

Reconstructing Z3 Proofs With KeY

Wolfram Pfeifer

07.05.2020

Karlsruhe Institute of Technology (KIT)
Institute of Theoretical Informatics

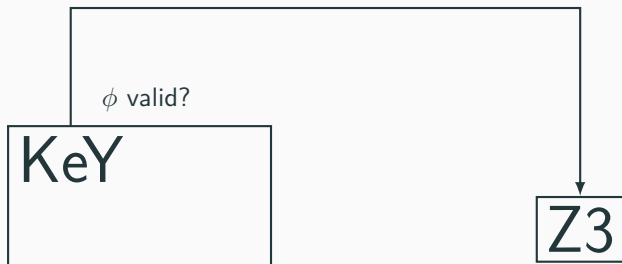
Motivation

Motivation: Current KeY Workflow with SMT Solvers

ϕ valid?

KeY

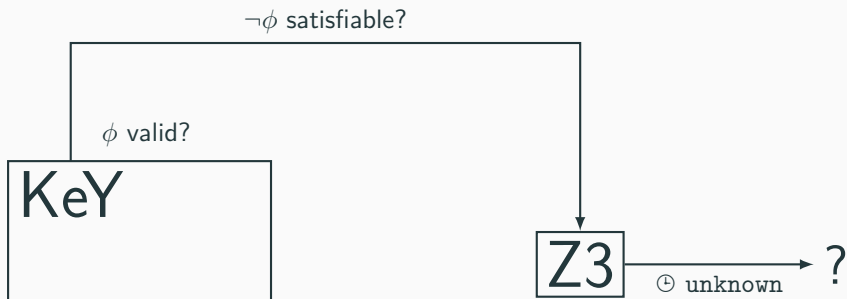
Motivation: Current KeY Workflow with SMT Solvers



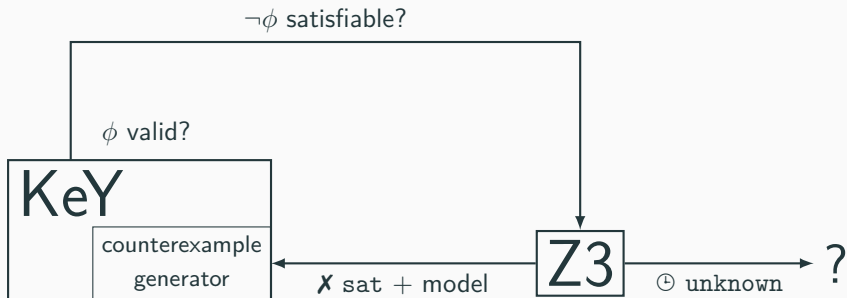
Motivation: Current KeY Workflow with SMT Solvers



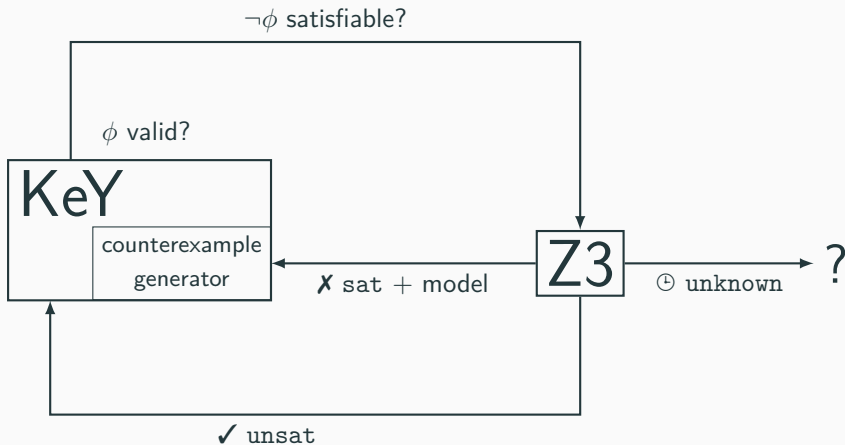
Motivation: Current KeY Workflow with SMT Solvers



Motivation: Current KeY Workflow with SMT Solvers



Motivation: Current KeY Workflow with SMT Solvers



Current Behavior

The screenshot shows the KeY 2.7 software interface. The title bar reads "KeY 2.7". The menu bar includes "File", "View", "Proof", "Options", "Interaction Logging", "Origin Tracking", and "About". The toolbar contains various icons for navigation and proof management. The main window is divided into several panes:

- Loaded Proofs:** Shows "Proofs" with "Env. with no model" and "project key".
- Goals:** Includes "Interaction Log", "Proof Search Strategy", "Proof", and "Info".
- Proof Tree:** Displays a tree structure with a root node labeled "0: OPEN GOAL".
- Sequent:** The main area for the current goal, containing the following logical expression:

```
==>  
  
  \exists S z7: (lives(z7) & killed(z7, agatha))  
  & lives(agatha)  
  & lives(butler)  
  & lives(charles)  
  
  & \forallall S z8: (lives(z8) -> z8 = agatha | (z8 = butler | z8 = charles))  
  & \forallall S z9; \forallall S z0: (killed(z9, z0) -> hates(z9, z0))  
  & \forallall S w1; \forallall S w2: (killed(w1, w2) -> !richer(w1, w2))  
  & \forallall S w3: (hates(agatha, w3) -> !hates(charles, w3))  
  & \forallall S w4: (!w4 = butler -> hates(agatha, w4))  
  & \forallall S w5: (!richer(w5, agatha) -> hates(butler, w5))  
  & \forallall S w6: (hates(agatha, w6) -> hates(butler, w6))  
  & \forallall S w7; \exists S w8; !hates(w7, w8)  
  & !agatha = butler  
  
-> killed(agatha, agatha)
```

At the bottom left, there is a "Loading proof" status bar. At the bottom right, the page number "3/25" is displayed.

Current Behavior

The screenshot displays the KeY 2.7 software interface. The main window shows the SMT solver's output, which includes a list of logical goals and their status. A modal dialog box titled "SMT Interface" is overlaid on the main window, indicating that the goal has been successfully solved.

KeY 2.7 Interface Elements:

- Menu Bar:** File, View, Proof, Options, Interaction Logging, Origin Tracking, About
- Toolbar:** Run Z3_NEW_TL, Undo, Redo, Copy, Paste, Find, etc.
- Left Panel:**
 - Loaded Proofs:** Shows a proof named "project.key" with the environment "Env. with no model".
 - Goals:** Includes "Interaction Log", "Proof Search Strategy", "Proof", and "Info".
 - Proof Tree:** Shows a tree structure with a root node labeled "0: OPEN GOAL".
- Main Window:** Displays the SMT solver's output, including the current goal and its solution.

SMT Solver Output (Current Goal):

```
==>
\exi
& live
& live
& live
& \for
& \for
& \for
& \for
& \for
& \for
& \for
& \for
& !agatha = butler
-> killed(agatha, agatha)
```

SMT Interface Dialog:

Finished.

Z3_NEW_TL	
Goal 0	Valid (0.100s)

Buttons: Discard, Apply

Additional Output (Right Side):

```
ha))
(z8 = butler | z8 = charles))
) -> hates(z9, z0))
) -> !richer(w1, w2))
(charles, w3))
a, w4))
s(butler, w5))
butler, w6))
```

Current Behavior

The screenshot displays the KeY 2.7 software interface. The main window has a menu bar (File, View, Proof, Options, Interaction Logging, Origin Tracking) and a toolbar. The 'Loaded Proofs' panel on the left shows a proof named 'project.key' with a green checkmark. The 'Proof Tree' panel below it shows a tree structure with a green checkmark and the text '0: SMT'. The 'Proof Statistics' dialog box is open in the center, displaying the following information:

Proved.

Nodes	1
Branches	1
Interactive steps	0
Symbolic execution steps	0
Automode time	0ms
Avg. time per step	0.0ms

Rule applications

Quantifier instantiations	0
One-step Simplifier apps	0
SMT solver apps	1
Dependency Contract apps	0
Operation Contract apps	0
Block/Loop Contract apps	0
Loop invariant adds	0

Buttons at the bottom of the dialog: Close, Export as CSV, Export as HTML.

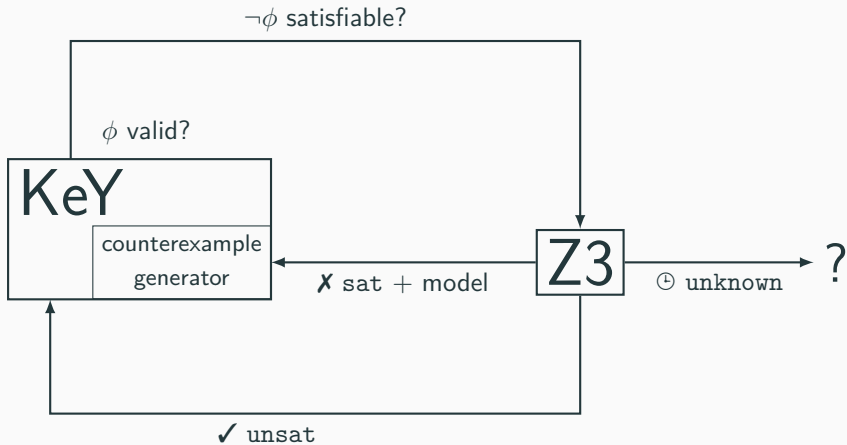
The background interface shows a 'Sequent' window with the following text:

```
Inner Node
==>
  \exis
  & lives
  & lives
  & lives
  & \fora
  & \fora
  & \fora
  & \fora
  & \fora
  & \fora
  & \fora
  & \fora
  & !agat
  -> killed(
```

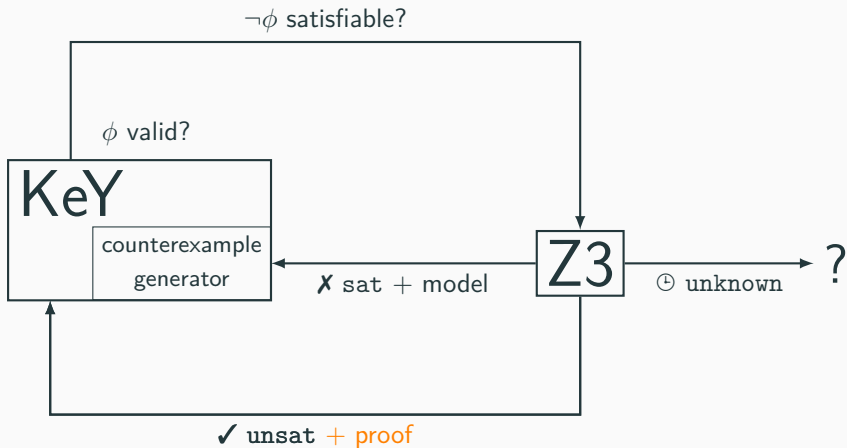
On the right side of the interface, there is a large text area containing the following code:

```
tha))
(z8 = butler | z8 = charles))
0) -> hates(z9, z0))
2) -> !richer(w1, w2))
s(charles, w3))
ha, w4))
es(butler, w5))
(butler, w6))
)
```

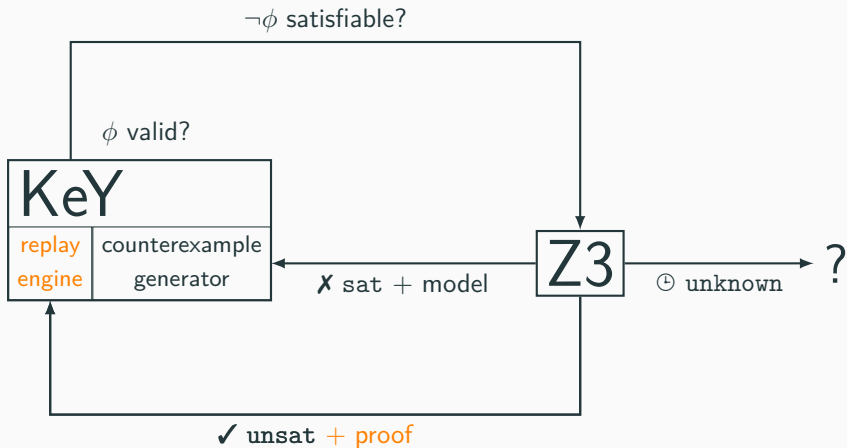
Replay Idea



Replay Idea



Replay Idea



Z3 Proofs

Input (SMT-LIB):

```
1 (set-option :produce-proofs true)
2
3 (declare-const p Bool)
4
5 (assert (not (or p true)))           ;  $\neg(p \vee \text{true})$ 
6
7 (check-sat)
8 (get-proof)
```

Z3 Proofs

Input (SMT-LIB):

```
1 (set-option :produce-proofs true)
2
3 (declare-const p Bool)
4
5 (assert (not (or p true)))           ;    $\neg(p \vee \text{true})$ 
6
7 (check-sat)
8 (get-proof)
```

Output (Z3-specific):

```
1 unsat
2 (mp (not-or-elim
3     (asserted (not (or p true)))
4     (not true))
5     (rewrite (= (not true) false))
6     false)
```


Z3 Proofs

$$\text{not-or-elim} \frac{\text{asserted} \frac{\Gamma \vdash \neg(p \vee \text{true})}{\Gamma \vdash \neg \text{true}}}{\Gamma \vdash \text{false}} \quad \frac{\Gamma \vdash \neg \text{true} \quad \vdash \neg \text{true} = \text{false}}{\Gamma \vdash \text{false}} \begin{array}{l} \text{rewrite} \\ \text{mp} \end{array}$$

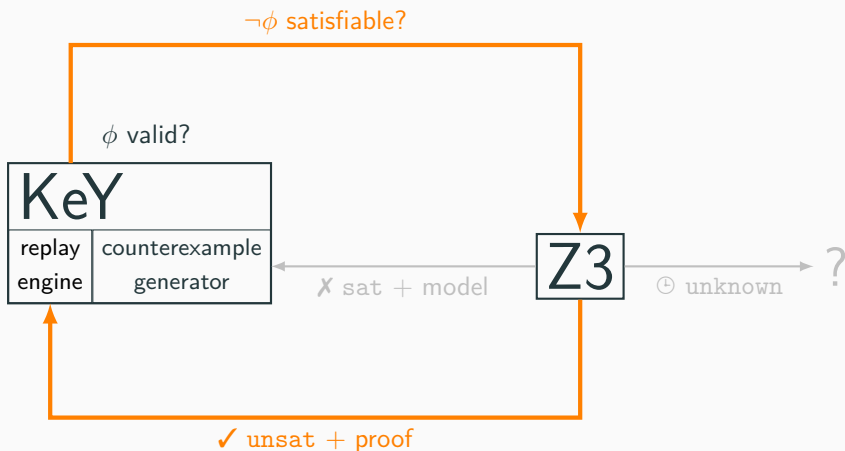
$$\Gamma = \{\neg(p \vee \text{true})\}$$

Output (Z3-specific):

```
1 unsat
2 (mp (not-or-elim
3     (asserted (not (or p true)))
4     (not true))
5     (rewrite (= (not true) false))
6     false)
```

The Replay Technique

Replay Technique: Round Trip



Problem: Different Proof Directions

$$\phi = p \vee \text{true}$$

Key (ϕ is valid):

$$\frac{\frac{\text{---}}{\Rightarrow p, \text{true}} \text{closeTrue}}{\Rightarrow p \vee \text{true}} \text{orRight}$$

Problem: Different Proof Directions

$$\phi = p \vee \text{true}$$

Key (ϕ is valid):

$$\frac{\frac{}{\Rightarrow p, \text{true}} \text{closeTrue}}{\Rightarrow p \vee \text{true}} \text{orRight}$$

↑ proof construction

Problem: Different Proof Directions

$$\phi = p \vee \text{true}$$

KeY (ϕ is valid):

$$\frac{\frac{\text{---}}{\Rightarrow p, \text{true}} \text{closeTrue}}{\Rightarrow p \vee \text{true}} \text{orRight}$$

↑ proof construction

Z3 ($\neg\phi$ is unsat):

$$\text{not-or-elim} \frac{\text{asserted} \frac{\text{---}}{\Gamma \vdash \neg(p \vee \text{true})}}{\Gamma \vdash \neg \text{true}} \quad \frac{\text{---}}{\vdash \neg \text{true} = \text{false}} \text{rewrite}}{\Gamma \vdash \text{false}} \text{mp}$$

where $\Gamma = \{\neg\phi\} = \{\neg(p \vee \text{true})\}$.

Problem: Different Proof Directions

$$\phi = p \vee \text{true}$$

KeY (ϕ is valid):

$$\frac{\frac{\text{---}}{\Rightarrow p, \text{true}} \text{closeTrue}}{\Rightarrow p \vee \text{true}} \text{orRight}$$

↑ proof construction

Z3 ($\neg\phi$ is unsat):

$$\text{not-or-elim} \frac{\text{asserted} \frac{\text{---}}{\Gamma \vdash \neg(p \vee \text{true})}}{\Gamma \vdash \neg \text{true}} \quad \frac{\text{---}}{\vdash \neg \text{true} = \text{false}} \text{rewrite}}{\Gamma \vdash \text{false}} \text{mp}$$

↓ proof construction

where $\Gamma = \{\neg\phi\} = \{\neg(p \vee \text{true})\}$.

Problem: Different Proof Directions

$$\phi = p \vee \text{true}$$

KeY (ϕ is valid):

$$\frac{\frac{\text{?}}{\neg(p \vee \text{true}) \implies \text{false}} \text{?}}{\implies p \vee \text{true}}$$

↑ proof construction

Z3 ($\neg\phi$ is unsat):

$$\text{not-or-elim} \frac{\text{asserted} \frac{\Gamma \vdash \neg(p \vee \text{true})}{\Gamma \vdash \neg \text{true}} \quad \frac{\Gamma \vdash \neg \text{true} \quad \vdash \neg \text{true} = \text{false}}{\Gamma \vdash \text{false}} \text{rewrite}}{\Gamma \vdash \text{false}} \text{mp}$$

↓ proof construction

where $\Gamma = \{\neg\phi\} = \{\neg(p \vee \text{true})\}$.

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\frac{\boxed{\vdash p} \quad \boxed{\vdash p \rightarrow q}}{\boxed{\vdash q}} \text{mp}$$

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\frac{\boxed{\vdash p} \quad \boxed{\vdash p \rightarrow q}}{\boxed{\vdash q}} \text{mp}$$

Replay in KeY:

$$\frac{\frac{\boxed{L}}{p, p \rightarrow q \Rightarrow q} \text{andLeft} \quad \frac{\boxed{\Rightarrow p} \quad \boxed{\Rightarrow p \rightarrow q}}{\Rightarrow p \wedge (p \rightarrow q)} \text{andRight}}{\frac{\Rightarrow p \wedge (p \rightarrow q), q}{\Rightarrow p \wedge (p \rightarrow q), q} \text{hideRight}} \text{cut} \\ \boxed{\Rightarrow q}$$

The single rule application is justified in L .

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\frac{\boxed{\vdash p} \quad \boxed{\vdash p \rightarrow q}}{\boxed{\vdash q}} \text{mp}$$

Replay in KeY:

$$\frac{\frac{\frac{\frac{\text{close}}{p, q \Rightarrow q}}{p, true \rightarrow q \Rightarrow q} \text{simplify}}{p, p \rightarrow q \Rightarrow q} \text{andLeft}}{p \wedge (p \rightarrow q) \Rightarrow q} \text{andRight}}{\boxed{\Rightarrow q}} \text{cut}$$

Labels in the diagram: *close*, *simplify*, *andLeft*, *andRight*, *hideRight*, *cut*, *repl_known_left*.

The single rule application is justified in *L*.

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\frac{\boxed{\vdash p} \quad \boxed{\vdash p \rightarrow q}}{\boxed{\vdash q}} \text{mp}$$

Replay in KeY:

$$\frac{\frac{\frac{\frac{\text{close}}{p, q \Rightarrow q}}{p, true \rightarrow q \Rightarrow q} \text{simplify}}{p, p \rightarrow q \Rightarrow q} \text{andLeft}}{p \wedge (p \rightarrow q) \Rightarrow q} \text{andRight}}{\frac{\frac{\Rightarrow p}{} \text{andRight} \quad \frac{\Rightarrow p \rightarrow q}{} \text{andRight}}{\Rightarrow p \wedge (p \rightarrow q)} \text{hideRight}}{\Rightarrow p \wedge (p \rightarrow q), q} \text{cut}} \text{repl_known_left}}{\Rightarrow q}$$

The single rule application is justified in L .

Assumption: Z3 terms can be translated one-to-one to KeY terms.

Replayed Proof Example

$\Gamma = \{\neg(p \vee \text{true})\}$

Z3 proof:

$$\frac{\frac{\text{asserted} \quad \Gamma \vdash \neg(p \vee \text{true})}{\Gamma \vdash \neg \text{true}} \quad \frac{}{\vdash \neg \text{true} = \text{false}}}{\Gamma \vdash \text{false}} \begin{array}{l} \text{not-or-elim} \\ \text{rewrite} \\ \text{mp} \end{array}$$

Replayed Proof Example

$$\Gamma = \{\neg(p \vee \text{true})\}$$

Z3 proof:

$$\frac{\text{asserted} \frac{\Gamma \vdash \neg(p \vee \text{true})}{\Gamma \vdash \neg \text{true}} \quad \text{rewrite} \frac{}{\vdash \neg \text{true} = \text{false}}}{\Gamma \vdash \text{false}} \text{mp}$$

Replayed in KeY (= is translated to \leftrightarrow):

$$\frac{\text{concrete_eq} \frac{\text{repl_known_left} \frac{\text{andLeft} \frac{\text{close} \frac{\Gamma, \neg \text{true}, \text{false} \Rightarrow \text{false}}{\Gamma, \neg \text{true}, \text{true} \leftrightarrow \text{false} \Rightarrow \text{false}}}{\Gamma, \neg \text{true}, \neg \text{true} \leftrightarrow \text{false} \Rightarrow \text{false}}}{\Gamma, \neg \text{true} \wedge (\neg \text{true} \leftrightarrow \text{false}) \Rightarrow \text{false}} \quad \text{notLeft} \frac{\text{orRight} \frac{\text{notRight} \frac{\text{close} \frac{\Gamma, \text{true} \Rightarrow p, \text{true}}{\Gamma \Rightarrow p, \text{true}, \neg \text{true}}}{\Gamma \Rightarrow p \vee \text{true}, \neg \text{true}}}{\Gamma, \neg(p \vee \text{true}) \Rightarrow \neg \text{true}} \quad \frac{\Gamma \Rightarrow \neg(p \vee \text{true})}{\Gamma \Rightarrow \neg(p \vee \text{true}), \neg \text{true}} \text{close}}{\Gamma \Rightarrow \neg \text{true}} \text{hideRight} \quad \vdots}{\Gamma \Rightarrow \neg \text{true} \leftrightarrow \text{false}} \text{cut} \quad \text{auto mode}}{\Gamma \Rightarrow \neg \text{true} \wedge (\neg \text{true} \leftrightarrow \text{false})} \text{hideRight}}{\Gamma \Rightarrow \neg \text{true} \wedge (\neg \text{true} \leftrightarrow \text{false}), \text{false}} \text{cut}}{\Gamma \Rightarrow \text{false}} \text{notRight}}{\Rightarrow p \vee \text{true}}$$

Translating Z3 Terms to KeY

Assumption: Z3 terms can be translated one-to-one to KeY terms.

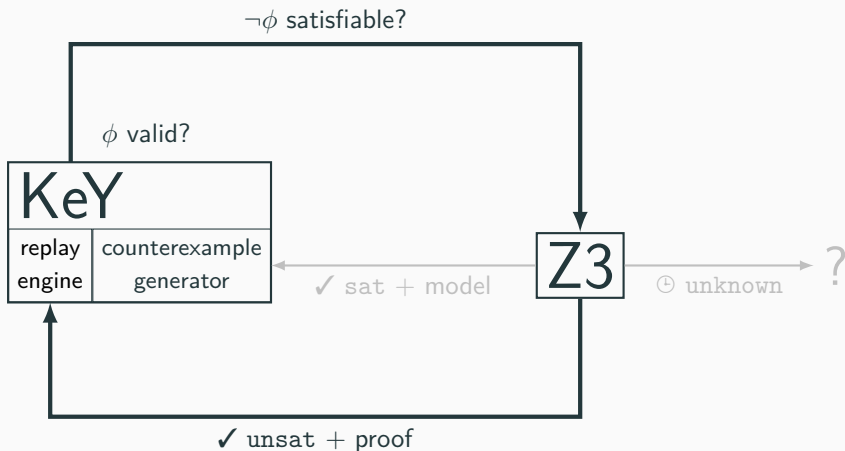
Translating Z3 Terms to KeY

Assumption: Z3 terms can be translated one-to-one to KeY terms.

Currently violated:

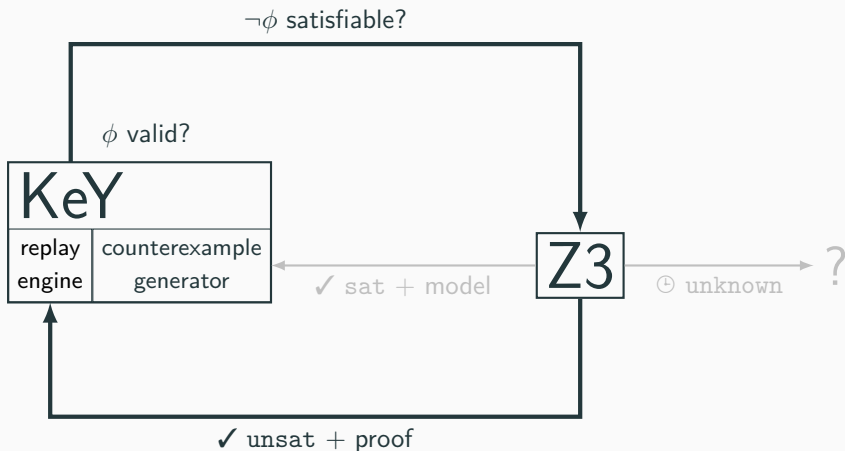
- (A) The type hierarchy axiomatisation introduces ***typeof*** and ***subtype***.
- (B) Z3 terms may contain an explicit ***equisatisfiability*** relation.
- (C) Z3 may introduce ***skolem functions***.

Replay Technique: Overview



Replay Technique: Overview

(A) type hierarchy axiomatization

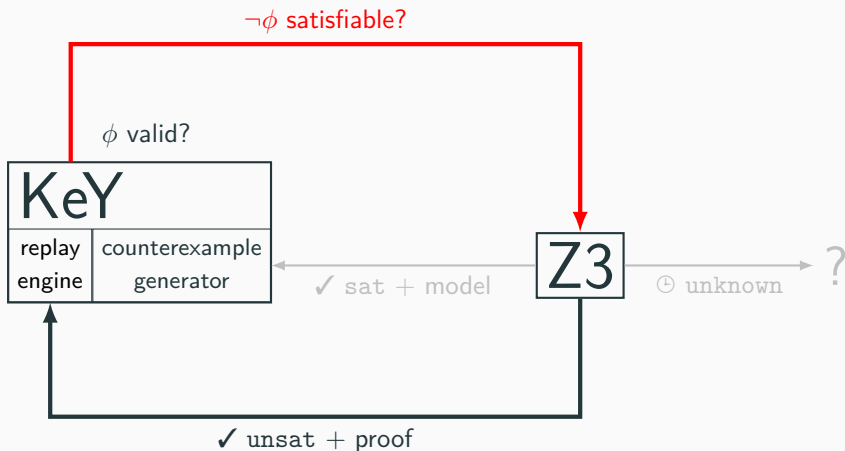


(B) equisatisfiability relation

(C) skolem functions

Replay Technique: Overview

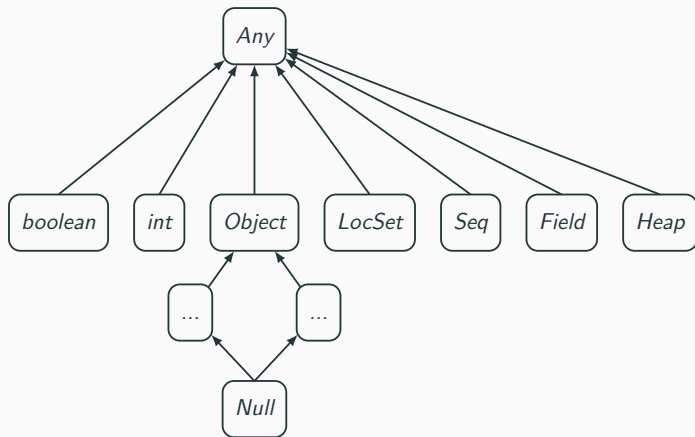
(A) type hierarchy axiomatization



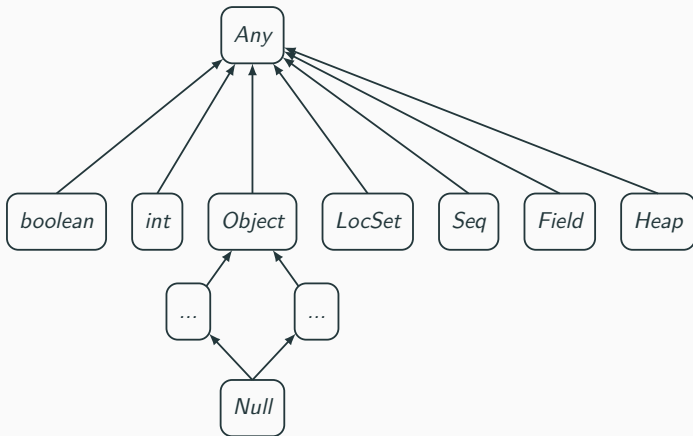
(B) equisatisfiability relation

(C) skolem functions

KeY Type Hierarchy

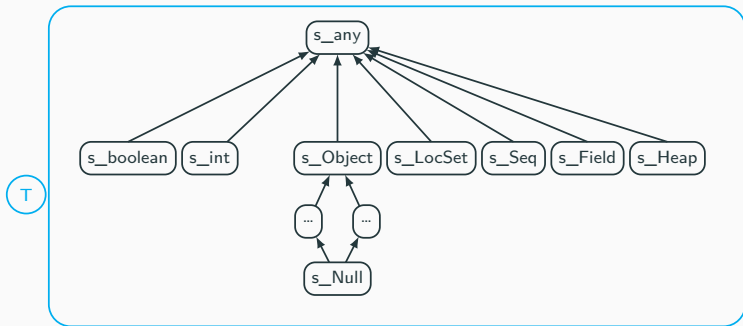


KeY Type Hierarchy



In contrast to that, Z3 has no inheritance.

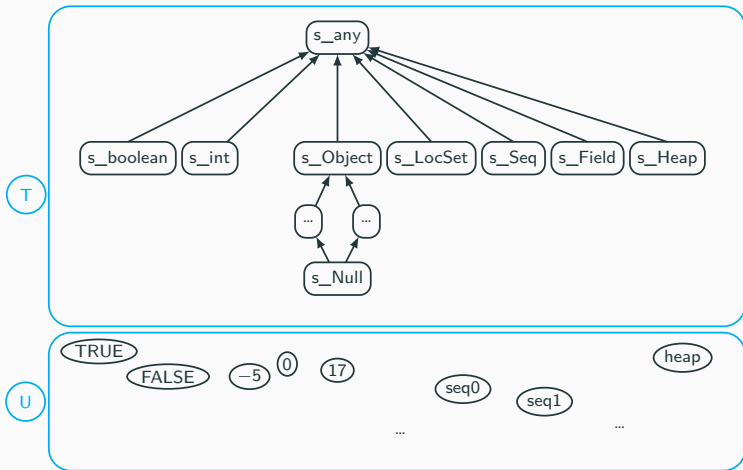
Existing Type Hierarchy Translation



— real Z3 sorts and their domains

← subtype relation (partial order)

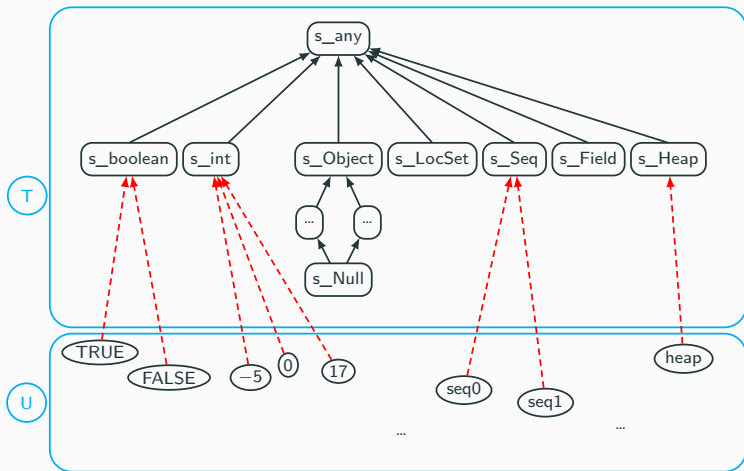
Existing Type Hierarchy Translation



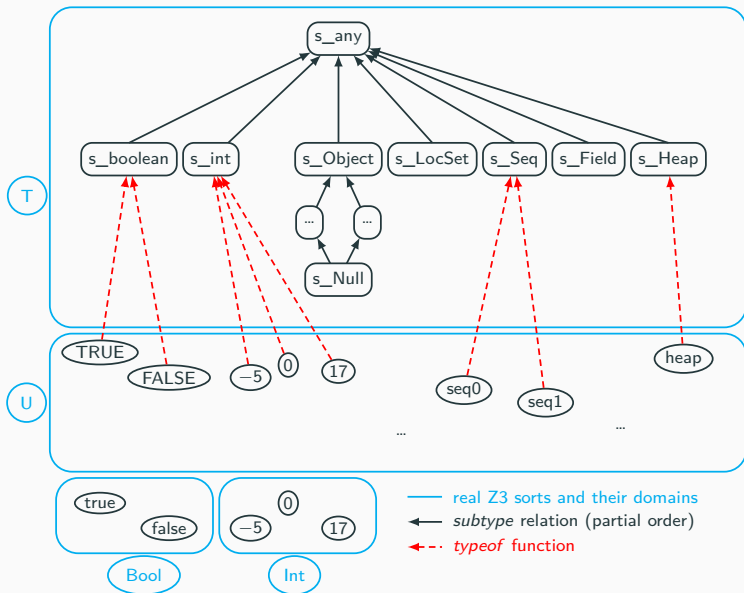
— real Z3 sorts and their domains

← subtype relation (partial order)

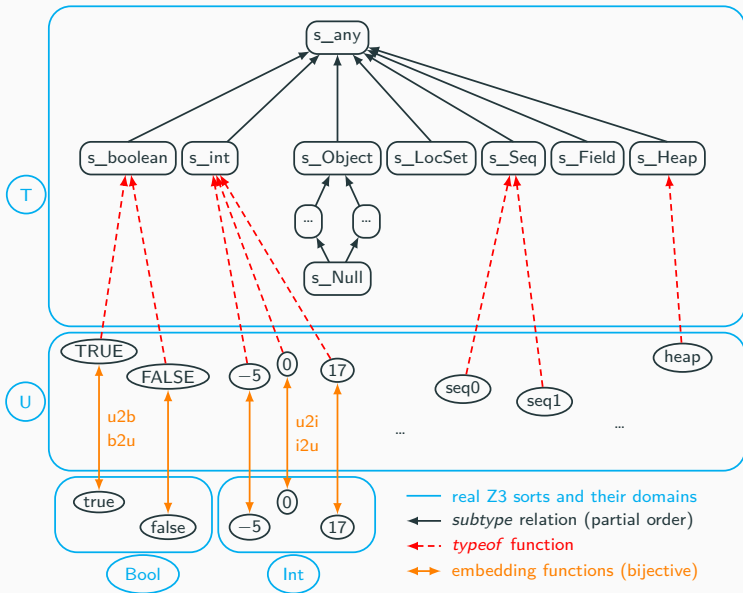
Existing Type Hierarchy Translation



Existing Type Hierarchy Translation



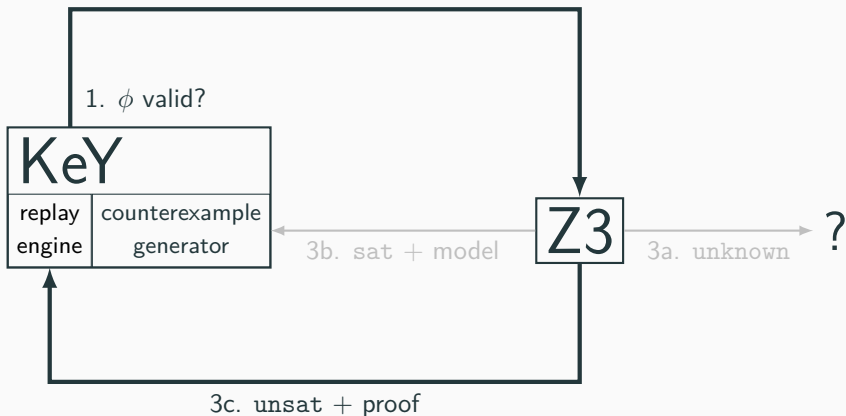
Existing Type Hierarchy Translation



Replay Technique: Overview

(A) type hierarchy axiomatization

2. $\neg\phi$ satisfiable?

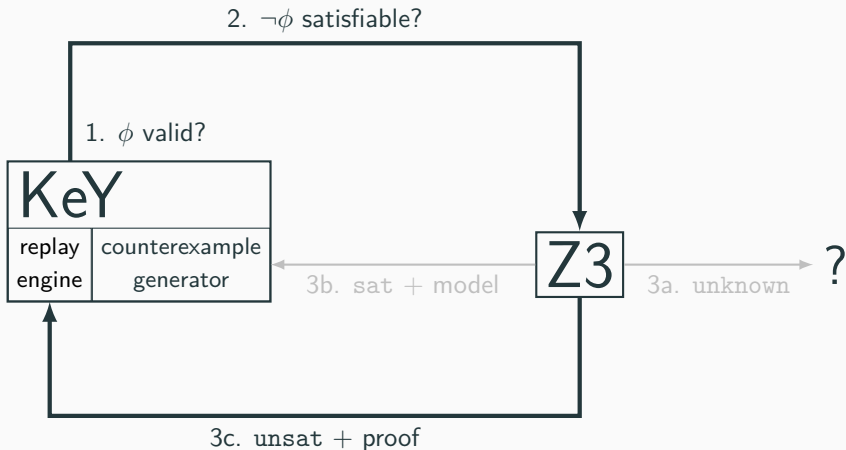


(B) equisatisfiability relation

(C) skolem functions

Replay Technique: Overview

(A) type hierarchy axiomatization ✓



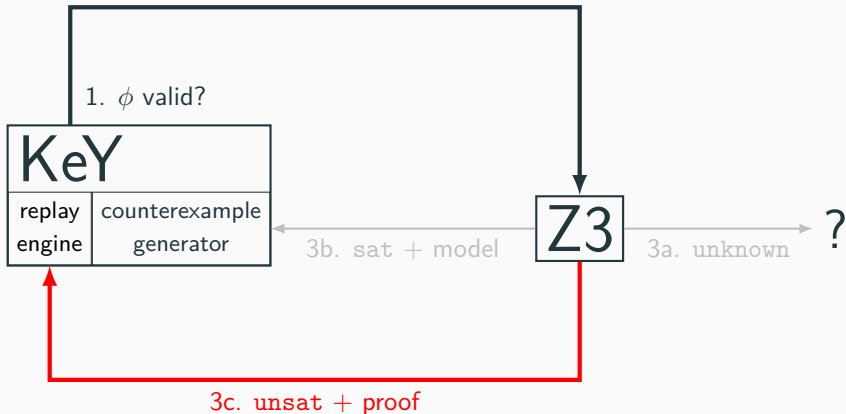
(B) equisatisfiability relation

(C) skolem functions

Replay Technique: Overview

(A) type hierarchy axiomatization ✓

2. $\neg\phi$ satisfiable?



(B) equisatisfiability relation

(C) skolem functions

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\frac{}{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{sk}$$

Z3 introduces skolem symbols via the sk rule in proof leaves:

$$\frac{\frac{\Gamma \vdash (\exists x. \phi(x)) \sim \phi(\mathbf{s})}{\dots} \text{sk}}{\dots \vdots \dots} \frac{\dots}{\Gamma \vdash \psi(\mathbf{s})} \dots$$

Z3 introduces skolem symbols via the sk rule in proof leaves:

$$\frac{\frac{\Gamma \vdash (\exists x. \phi(x)) \sim \phi(\mathbf{s})}{\dots} \text{sk}}{\dots \vdots \dots} \frac{\dots \vdots \dots}{\Gamma \vdash \psi(\mathbf{s})} \dots$$

Hilbert choice operator: $\epsilon x. \phi(x)$

Z3 introduces skolem symbols via the sk rule in proof leaves:

$$\frac{\frac{\Gamma \vdash (\exists x. \phi(x)) \sim \phi(\mathbf{s})}{\dots} \text{sk}}{\dots \vdots \dots} \frac{\dots}{\Gamma \vdash \psi(\mathbf{s})} \dots$$

Hilbert choice operator: $\epsilon x. \phi(x)$

Defining axiom:

$$\exists x. \phi(x) \leftrightarrow \phi(\epsilon x. \phi(x))$$

Skolemization Using Hilbert Choice Operator

For a skolem constant s :

$$\frac{\dots \quad \frac{\Gamma \vdash (\exists x. \phi(x)) \sim \phi(\mathbf{s})}{\dots} \text{sk} \quad \dots}{\dots \quad \frac{\Gamma \vdash \psi(\mathbf{s})}{\dots} \quad \dots} \dots$$
$$\frac{\dots \quad \frac{\Gamma \vdash (\exists x. \phi(x)) \sim \phi(\epsilon x. \phi(x))}{\dots} \text{sk} \quad \dots}{\dots \quad \frac{\Gamma \vdash \psi(\epsilon x. \phi(x))}{\dots} \quad \dots} \dots$$

Skolemization Using Hilbert Choice Operator

For a skolem constant s :

$$\frac{\frac{\Gamma \vdash (\exists x. \phi(x)) \sim \phi(s)}{\dots} \text{sk}}{\dots} \dots \frac{\Gamma \vdash \psi(s)}{\dots} \dots$$
$$\frac{\frac{\Gamma \vdash (\exists x. \phi(x)) \leftrightarrow \phi(\epsilon x. \phi(x))}{\dots} \text{sk}}{\dots} \dots \frac{\Gamma \vdash \psi(\epsilon x. \phi(x))}{\dots} \dots$$

Skolemization Using Hilbert Choice Operator

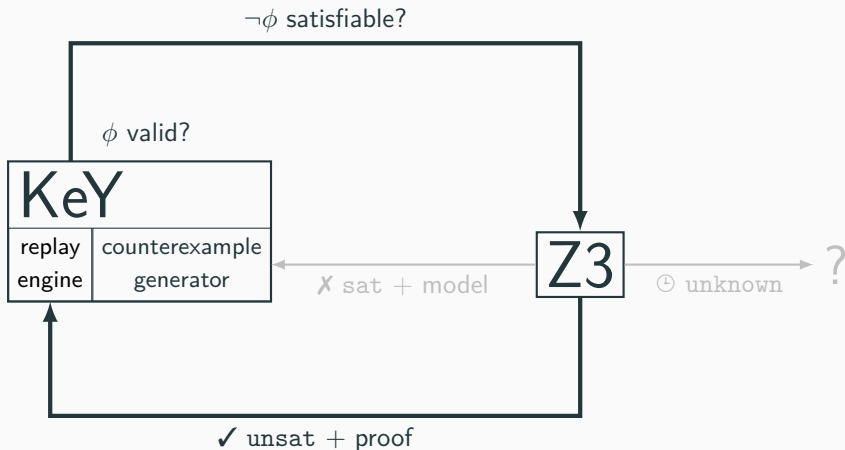
For a skolem constant s :

$$\frac{\frac{\Gamma \vdash (\exists x. \phi(x)) \sim \phi(s)}{\dots} \text{sk}}{\dots} \dots \frac{\Gamma \vdash \psi(s)}{\dots} \dots$$
$$\frac{\frac{\Gamma \vdash (\exists x. \phi(x)) \leftrightarrow \phi(\epsilon x. \phi(x))}{\dots} \text{sk}}{\dots} \dots \frac{\Gamma \vdash \psi(\epsilon x. \phi(x))}{\dots} \dots$$

This works also for skolem functions!

Replay Technique: Overview

(A) type hierarchy axiomatization ✓

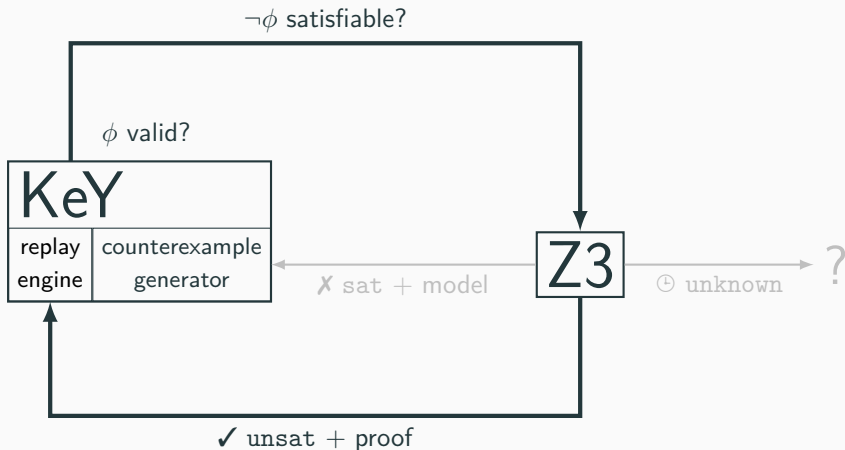


(B) equisatisfiability relation

(C) skolem functions

Replay Technique: Overview

(A) type hierarchy axiomatization ✓



(B) equisatisfiability relation ✓

(C) skolem functions ✓

Replay Technique: Further Notes

- Free variables in Z3 are directly translated to skolem symbols.

Replay Technique: Further Notes

- Free variables in Z3 are directly translated to skolem symbols.
- Translation is defined for 31 of 42 Z3 rules.

Replay Technique: Further Notes

- Free variables in Z3 are directly translated to skolem symbols.
- Translation is defined for 31 of 42 Z3 rules.
- Replay of theory reasoning needs automatic proof search in KeY.

Replay Technique: Further Notes

- Free variables in Z3 are directly translated to skolem symbols.
- Translation is defined for 31 of 42 Z3 rules.
- Replay of theory reasoning needs automatic proof search in KeY.
- Some schematic proof rules are currently replayed with automatic proof search.

Replay Technique: Further Notes

- Free variables in Z3 are directly translated to skolem symbols.
- Translation is defined for 31 of 42 Z3 rules.
- Replay of theory reasoning needs automatic proof search in KeY.
- Some schematic proof rules are currently replayed with automatic proof search.
- By construction, the technique is sound.

Replay Technique: Further Notes

- Free variables in Z3 are directly translated to skolem symbols.
- Translation is defined for 31 of 42 Z3 rules.
- Replay of theory reasoning needs automatic proof search in KeY.
- Some schematic proof rules are currently replayed with automatic proof search.
- By construction, the technique is sound.
- The technique is total (under assumptions).

Implementation

Prototype implementation (4 100 lines of Java code, 50 classes)

Prototype implementation (4 100 lines of Java code, 50 classes)

KeY Demo!

Evaluation

Evaluation 1: Time Statistics

Problem Name	KeY	Replay			Total
		Transl. + Z3	Replay	(Auto)	
doubleNeg	110	63	16	0	79
andCommutes	110	47	31	0	78
liarsville	110	62	110	94	170
inequations2	110	62	140	94	200
generalProjection	110	62	970	780	1 000
project (Aunt Agatha)	1 000	170	13 000	7 500	13 000
PUZ001p1	3 000	380	83 000	51 000	83 000

(time in ms)

Evaluation 1: Sharing Statistics

Problem Name	KeY Steps	Z3			Replay	
		Proof Lines	Shared Terms	Shared Sub- Proofs	Steps	(Auto)
doubleNeg	4	17	12	6	180	93
andCommutes	6	18	9	8	200	110
liarsville	14	66	19	2	160	110
inequations2	27	22	3	1	520	430
generalProjection	11	60	370	35	1 000	540
project (Aunt Agatha)	150	770	1 800	230	16 000	12 700
PUZ001p1	3 200	1 200	5 600	380	51 000	44 900

There are formulas where KeY does not find a proof!

There are formulas where KeY does not find a proof!

Example:

$$(\exists x.p(x) \rightarrow \exists x.q(x)) \rightarrow \exists x.(p(x) \rightarrow q(x))$$

Conclusion

Summary

There is a replay technique (sound, total) for replaying Z3 proofs in KeY.

Summary

There is a replay technique (sound, total) for replaying Z3 proofs in KeY.

Challenge	Solution
symbols with no counterpart in KeY occur in Z3 proofs	use different logical encoding (axioms, Hilbert operator, ...)
different proof directions	locally “invert” proof direction via cut rule. Hilbert operator
theory reasoning is a black box	use automatic proof search of KeY

Summary

There is a replay technique (sound, total) for replaying Z3 proofs in KeY.

Challenge	Solution
symbols with no counterpart in KeY occur in Z3 proofs	use different logical encoding (axioms, Hilbert operator, ...)
different proof directions	locally “invert” proof direction via cut rule. Hilbert operator
theory reasoning is a black box	use automatic proof search of KeY

There is working prototype.

Future work ideas: Use only some information from the proof (e.g. quantifier instantiations, theory lemmas).

Appendix

Sharing of Terms and Sub-Proofs in Z3

Input (SMT-LIB):

```
1 (set-option :produce-proofs true)
2
3 (declare-const p Bool)
4
5 (assert (not (or p (not p) (not p)))) ;  $\neg(p \vee \neg p \vee \neg p)$ 
6
7 (check-sat)
8 (get-proof)
```

Sharing of Terms and Sub-Proofs in Z3

Input (SMT-LIB):

```
1 (set-option :produce-proofs true)
2
3 (declare-const p Bool)
4
5 (assert (not (or p (not p) (not p)))) ;  $\neg(p \vee \neg p \vee \neg p)$ 
6
7 (check-sat)
8 (get-proof)
```

Output with sharing:

```
1 unsat
2 (let ((a!1 (not p)))
3 (let ((a!2 (or p a!1 a!1)))
4 (let ((a!3 (not a!2)))
5 (let ((a!4 (asserted a!3)))
6 (let ((a!5 (not-or-elim a!4 a!1))
7 (a!6 (not-or-elim a!4 p)))
8 (unit-resolution a!5 a!6 false))))))
```

Output “un-shared”:

```
1 unsat
2 (unit-resolution
3 (not-or-elim
4 (asserted (not (or p (not p) (not p))))
5 (not p))
6 (not-or-elim
7 (asserted (not (or p (not p) (not p))))
8 p)
9 false)
```

Skolem Functions

The skolem symbol s depends on the free variable y :

$$\frac{\frac{\frac{\dots}{\vdots} \dots}{\Gamma \vdash (\exists x. \phi(x, y)) \sim \phi(s(y))} \text{sk}}{\dots} \dots$$
$$\frac{\frac{\frac{\dots}{\vdots} \dots}{\Gamma \vdash \chi(y) \sim \rho(y)} \text{quant-intro}}{\Gamma \vdash Q(y). \chi(y) \sim Q(y). \rho(y)} \dots$$
$$\frac{\dots}{\vdots} \dots$$
$$\frac{\dots}{\Gamma \vdash \psi(s(t))} \dots$$

Skolem Functions

The skolem symbol s depends on the free variable y :

$$\begin{array}{c}
 \frac{\dots}{\dots} \vdots \dots \\
 \frac{\Gamma \vdash (\exists x. \phi(x, y)) \sim \phi(s(y))}{\dots} \text{sk} \\
 \dots \\
 \frac{\Gamma \vdash \chi(y) \sim \rho(y)}{\Gamma \vdash Q(y). \chi(y) \sim Q(y). \rho(y)} \text{quant-intro} \\
 \dots \\
 \dots \\
 \vdots \\
 \Gamma \vdash \psi(s(t)) \dots
 \end{array}$$

$$\begin{array}{c}
 \dots \\
 \dots \\
 \frac{\Gamma \vdash (\exists x. \phi(x, y)) \leftrightarrow \phi(\epsilon x. \phi(x, y))}{\dots} \text{sk} \\
 \dots \\
 \vdots \\
 \dots \\
 \frac{\Gamma \vdash \chi(y) \leftrightarrow \rho(y)}{\Gamma \vdash Q(y). \chi(y) \leftrightarrow Q(y). \rho(y)} \text{quant-intro} \\
 \dots \\
 \dots \\
 \vdots \\
 \dots \\
 \Gamma \vdash \psi([y/t] \epsilon x. \phi(x, y)) \dots
 \end{array}$$

Formulas Where KeY does not find a proof automatically

$$(\exists x.p(x) \rightarrow \exists x.q(x)) \rightarrow \exists x.(p(x) \rightarrow q(x))$$

Formulas Where KeY does not find a proof automatically

$$(\exists x.p(x) \rightarrow \exists x.q(x)) \rightarrow \exists x.(p(x) \rightarrow q(x))$$

$$\begin{array}{c}
 \frac{\frac{\frac{\frac{\frac{\psi(c), \phi(c) \implies \psi(c)}{\psi(c) \implies \phi(c) \rightarrow \psi(c)}{\text{close}}}{\psi(c) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\text{impLeft}}}{\psi(c) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\text{exRight}}}{\implies \exists x.\phi(x), \exists x.(\phi(x) \rightarrow \psi(x))} \text{?} \quad \frac{\psi(c) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\exists x.\psi(x) \implies \exists x.(\phi(x) \rightarrow \psi(x))} \text{exLeft}}{\implies \exists x.\phi(x), \exists x.(\phi(x) \rightarrow \psi(x))} \text{?} \quad \frac{\exists x.\psi(x) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\exists x.\phi(x) \rightarrow \exists x.\psi(x) \implies \exists x.(\phi(x) \rightarrow \psi(x))} \text{impLeft}}{\implies (\exists x.\phi(x) \rightarrow \exists x.\psi(x)) \rightarrow \exists x.(\phi(x) \rightarrow \psi(x))} \text{impRight}
 \end{array}$$

Formulas Where KeY does not find a proof automatically

$$(\exists x.p(x) \rightarrow \exists x.q(x)) \rightarrow \exists x.(p(x) \rightarrow q(x))$$

$$\frac{\frac{\frac{\frac{\frac{\frac{\psi(c), \phi(c) \implies \psi(c)}{\psi(c) \implies \phi(c) \rightarrow \psi(c)}{\text{close}}}{\psi(c) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\text{impLeft}}}{\psi(c) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\text{exRight}}}{\implies \exists x.\phi(x), \exists x.(\phi(x) \rightarrow \psi(x)) \quad ?}{\implies \exists x.\psi(x) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\text{exLeft}}}{\implies (\exists x.\phi(x) \rightarrow \exists x.\psi(x)) \implies \exists x.(\phi(x) \rightarrow \psi(x))}{\text{impLeft}}}$$

Problem: No ground term available for quantifier instantiation.

Formulas Where KeY does not find a proof automatically (2)

$$\exists x.\exists y.(r(x,y) \leftrightarrow r(y,x))$$

Formulas Where KeY does not find a proof automatically (2)

$$\exists x.\exists y.(r(x,y) \leftrightarrow r(y,x))$$

$$\frac{\quad ?}{\implies \exists x.\exists y.(r(x,y) \leftrightarrow r(y,x))} ?$$

Formulas Where KeY does not find a proof automatically (2)

$$\exists x.\exists y.(r(x,y) \leftrightarrow r(y,x))$$

$$\frac{\quad ?}{\implies \exists x.\exists y.(r(x,y) \leftrightarrow r(y,x))} ?$$

Problem: No ground term available for quantifier instantiation.

Existing Type Hierarchy Axiomatization

subtype relation is a partial order:

reflexive:

$$\forall t : T. \text{subtype}(t, t)$$

antisymmetric:

$$\forall t_1 : T, t_2 : T. \text{subtype}(t_1, t_2) \wedge \text{subtype}(t_2, t_1) \rightarrow t_1 = t_2$$

transitive:

$$\forall t_1 : T, t_2 : T, t_3 : T. \text{subtype}(t_1, t_2) \wedge \text{subtype}(t_2, t_3) \rightarrow \text{subtype}(t_1, t_3)$$

cast:

$$\forall u : U, t : T. \text{subtype}(\text{typeof}(\text{cast}(t, u)), t)$$

$$\forall u : U, t : T. \text{subtype}(\text{typeof}(u), t) \rightarrow \text{cast}(t, u) = u$$

for each sort s with n child sorts c_1, \dots, c_n :

sort “definition”:

$$\text{instanceof}(t, u)$$

for every subsort c_i :

$$\text{subtype}(c_i, s)$$

Existing Type Hierarchy Axiomatization

subtype relation is a partial order:

reflexive:

$$\forall t : T. \text{subtype}(t, t)$$

antisymmetric:

$$\forall t_1 : T, t_2 : T. \text{subtype}(t_1, t_2) \wedge \text{subtype}(t_2, t_1) \rightarrow t_1 = t_2$$

transitive:

$$\forall t_1 : T, t_2 : T, t_3 : T. \text{subtype}(t_1, t_2) \wedge \text{subtype}(t_2, t_3) \rightarrow \text{subtype}(t_1, t_3)$$

cast:

$$\forall u : U, t : T. \text{subtype}(\text{typeof}(\text{cast}(t, u)), t)$$

$$\forall u : U, t : T. \text{subtype}(\text{typeof}(u), t) \rightarrow \text{cast}(t, u) = u$$

for each sort s with n child sorts c_1, \dots, c_n :

sort “definition”:

$$\text{instanceof}(t, u)$$

for every subsort c_i :

$$\text{subtype}(c_i, s)$$

$O(\#\text{sorts}^2)$ axioms

New Type Hierarchy Axiomatization

for each sort s with n child sorts c_1, \dots, c_n :

sort "definition":

$$\forall u : U. \text{exactInstanceof}(s, u) \vee \text{instanceof}(c_1, u) \vee \dots \vee \text{instanceof}(c_n, u) \rightarrow \text{instanceof}(s, u)$$

cast:

$$\forall u : U. \text{instanceof}(s, \text{cast}(s, u))$$

$$\forall u : U. \text{instanceof}(s, u) \rightarrow \text{cast}(s, u) = u$$

typeguard:

$$\forall u : U. \text{typeguard}(s, u) \leftrightarrow \text{instanceof}(s, u)$$

for every subsort c_i :

at most one type:

$$\forall u : U. \neg \text{exactInstanceof}(s, u) \vee \neg \text{instanceof}(c_i, u)$$

additional axiom for sort_any only:

for every subsort pair (c_i, c_j) , subsorts are disjoint:

$$\forall u : U. \neg \text{instanceof}(c_i, u) \vee \neg \text{instanceof}(c_j, u) \quad (i \neq j)$$

additional axioms for object sorts only:

for every subsort pair (c_i, c_j) subsorts are disjoint (except null):

$$\forall u : U. \text{instanceof}(c_i, u) \wedge \text{instanceof}(c_j, u) \rightarrow u = \text{null} \quad (i \neq j)$$

New Type Hierarchy Axiomatization

for each sort s with n child sorts c_1, \dots, c_n :

sort "definition":

$$\forall u : U. \text{exactInstanceof}(s, u) \vee \text{instanceof}(c_1, u) \vee \dots \vee \text{instanceof}(c_n, u) \rightarrow \text{instanceof}(s, u)$$

cast:

$$\forall u : U. \text{instanceof}(s, \text{cast}(s, u))$$

$$\forall u : U. \text{instanceof}(s, u) \rightarrow \text{cast}(s, u) = u$$

typeguard:

$$\forall u : U. \text{typeguard}(s, u) \leftrightarrow \text{instanceof}(s, u)$$

for every subsort c_i :

at most one type:

$$\forall u : U. \neg \text{exactInstanceof}(s, u) \vee \neg \text{instanceof}(c_i, u)$$

additional axiom for sort_ only:

for every subsort pair (c_i, c_j) , subsorts are disjoint:

$$\forall u : U. \neg \text{instanceof}(c_i, u) \vee \neg \text{instanceof}(c_j, u) \quad (i \neq j)$$

$O(\#\text{sorts}^3)$ axioms

additional axioms for object sorts only:

for every subsort pair (c_i, c_j) subsorts are disjoint (except null):

$$\forall u : U. \text{instanceof}(c_i, u) \wedge \text{instanceof}(c_j, u) \rightarrow u = \text{null} \quad (i \neq j)$$

Equisatisfiability Relation

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\frac{}{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{sk}$$

Equisatisfiability Relation

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\frac{}{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{sk}$$

Semantics (free variables \bar{x} and \bar{y} , first order structures $\mathcal{M}_1, \mathcal{M}_2$):

$$\begin{aligned} \phi(\bar{x}) \sim \psi(\bar{y}) \quad \text{iff} \quad & \text{for all } \beta: && \text{there is some } \mathcal{M}_1 : (\mathcal{M}_1, \beta) \models \phi(\bar{x}) \\ & && \text{iff} \quad \text{there is some } \mathcal{M}_2 : (\mathcal{M}_2, \beta) \models \psi(\bar{y}) \end{aligned}$$

Equisatisfiability Relation

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\frac{}{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{sk}$$

Semantics (free variables \bar{x} and \bar{y} , first order structures $\mathcal{M}_1, \mathcal{M}_2$):

$$\begin{aligned} \phi(\bar{x}) \sim \psi(\bar{y}) \quad \text{iff} \quad & \text{for all } \beta: \quad \text{there is some } \mathcal{M}_1 : (\mathcal{M}_1, \beta) \models \phi(\bar{x}) \\ & \text{iff} \quad \text{there is some } \mathcal{M}_2 : (\mathcal{M}_2, \beta) \models \psi(\bar{y}) \end{aligned}$$

Examples:

$$f(x) = 5 \sim \text{true}$$

Equisatisfiability Relation

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\overline{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{ sk}$$

Semantics (free variables \bar{x} and \bar{y} , first order structures $\mathcal{M}_1, \mathcal{M}_2$):

$$\begin{aligned} \phi(\bar{x}) \sim \psi(\bar{y}) \quad \text{iff} \quad \text{for all } \beta: & \quad \text{there is some } \mathcal{M}_1 : (\mathcal{M}_1, \beta) \models \phi(\bar{x}) \\ & \quad \text{iff} \quad \text{there is some } \mathcal{M}_2 : (\mathcal{M}_2, \beta) \models \psi(\bar{y}) \end{aligned}$$

Examples:

$$f(x) = 5 \sim \text{true}$$

$$f(x) = 5 \wedge f(x) = 6 \sim \text{false}$$

Equisatisfiability Relation

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\overline{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{ sk}$$

Semantics (free variables \bar{x} and \bar{y} , first order structures $\mathcal{M}_1, \mathcal{M}_2$):

$$\begin{aligned} \phi(\bar{x}) \sim \psi(\bar{y}) \quad \text{iff} \quad & \text{for all } \beta: \quad \text{there is some } \mathcal{M}_1 : (\mathcal{M}_1, \beta) \models \phi(\bar{x}) \\ & \text{iff} \quad \text{there is some } \mathcal{M}_2 : (\mathcal{M}_2, \beta) \models \psi(\bar{y}) \end{aligned}$$

Examples:

$$f(x) = 5 \sim \text{true}$$

$$f(x) = 5 \wedge f(x) = 6 \sim \text{false}$$

$$f(x) = 5 \sim f(x) = 6$$

Equisatisfiability Relation

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\overline{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{ sk}$$

Semantics (free variables \bar{x} and \bar{y} , first order structures $\mathcal{M}_1, \mathcal{M}_2$):

$$\begin{aligned} \phi(\bar{x}) \sim \psi(\bar{y}) \quad \text{iff} \quad \text{for all } \beta: & \quad \text{there is some } \mathcal{M}_1 : (\mathcal{M}_1, \beta) \models \phi(\bar{x}) \\ & \quad \text{iff} \quad \text{there is some } \mathcal{M}_2 : (\mathcal{M}_2, \beta) \models \psi(\bar{y}) \end{aligned}$$

Examples:

$$\begin{aligned} f(x) = 5 &\sim \text{true} & x = 5 &\approx x = 6 \\ f(x) = 5 \wedge f(x) = 6 &\sim \text{false} \\ f(x) = 5 &\sim f(x) = 6 \end{aligned}$$

Equisatisfiability Relation

Explicit equisatisfiability relation (\sim) in Z3 terms:

$$\overline{\vdash (\exists x. \phi(x)) \sim \phi(s)} \text{ sk}$$

Semantics (free variables \bar{x} and \bar{y} , first order structures $\mathcal{M}_1, \mathcal{M}_2$):

$$\begin{aligned} \phi(\bar{x}) \sim \psi(\bar{y}) \quad \text{iff} \quad \text{for all } \beta: & \quad \text{there is some } \mathcal{M}_1 : (\mathcal{M}_1, \beta) \models \phi(\bar{x}) \\ & \quad \text{iff} \quad \text{there is some } \mathcal{M}_2 : (\mathcal{M}_2, \beta) \models \psi(\bar{y}) \end{aligned}$$

Examples:

$$\begin{aligned} f(x) = 5 &\sim \text{true} & x = 5 &\approx x = 6 \\ f(x) = 5 \wedge f(x) = 6 &\sim \text{false} & x = 5 &\approx y = 6 \\ f(x) = 5 &\sim f(x) = 6 \end{aligned}$$