Set of Support for Theory Reasoning

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

\textsuperscript{1}School of Computer Science, University of Manchester, UK
\textsuperscript{2}TU Wien, Vienna, Austria

IWIL 2017 – Maun, May 7, 2017
Consider the following toy theory problem

\[ f(1 + a) < a, \quad \forall x. (x < f(x + 1)) \]
Consider the following toy theory problem

\[ f(1 + a) < a, \quad \forall x. (x < f(x + 1)) \]

can be refuted by Vampire via the following derivation:

\[
\begin{align*}
\begin{array}{l}
\text{x + y = y + x} \\
\text{x < f(x + 1)} \\
\hline
\text{x < f(1 + x)}
\end{array}
\quad \begin{array}{l}
\neg x < y \lor \neg y < z \lor x < z \\
\hline
\neg (x < f(1 + a)) \lor x < a
\end{array}
\quad \begin{array}{l}
f(1 + a) < a \\
\hline
a < a
\end{array}
\quad \begin{array}{l}
\neg (x < x)
\hline
\bot
\end{array}
\end{align*}
\]
Consider the following toy theory problem

\[ f(1 + a) < a, \quad \forall x.(x < f(x + 1)) \]

can be refuted by Vampire via the following derivation:

\[
\begin{align*}
  x + y &= y + x \\
  x < f(x + 1) &\quad \Rightarrow \quad x < f(1 + x) \\
  \neg x < y \lor \neg y < z \lor x < z &\quad \Rightarrow \quad \neg(x < f(1 + a)) \lor x < a \\
  a < a &\quad \Rightarrow \quad \neg(x < x)
\end{align*}
\]

However, in the meantime, the theory axioms may also yield:

\[ \neg(x < y) \lor \neg(y < x) \]

or (perhaps less usefully):

\[ \neg(x_0 < x_1) \lor \neg(x_2 < x_0) \lor \neg(x_1 < x_3) \lor \neg(x_4 < x_5) \lor \neg(x_3 < x_4) \lor \neg(x_5 < x_2) \]
Inferences between axioms

Example problem ARI176=1 from TPTP

\[ 3x + 5y \neq 22 \]

can be shown unsatisfiable using axioms

\[ x + y = y + x, \quad x + (y + z) = (x + y) + z, \quad x \times 1 = x, \quad x \times (y + z) = (x \times y) + (x \times z) \]
Inferences between axioms

Example problem ARI176=1 from TPTP

\[ 3x + 5y \neq 22 \]

can be shown unsatisfiable using axioms

\[ x + y = y + x, \quad x + (y + z) = (x + y) + z, \quad x \cdot 1 = x, \quad x \cdot (y + z) = (x \cdot y) + (x \cdot z) \]

The derivation starts by:

\[
\begin{align*}
    x \cdot 1 &= x \\
    x \cdot (y + z) &= (x \cdot y) + (x \cdot z) \\
    x \cdot (1 + y) &= x + (x \cdot y) \\
    x + (y + z) &= (x + y) + z \\
    (x \cdot (1 + y)) + z &= x + ((x \cdot y) + z)
\end{align*}
\]
Inferences between axioms

Example problem ARI176=1 from TPTP

\[ 3x + 5y \neq 22 \]

can be shown unsatisfiable using axioms

\[ x + y = y + x, \quad x + (y + z) = (x + y) + z, \quad x \cdot 1 = x, \quad x \cdot (y + z) = (x \cdot y) + (x \cdot z) \]

The derivation starts by:

\[
\begin{align*}
  x \cdot 1 &= x \\
  x \cdot (y + z) &= (x \cdot y) + (x \cdot z) \\
  x \cdot (1 + y) &= x + (x \cdot y) \\
  x + (y + z) &= (x + y) + z \\
  (x \cdot (1 + y)) + z &= x + ((x \cdot y) + z)
\end{align*}
\]

The problem cannot be solved in Vampire in reasonable time without first combining axioms among themselves
One useful technique for reasoning with theories and quantifiers is the addition of theory axioms.
One useful technique for reasoning with theories and quantifiers is the **addition of theory axioms**.

Quite successful in many cases. However, many axioms can be “explosive”.
One useful technique for reasoning with theories and quantifiers is the addition of theory axioms. Quite successful in many cases. However, many axioms can be “explosive”. Set of support is a well known idea to prevent explosion.
One useful technique for reasoning with theories and quantifiers is the *addition* of theory axioms.

Quite successful in many cases. However, many axioms can be “explosive”.

*Set of support* is a well known idea to prevent explosion.

Idea 1: apply SOS for theory reasoning.
One useful technique for reasoning with theories and quantifiers is the addition of theory axioms. Quite successful in many cases. However, many axioms can be “explosive”. Set of support is a well known idea to prevent explosion. Idea 1: apply SOS for theory reasoning. Idea 2: fine-tune this by allowing limited reasoning among theory axioms.
One useful technique for reasoning with theories and quantifiers is the addition of theory axioms. Quite successful in many cases. However, many axioms can be "explosive". Set of support is a well known idea to prevent explosion. Idea 1: apply SOS for theory reasoning. Idea 2: fine-tune this by allowing limited reasoning among theory axioms. Preliminary evaluation of the technique.
1 Saturation and Theory Reasoning in Vampire

2 The Set of Support Strategy

3 Set of Support for Theory Reasoning

4 Conclusion
Saturation-based Theorem Proving

Compute deductive closure of the input $N$ wrt inferences $\mathcal{I}$:
Saturation-based Theorem Proving

Compute deductive closure of the input $N$ wrt inferences $\mathcal{I}$:

- clause selection schemes
- further aspects: literal selection, ordering restrictions, ...
- completeness considerations
Theory Reasoning in Vampire

Main focus

Reasoning with quantifiers and theories
Main focus

Reasoning with quantifiers and theories

Current arsenal:
Main focus

Reasoning with quantifiers and theories

Current arsenal:

- Evaluation of ground interpreted terms:
  \[ 1 + 1 \implies 2, \ 1 < 1 \implies false, \ldots \]
Main focus

Reasoning with quantifiers and theories

Current arsenal:

- Evaluation of ground interpreted terms:
  \[ 1 + 1 \rightarrow 2, \ 1 < 1 \rightarrow false, \ldots \]

- Interpreted operations treated specially by ordering
Main focus
Reasoning with quantifiers and theories

Current arsenal:

- Evaluation of ground interpreted terms:
  \[1 + 1 \rightarrow 2, \ 1 < 1 \rightarrow \text{false}, \ldots\]
- Interpreted operations treated specially by ordering
- Normalization of interpreted operations, i.e. only use \(<\)
Main focus
Reasoning with quantifiers and theories

Current arsenal:
- Evaluation of ground interpreted terms:
  \[1 + 1 \rightarrow 2, \quad 1 < 1 \rightarrow \text{false}, \ldots\]
- Interpreted operations treated specially by ordering
- Normalization of interpreted operations, i.e. only use \(<\)
- Theory axioms
  - hand-crafted set
  - either all added or none added (based on an option)
Main focus

Reasoning with quantifiers and theories

Current arsenal:

- Evaluation of ground interpreted terms:
  \[ 1 + 1 \rightarrow 2, \ 1 < 1 \rightarrow \text{false}, \ldots \]
- Interpreted operations treated specially by ordering
- Normalization of interpreted operations, i.e. only use \(<\n- Theory axioms
  - hand-crafted set
  - either all added or none added (based on an option)
- AVATAR with an SMT solver
  - Idea: Vampire only explores theory-consistent ground sub-problems
Main focus

Reasoning with quantifiers and theories

Current arsenal:

- Evaluation of ground interpreted terms:
  \[1 + 1 \rightarrow 2, \ 1 < 1 \rightarrow false, \ldots\]
- Interpreted operations treated specially by ordering
- Normalization of interpreted operations, i.e. only use <
- Theory axioms
  - hand-crafted set
  - either all added or none added (based on an option)
- AVATAR with an SMT solver
  - Idea: Vampire only explores theory-consistent ground sub-problems
- Theory Instantiation and Unification with Abstraction
Theory Reasoning in Vampire

Main focus
Reasoning with quantifiers and theories

Current arsenal:

- Evaluation of ground interpreted terms:
  \[ 1 + 1 \rightarrow 2, \ 1 < 1 \rightarrow false, \ldots \]
- Interpreted operations treated specially by ordering
- Normalization of interpreted operations, i.e. only use \(<\)
- **Theory axioms**
  - hand-crafted set
  - either all added or none added (based on an option)
- AVATAR with an SMT solver
  - Idea: Vampire only explores theory-consistent ground sub-problems
- Theory Instantiation and Unification with Abstraction
Some axioms

\begin{align*}
  x + (y + z) &= (x + y) + z \\
  x + y &= y + x \\
  -x &= x \\
  x \cdot 0 &= 0 \\
  x \cdot 1 &= x \\
  (x \cdot y) + (x \cdot z) &= x \cdot (y + z) \\
  x < y \lor y < x \lor x = y \\
  \neg (x < y) \lor x + z < y + z \\
  x < y \lor y < x + 1 \text{ (for ints)}
\end{align*}

\begin{align*}
  x + 0 &= x \\
  -(x + y) &= (-x + -y) \\
  x + (-x) &= 0 \\
  x \cdot (y \cdot z) &= (x \cdot y) \cdot z \\
  x \cdot y &= y \cdot x \\
  \neg (x < y) \lor \neg (y < z) \lor \neg (x < z) \\
  \neg (x < y) \lor \neg (y < x + 1) \\
  \neg (x < x) \\
  x &= 0 \lor (y \cdot x)/x = y \text{ (for reals)}
\end{align*}
Axioms can be “explosive”

```prolog
ARl581=1.p
tff(mix_quant_ineq_sys_solvable_2,conjecture,(  
  ! [X: $int] : ( $less(5,X) =>  
    ? [Y: $int] : ( $less(Y,3) & $less(7,$sum(X,Y))))))).
```

- default strategy with all axioms: not solved in 60 s
- remove commutativity of \(+\): solved instantly
Axioms can be “explosive”

ARl581=1.p

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

- default strategy with all axioms: not solved in 60 s
- remove commutativity of +: solved instantly

SYN000=2.p

- “test tptp theory syntax” benchmark
- Vampire in default: 223 clauses (90 theory consequences, 1 used in the proof)
- negate the conjecture, run for 10 s: 456 973 clauses (98 % are consequences of theory axioms)
Outline

1. Saturation and Theory Reasoning in Vampire
2. The Set of Support Strategy
3. Set of Support for Theory Reasoning
4. Conclusion
The Set of Support Strategy

Basic idea:
- split the input clauses into a set of support and the rest
- restrict inferences to involve at least one premise from SOS
- new clauses are added to SOS

“Every inference must have an ancestor in the initial SOS.”
The Set of Support Strategy

Basic idea:
- split the input clauses into a **set of support** and the rest
- restrict inferences to involve at least one premise from SOS
- new clauses are added to SOS

“Every inference must have an ancestor in the initial SOS.”

In practice:
- just put non-SOS clauses directly to active
The Set of Support Strategy

Basic idea:

- split the input clauses into a set of support and the rest
- restrict inferences to involve at least one premise from SOS
- new clauses are added to SOS

“Every inference must have an ancestor in the initial SOS.”

In practice:

- just put non-SOS clauses directly to active
- define SOS = clauses from the conjecture
  - Note: benchmarks without explicit conjecture SOS-suck
Vampire’s `-sos` option values:

- **off**: do not use SOS
- **on**: standard SOS
- **all**: SOS + select all literals of clauses in “initially active”
Vampire’s -sos option values:

- off: do not use SOS
- on: standard SOS
- all: SOS + select all literals of clauses in “initially active”

Experiment (relevant TPTP v6.4.0, 300 s)

<table>
<thead>
<tr>
<th></th>
<th>competition mode</th>
<th>competition mode with sos=off</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solved</td>
<td>11 948</td>
<td>11 613</td>
</tr>
<tr>
<td>Uniques</td>
<td>422</td>
<td>87</td>
</tr>
</tbody>
</table>
Outline

1. Saturation and Theory Reasoning in Vampire
2. The Set of Support Strategy
3. Set of Support for Theory Reasoning
4. Conclusion
SOS and theory axioms

- the whole input problem is the SOS
- added theory axioms go directly to active
- new, fourth `-sos` option value: `theory`
SOS and theory axioms
- the whole input problem is the SOS
- added theory axioms go directly to active
- new, fourth -sos option value: theory
- Also applies to problems without explicit conjecture!
SOS for Theories

SOS and theory axioms

- the whole input problem is the SOS
- added theory axioms go directly to active
- new, fourth -sos option value: theory
- Also applies to problems without explicit conjecture!

Experiment (relevant SMTLIB, default strategy, 60 s)

<table>
<thead>
<tr>
<th></th>
<th>default mode</th>
<th>default mode + sos=theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solved</td>
<td>32 769</td>
<td>32 522</td>
</tr>
<tr>
<td>Uniques</td>
<td>641</td>
<td>394</td>
</tr>
</tbody>
</table>
How deep is theory reasoning?

Mining proofs for statistics:

- record maximum derivation depth of a pure theory consequence used in the proof
How deep is theory reasoning?

Mining proofs for statistics:

- record maximum derivation depth of a pure theory consequence used in the proof

### Experiment (relevant SMTLIB, default strategy, 60 s)

<table>
<thead>
<tr>
<th>Depth</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31,959</td>
</tr>
<tr>
<td>1</td>
<td>209</td>
</tr>
<tr>
<td>2</td>
<td>304</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
</tr>
<tr>
<td>5</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
</tr>
</tbody>
</table>
What do useful pure theory consequences look like?

Example (deep pure theory consequences)

\[ 0 < x \lor x < 4 \]

from UFLIA/sledgehammer/TwoSquares/z3.637729.smt2

\[ \neg((x + (y + ((-x) + 2.0))) < y) \quad \text{and} \quad \neg(2.0 + x < x) \]

from NRA/keymaera/ETCS-essentials-live-range2.proof-node1388.smt2

Note that: large constants must be obtained by combining the basic axioms, a clumsy search for a useful instance?
What do useful pure theory consequences look like?

Example (deep pure theory consequences)

\[ 0 < x \lor x < 4 \]

from UFLIA/sledgehammer/TwoSquares/z3.637729.smt2

\[ \neg ((x + (y + ((-x) + 2.0))) < y) \quad \text{and} \quad \neg (2.0 + x < x) \]

from NRA/keymaera/ETCS-essentials-live-range2.proof-node1388.smt2

Note that:

- large constants must be obtained by combining the basic axioms
What do useful pure theory consequences look like?

Example (deep pure theory consequences)

\[ 0 < x \lor x < 4 \]

from UFLIA/sledgehammer/TwoSquares/z3.637729.smt2

\[ \neg((x + (y + ((-x) + 2.0))) < y) \quad \text{and} \quad \neg(2.0 + x < x) \]

from NRA/keymaera/ETCS-essentials-live-range2.proof-node1388.smt2

Note that:

- large constants must be obtained by combining the basic axioms
- a clumsy search for a useful instance?
Explicitly limiting depth of pure theory consequences

<table>
<thead>
<tr>
<th>Depth</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32 522</td>
<td>32 253</td>
<td>32 130</td>
<td>32 061</td>
<td>32 162</td>
<td>32 040</td>
<td>31 959</td>
</tr>
<tr>
<td>1</td>
<td>552</td>
<td>237</td>
<td>209</td>
<td>216</td>
<td>208</td>
<td>209</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>551</td>
<td>314</td>
<td>310</td>
<td>307</td>
<td>304</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>312</td>
<td>254</td>
<td>212</td>
<td>200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>69</td>
<td>48</td>
<td>49</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>61</td>
<td>21</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>26</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>32 522</td>
<td>32 805</td>
<td>32 918</td>
<td>32 896</td>
<td>33 072</td>
<td>32 863</td>
<td>32 769</td>
</tr>
</tbody>
</table>
Some further observations

Let us denote the depth threshold $T$

- solved with $T = n$ can still be solvable with $T = m < n$
Let us denote the depth threshold $T$

- solved with $T = n$ can still be solvable with $T = m < n$
- decreasing $T$ can dramatically decrease the solution time and length of the found proof
Let us denote the depth threshold \( T \)

- solved with \( T = n \) can still be solvable with \( T = m < n \)
- decreasing \( T \) can dramatically decrease the solution time and length of the found proof
- decreasing \( T \) can also dramatically increase the solution time and length of the found proof
Some further observations

Let us denote the depth threshold $T$

- solved with $T = n$ can still be solvable with $T = m < n$
- decreasing $T$ can dramatically decrease the solution time and length of the found proof
- decreasing $T$ can also dramatically increase the solution time and length of the found proof

<table>
<thead>
<tr>
<th>Experiment (relevant SMTLIB, smtcomp mode, 1800 s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>competition mode</td>
</tr>
<tr>
<td>Solved</td>
</tr>
<tr>
<td>Uniques</td>
</tr>
</tbody>
</table>
Conclusion

Summary

- adapted SOS for dealing with theory axioms
- tuned by a derivation depth parameter
- promising initial experiments

Ideas and plans for future work:
- better understand relations to other theory reasoning techniques
- what are the useful (deep) theory consequences?
- could they be precomputed?
- distinguish “explosiveness” of axioms on case by case basis

Thank you for your attention!
Conclusion

Summary
- adapted SOS for dealing with theory axioms
- tuned by a derivation depth parameter
- promising initial experiments

Ideas and plans for future work:
- better understand relations to other theory reasoning techniques

Thank you for your attention!
Summary

- adapted SOS for dealing with theory axioms
- tuned by a derivation depth parameter
- promising initial experiments

Ideas and plans for future work:

- better understand relations to other theory reasoning techniques
- what are the useful (deep) theory consequences? could they be precomputed?
Conclusion

Summary

- adapted SOS for dealing with theory axioms
- tuned by a derivation depth parameter
- promising initial experiments

Ideas and plans for future work:

- better understand relations to other theory reasoning techniques
- what are the useful (deep) theory consequences? could they be precomputed?
- distinguish “explosiveness” of axioms on case by case basis

Thank you for your attention!
Conclusion

Summary

- adapted SOS for dealing with theory axioms
- tuned by a derivation depth parameter
- promising initial experiments

Ideas and plans for future work:

- better understand relations to other theory reasoning techniques
- what are the useful (deep) theory consequences? could they be precomputed?
- distinguish “explosiveness” of axioms on case by case basis

Thank you for your attention!