Proof Reconstruction for Z3 in Isabelle/HOL

Sascha Böhme
Technische Universität München

3. August 2009
User perspective:

▶ SMT as “black-box” technology
User perspective:
- SMT as “black-box” technology

Additional information:
- satisfiability: model

Our aim:
- certify proofs of Z3
  with Isabelle/HOL
User perspective:
- SMT as “black-box” technology

Additional information:
- satisfiability: model
- unsatisfiability: proof

Our aim:
- certify proofs of Z3 with Isabelle/HOL
User perspective:
- SMT as “black-box” technology

Additional information:
- satisfiability: model
- unsatisfiability: proof

Increased confidence:
- checkable certificates
User perspective:
- SMT as “black-box” technology

Additional information:
- satisfiability: model
- unsatisfiability: proof

Increased confidence:
- checkable certificates

Our aim:
- certify proofs of Z3
- with Isabelle/HOL
A Quick Glance at Isabelle/HOL

LCF

Theorems

LCF kernel:
- abstract type: theorems
- operations: basic inference rules
- small
A Quick Glance at Isabelle/HOL

Higher-order logic (HOL)

LCF

LCF kernel:

- abstract type: theorems
- operations: basic inference rules
- small
A Quick Glance at Isabelle/HOL

Higher-order logic (HOL)

Proof tools:
- term rewriting (simplifier)
- tableaux prover (blast)
- decision procedures: linear arithmetic, quantifier elimination

LCF

LCF kernel:
- abstract type: theorems
- operations: basic inference rules
- small

Theorems
Z3 Terms

Language: many-sorted first-order logic

Terms: $t, s$

- variables: $x, y$
- applications: $f \ t_1 \ldots t_n$
  - logical connectives: $true, false, \neg, \land, \lor, \rightarrow, \leftrightarrow, \sim$
- quantifiers: $\forall, \exists$
- terms of sort $bool$ (formulas): $P$

Equisatisfiability:

$\neg x \lor false \equiv \exists x. \neg x \lor false \leftrightarrow \exists y. \neg y$

Natural mapping into higher-order logics (Isabelle/HOL)

- equisatisfiability: representable as equivalence with one exception: Skolemization
Z3 Terms

Language: many-sorted first-order logic

Terms: $t, s$

- variables: $x, y$
- applications: $f \ t_1 \ldots t_n$
  - logical connectives: $true, false, \neg, \land, \lor, \rightarrow, \leftrightarrow, \sim$
- quantifiers: $\forall, \exists$
- terms of sort $bool$ (formulas): $P$

Equisatisfiability:

$$(\neg x \lor false) \sim (\neg y) \equiv (\exists x. \neg x \lor false) \leftrightarrow (\exists y. \neg y)$$
Z3 Terms

Language: many-sorted first-order logic

Terms: $t, s$

- variables: $x, y$
- applications: $f \ t_1 \ldots t_n$
  - logical connectives: $true, false, \neg, \land, \lor, \rightarrow, \leftrightarrow, \sim$
- quantifiers: $\forall, \exists$
- terms of sort $bool$ (formulas): $P$

Equisatisfiability:

$$\sim(x \lor false) \equiv (\exists x. \neg x \lor false) \leftrightarrow (\exists y. \neg y)$$

Natural mapping into higher-order logics (Isabelle/HOL)

- equisatisfiability: representable as equivalence with one exception: Skolemization
Natural deduction style:

\[
\begin{array}{c}
\Gamma_1 \vdash P_1 \\
\Gamma_2 \vdash P_1 \leftrightarrow P_2
\end{array}
\]

\[
\Gamma_1 \cup \Gamma_2 \vdash P_2 \quad \text{mp} \leftrightarrow
\]
Natural deduction style:

\[
\Gamma_1 \vdash P_1 \quad \Gamma_2 \vdash P_1 \leftrightarrow P_2 \\
\frac{}{\Gamma_1 \cup \Gamma_2 \vdash P_2 \quad \text{mp}}
\]

Proof trees:

- \(\neg true \vdash \neg true\) asserted
- \(\vdash \neg true \leftrightarrow false\) rewrite
- \(\neg true \vdash false\) mp
Z3 Proofs

Natural deduction style:

\[
\Gamma_1 \vdash P_1 \quad \Gamma_2 \vdash P_1 \leftrightarrow P_2 \quad \text{mp} \leftrightarrow \\
\Gamma_1 \cup \Gamma_2 \vdash P_2
\]

Proof trees:

\[
\begin{array}{c}
\neg \text{true} \vdash \neg \text{true} \\
\text{asserted}
\end{array}
\quad
\begin{array}{c}
\vdash \neg \text{true} \leftrightarrow \text{false} \\
\text{rewrite}
\end{array}
\quad
\begin{array}{c}
\neg \text{true} \vdash \text{false} \\
\text{mp} \leftrightarrow
\end{array}
\]

28 proof rules:

- core logic: asserted, unit, ...
- equality: refl, trans, ...
- quantifiers: quant-inst, elim-unused, ...
- theories: rewrite, th-lemma
Proof Reconstruction

\[
\begin{align*}
\neg \text{true} \vdash \neg \text{true} & \quad \text{asserted} \\
\vdash \neg \text{true} \leftrightarrow \text{false} & \quad \text{rewrite} \\
\neg \text{true} \vdash \text{false} & \quad \text{mp} \leftrightarrow
\end{align*}
\]
Proof Reconstruction

- \( \neg true \vdash \neg true \) asserted
- \( \vdash \neg true \leftrightarrow false \) rewrite
- \( \neg true \vdash false \) mp

- bottom-up
- one method for every rule
Proof Reconstruction

\[ \neg \text{true} \vdash \neg \text{true} \]

\[ \vdash \neg \text{true} \iff \text{false} \]

\[ \neg \text{true} \vdash \text{false} \]

- bottom-up
- one method for every rule
Proof Reconstruction

\[
\neg \text{true} \vdash \neg \text{true} \\
\vdash \neg \text{true} \leftrightarrow \text{false} \\
\neg \text{true} \vdash \text{false}
\]

- bottom-up
- one method for every rule
Proof Reconstruction

\[ \neg \text{true} \vdash \neg \text{true} \]

\[ \vdash \neg \text{true} \leftrightarrow \text{false} \]

\[ \neg \text{true} \vdash \text{false} \]

- bottom-up
- one method for every rule
- all inferences certified by LCF kernel
Proof Reconstruction

(asserted) \( \neg \text{true} \vdash \neg \text{true} \) rewrite

\( \vdash \neg \text{true} \leftrightarrow \text{false} \)

\( \neg \text{true} \vdash \text{false} \) (mp)

- bottom-up
- one method for every rule
- all inferences certified by LCF kernel
- additional checks
Proof Reconstruction

\[
\neg\text{true} \vdash \neg\text{true} \quad \text{asserted} \quad \neg\text{true} \leftrightarrow \text{false} \quad \text{rewrite} \quad \frac{\neg\text{true} \vdash \text{false}}{\neg\text{true} \vdash \text{false}} \quad \text{mp} \\
\]

- bottom-up
- one method for every rule
- all inferences certified by LCF kernel
- additional checks (for debugging)
Reconstruction Methods
Reconstruction Methods

- basic inference rules of Isabelle (2 rules)
Reconstruction Methods

- basic inference rules of Isabelle  
  (2 rules)

- Isabelle theorem and resolution  
  (8 rules)

\[ \Gamma_1 \vdash P_1 \quad \Gamma_2 \vdash P_1 \leftrightarrow P_2 \]
\[ \Gamma_1 \cup \Gamma_2 \vdash P_2 \]

\[ P_1 \Rightarrow P_1 \leftrightarrow P_2 \Rightarrow P_2 \]
Reconstruction Methods

- basic inference rules of Isabelle (2 rules)
- Isabelle theorem and resolution (8 rules)

\[ \begin{align*}
\Gamma_1 & \vdash P_1 & \Gamma_2 & \vdash P_1 \leftrightarrow P_2 \\
\Gamma_1 \cup \Gamma_2 & \vdash \text{mp} \\
\end{align*} \]

- Isabelle proof tools (simplifier, blast) (9 rules)

\[ P_1 \implies P_1 \leftrightarrow P_2 \implies P_2 \]
Reconstruction Methods

- basic inference rules of Isabelle (2 rules)
- Isabelle theorem and resolution (8 rules)
  \[ \Gamma_1 \vdash P_1 \quad \Gamma_2 \vdash P_1 \leftrightarrow P_2 \quad \text{mp} \leftrightarrow \]
  \[ \Gamma_1 \cup \Gamma_2 \vdash P_2 \]
  \[ P_1 \implies P_1 \leftrightarrow P_2 \implies P_2 \]
- Isabelle proof tools (simplifier, blast) (9 rules)
- specialized treatment (9 rules)
  - in some cases: optimizations
Congruence

\[ \Gamma_1 \vdash t_1 = s_1 \quad \ldots \quad \Gamma_n \vdash t_n = s_n \quad \bigcup_{i \leq n} \Gamma_i \vdash f \ t_1 \ldots t_n = f \ s_1 \ldots s_n \quad \text{mono} \]

In principle: provable by simplifier (term rewriting)

But: one of the central rules!
  ▶ optimization is worthwhile

Thus: combination of
  ▶ congruence: \( f = g \implies x = y \implies f \ x = g \ y \)
  ▶ reflexivity: \( t = t \)
Skolemization

Example:

\[ \vdash (\exists x. P \times y) \sim P (f \ y) \ y \]
Skolemization

Example:

⊢ (\exists x. P \times y) \sim P (f \times y) \times y

With Hilbert choice operator \( \varepsilon \):

\[ f = (\lambda y. \varepsilon x. P \times y) \vdash (\exists x. P \times y) \leftrightarrow P (f \times y) \times y \]
Skolemization

Example:

\[ \vdash (\exists x. P \times y) \sim P (f \times y) y \]

With Hilbert choice operator \( \varepsilon \):

\[ f = (\lambda y. \varepsilon x. P \times y) \vdash (\exists x. P \times y) \leftrightarrow P (f \times y) y \]

At the end of reconstruction:

\[ \Gamma, f = (\lambda y. \varepsilon x. P \times y) \vdash false \]
Skolemization

Example:

\[ \vdash (\exists x. P \times y) \sim P (f \cdot y) \cdot y \]

With Hilbert choice operator \( \varepsilon \):

\[ f = (\lambda y. \varepsilon x. P \times y) \vdash (\exists x. P \times y) \leftrightarrow P (f \cdot y) \cdot y \]

At the end of reconstruction:

\[ \Gamma, f = (\lambda y. \varepsilon x. P \times y) \vdash false \]

\[ \Gamma \vdash f = (\lambda y. \varepsilon x. P \times y) \rightarrow false \]
Skolemization

Example:

$$\vdash (\exists x. P \times y) \sim P (f \times y) \times y$$

With Hilbert choice operator $\varepsilon$:

$$f = (\lambda y. \varepsilon x. P \times y) \vdash (\exists x. P \times y) \leftrightarrow P (f \times y) \times y$$

At the end of reconstruction:

$$\Gamma, f = (\lambda y. \varepsilon x. P \times y) \vdash \text{false}$$

$$\Gamma \vdash f = (\lambda y. \varepsilon x. P \times y) \rightarrow \text{false}$$

$$\Gamma \vdash (\lambda y. \varepsilon x. P \times y) = (\lambda y. \varepsilon x. P \times y) \rightarrow \text{false}$$
Skolemization

Example:

\[ \vdash (\exists x. P \times y) \sim P (f \times y) y \]

With Hilbert choice operator \( \varepsilon \):

\[ f = (\lambda y. \varepsilon x. P \times y) \vdash (\exists x. P \times y) \leftrightarrow P (f \times y) y \]

At the end of reconstruction:

\[ \Gamma, f = (\lambda y. \varepsilon x. P \times y) \vdash false \]

\[ \Gamma \vdash f = (\lambda y. \varepsilon x. P \times y) \rightarrow false \]

\[ \Gamma \vdash (\lambda y. \varepsilon x. P \times y) = (\lambda y. \varepsilon x. P \times y) \rightarrow false \]

\[ \Gamma \vdash false \]
Theories

Rewriting (**rewrite**):

\[ \vdash f \ t_1 \ldots t_n = s \]

- in general: apply rules of \( f \)
- simplifier, linear arithmetic, specialized procedures
Theories

Rewriting (\texttt{rewrite}):\[
\Gamma \vdash f \ t_1 \ldots t_n = s
\]

\begin{itemize}
  \item in general: apply rules of \( f \)
  \item simplifier, linear arithmetic, specialized procedures
\end{itemize}

Theory reasoning (\texttt{th-lemma}):\[
\Gamma_1 \vdash P_1 \quad \ldots \quad \Gamma_n \vdash P_n
\]
\[
\Gamma_1 \vdash P_1 \ldots \Gamma_n \vdash P_n
\]
\[
\vdash \bigvee_{i \in I} P_i
\]

\begin{itemize}
  \item linear arithmetics: Fourier-Motzkin elimination
  \item arrays: simplifier
\end{itemize}
Experimental Results

- 5 SMT-LIB logics
- 100 unsatisfiably benchmarks (randomly selected)
- timeout: Z3: 2 minutes, Isabelle/HOL: 10 minutes

<table>
<thead>
<tr>
<th>Logic</th>
<th>Solved by Z3</th>
<th>Success</th>
<th>Failure</th>
<th>Timeout</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>QF_UF</td>
<td>96</td>
<td>33</td>
<td>27</td>
<td>36</td>
<td>6.5</td>
</tr>
<tr>
<td>QF_UFLIA</td>
<td>99</td>
<td>93</td>
<td>0</td>
<td>6</td>
<td>29.6</td>
</tr>
<tr>
<td>QF_UFLRA</td>
<td>100</td>
<td>43</td>
<td>0</td>
<td>57</td>
<td>558.3</td>
</tr>
<tr>
<td>AUFLIA</td>
<td>100</td>
<td>50</td>
<td>31</td>
<td>19</td>
<td>81.3</td>
</tr>
<tr>
<td>AUFLIRA</td>
<td>100</td>
<td>81</td>
<td>6</td>
<td>13</td>
<td>24.3</td>
</tr>
</tbody>
</table>

Bad performance:
- only few optimizations implemented
- huge formulas of benchmarks
Proof Reconstruction Failures

- Incomplete documentation of rewrite:
  
  *The head function symbol of the left-hand side is interpreted.*
Proof Reconstruction Failures

- Incomplete documentation of \texttt{rewrite}:

  *The head function symbol of the left-hand side is interpreted.*

  But:

  \[
  \vdash P \land (\forall x : \text{int}. x > 0) \leftrightarrow \text{false} \land P \quad \text{\texttt{rewrite}}
  \]
Proof Reconstruction Failures

- Incomplete documentation of **rewrite**:

  *The head function symbol of the left-hand side is interpreted.*

  But:

  \[
  \vdash P \land (\forall x : \text{int}. \ x > 0) \leftrightarrow \text{false} \land P
  \]

  \[
  \vdash (P_1 \land P_2) \leftrightarrow \neg(\neg P_1 \lor \neg P_2)
  \]
Proof Reconstruction Failures

- Incomplete documentation of **rewrite**:  
  *The head function symbol of the left-hand side is interpreted.*

  But:
  \[
  \vdash P \land (\forall x : \text{int. } x > 0) \iff \text{false} \land P \quad \text{rewrite}
  \]

  \[
  \vdash (P_1 \land P_2) \iff \neg(\neg P_1 \lor \neg P_2) \quad \text{rewrite}
  \]

- **Unit resolution:**

  \[
  \Gamma_1 \vdash P_1 \lor P_2 \lor P_1 \quad \Gamma_2 \vdash \neg P_2
  \]

  \[
  \Gamma_1 \cup \Gamma_2 \vdash P_1 \quad \text{unit}
  \]
Proof Reconstruction Failures

- Incomplete documentation of **rewrite**:
  
  The head function symbol of the left-hand side is interpreted.
  
  But:
  
  \[
  \vdash P \land (\forall x : \text{int}. x > 0) \leftrightarrow false \land P \quad \text{rewrite}
  \]
  
  \[
  \vdash (P_1 \land P_2) \leftrightarrow \neg(\neg P_1 \lor \neg P_2) \quad \text{rewrite}
  \]

- Unit resolution:
  
  \[
  \Gamma_1 \vdash P_1 \lor P_2 \lor P_1 \quad \Gamma_2 \vdash \neg P_2 \quad \text{unit}
  \]
  
  \[
  \Gamma_1 \cup \Gamma_2 \vdash P_1
  \]

- Transitivity:
  
  \[
  \Gamma_1 \vdash s = t \quad \Gamma_2 \vdash u = t \quad \text{trans}
  \]
  
  \[
  \Gamma_1 \cup \Gamma_2 \vdash s = u
  \]
Conclusion

Proof reconstruction for Z3:
▶ in Isabelle/HOL: certification by LCF kernel
▶ challenges: equisatisfiability, huge formulas
▶ helped to debug Z3 proof generation

Future work:
▶ improve performance
▶ integrate into Isabelle/HOL
▶ consider further theories