th>
<th>E</th>
<th>Geo-III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Av. CPU Time</td>
<td>27.61</td>
<td>27.91</td>
<td>36.57</td>
<td>11.09</td>
<td>86.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Large Theory Batch Problems</th>
<th>Vampire</th>
<th>MalMig</th>
<th>E</th>
<th>iProver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solved: 1600</td>
<td>1208/1600</td>
<td>837/1600</td>
<td>799/1600</td>
<td>352/1600</td>
</tr>
<tr>
<td>Av. WC Time</td>
<td>6.51</td>
<td>8.30</td>
<td>7.40</td>
<td>13.04</td>
</tr>
<tr>
<td>Solutions</td>
<td>1208/1600</td>
<td>837/1600</td>
<td>799/1600</td>
<td>352/1600</td>
</tr>
</tbody>
</table>
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("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=minisat:sos=all:spl=off:sp=occurrence_190");
    } 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("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");
    }
...
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 ($1 + 1 \rightarrow 2$)
- 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

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

Reger, Suda
When Should We Add Theory Axioms And Which Ones?
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. \(X \leq Y \lor Y \leq X\)
10. \(X \preceq Y \lor Y \preceq X \lor X = Y\)
11. \(X \preceq Y \lor Y \preceq Z \lor X \leq Z\)
12. \(X \leq Y \lor Y + 1 \leq X\)
13. \(X \preceq Y \lor Y + 1 \preceq X\)
14. \(X + 1 \preceq X\)
15. \(X \preceq Y \lor X + Z \leq Y + Z\)
## 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, 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**

\[
S(0,0,0,0,0,0,0,0,0,0,0,0) = 0 \\
S(0,1,1,0,0,1,1,0,1,1,0,1,1) = 115 \\
S(*,*,*,*,*,*,*,*,*,*,*,*) = 142 \\
C(*,*,*,*,*,*,*,*,*,*,*,*) = 15
\]
Reducing the complexity without losing solutions

**-notations**

\[ S(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) = 0 \]
\[ S(0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1) = 115 \]
\[ S(*, *, *, *, *, *, *, *, *, *, *, *) = 142 \]
\[ C(*, *, *, *, *, *, *, *, *, *, *, *, *) = 15 \]

**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. \( S(a) = S(a[i \rightarrow v]) \) then recurse on \( a[i \rightarrow v] \)
- otherwise report \( a \) and \( C(a) \)
Reducing the complexity – results

Four winners

\[
\begin{align*}
  a_1 &= (0, 0, *, *, 0, 0, *, 1, *, *, *, 0, *, *) \\
  a_2 &= (1, *, *, *, 0, 0, 1, 0, *, *, *, 0, *, *) \\
  a_3 &= (1, 0, *, *, 0, 0, *, 0, *, *, 0, *, *) \\
  a_4 &= (1, 0, *, 1, *, 0, *, 1, *, *, *, *, 0, *) \\
  C(a_i) &= 9, S(a_i) = 142
\end{align*}
\]
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

<table>
<thead>
<tr>
<th>ord</th>
<th>contrib</th>
<th>choices</th>
<th>best</th>
<th>strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>115</td>
<td>1</td>
<td>115</td>
<td>(0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1)</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>93</td>
<td>(0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1)</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>5</td>
<td>87</td>
<td>(0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1)</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>38</td>
<td>90</td>
<td>(1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1)</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>17</td>
<td>49</td>
<td>(1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0)</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>459</td>
<td>100</td>
<td>(1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1)</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>450</td>
<td>88</td>
<td>(0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1)</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>229</td>
<td>85</td>
<td>(0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1)</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>166</td>
<td>67</td>
<td>(1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0)</td>
</tr>
</tbody>
</table>

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!