Reconstructing Z3 Proofs With KeY

Wolfram Pfeifer

07.05.2020

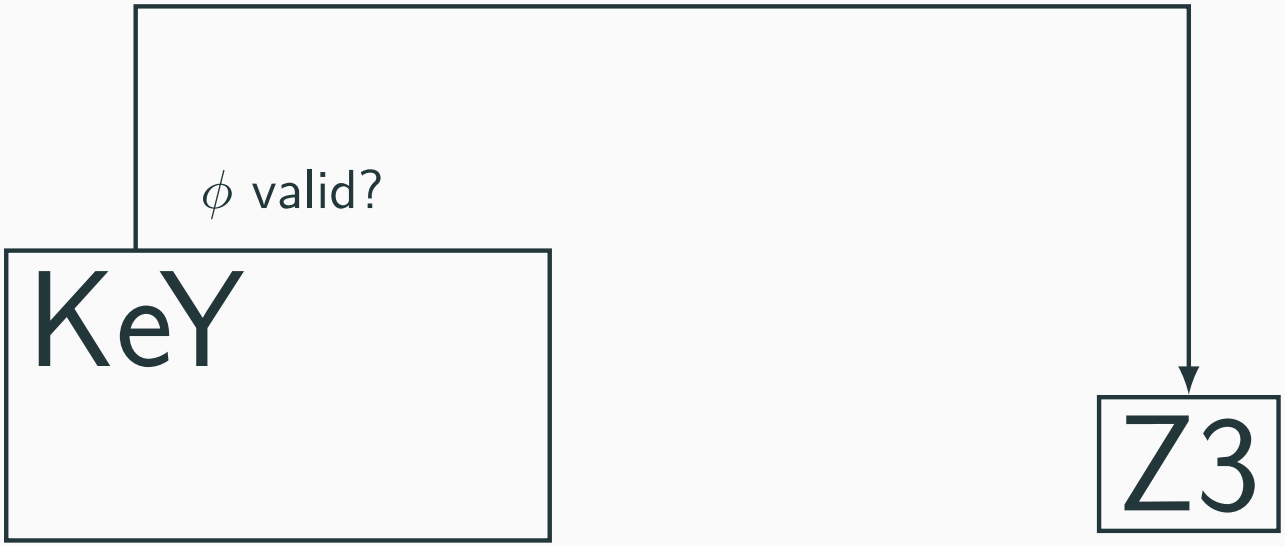
Karlsruhe Institute of Technology (KIT)
Institute of Theoretical Informatics

Motivation

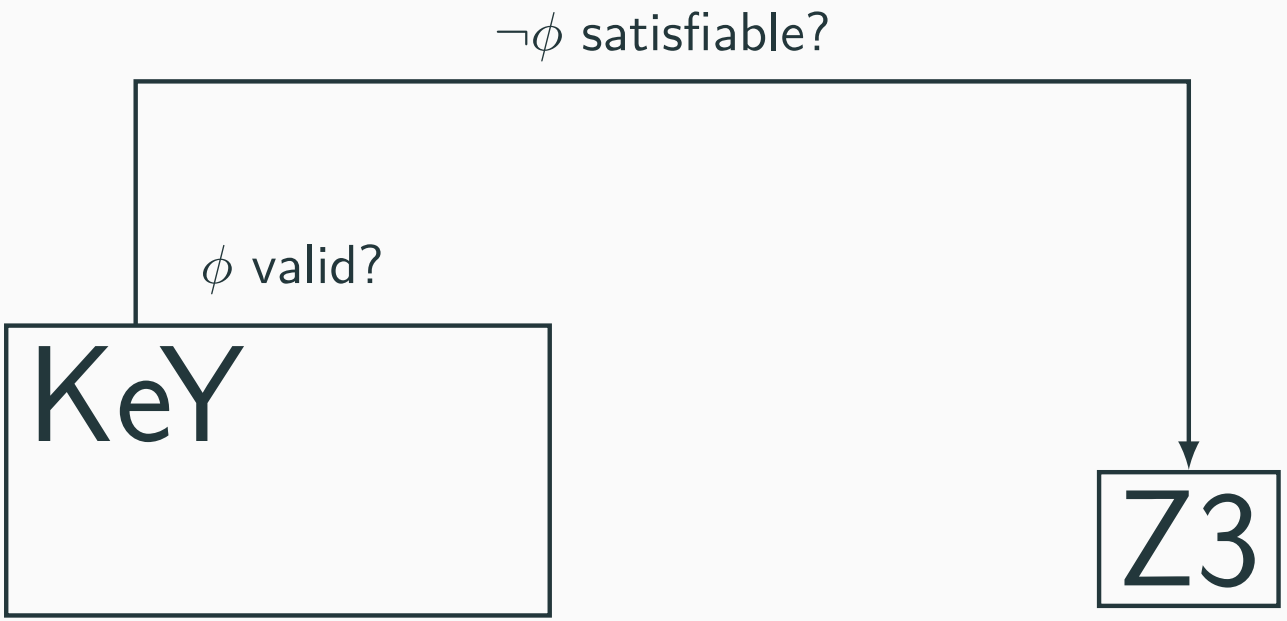
Motivation: Current KeY Workflow with SMT Solvers

ϕ valid?

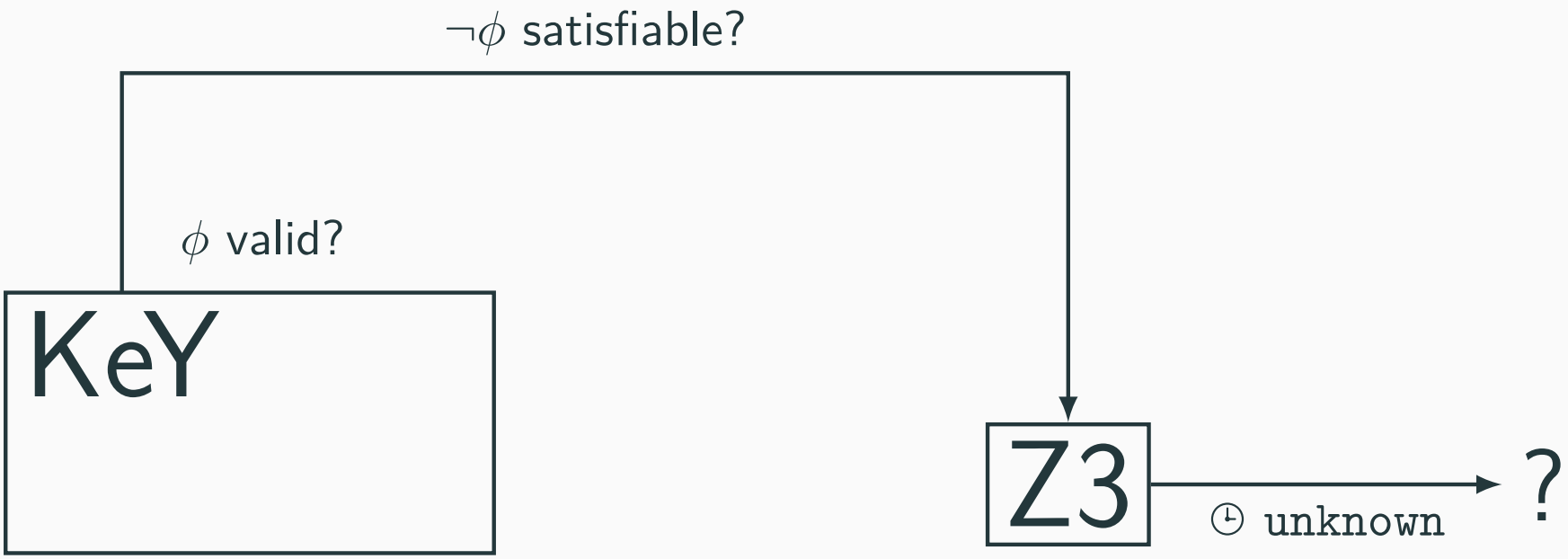
Motivation: Current KeY Workflow with SMT Solvers



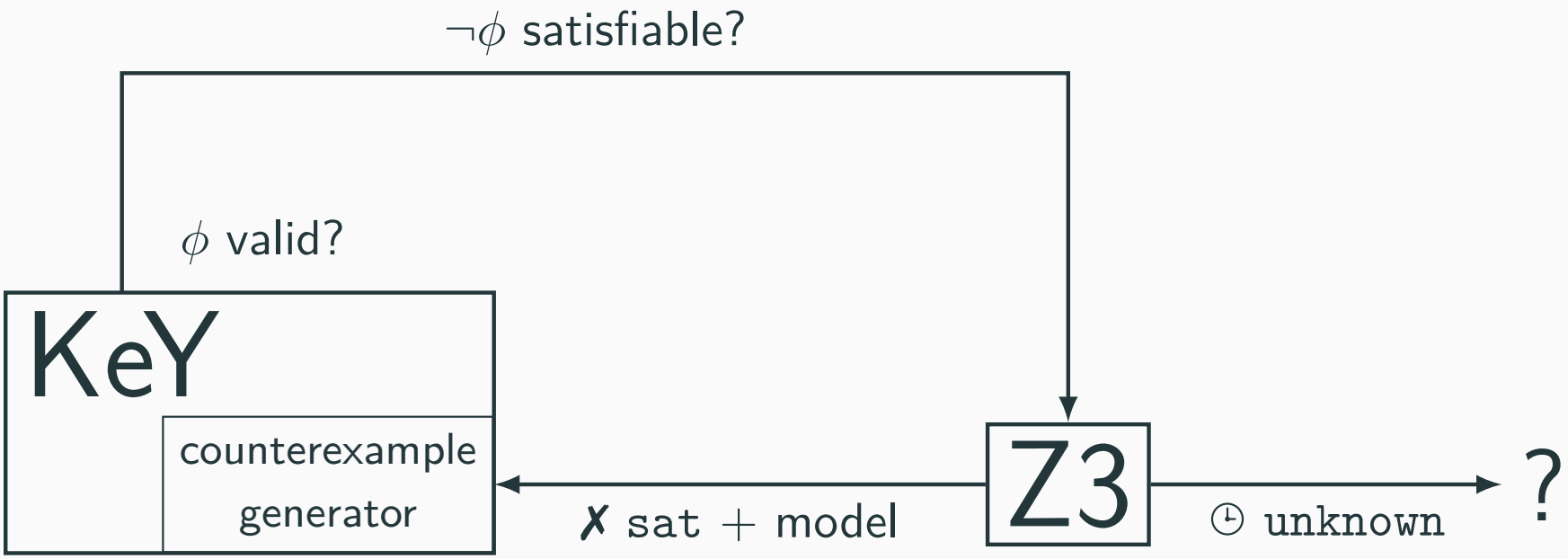
Motivation: Current KeY Workflow with SMT Solvers



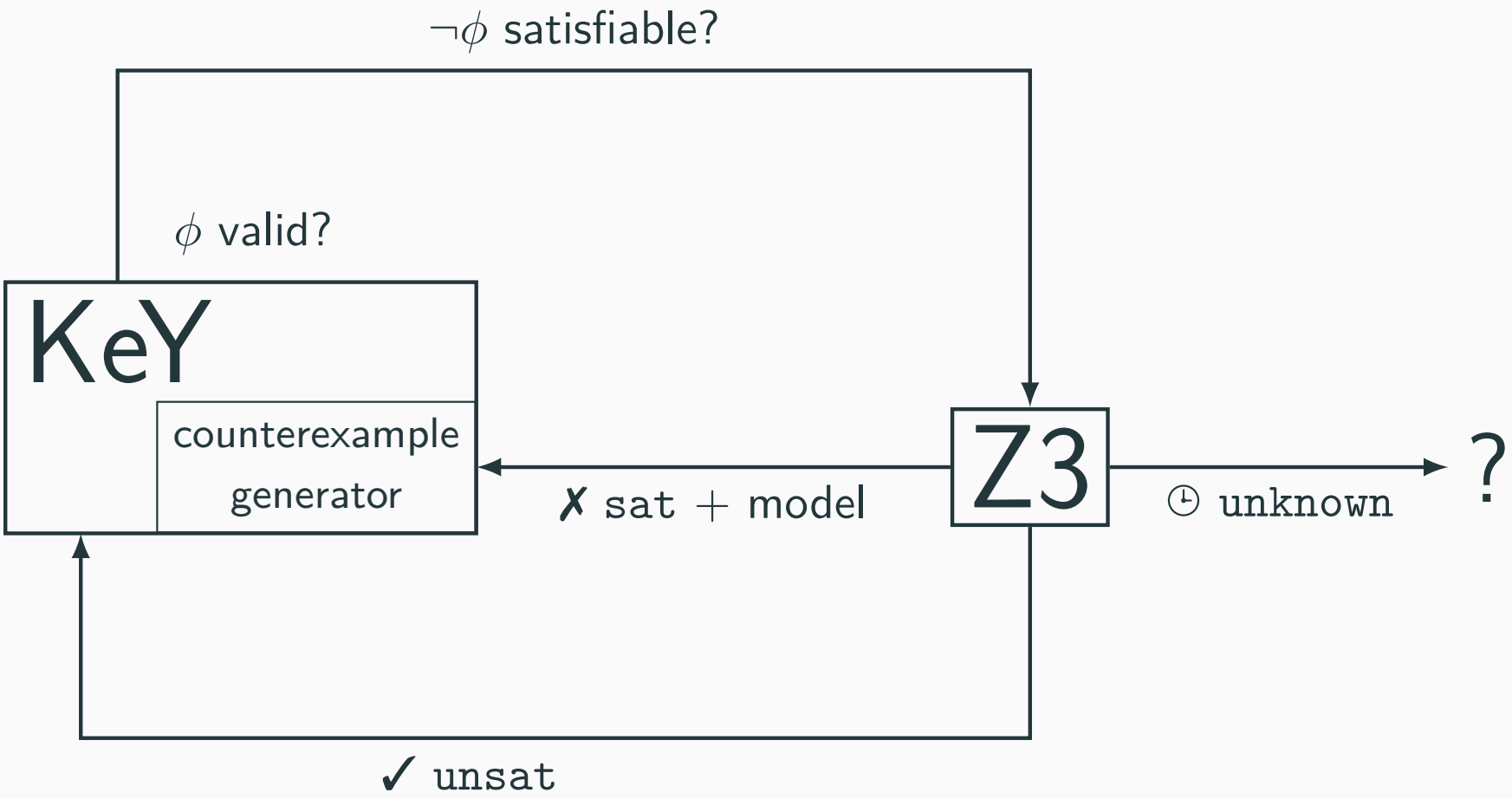
Motivation: Current KeY Workflow with SMT Solvers



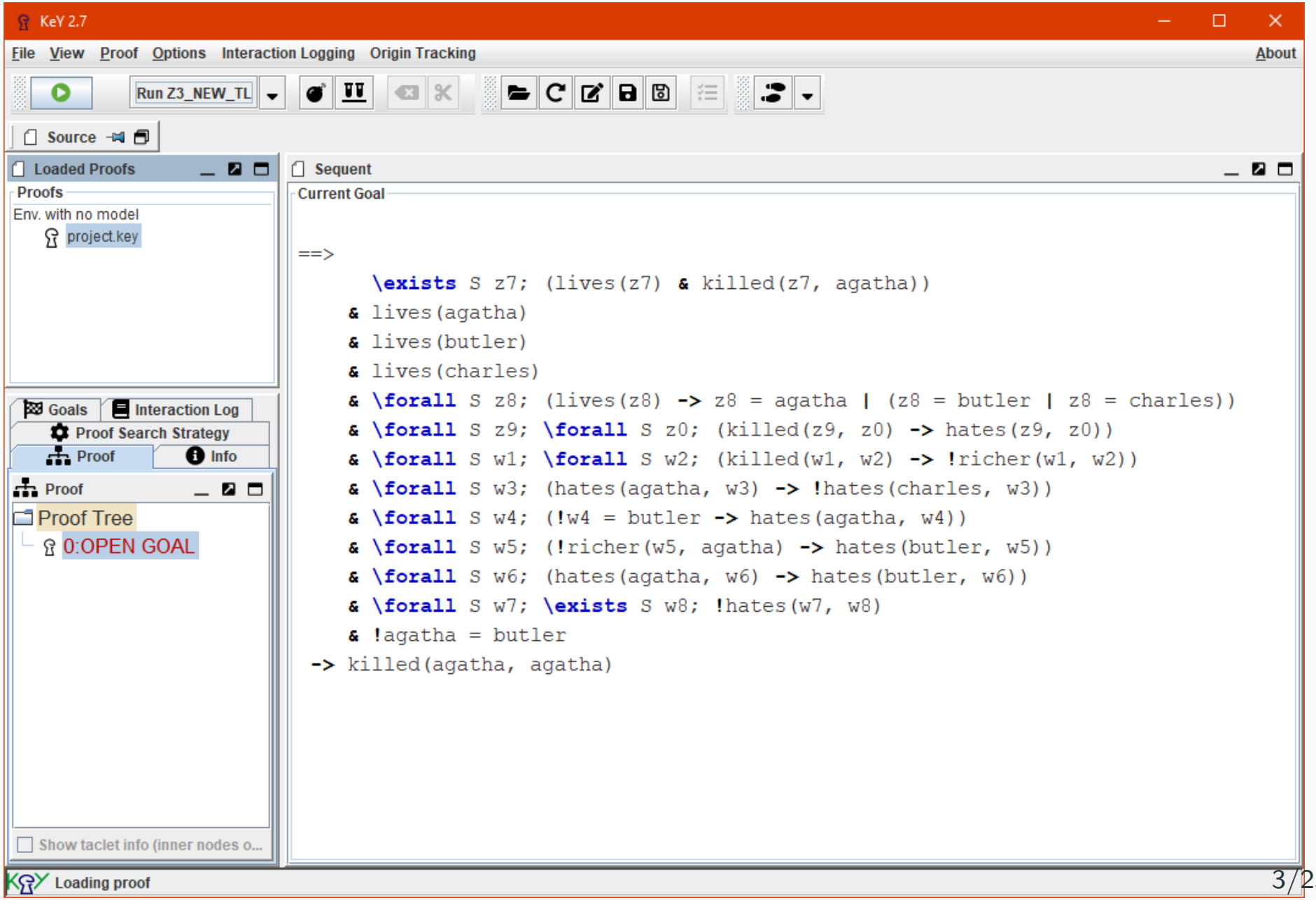
Motivation: Current KeY Workflow with SMT Solvers



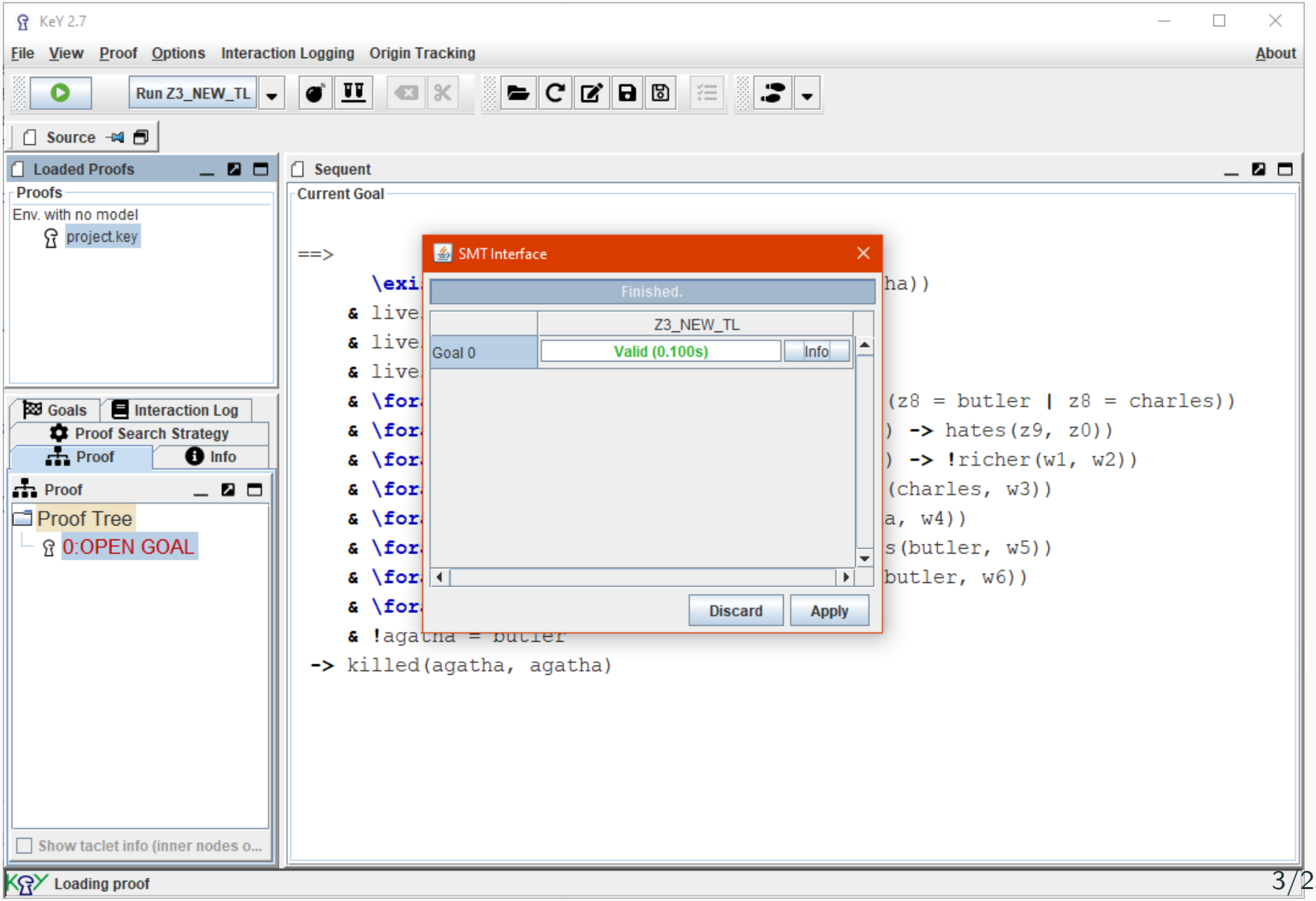
Motivation: Current KeY Workflow with SMT Solvers



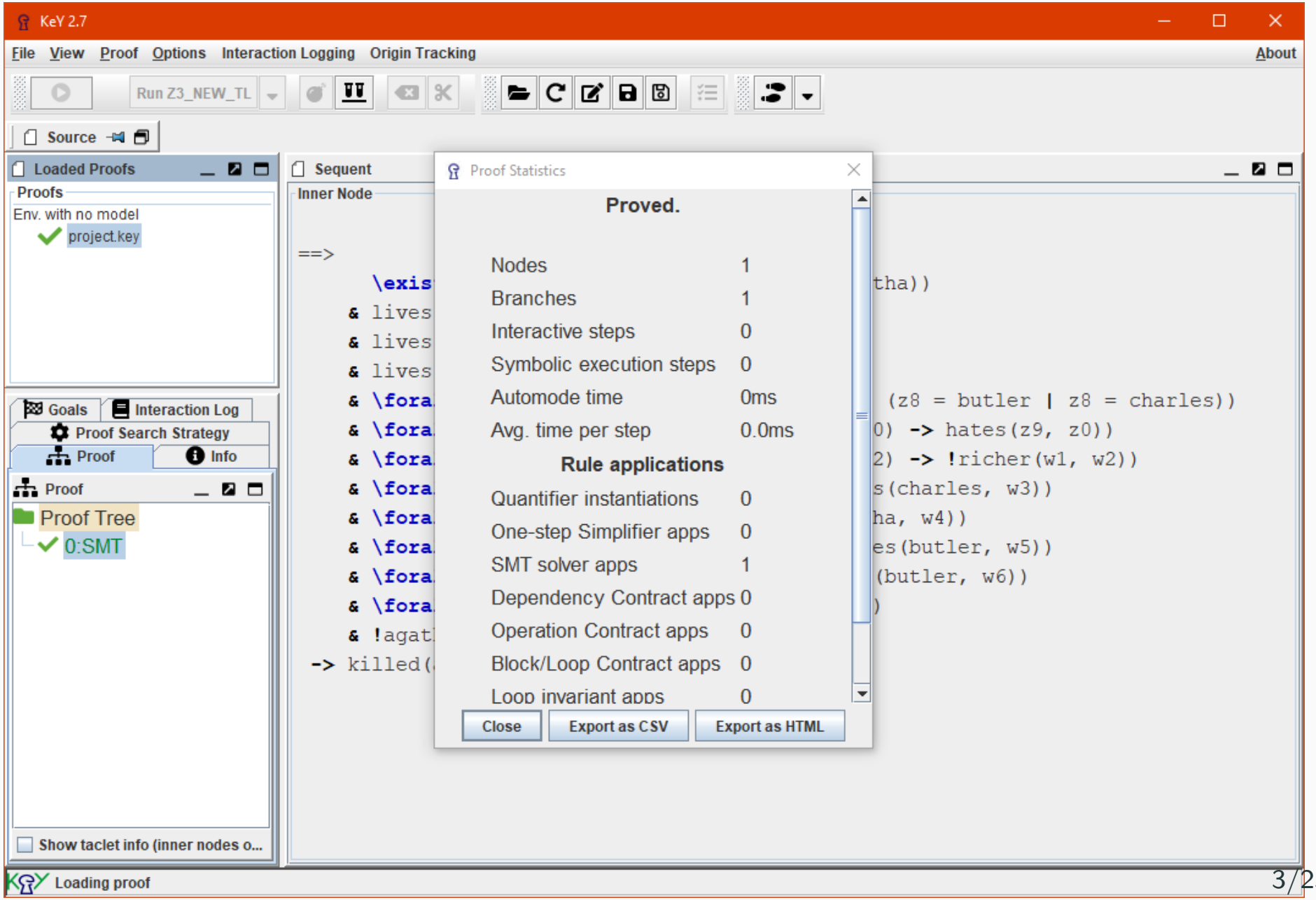
Current Behavior



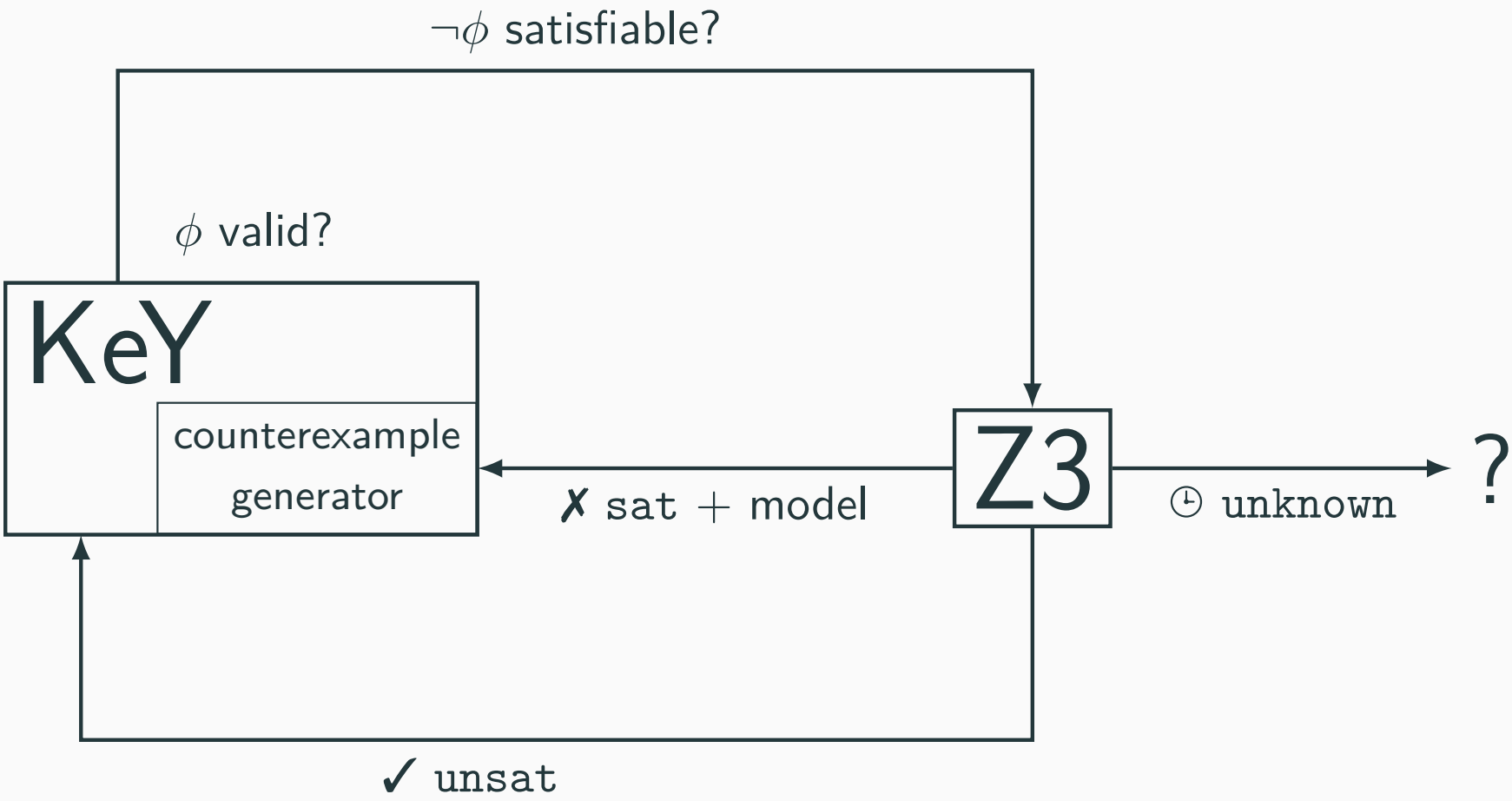
Current Behavior



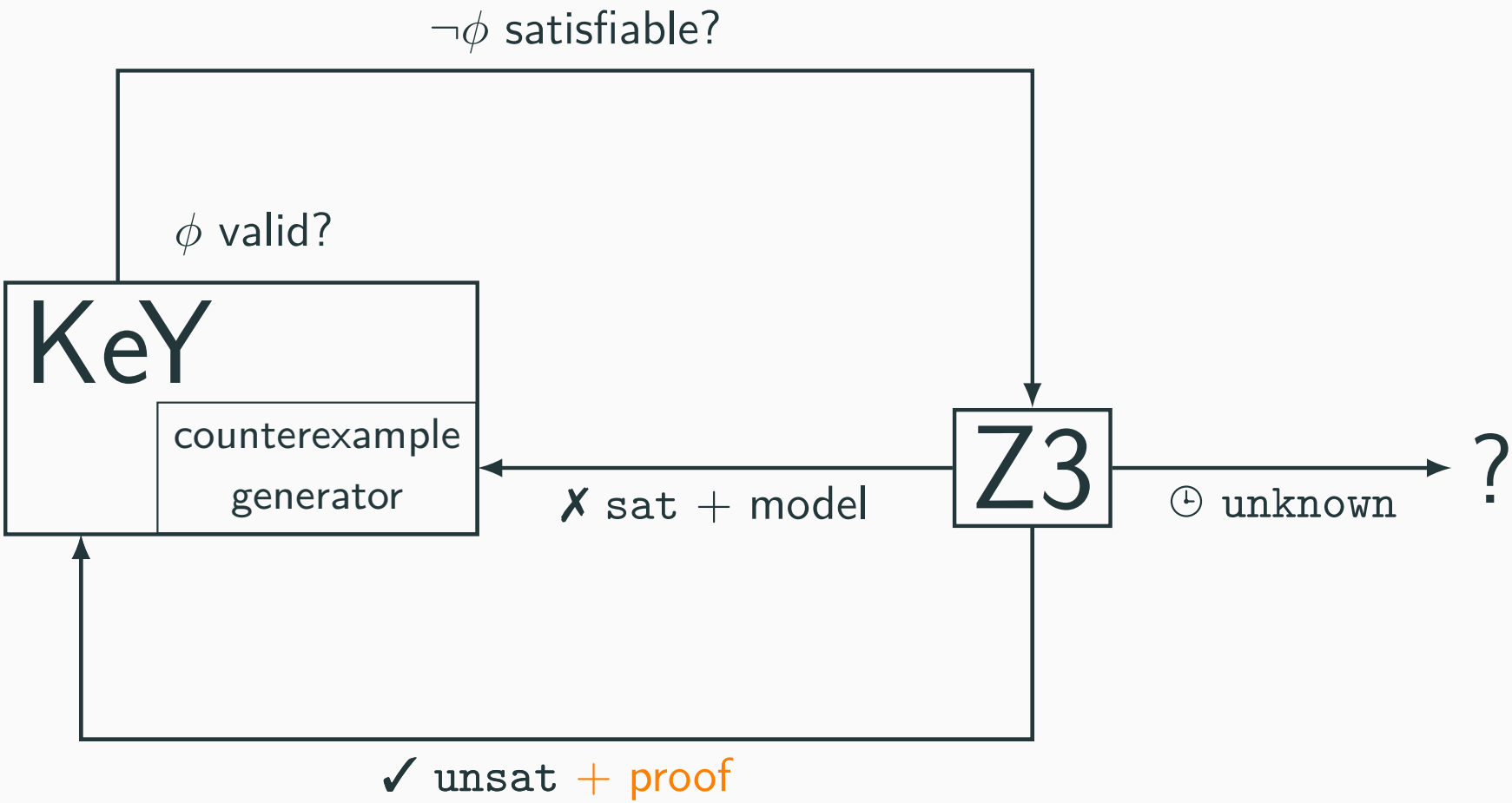
Current Behavior



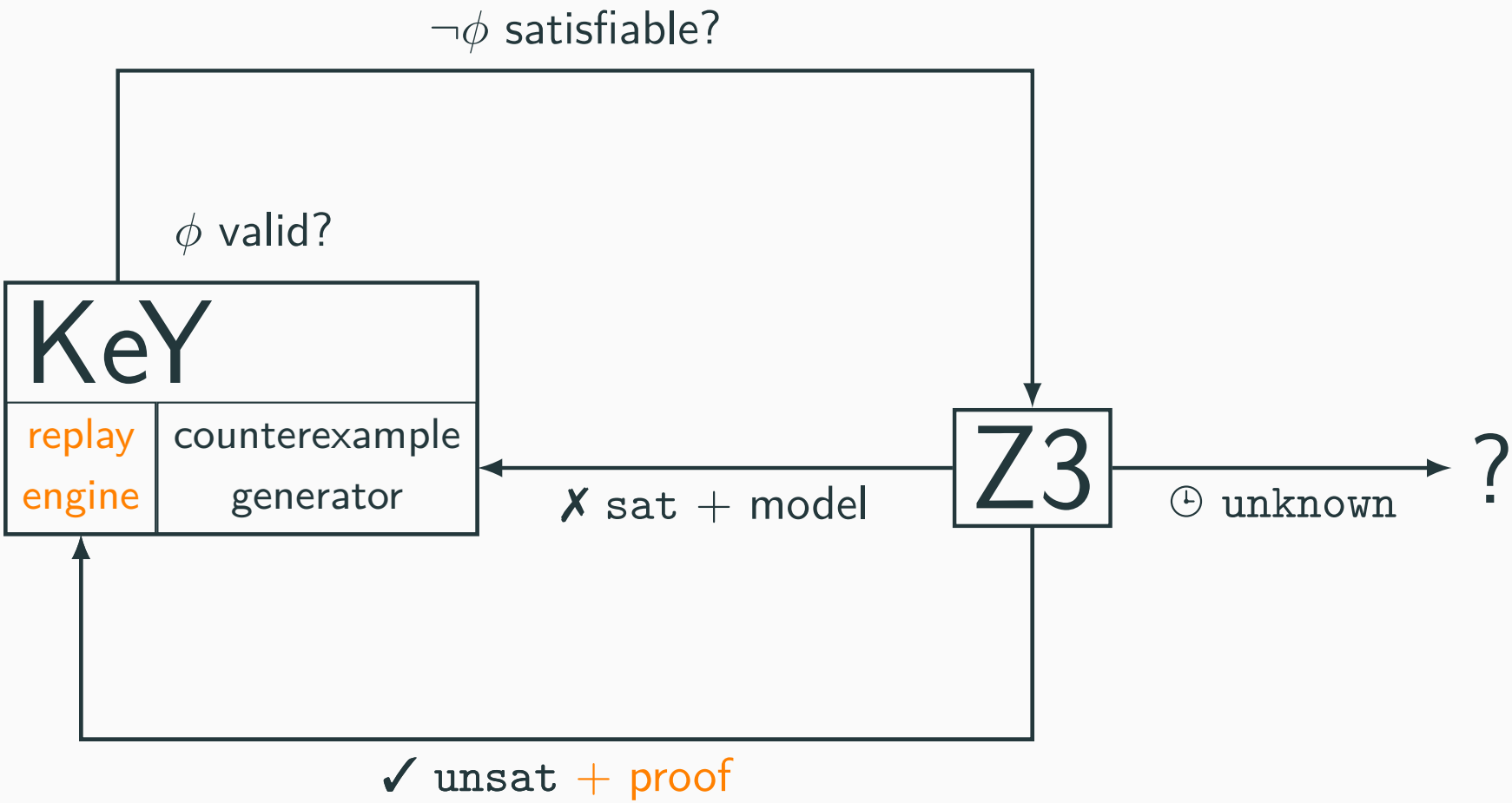
Replay Idea


$\neg\phi$ satisfiable?
ϕ valid?
KeY
counterexample generator
Z3
✗ sat + model
unknown
?
✓ unsat

Replay Idea


$\neg\phi$ satisfiable?
ϕ valid?
KeY
counterexample generator
Z3
✗ sat + model
unknown
?
✓ unsat + proof

Replay Idea


$\neg\phi$ satisfiable?
ϕ valid?
KeY
replay engine
counterexample generator
Z3
✗ sat + model
unknown
?
✓ unsat + 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)))                    ;     ¬(p ∨ 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)))                    ;     ¬(p ∨ 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

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

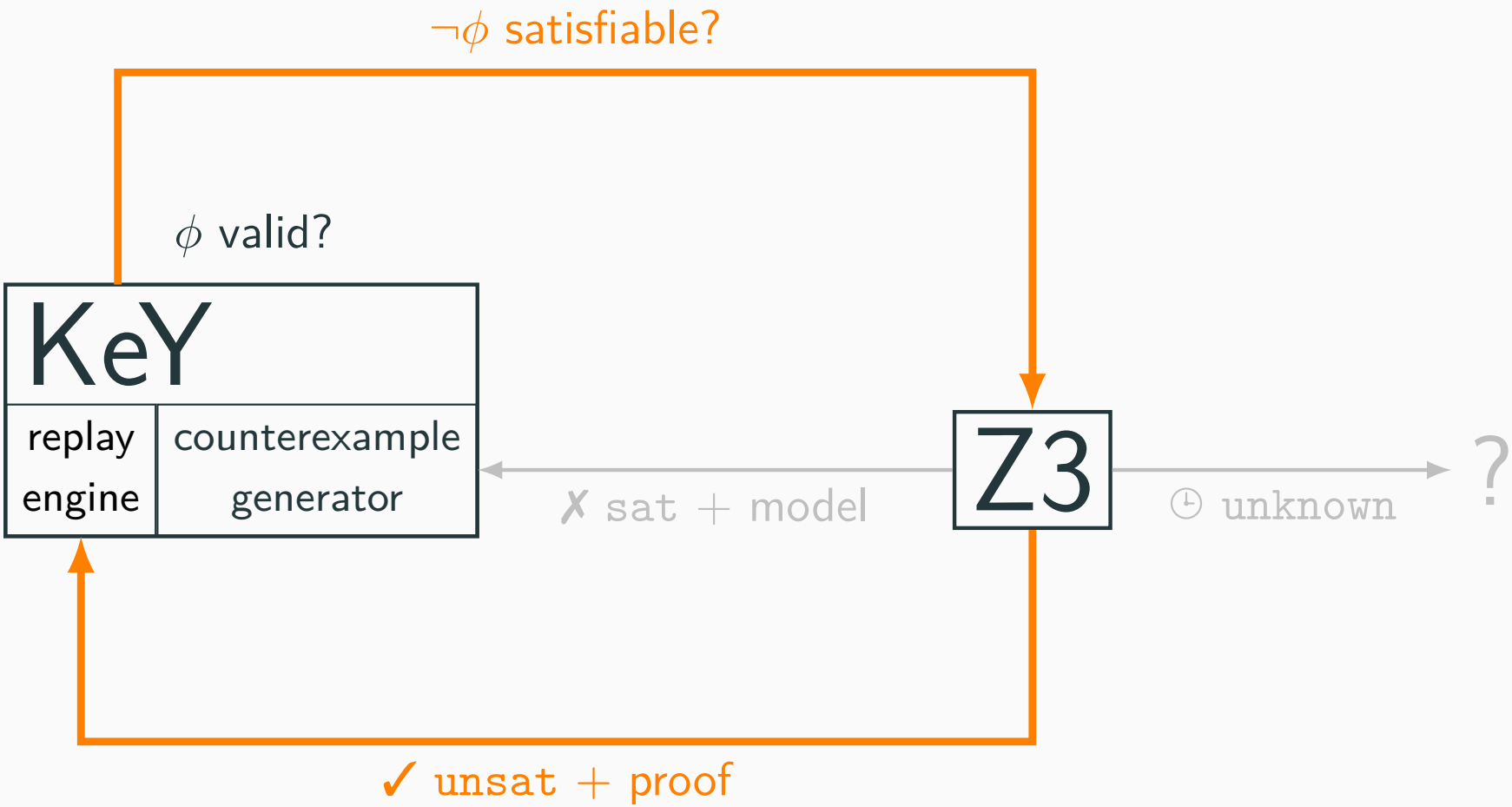
$\Gamma = \{\neg(p \vee \mathit{true})\}$

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

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

KeY (ϕ is valid):

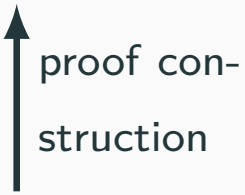
$$\dfrac{\dfrac{}{\Longrightarrow p, \mathit{true}}\,\mathit{closeTrue}}{\Longrightarrow p \vee \mathit{true}}\,\mathit{orRight}$$

Problem: Different Proof Directions

$\phi = p \vee true$

KeY (ϕ is valid):

$$\dfrac{\dfrac{}{\Longrightarrow p, true}\, closeTrue}{\Longrightarrow p \vee true}\, orRight$$

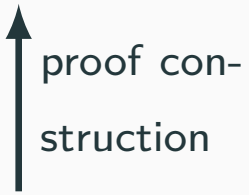


Problem: Different Proof Directions

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

KeY (ϕ is valid):

$$\dfrac{\dfrac{}{\Longrightarrow p, \mathit{true}}\,\mathit{closeTrue}}{\Longrightarrow p \vee \mathit{true}}\,\mathit{orRight}$$



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

$$\dfrac{\dfrac{\texttt{asserted}\,\dfrac{}{\Gamma \vdash \neg(p \vee \mathit{true})}}{\texttt{not-or-elim}\,\dfrac{}{\Gamma \vdash \neg \mathit{true}}} \qquad \dfrac{}{\vdash \neg \mathit{true} = \mathit{false}}\,\texttt{rewrite}}{\Gamma \vdash \mathit{false}}\,\texttt{mp}$$

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

Problem: Different Proof Directions

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

KeY (ϕ is valid):

$$\dfrac{\dfrac{}{\Longrightarrow p, \mathit{true}}\,\mathit{closeTrue}}{\Longrightarrow p \vee \mathit{true}}\,\mathit{orRight}$$

proof construction

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

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \neg(p \vee \mathit{true})}\,\texttt{asserted}}{\Gamma \vdash \neg\mathit{true}}\,\texttt{not-or-elim} \qquad \dfrac{}{\vdash \neg\mathit{true} = \mathit{false}}\,\texttt{rewrite}}{\Gamma \vdash \mathit{false}}\,\texttt{mp}$$

proof construction

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

Problem: Different Proof Directions

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

KeY (ϕ is valid):

$$\dfrac{\dfrac{?}{\neg(p \vee \mathit{true}) \Longrightarrow \mathit{false}}\,?}{\Longrightarrow p \vee \mathit{true}}$$

proof construction

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

$$\dfrac{\dfrac{\texttt{asserted}\dfrac{}{\Gamma \vdash \neg(p \vee \mathit{true})}}{\texttt{not-or-elim}\dfrac{}{\Gamma \vdash \neg\mathit{true}}} \qquad \dfrac{}{\vdash \neg\mathit{true} = \mathit{false}}\,\texttt{rewrite}}{\Gamma \vdash \mathit{false}}\,\texttt{mp}$$

proof construction

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

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\dfrac{\vdash p \qquad \vdash p \to q}{\vdash q}\ \mathtt{mp}$$

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\frac{\vdash p \qquad \vdash p \to q}{\vdash q}\ \mathtt{mp}$$

Replay in KeY:

$$\dfrac{\dfrac{\dfrac{L}{p, p \to q \Longrightarrow q}}{p \wedge (p \to q) \Longrightarrow q}\ \mathit{andLeft} \qquad \dfrac{\dfrac{\dfrac{\Longrightarrow p \qquad \Longrightarrow p \to q}{\Longrightarrow p \wedge (p \to q)}\ \mathit{andRight}}{\Longrightarrow p \wedge (p \to q), q}\ \mathit{hideRight}}{\Longrightarrow q}\ \mathit{cut}$$

The single rule application is justified in L.

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\frac{\vdash p \qquad \vdash p \to q}{\vdash q}\ \mathtt{mp}$$

Replay in KeY:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p, q \Longrightarrow q}\ \text{close}}{p, \mathit{true} \to q \Longrightarrow q}\ \text{simplify}}{p, p \to q \Longrightarrow q}\ \mathit{repl\_known\_left}}{p \wedge (p \to q) \Longrightarrow q}\ \mathit{andLeft} \qquad \dfrac{\dfrac{\Longrightarrow p \qquad \Longrightarrow p \to q}{\Longrightarrow p \wedge (p \to q)}\ \mathit{andRight}}{\Longrightarrow p \wedge (p \to q), q}\ \mathit{hideRight}$$

$$\frac{p \wedge (p \to q) \Longrightarrow q \qquad \Longrightarrow p \wedge (p \to q), q}{\Longrightarrow q}\ \mathit{cut}$$

The single rule application is justified in L.

Solution: Invert Direction Locally with Cut

Z3 rule (contexts omitted for readability):

$$\frac{\vdash p \qquad \vdash p \to q}{\vdash q}\ \mathtt{mp}$$

Replay in KeY:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p, q \Longrightarrow q}\ \text{close}}{p, \mathit{true} \to q \Longrightarrow q}\ \text{simplify}}{p, p \to q \Longrightarrow q}\ \mathit{repl\_known\_left}}{p \wedge (p \to q) \Longrightarrow q}\ \mathit{andLeft} \qquad \dfrac{\dfrac{\Longrightarrow p \qquad \Longrightarrow p \to q}{\Longrightarrow p \wedge (p \to q)}\ \mathit{andRight}}{\Longrightarrow p \wedge (p \to q), q}\ \mathit{hideRight}$$

$$\frac{p \wedge (p \to q) \Longrightarrow q \qquad \Longrightarrow p \wedge (p \to q), q}{\Longrightarrow q}\ \mathit{cut}$$

The single rule application is justified in L.

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

Replayed Proof Example

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

Z3 proof:

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \neg(p \vee \mathit{true})}\,\texttt{asserted}}{\Gamma \vdash \neg\mathit{true}}\,\texttt{not-or-elim} \qquad \dfrac{}{\vdash \neg\mathit{true} = \mathit{false}}\,\texttt{rewrite}}{\Gamma \vdash \mathit{false}}\,\texttt{mp}$$

Replayed Proof Example

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

Z3 proof:

$$\dfrac{\dfrac{}{\Gamma \vdash \neg(p \vee \mathit{true})}\,\texttt{asserted}}{\Gamma \vdash \neg \mathit{true}}\,\texttt{not-or-elim} \qquad \dfrac{}{\vdash \neg \mathit{true} = \mathit{false}}\,\texttt{rewrite}$$
$$\dfrac{\Gamma \vdash \neg \mathit{true} \qquad \vdash \neg \mathit{true} = \mathit{false}}{\Gamma \vdash \mathit{false}}\,\texttt{mp}$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \neg \mathit{true}, \mathit{false} \Longrightarrow \mathit{false}}\,\mathit{close}}{\Gamma, \neg \mathit{true}, \mathit{true} \leftrightarrow \mathit{false} \Longrightarrow \mathit{false}}\,\mathit{concrete\_eq}}{\Gamma, \neg \mathit{true}, \neg \mathit{true} \leftrightarrow \mathit{false} \Longrightarrow \mathit{false}}\,\mathit{repl\_known\_left}}{\Gamma, \neg \mathit{true} \wedge (\neg \mathit{true} \leftrightarrow \mathit{false}) \Longrightarrow \mathit{false}}\,\mathit{andLeft}}{\Gamma \Longrightarrow \mathit{false}}\,\mathit{cut} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \mathit{true} \Longrightarrow p, \mathit{true}}\,\mathit{close}}{\Gamma \Longrightarrow p, \mathit{true}, \neg \mathit{true}}\,\mathit{notRight}}{\Gamma \Longrightarrow p \vee \mathit{true}, \neg \mathit{true}}\,\mathit{orRight}}{\Gamma, \neg(p \vee \mathit{true}) \Longrightarrow \neg \mathit{true}}\,\mathit{notLeft}}{\Gamma \Longrightarrow \neg(p \vee \mathit{true}), \neg \mathit{true}}\,\mathit{hideRight} \qquad \dfrac{}{\Gamma \Longrightarrow \neg(p \vee \mathit{true})}\,\mathit{close}}{\Gamma \Longrightarrow \neg \mathit{true}}\,\mathit{cut} \qquad \dfrac{\vdots}{\Gamma \Longrightarrow \neg \mathit{true} \leftrightarrow \mathit{false}}\,\mathit{auto\ mode}}{\Gamma \Longrightarrow \neg \mathit{true} \wedge (\neg \mathit{true} \leftrightarrow \mathit{false})}\,\mathit{andRight}$$
$$\dfrac{\Gamma \Longrightarrow \neg \mathit{true} \wedge (\neg \mathit{true} \leftrightarrow \mathit{false})}{\Gamma \Longrightarrow \neg \mathit{true} \wedge (\neg \mathit{true} \leftrightarrow \mathit{false}), \mathit{false}}\,\mathit{hideRight} \qquad \mathit{cut}$$
$$\dfrac{\Gamma \Longrightarrow \mathit{false}}{\Longrightarrow p \vee \mathit{true}}\,\mathit{notRight}$$

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.

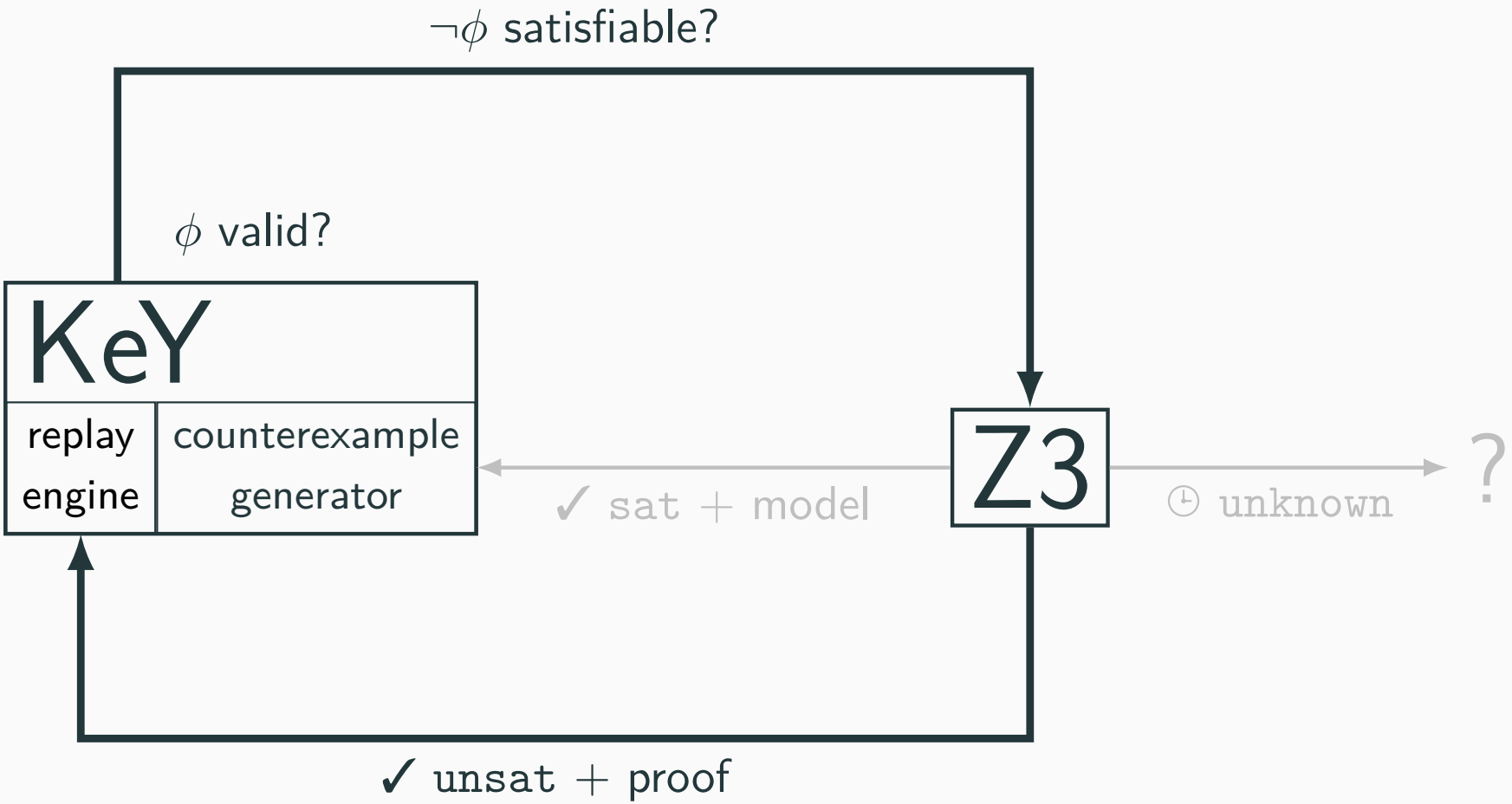
Currently violated:

(A) The type hierarchy axiomatisation introduces ***typeof*** **and** ***subtype***.

(B) Z3 terms may contain an explicit **equisatisfiability** relation.

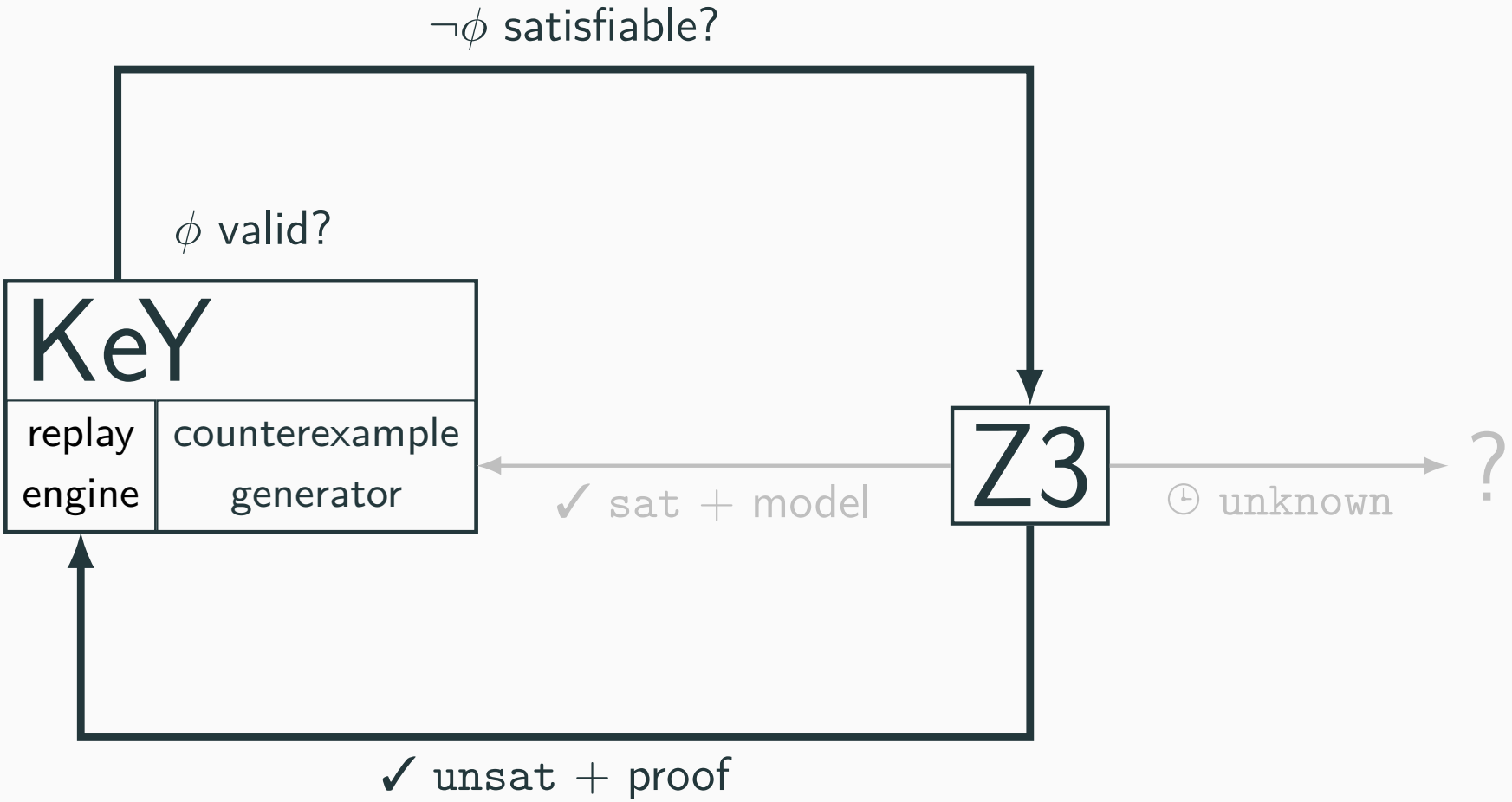
(C) Z3 may introduce **skolem functions**.

Replay Technique: Overview



Replay Technique: Overview

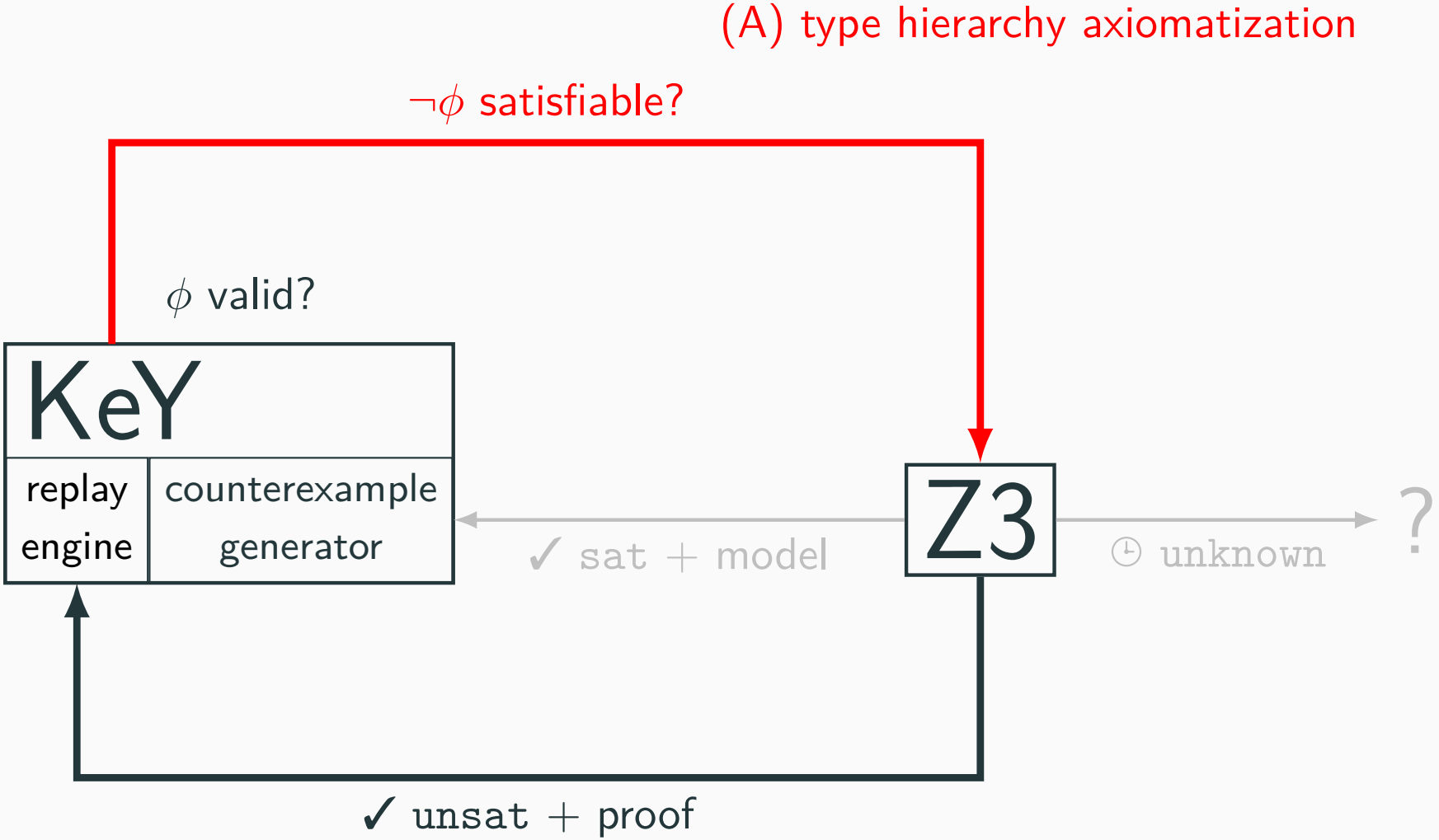
(A) type hierarchy axiomatization



(B) equisatisfiability relation

(C) skolem functions

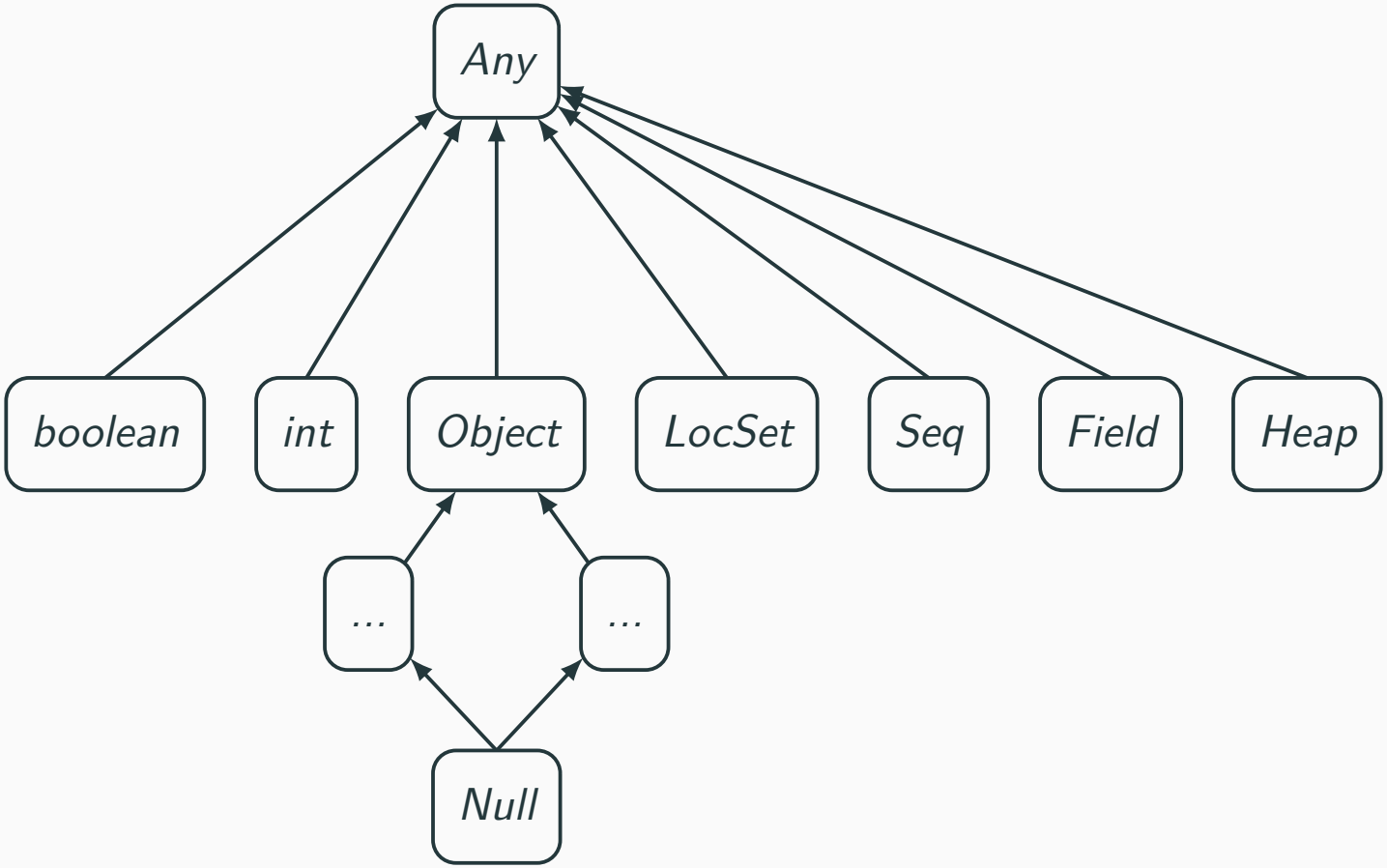
Replay Technique: Overview



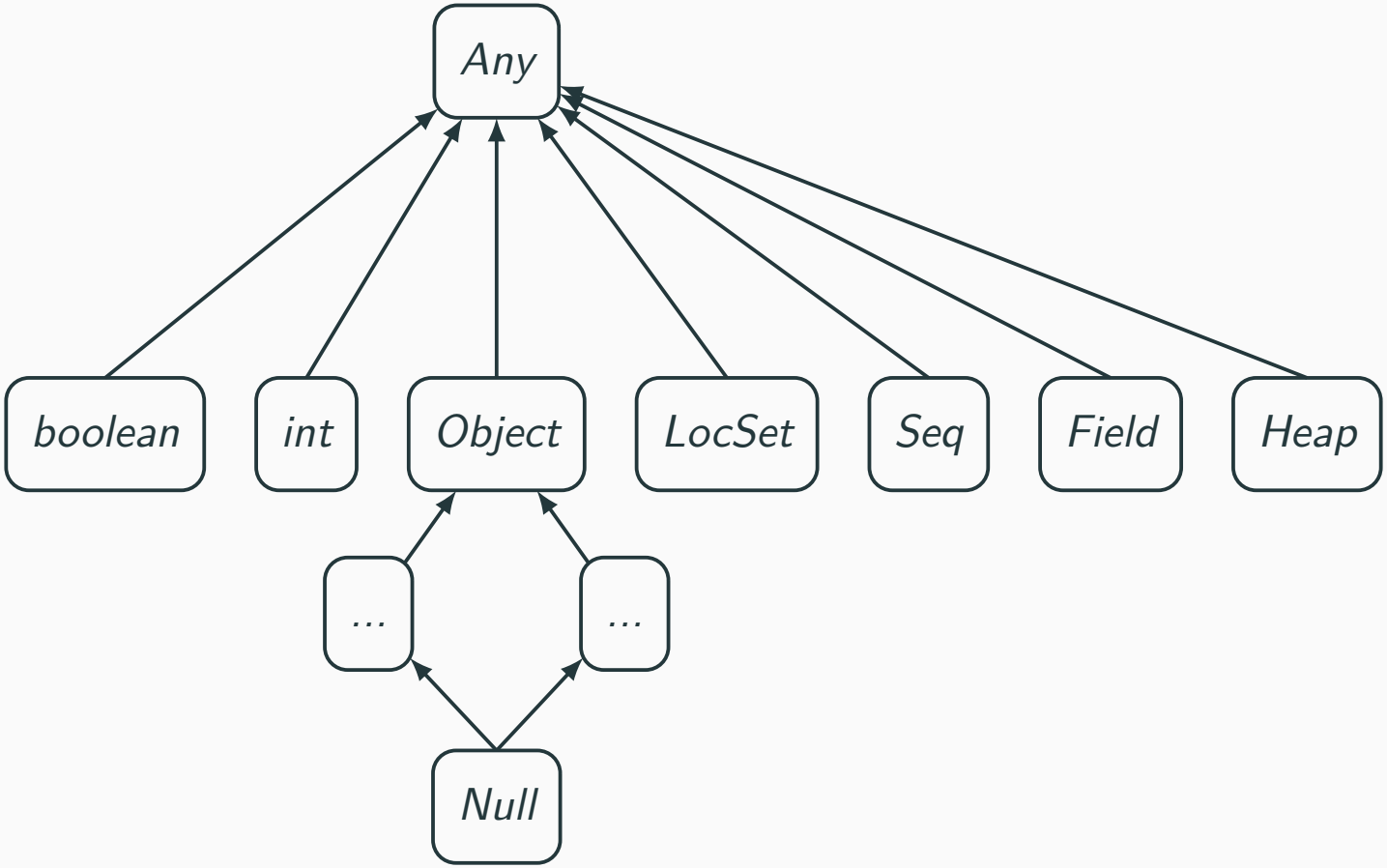
(B) equisatisfiability relation

(C) skolem functions

KeY Type Hierarchy

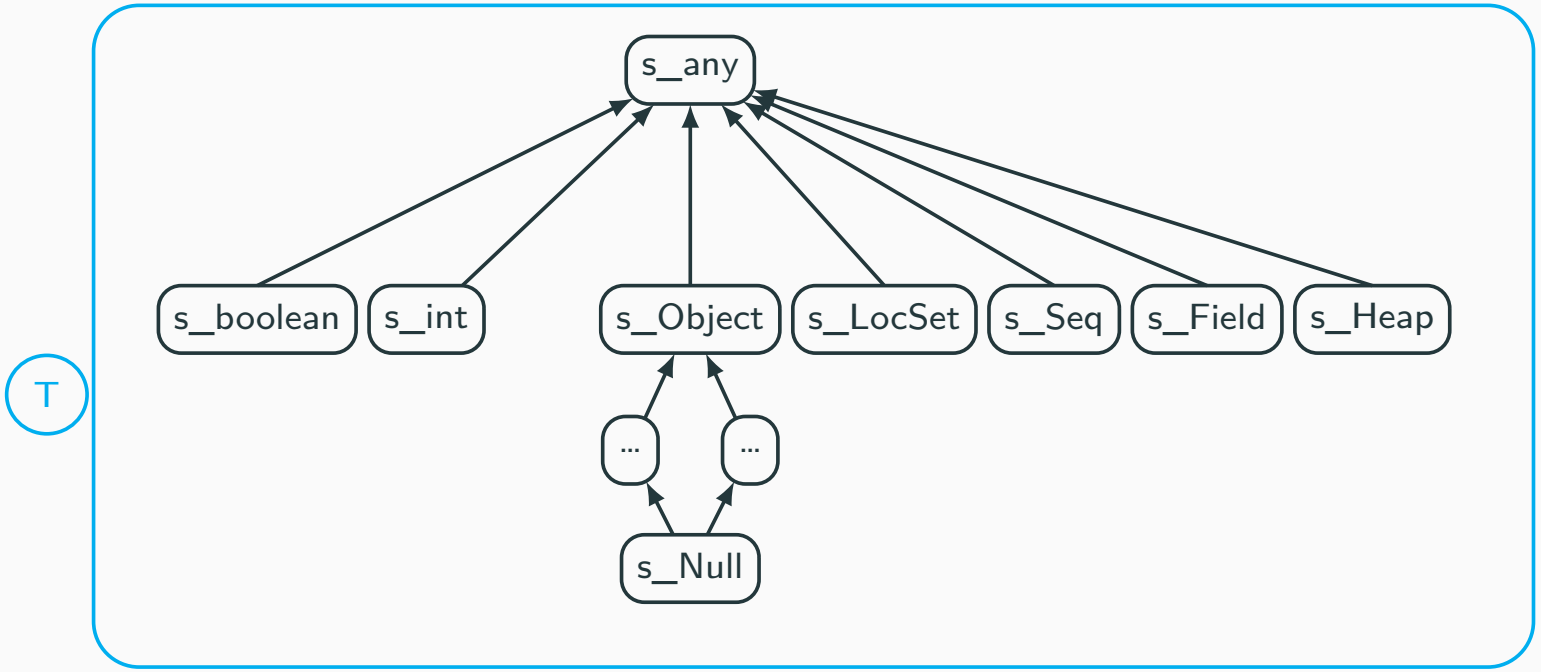


KeY Type Hierarchy



In contrast to that, Z3 has no inheritance.

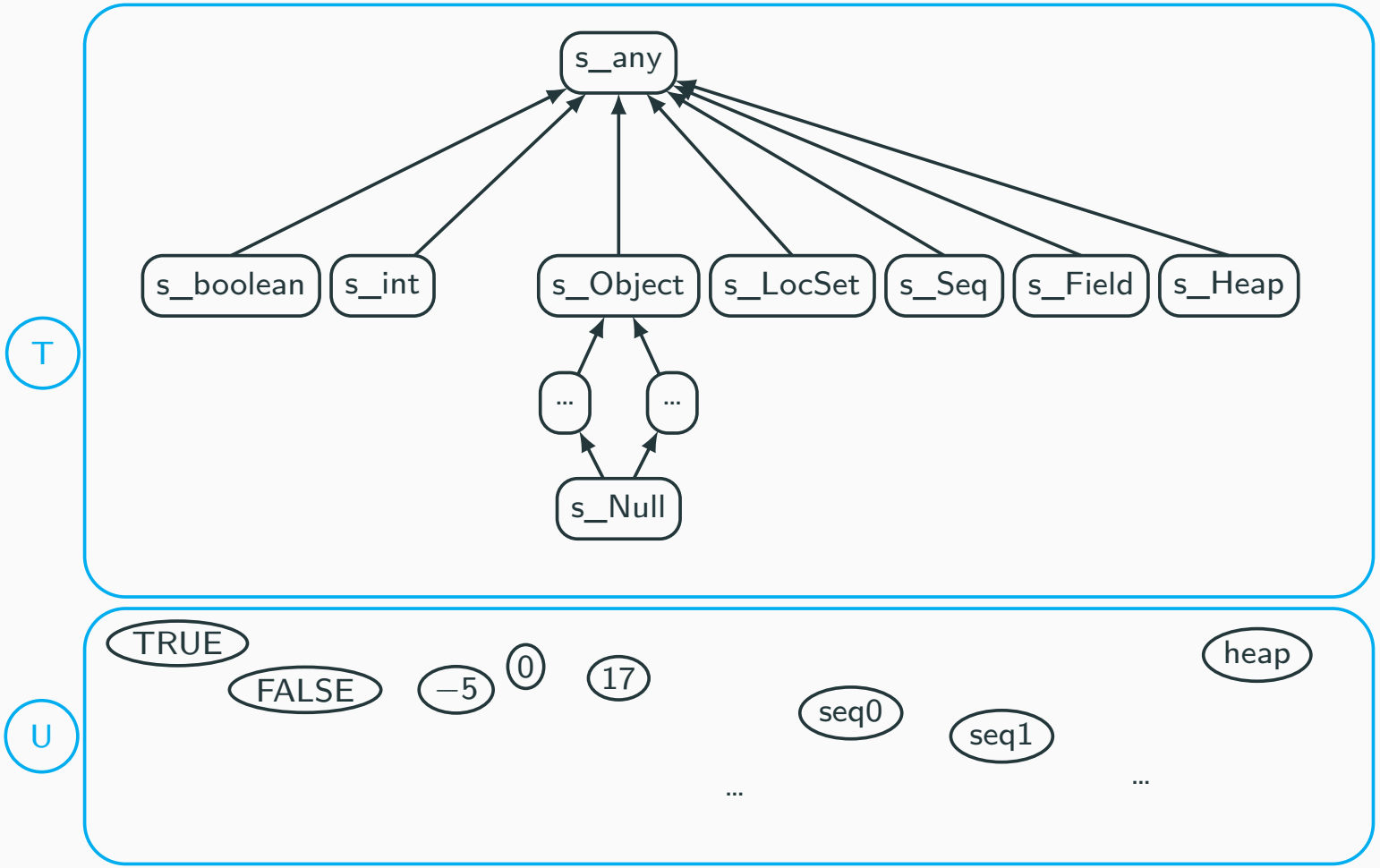
Existing Type Hierarchy Translation



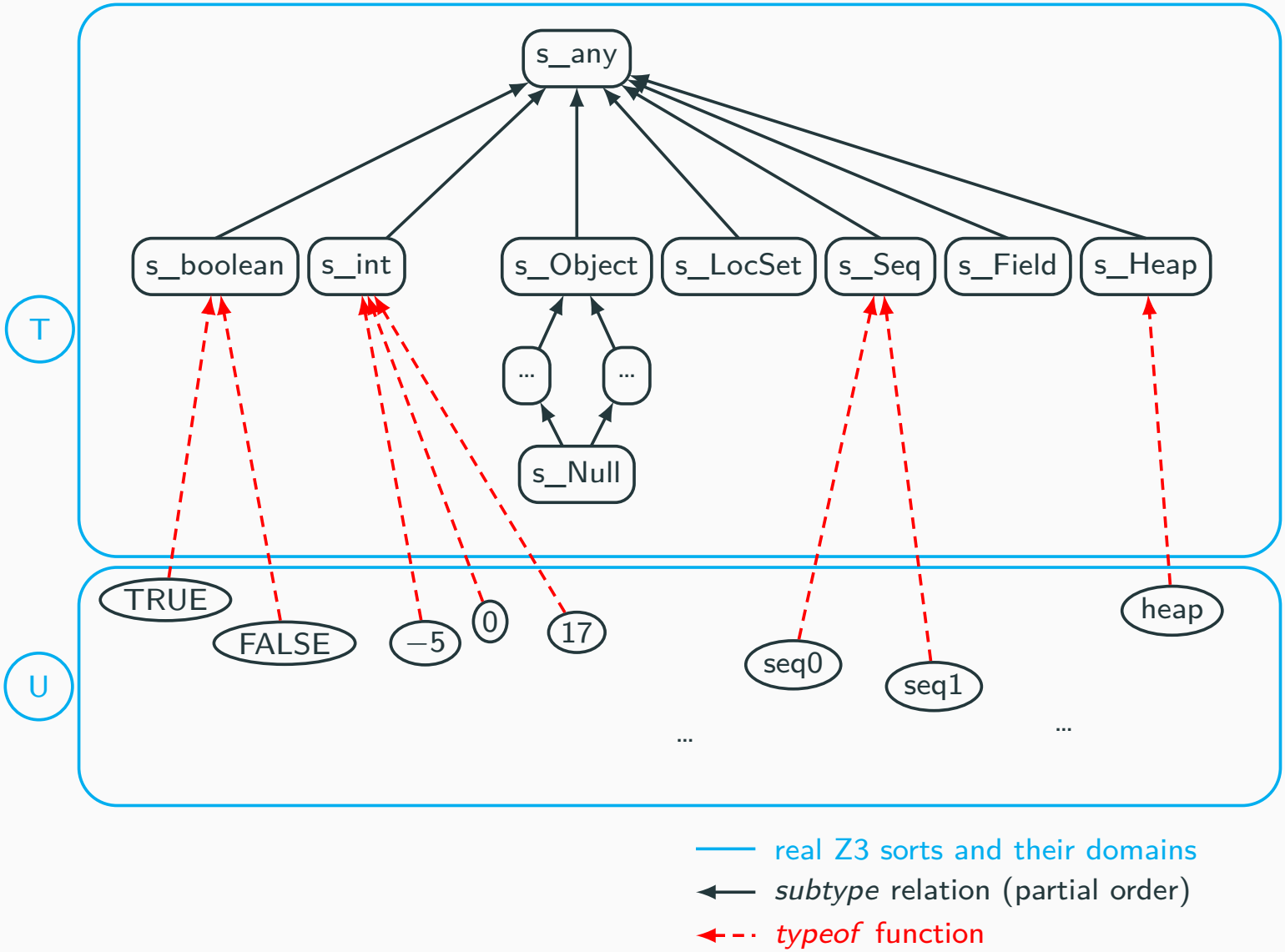
real Z3 sorts and their domains

subtype relation (partial order)

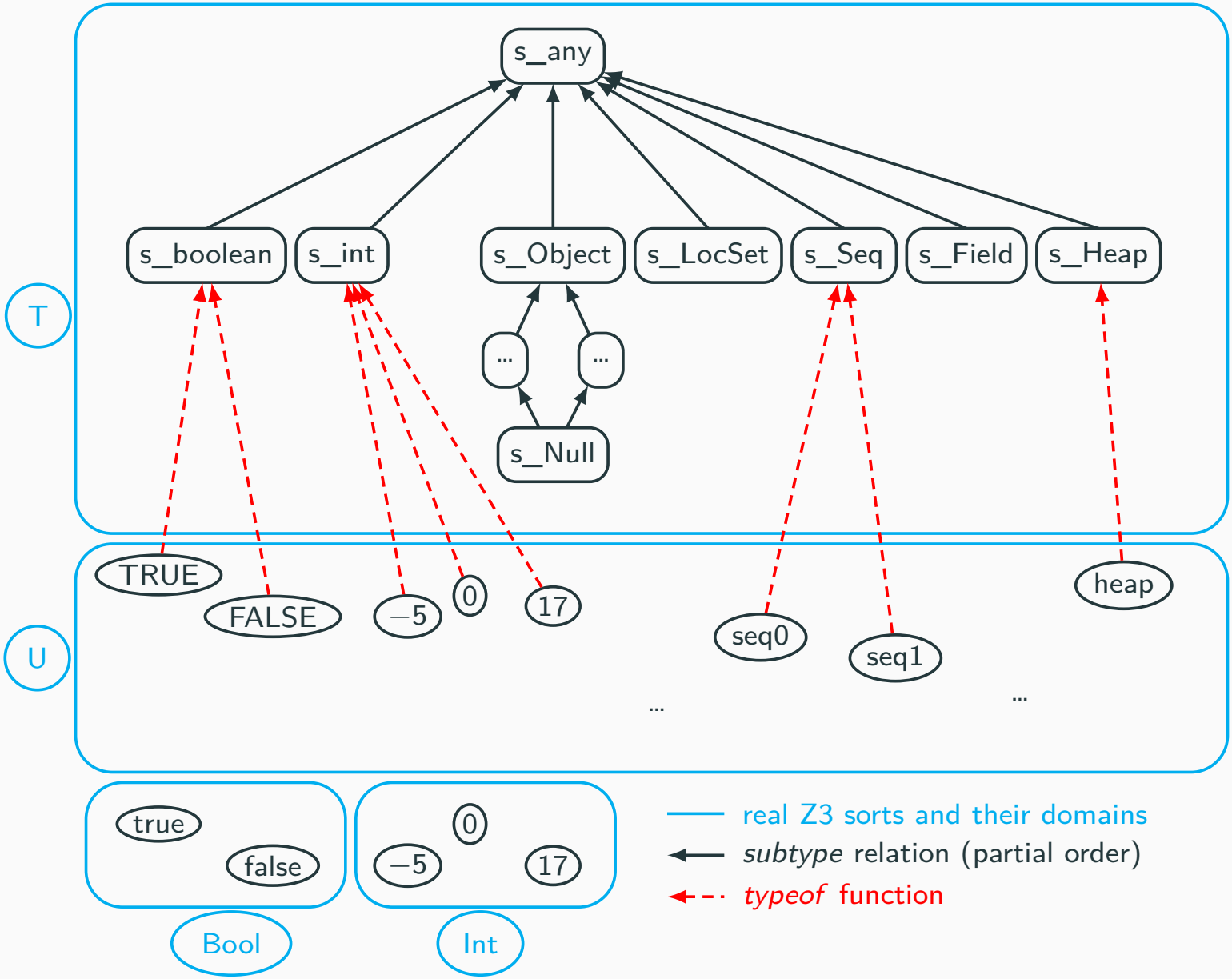
Existing Type Hierarchy Translation



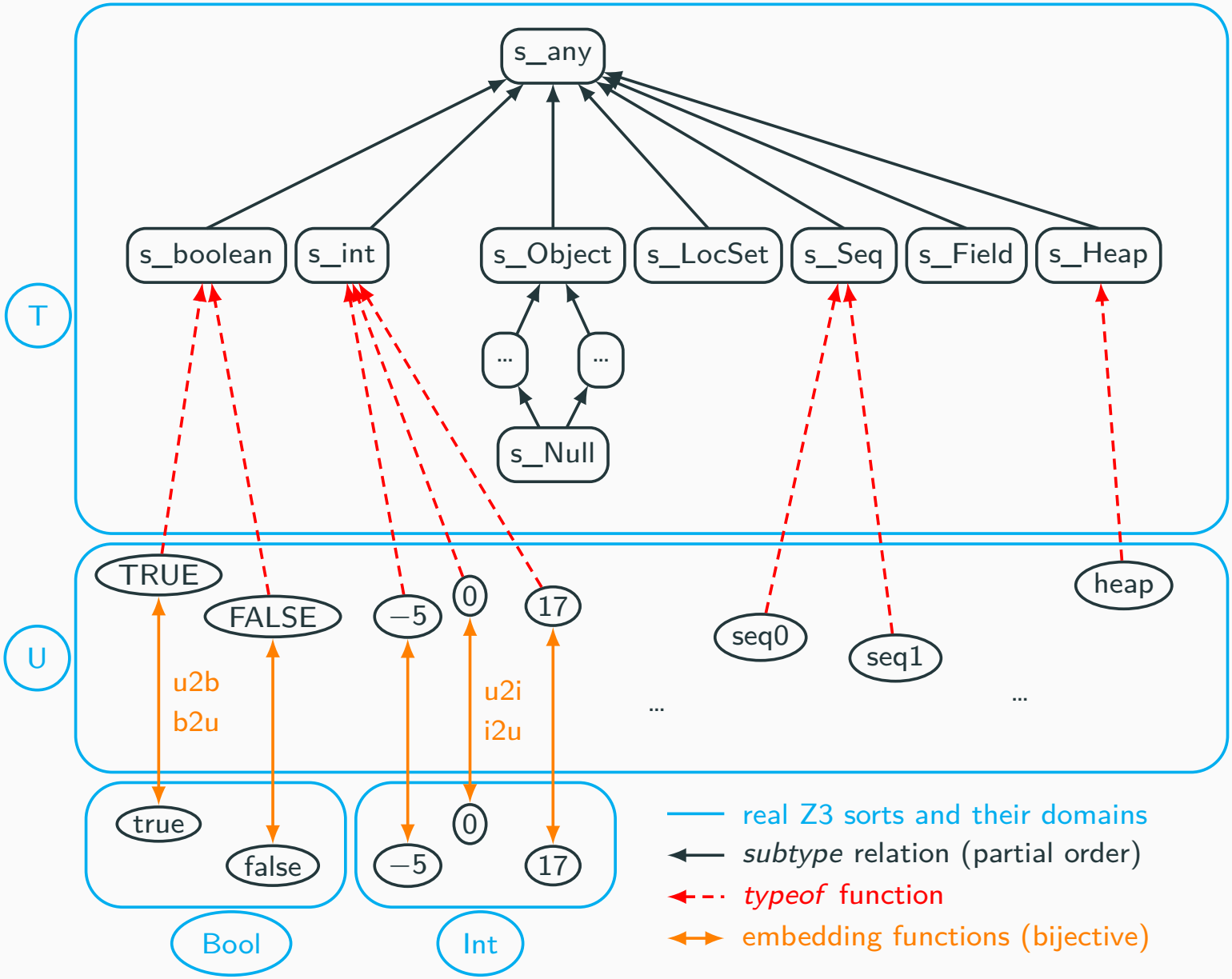
Existing Type Hierarchy Translation



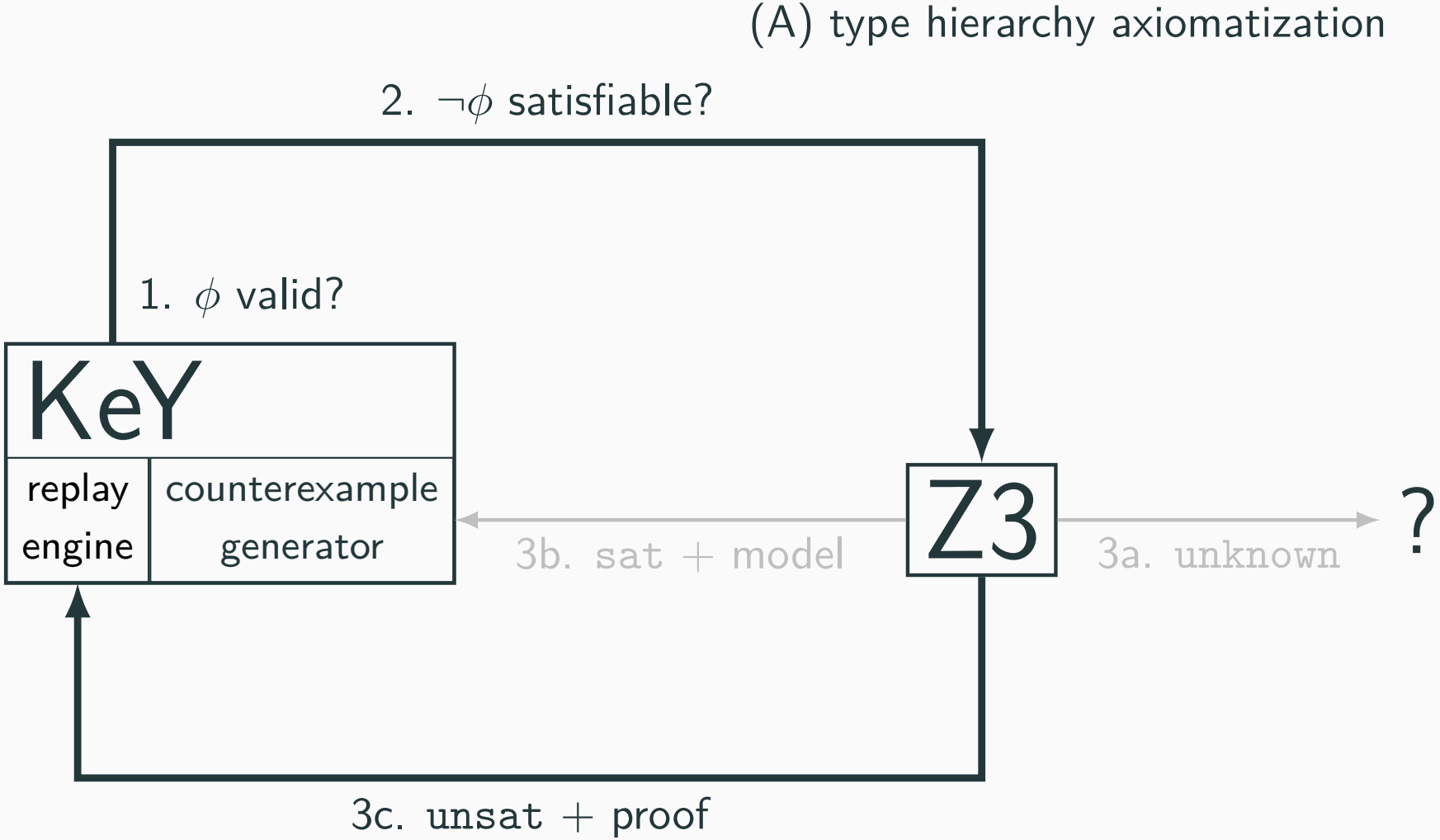
Existing Type Hierarchy Translation



Existing Type Hierarchy Translation

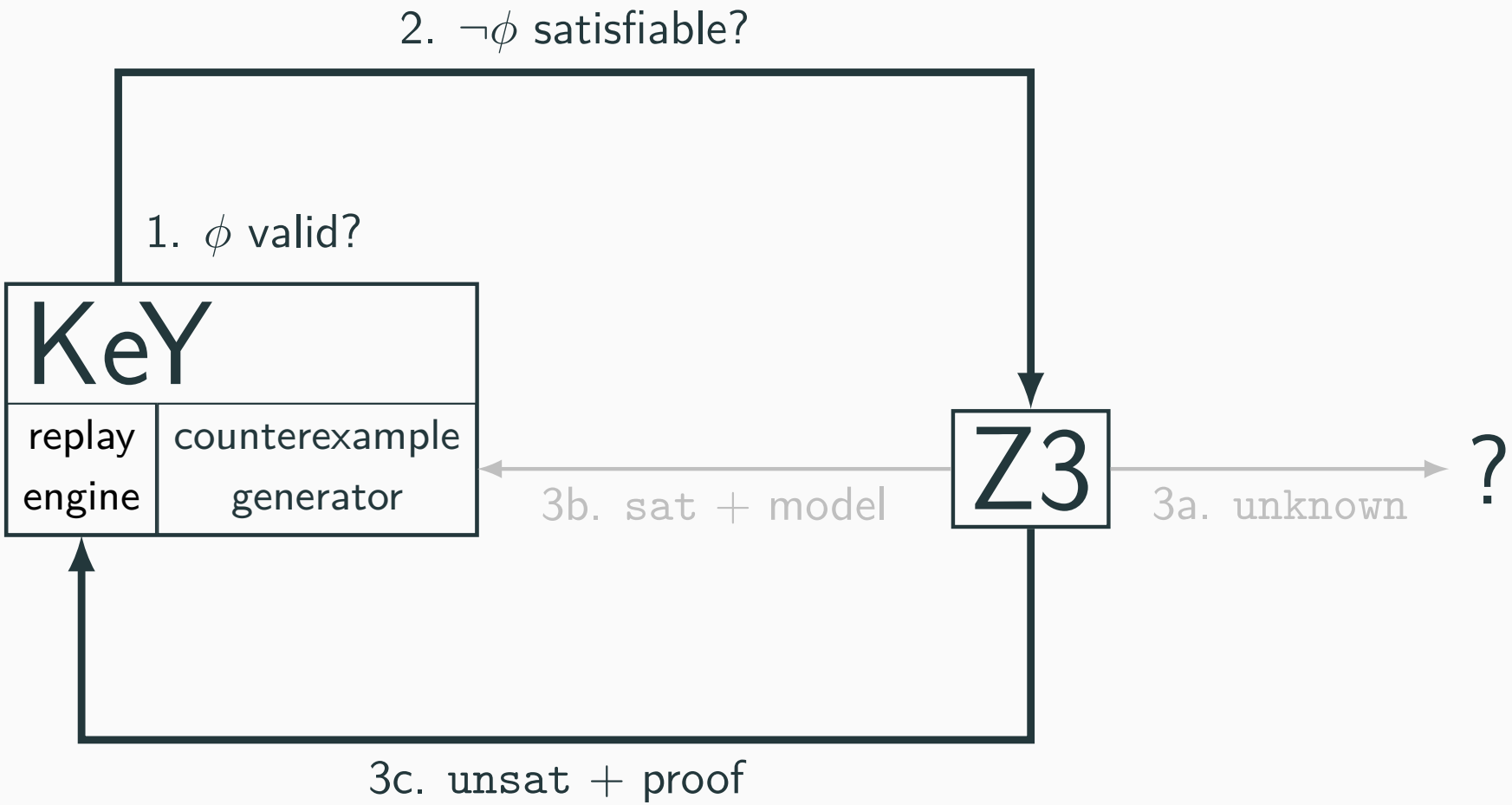


Replay Technique: Overview



Replay Technique: Overview

(A) type hierarchy axiomatization ✓

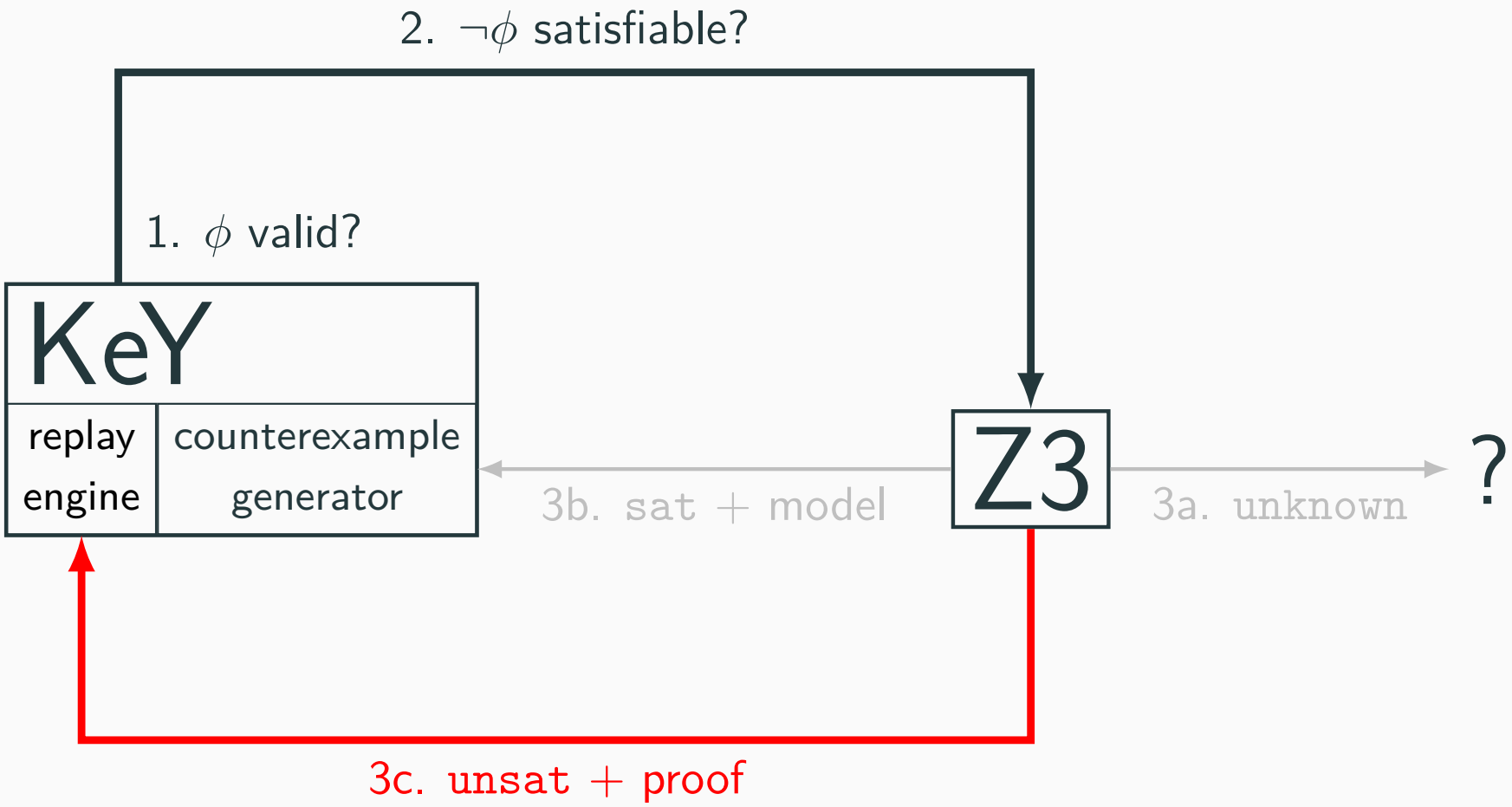


(B) equisatisfiability relation

(C) skolem functions

Replay Technique: Overview

(A) type hierarchy axiomatization ✓



(B) equisatisfiability relation

(C) skolem functions

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

Skolemization

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

$$\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \sim \phi(s)}\ \texttt{sk}}{\dfrac{\vdots}{\dfrac{\ldots \qquad\qquad \ldots}{\dfrac{\Gamma \vdash \psi(s)}{\vdots}}}}$$

Skolemization

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

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \sim \phi(s)}\,\mathtt{sk}}{\vdots}\,\ldots}{\dfrac{\ldots \qquad \vdots \qquad \ldots}{\dfrac{\Gamma \vdash \psi(s)}{\vdots}\,\ldots}\,\ldots}$$

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

Skolemization

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

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \sim \phi(s)}\,\texttt{sk}}{\vdots}\,\ldots}{\dfrac{\ldots \qquad \vdots \qquad \ldots}{\dfrac{\Gamma \vdash \psi(s)}{\vdots}\,\ldots}\,\ldots}$$

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

Defining axiom:

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

Skolemization Using Hilbert Choice Operator

For a skolem constant s:

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \sim \phi(s)}\,\mathtt{sk}}{\vdots}\,\dots \quad \dots}{\Gamma \vdash \psi(s)}\,\dots \qquad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \sim \phi(\epsilon x.\ \phi(x))}\,\mathtt{sk}}{\vdots}\,\dots \quad \dots}{\Gamma \vdash \psi(\epsilon x.\ \phi(x))}\,\dots$$

Skolemization Using Hilbert Choice Operator

For a skolem constant s:

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \sim \phi(s)}\,\mathtt{sk}}{\vdots}\,\dots \quad \cdots \quad \cdots}{\dfrac{\Gamma \vdash \psi(s)}{\vdots}\,\dots}\,\dots$$

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \leftrightarrow \phi(\epsilon x.\ \phi(x))}\,\mathtt{sk}}{\vdots}\,\dots \quad \cdots \quad \cdots}{\dfrac{\Gamma \vdash \psi(\epsilon x.\ \phi(x))}{\vdots}\,\dots}\,\dots$$

Skolemization Using Hilbert Choice Operator

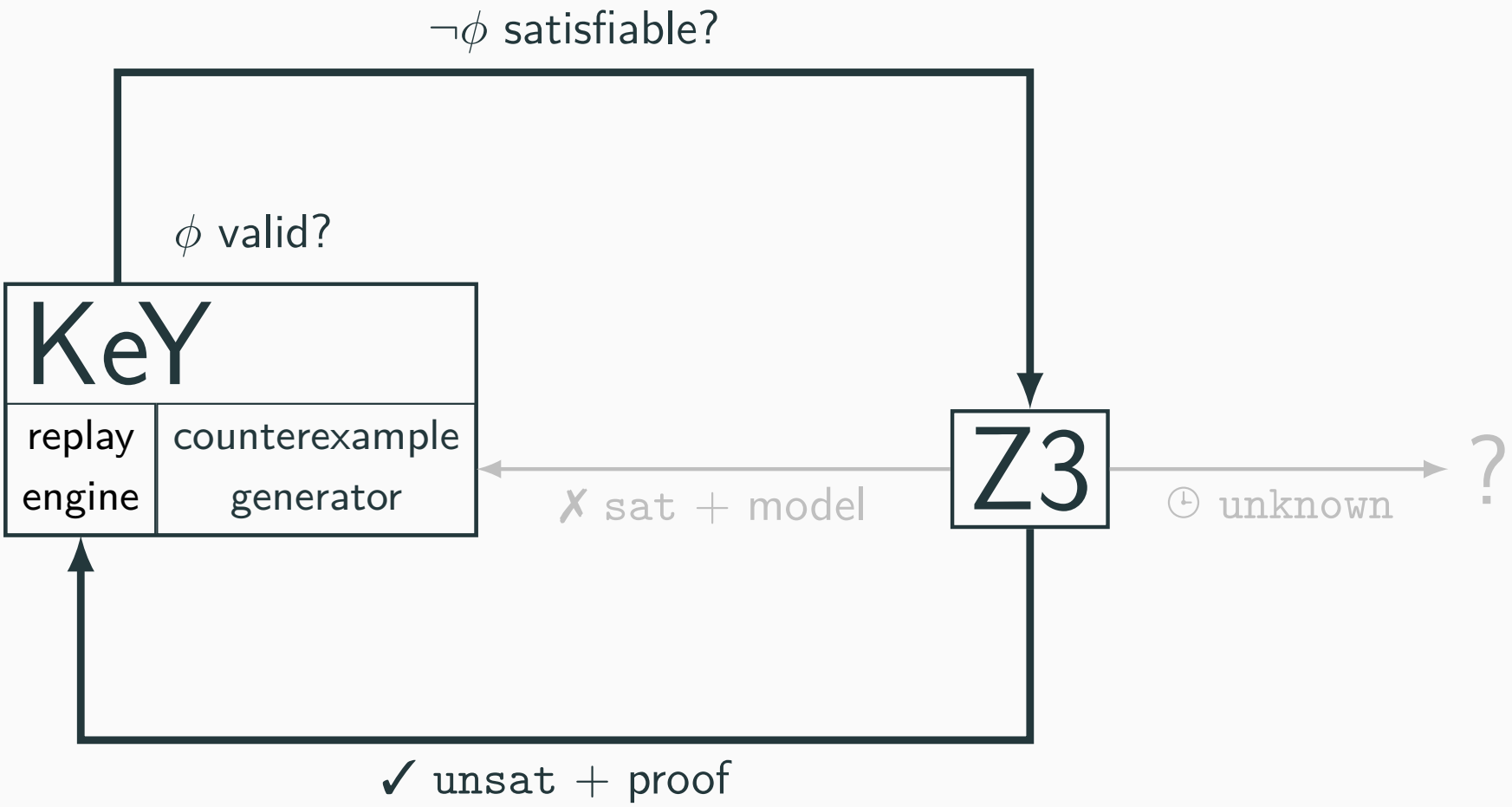
For a skolem constant s:

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \sim \phi(s)}\,\mathtt{sk}}{\vdots}\,\ldots \qquad \ldots}{\dfrac{\Gamma \vdash \psi(s)}{\vdots}\,\ldots}\,\ldots \qquad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x)) \leftrightarrow \phi(\epsilon x.\ \phi(x))}\,\mathtt{sk}}{\vdots}\,\ldots \qquad \ldots}{\dfrac{\Gamma \vdash \psi(\epsilon x.\ \phi(x))}{\vdots}\,\ldots}\,\ldots$$

This works also for skolem functions!

Replay Technique: Overview

(A) type hierarchy axiomatization ✓

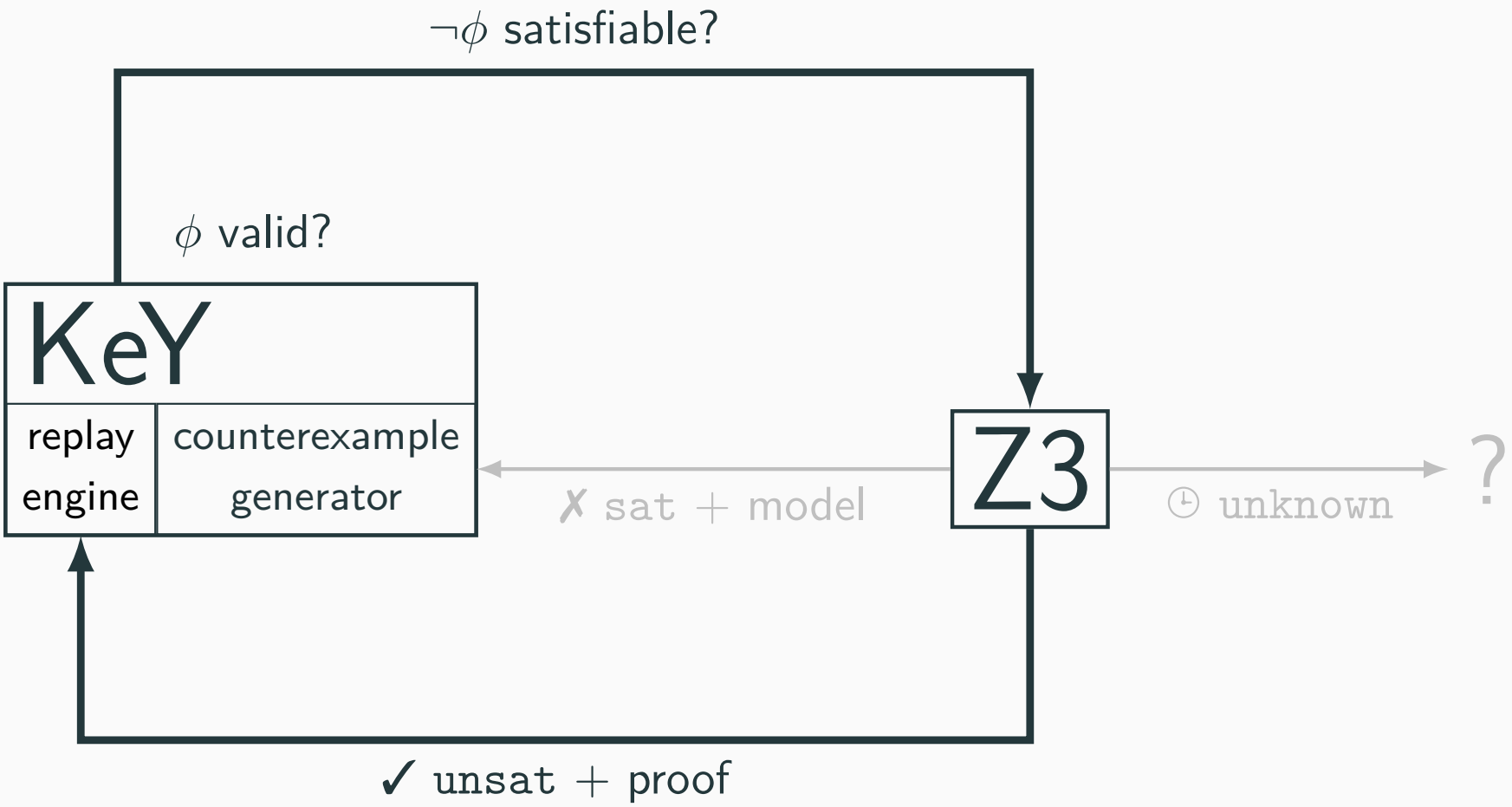


(B) equisatisfiability relation

(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

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			
		Transl. + Z3	Replay	(Auto)	Total
`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	Z3			Replay	
	Steps	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

Evaluation 2

There are formulas where KeY does not find a proof!

Evaluation 2

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))))        ;    ¬(p ∨ ¬p ∨ ¬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))))        ;    ¬(p ∨ ¬p ∨ ¬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:

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

Skolem Functions

The skolem symbol s depends on the free variable y:

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

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (\exists x.\ \phi(x,y)) \leftrightarrow \phi(\boxed{\epsilon x.\ \phi(x,y)})}\,\texttt{sk}}{\vdots}\,\dots \quad \dots \quad \dots}{\dfrac{\dfrac{\Gamma \vdash \chi(y) \leftrightarrow \rho(y)}{\Gamma \vdash Q\boxed{y}.\,\chi(y) \leftrightarrow Q\boxed{y}.\,\rho(y)}\,\texttt{quant-intro}}{\vdots}\,\dots \quad \dots \quad \dots}$$
$$\Gamma \vdash \psi(\boxed{[y/t]}\,\boxed{\epsilon x.\ \phi(x,y)})$$

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

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

$$\dfrac{\dfrac{?}{\Longrightarrow \exists x.\phi(x), \exists x.(\phi(x) \rightarrow \psi(x))}\,? \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\psi(c), \phi(c) \Longrightarrow \psi(c)}\,\text{close}}{\psi(c) \Longrightarrow \phi(c) \rightarrow \psi(c)}\,\text{impLeft}}{\psi(c) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{exRight}}{\exists x.\psi(x) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{exLeft}}{\exists x.\phi(x) \rightarrow \exists x.\psi(x) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{impLeft}$$

$$\dfrac{\exists x.\phi(x) \rightarrow \exists x.\psi(x) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}{\Longrightarrow (\exists x.\phi(x) \rightarrow \exists x.\psi(x)) \rightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{impRight}$$

Formulas Where KeY does not find a proof automatically

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

$$\dfrac{\dfrac{?}{\Longrightarrow \exists x.\phi(x), \exists x.(\phi(x) \rightarrow \psi(x))}\,? \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\psi(c), \phi(c) \Longrightarrow \psi(c)}\,\text{close}}{\psi(c) \Longrightarrow \phi(c) \rightarrow \psi(c)}\,\text{impLeft}}{\psi(c) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{exRight}}{\exists x.\psi(x) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{exLeft}}{\exists x.\phi(x) \rightarrow \exists x.\psi(x) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{impLeft}$$

$$\dfrac{\exists x.\phi(x) \rightarrow \exists x.\psi(x) \Longrightarrow \exists x.(\phi(x) \rightarrow \psi(x))}{\Longrightarrow (\exists x.\phi(x) \rightarrow \exists x.\psi(x)) \rightarrow \exists x.(\phi(x) \rightarrow \psi(x))}\,\text{impRight}$$

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)

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

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

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

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

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

Problem: No ground term available for quantifier instantiation.

Existing Type Hierarchy Axiomatization

subtype relation is a partial order:

reflexive:

$$\forall t : T.\ subtype(t, t)$$

antisymmetric:

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

transitive:

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

cast:

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

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

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

sort "definition":

$$instanceof(t, u)$$

for every subsort c_i:

$$subtype(c_i, s)$$

Existing Type Hierarchy Axiomatization

subtype relation is a partial order:

reflexive:

$$\forall t : T.\ subtype(t, t)$$

antisymmetric:

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

transitive:

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

cast:

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

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

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

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

sort "definition":

$$instanceof(t, u)$$

for every subsort c_i:

$$subtype(c_i, s)$$

New Type Hierarchy Axiomatization

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

sort "definition":

$$\forall u : U.\ exactInstanceof(s, u) \vee instanceof(c_1, u) \vee \ldots \vee instanceof(c_n, u) \rightarrow instanceof(s, u)$$

cast:

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

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

typeguard:

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

for every subsort c_i:

at most one type:

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

additional axiom for sort_any only:

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

$$\forall u : U.\ \neg instanceof(c_i, u) \vee \neg instanceof(c_j, u) \qquad (i \neq j)$$

additional axioms for object sorts only:

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

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

New Type Hierarchy Axiomatization

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

sort "definition":

$$\forall u : U.\ exactInstanceof(s, u) \vee instanceof(c_1, u) \vee \ldots \vee instanceof(c_n, u) \rightarrow instanceof(s, u)$$

cast:

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

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

typeguard:

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

for every subsort c_i:

at most one type:

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

additional axiom for sort_any only:

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

$$\forall u : U.\ \neg instanceof(c_i, u) \vee \neg instanceof(c_j, u) \qquad (i \neq j)$$

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

additional axioms for object sorts only:

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

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

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

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

$$\phi(\overline{x}) \sim \psi(\overline{y}) \quad \text{iff} \quad \text{for all } \beta\text{:} \quad \text{there is some } \mathscr{M}_1 : \ (\mathscr{M}_1, \beta) \vDash \phi(\overline{x})$$
$$\text{iff} \quad \text{there is some } \mathscr{M}_2 : \ (\mathscr{M}_2, \beta) \vDash \psi(\overline{y})$$

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

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

$$\phi(\overline{x}) \sim \psi(\overline{y}) \quad \text{iff} \quad \text{for all } \beta\text{:} \quad \text{there is some } \mathscr{M}_1 : \ (\mathscr{M}_1, \beta) \vDash \phi(\overline{x})$$
$$\text{iff} \quad \text{there is some } \mathscr{M}_2 : \ (\mathscr{M}_2, \beta) \vDash \psi(\overline{y})$$

Examples:

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

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

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

$$\phi(\overline{x}) \sim \psi(\overline{y}) \quad \text{iff} \quad \text{for all } \beta\text{:} \quad \text{there is some } \mathscr{M}_1 : \ (\mathscr{M}_1, \beta) \vDash \phi(\overline{x})$$
$$\text{iff} \quad \text{there is some } \mathscr{M}_2 : \ (\mathscr{M}_2, \beta) \vDash \psi(\overline{y})$$

Examples:

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

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

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

$$\phi(\overline{x}) \sim \psi(\overline{y}) \quad \text{iff} \quad \text{for all } \beta\text{:} \quad \text{there is some } \mathscr{M}_1 : \ (\mathscr{M}_1, \beta) \vDash \phi(\overline{x})$$
$$\text{iff} \quad \text{there is some } \mathscr{M}_2 : \ (\mathscr{M}_2, \beta) \vDash \psi(\overline{y})$$

Examples:

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

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

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

$$\phi(\overline{x}) \sim \psi(\overline{y}) \quad \text{iff} \quad \text{for all } \beta\text{:} \quad \text{there is some } \mathscr{M}_1 : \ (\mathscr{M}_1, \beta) \vDash \phi(\overline{x})$$
$$\text{iff} \quad \text{there is some } \mathscr{M}_2 : \ (\mathscr{M}_2, \beta) \vDash \psi(\overline{y})$$

Examples:

$$f(x) = 5 \sim \mathit{true} \qquad\qquad x = 5 \nsim x = 6$$
$$f(x) = 5 \wedge f(x) = 6 \sim \mathit{false}$$
$$f(x) = 5 \sim f(x) = 6$$

Equisatisfiability Relation

Explicit equisatisfiability relation ($\sim$) in Z3 terms:

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

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

$$\phi(\overline{x}) \sim \psi(\overline{y}) \quad \text{iff} \quad \text{for all } \beta\text{:} \quad \text{there is some } \mathscr{M}_1 : \ (\mathscr{M}_1, \beta) \vDash \phi(\overline{x})$$
$$\text{iff} \quad \text{there is some } \mathscr{M}_2 : \ (\mathscr{M}_2, \beta) \vDash \psi(\overline{y})$$

Examples:

$$f(x) = 5 \sim \mathit{true} \qquad x = 5 \nsim x = 6$$
$$f(x) = 5 \wedge f(x) = 6 \sim \mathit{false} \qquad x = 5 \nsim y = 6$$
$$f(x) = 5 \sim f(x) = 6$$