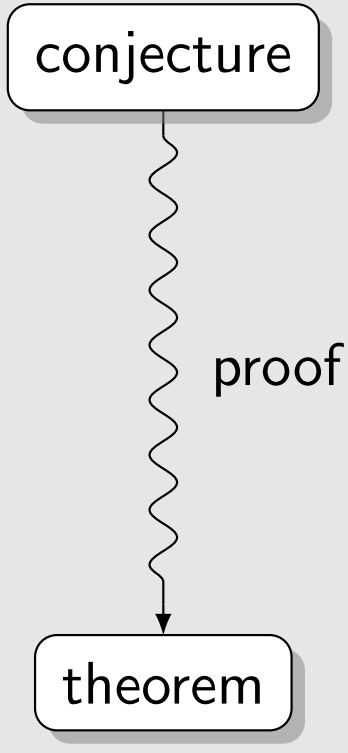
# Fast LCF-Style Proof Reconstruction for Z3

Sascha Böhme and Tjark Weber



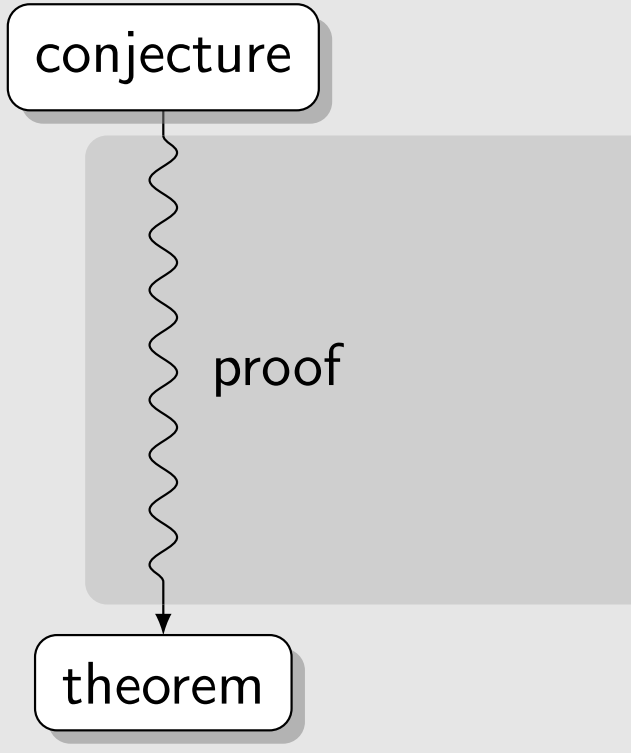
July 14, 2010

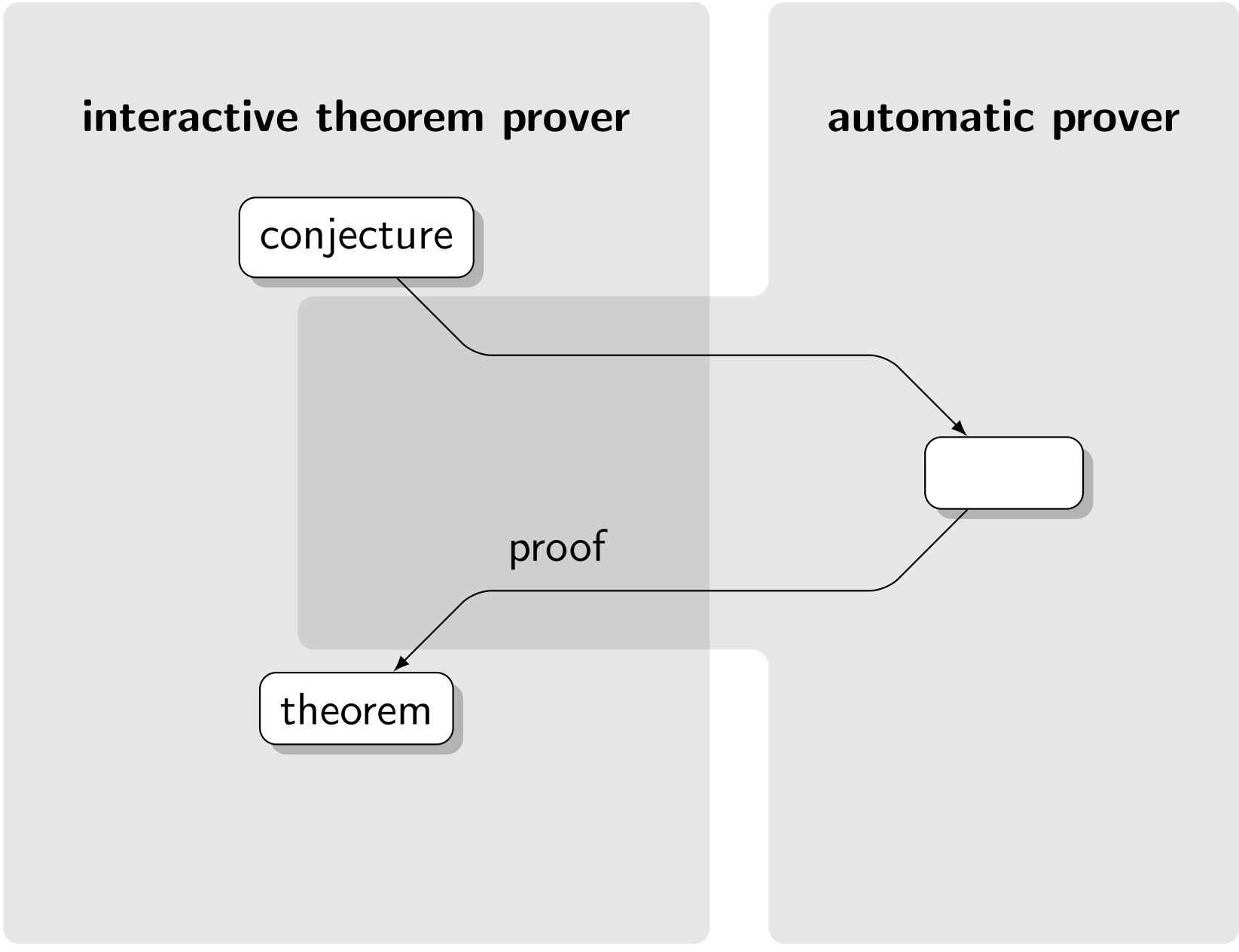
## interactive theorem prover

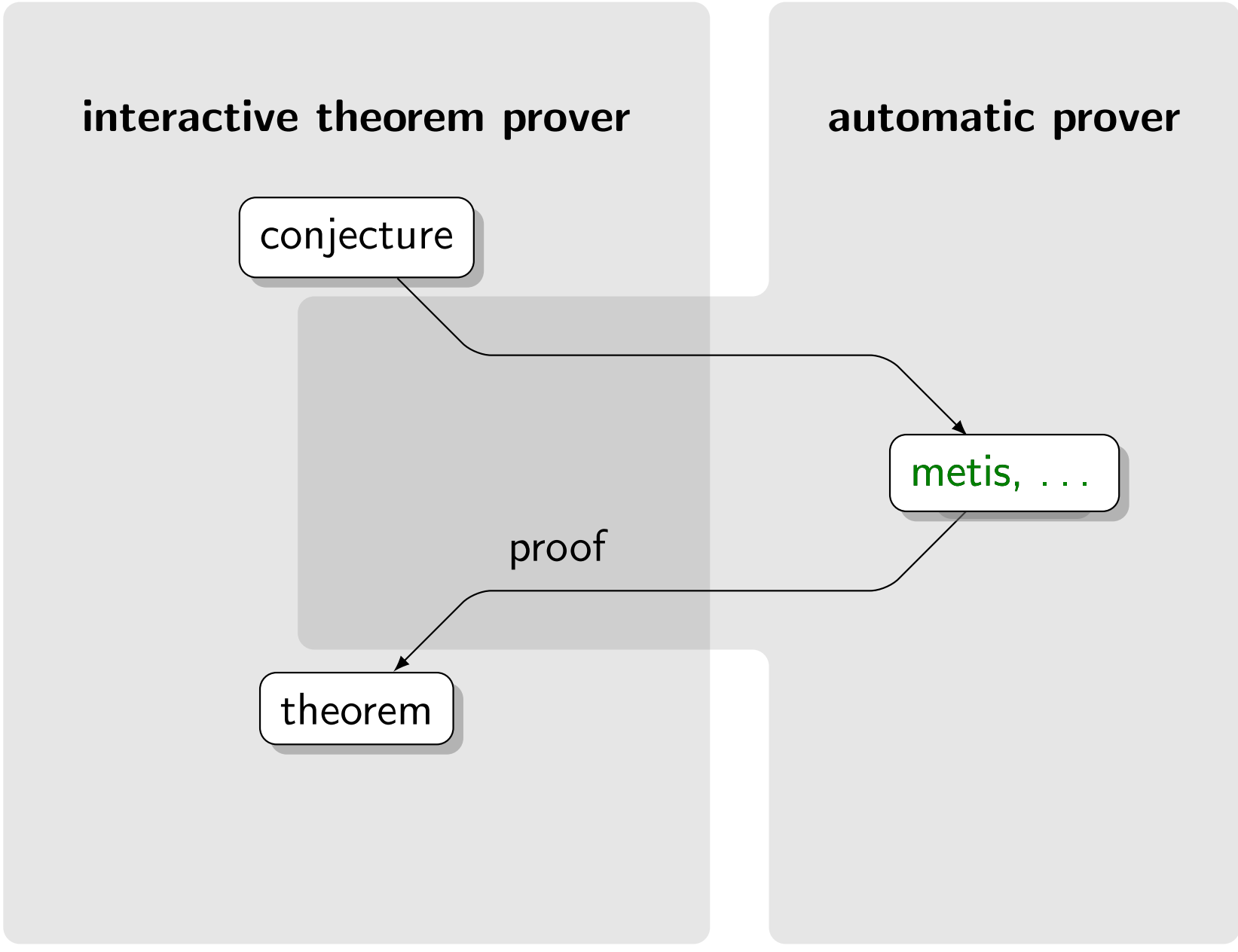


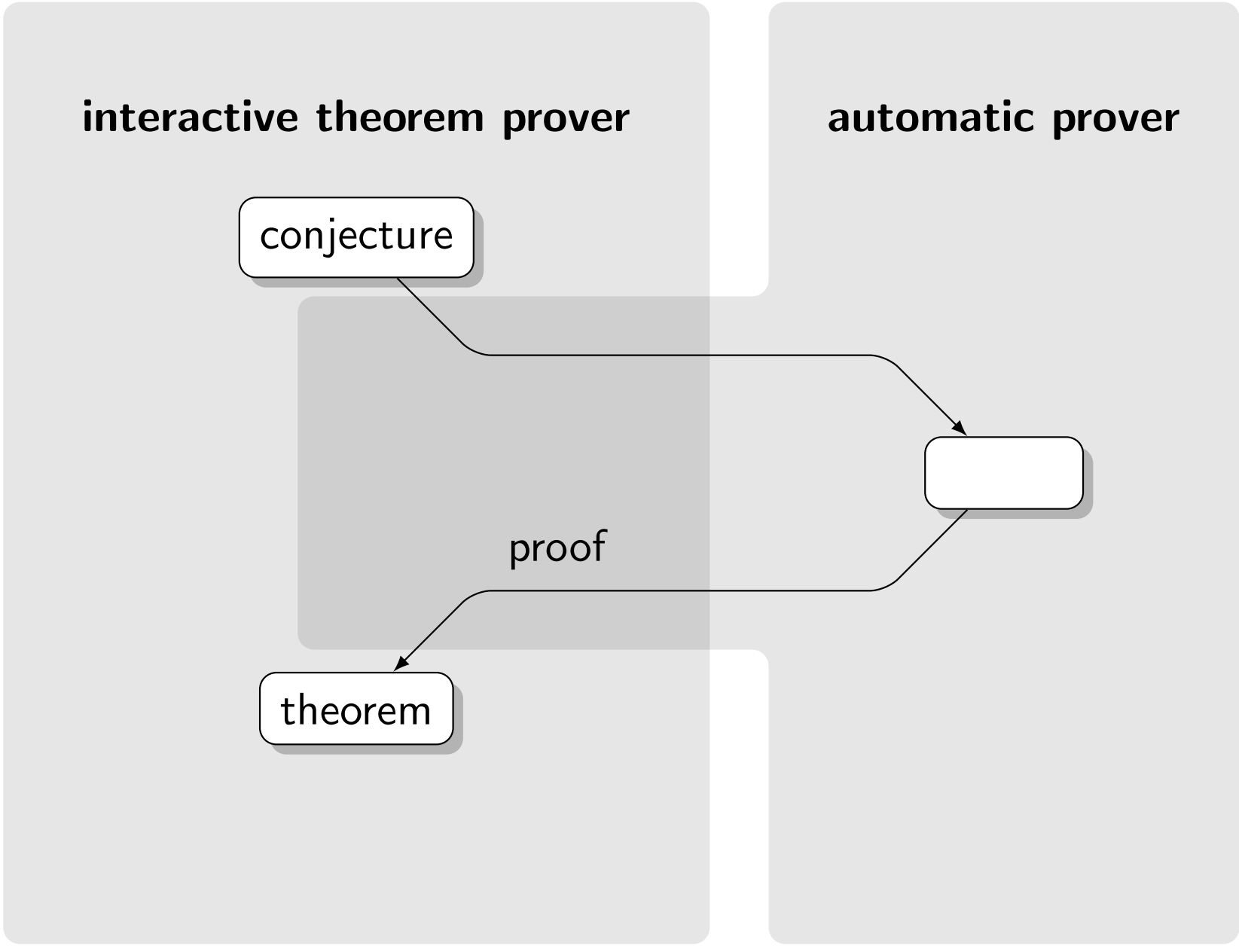
**interactive theorem prover**

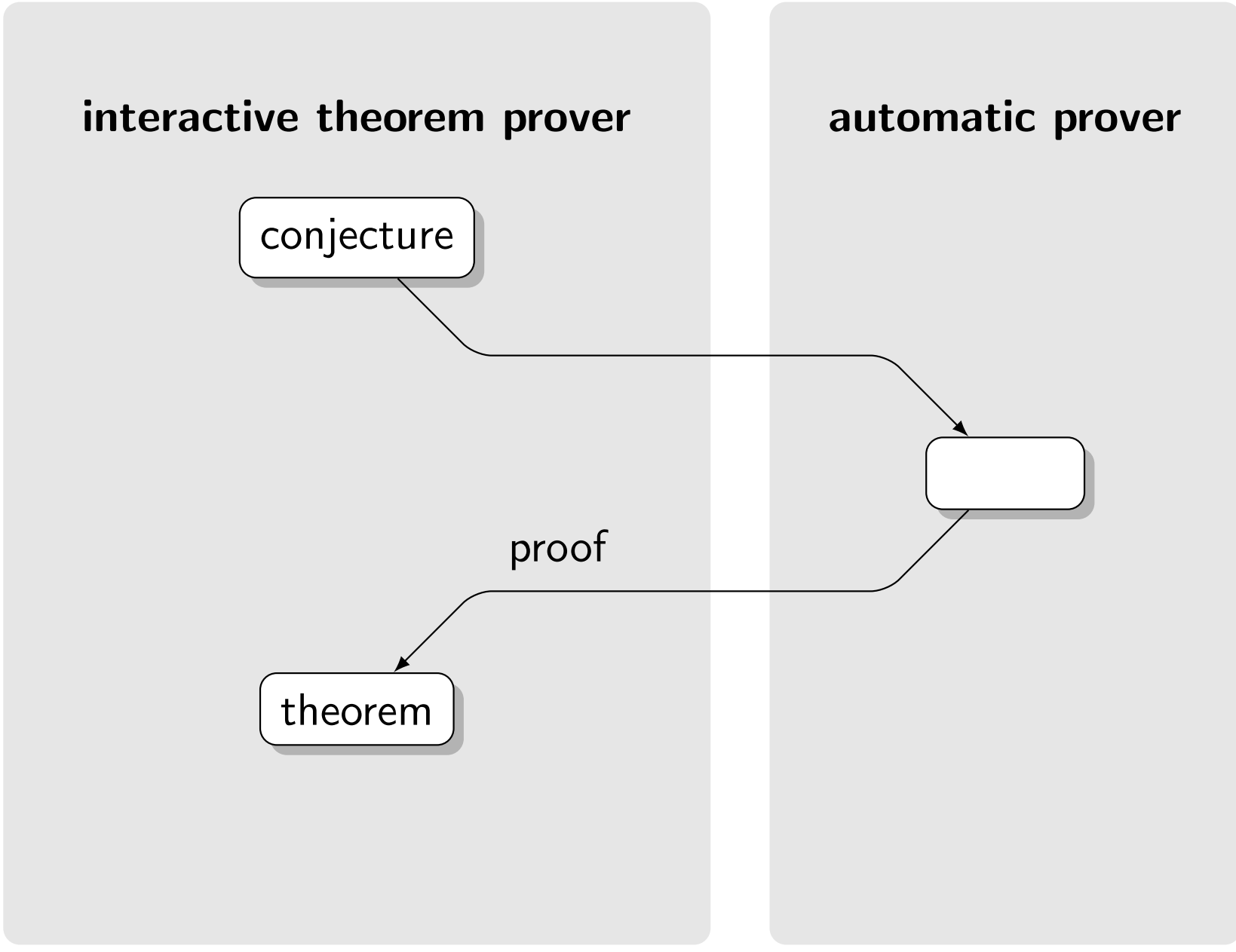
**automatic prover**

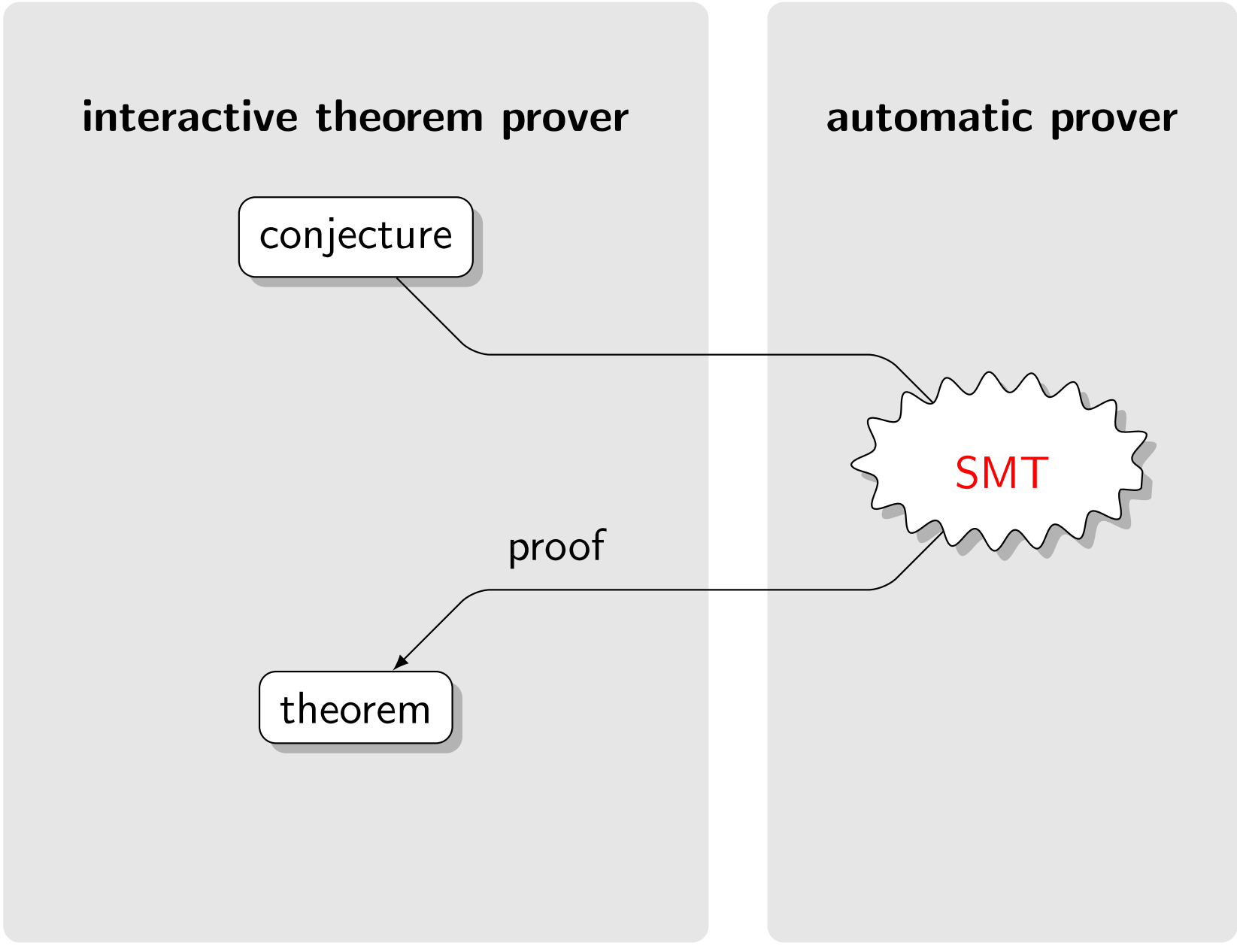



interactive theorem prover
automatic prover
conjecture
proof
theorem

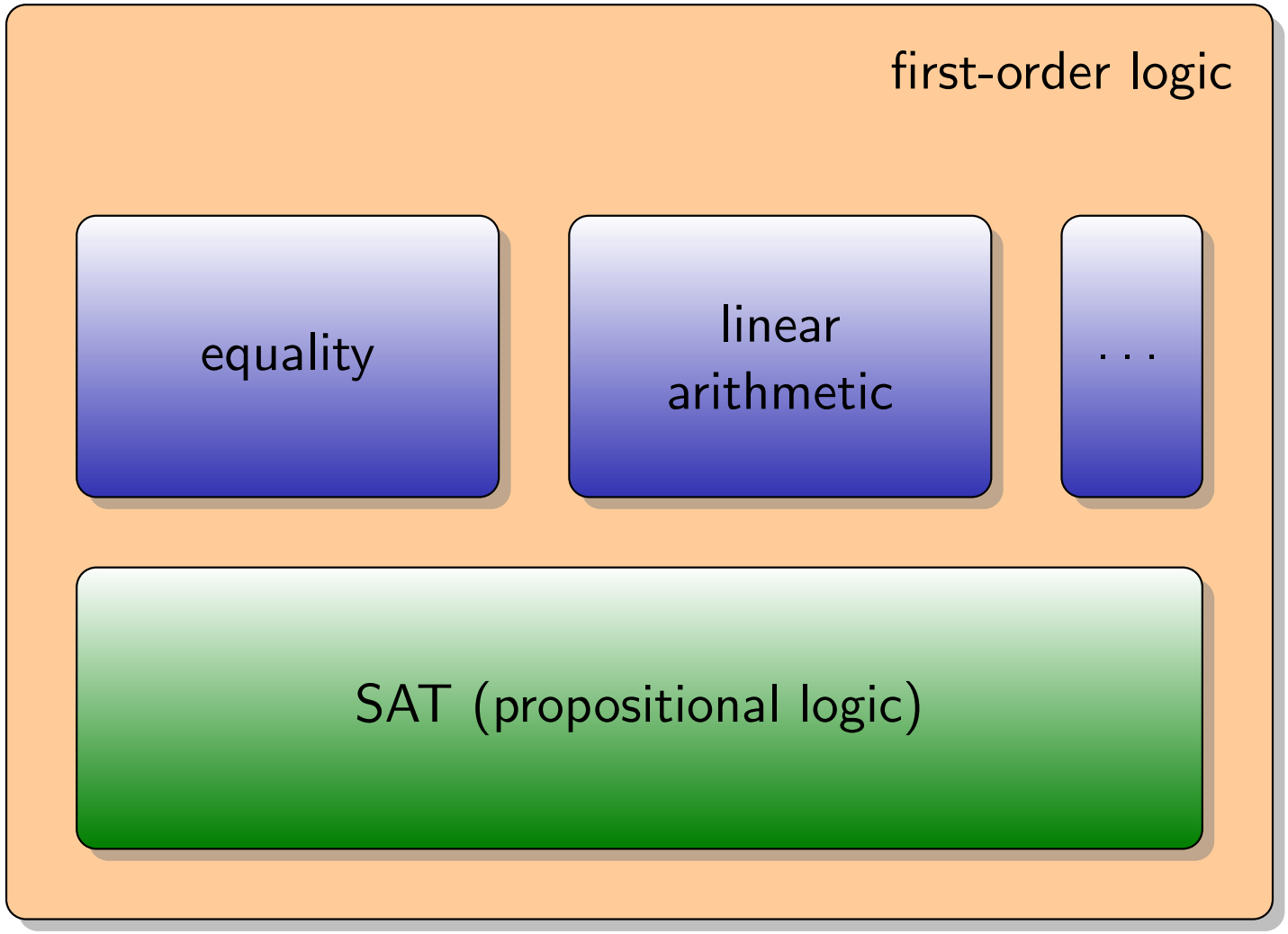

interactive theorem prover
automatic prover
conjecture
metis, . . .
proof
theorem

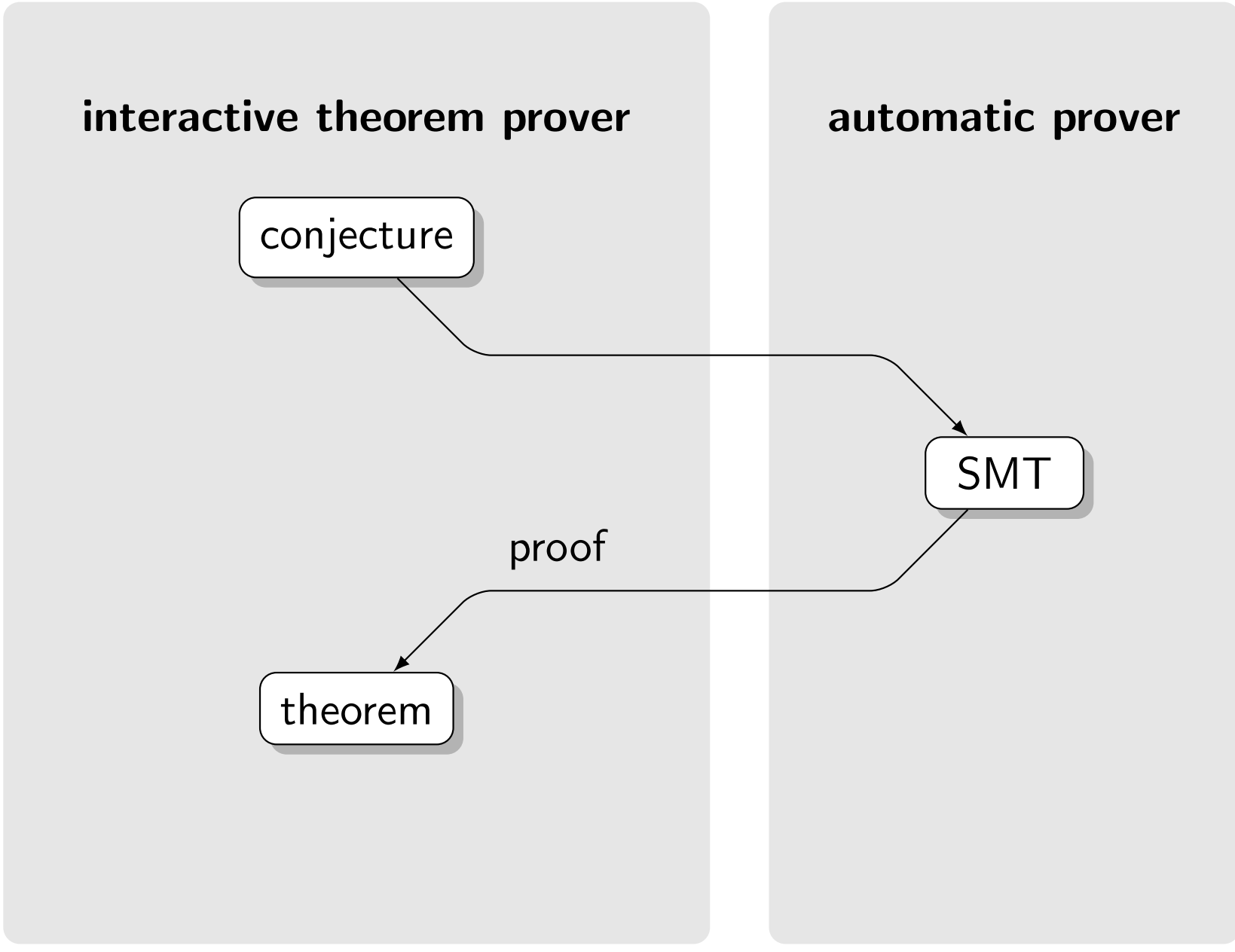

interactive theorem prover
automatic prover
conjecture
proof
theorem

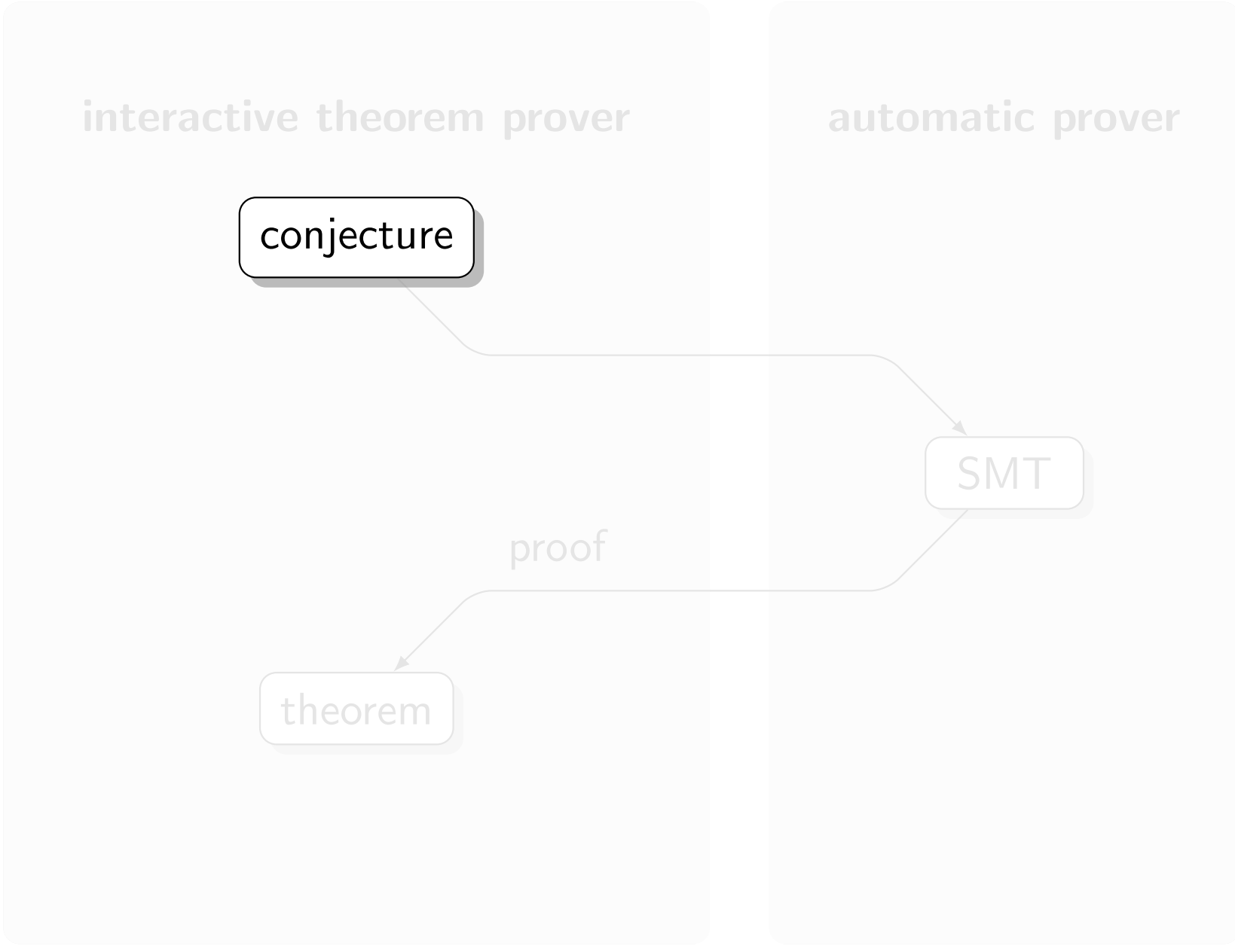

interactive theorem prover
automatic prover
conjecture
proof
theorem


interactive theorem prover
automatic prover
conjecture
SMT
proof
theorem

# SMT: Satisfiability Modulo Theories

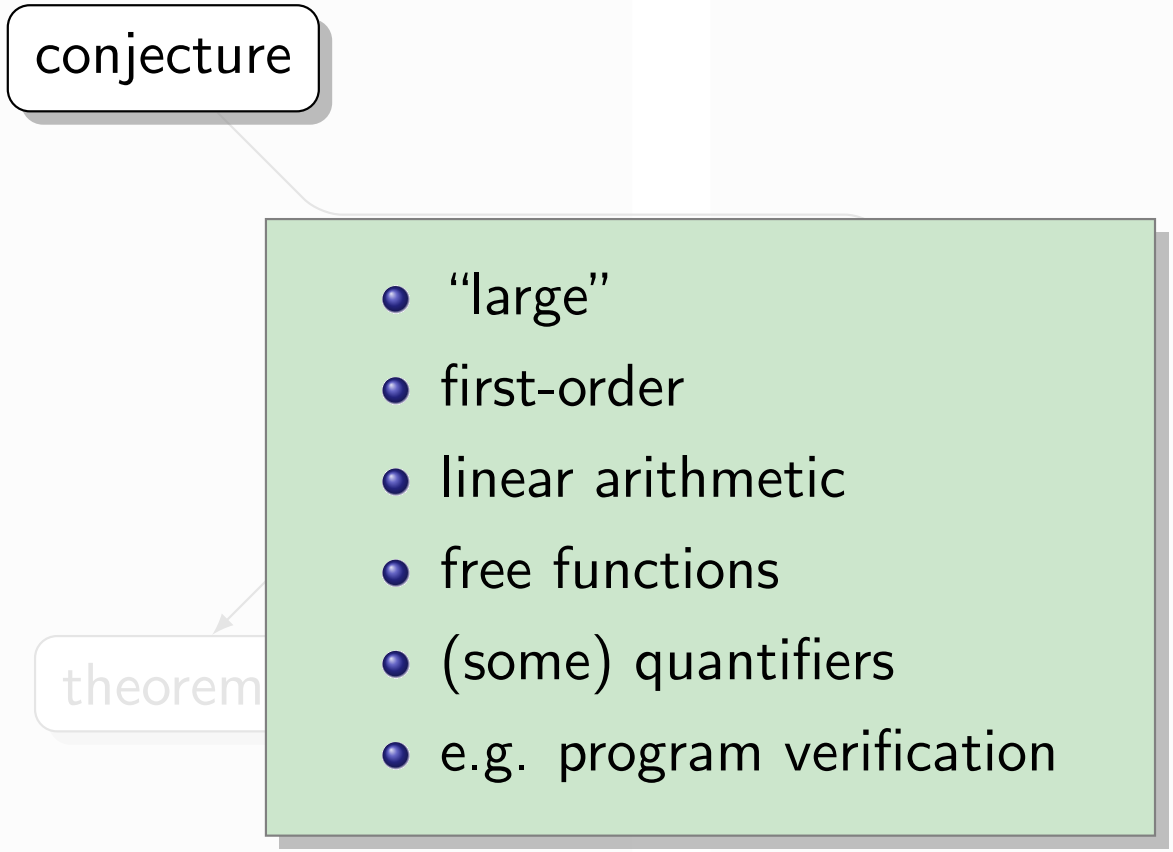


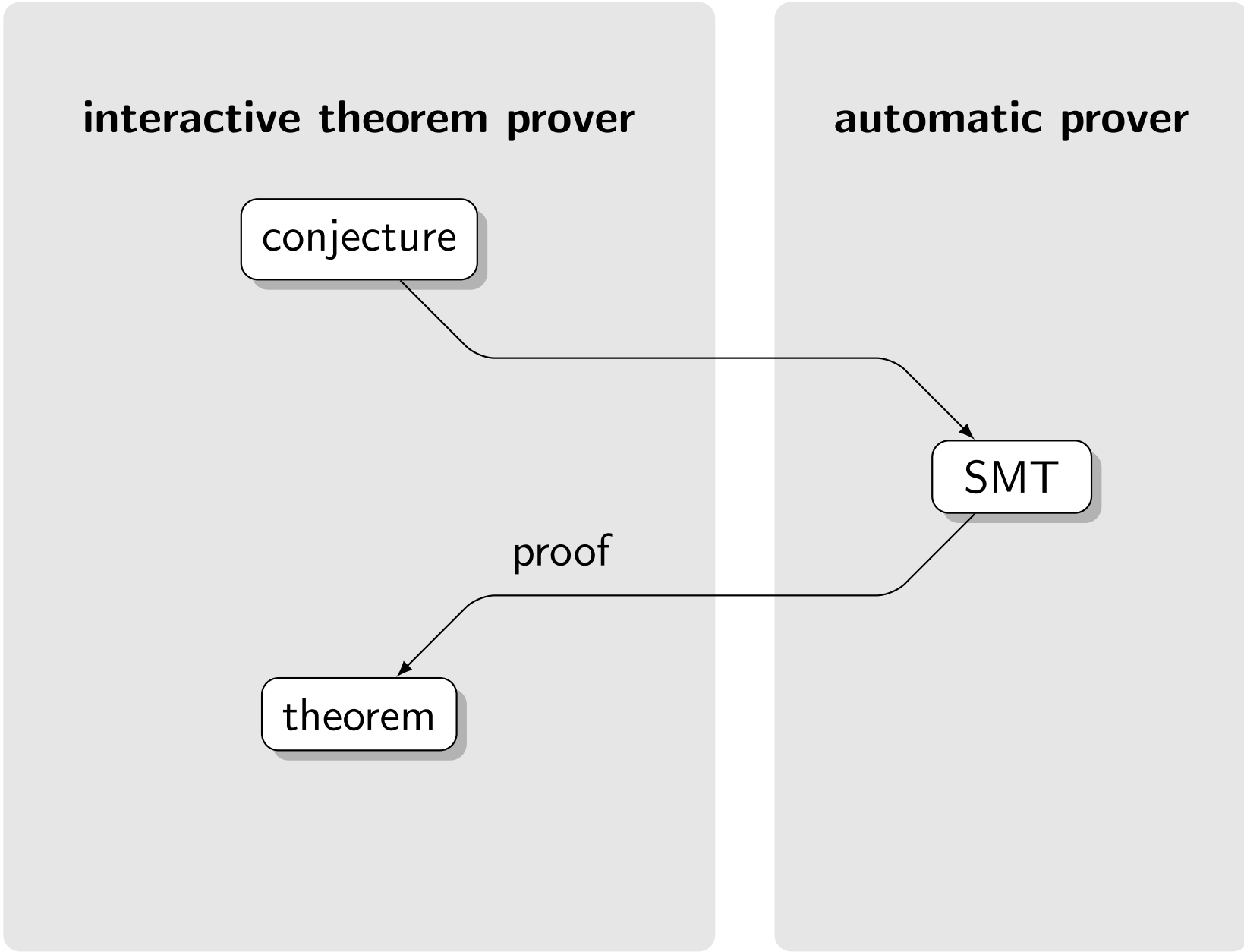

interactive theorem prover
automatic prover
conjecture
SMT
proof
theorem

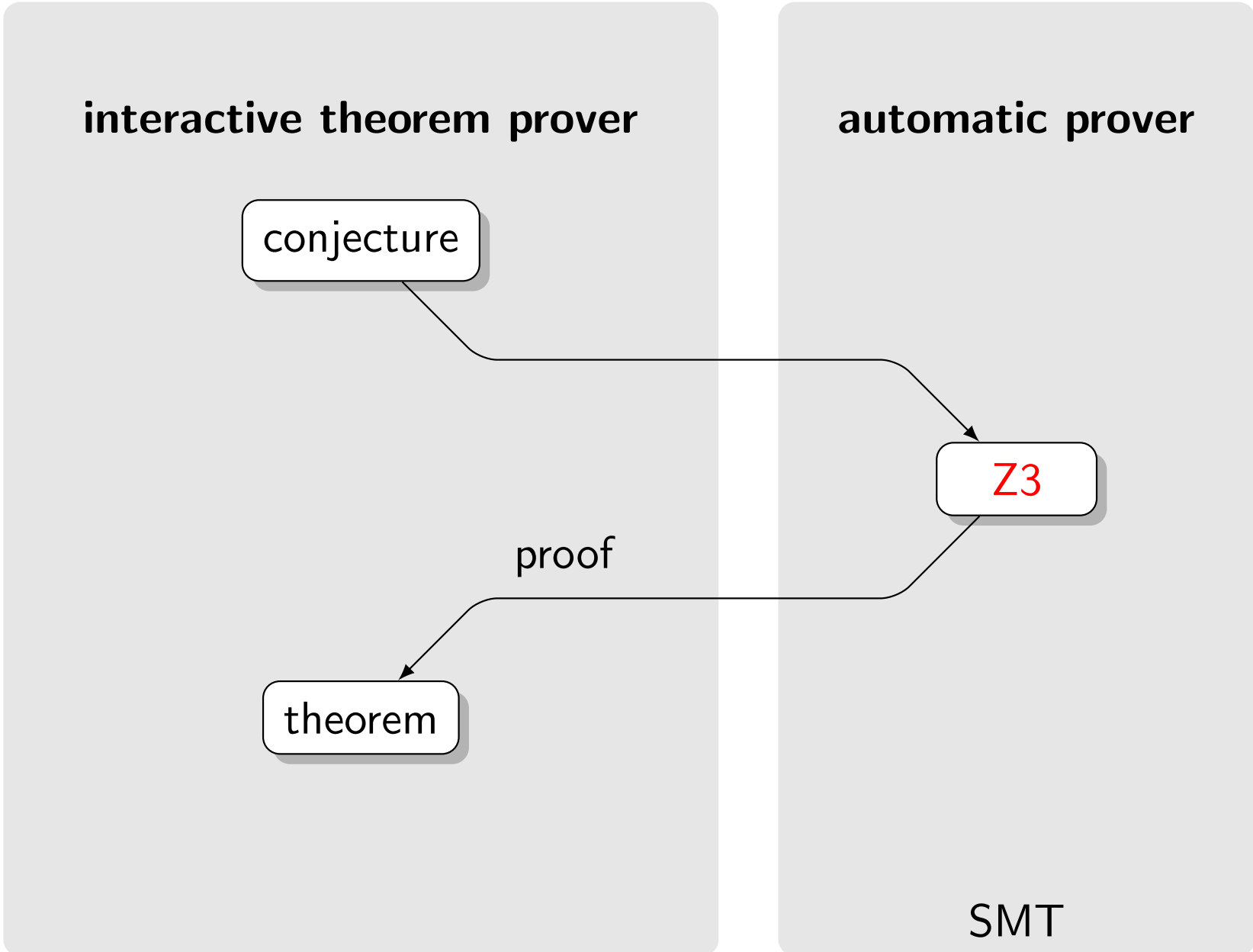

interactive theorem prover
automatic prover
conjecture
SMT
proof
theorem

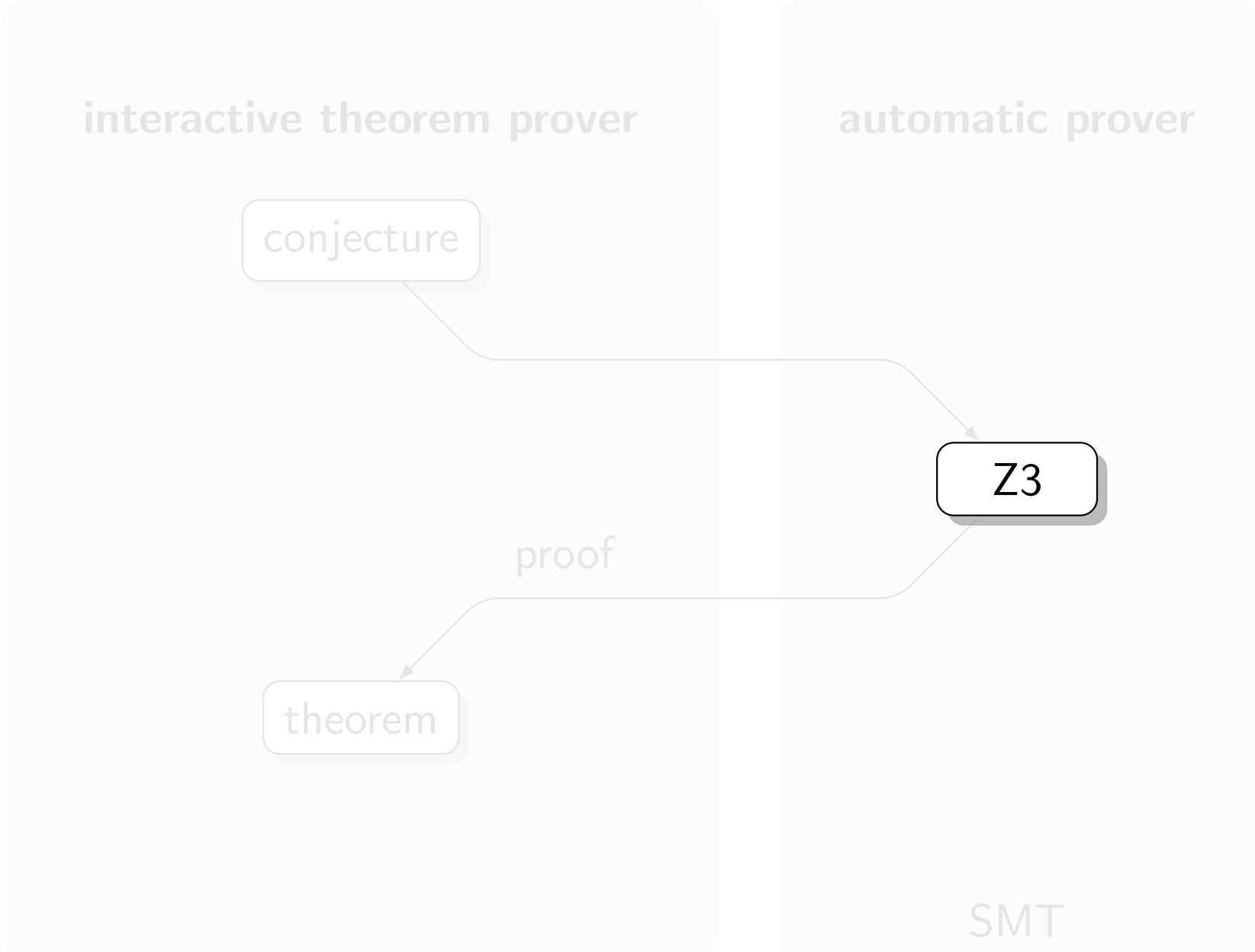
interactive theorem prover

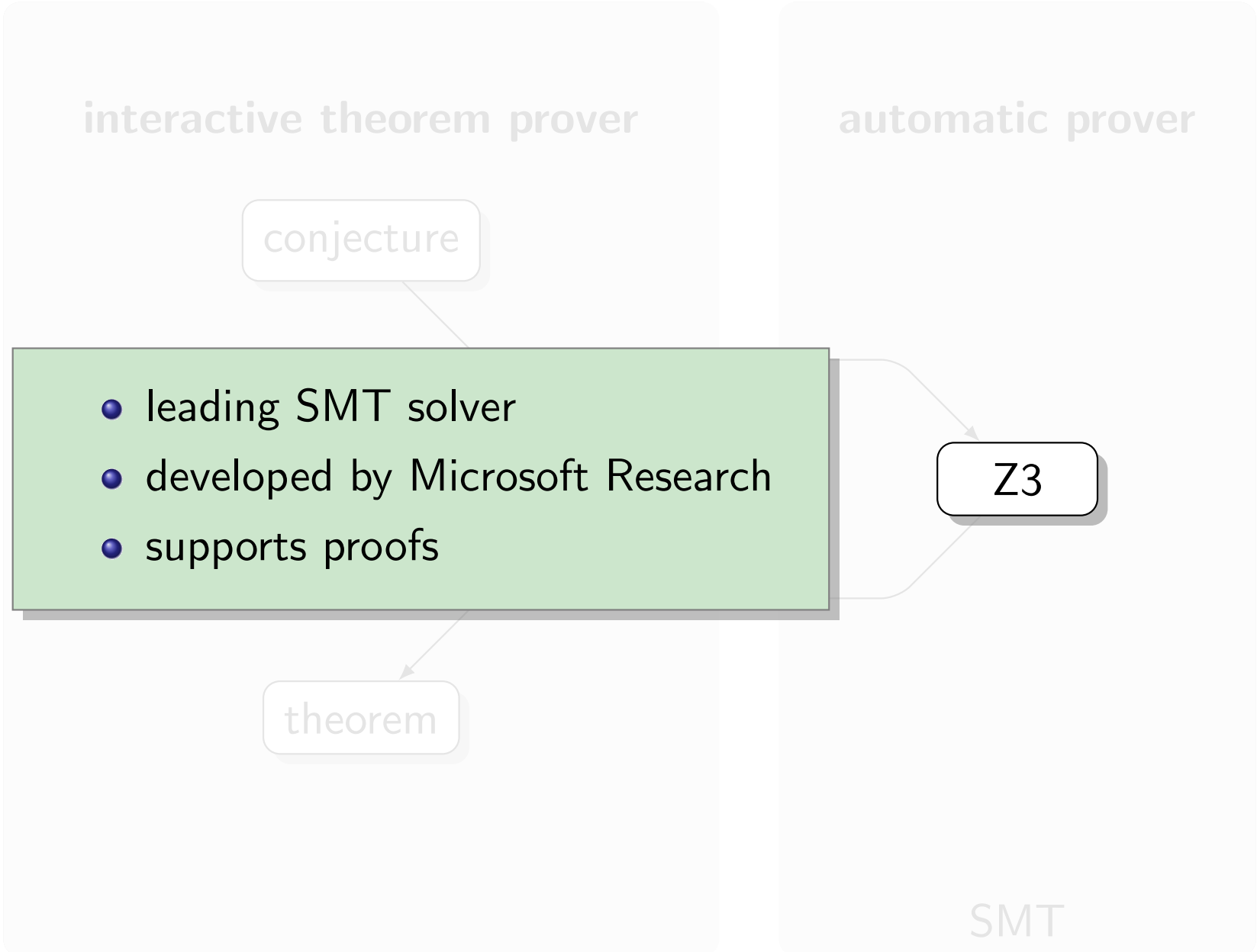
automatic prover

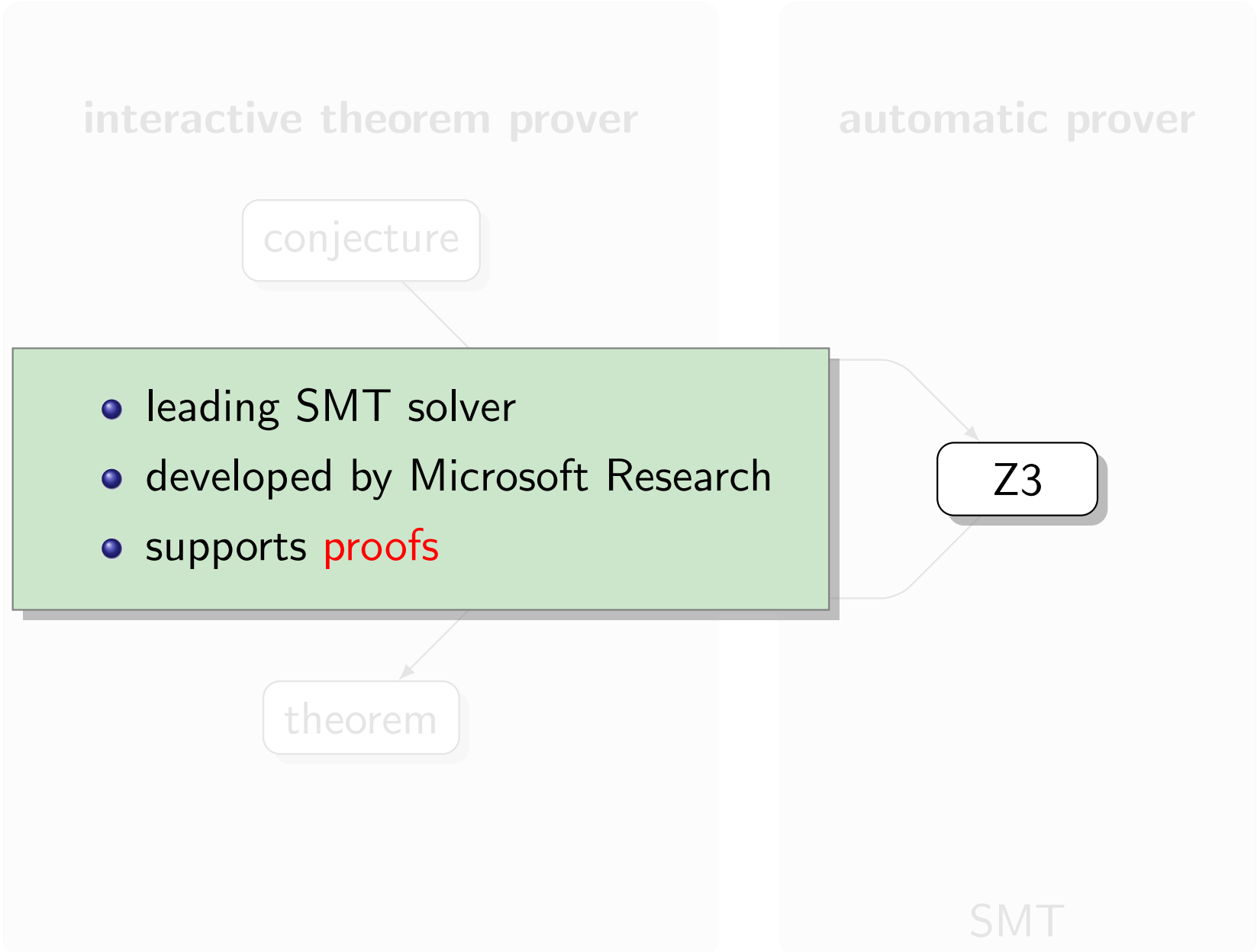


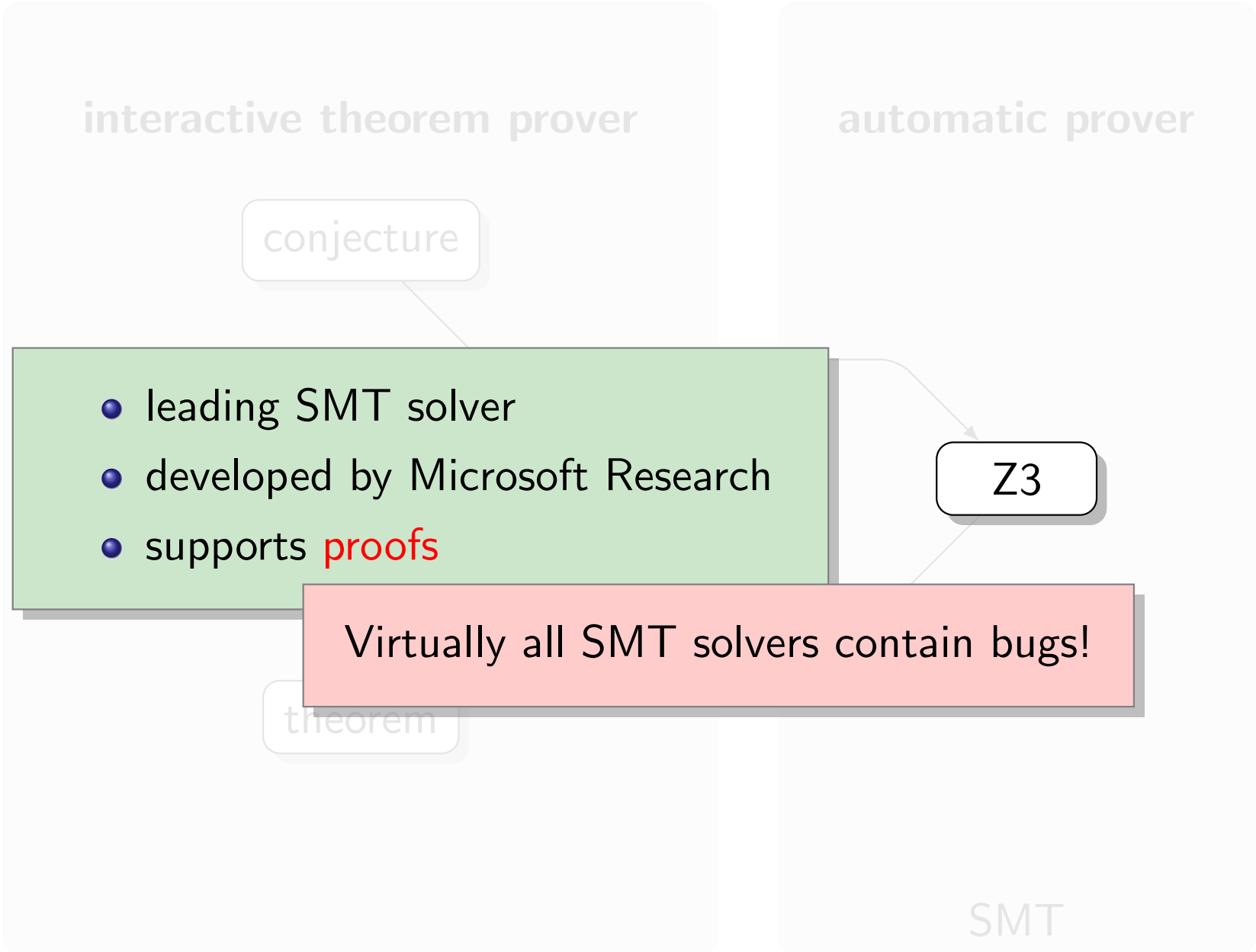

interactive theorem prover
automatic prover
conjecture
SMT
proof
theorem

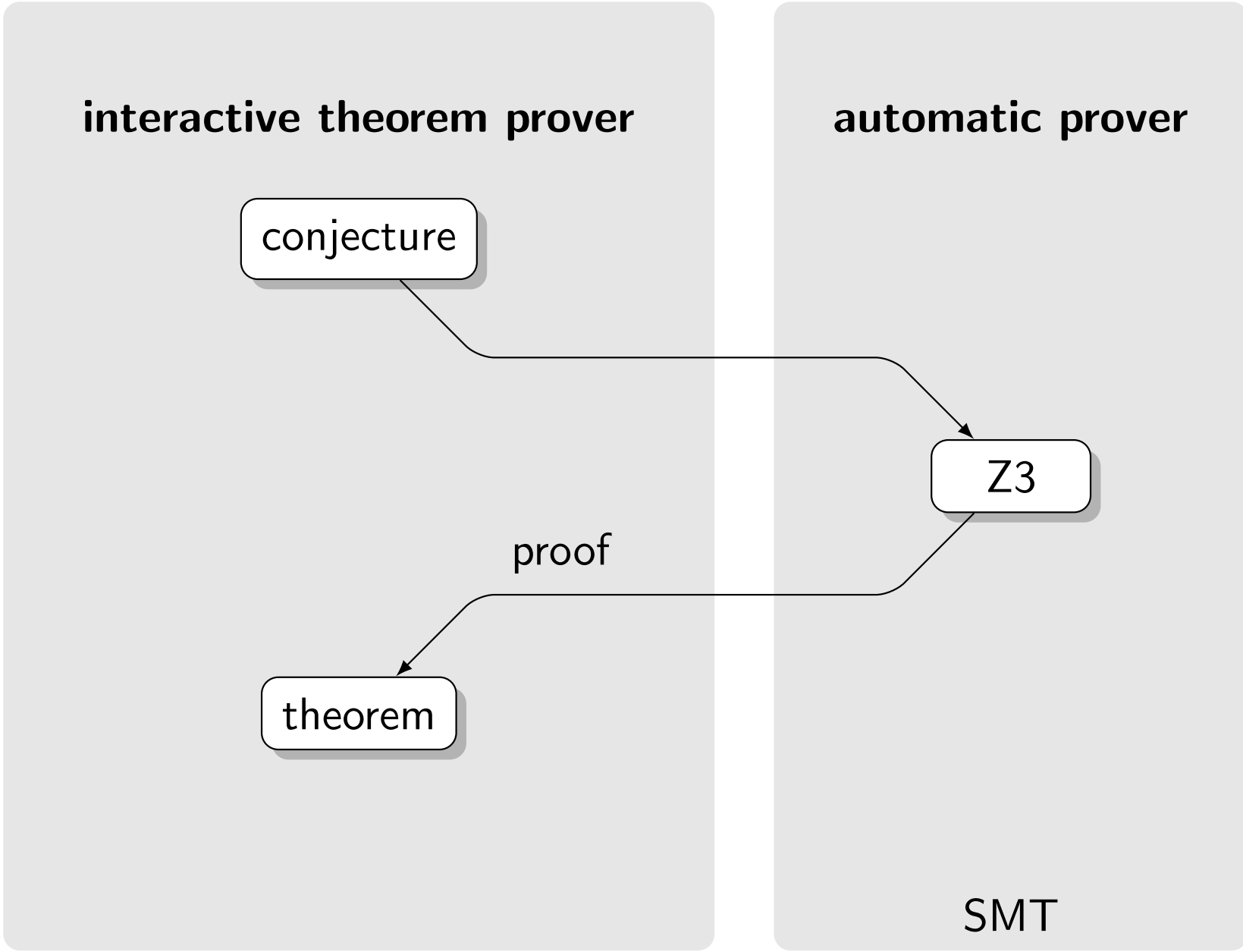

interactive theorem prover
automatic prover
conjecture
Z3
proof
theorem
SMT

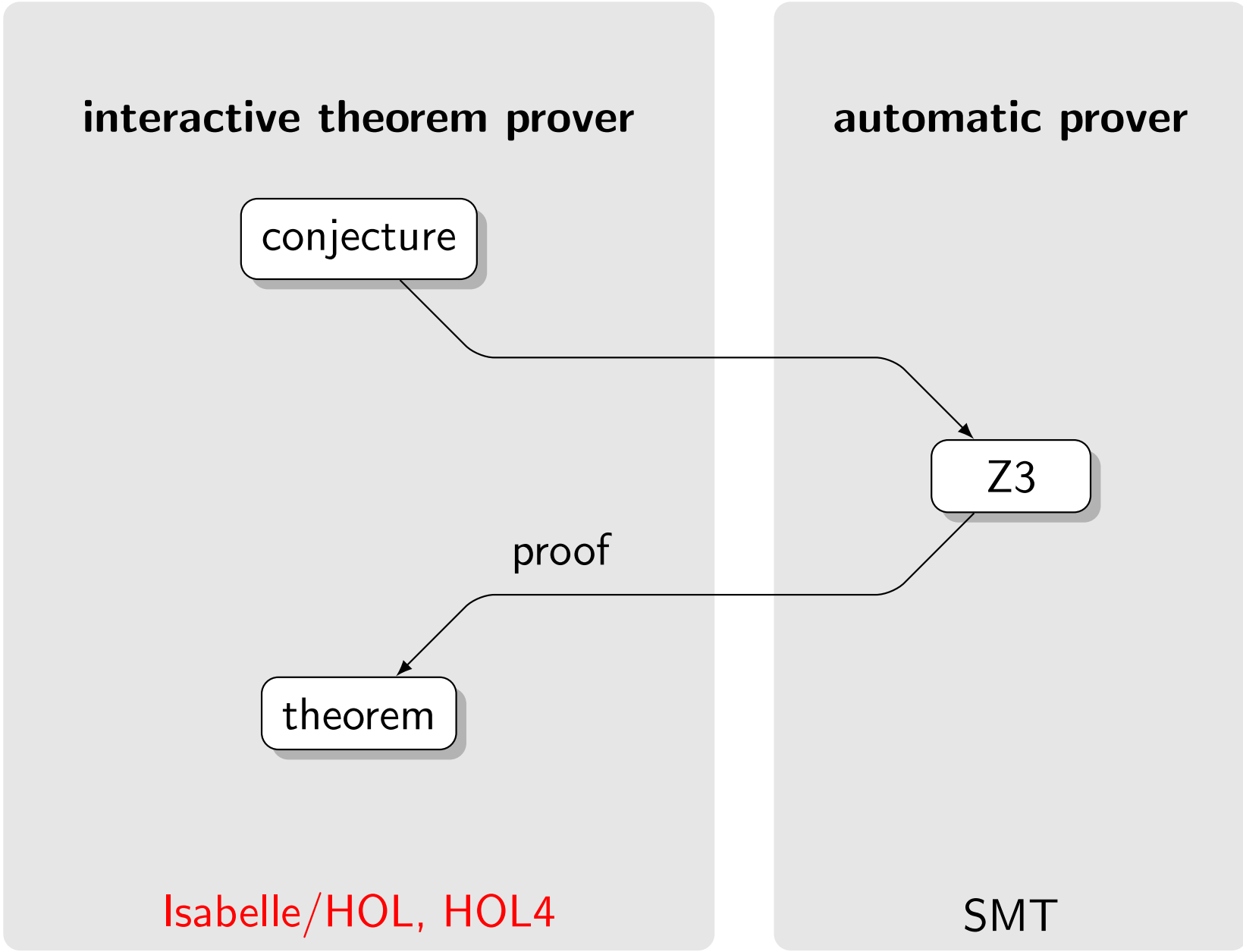

interactive theorem prover
automatic prover
conjecture
Z3
proof
theorem
SMT

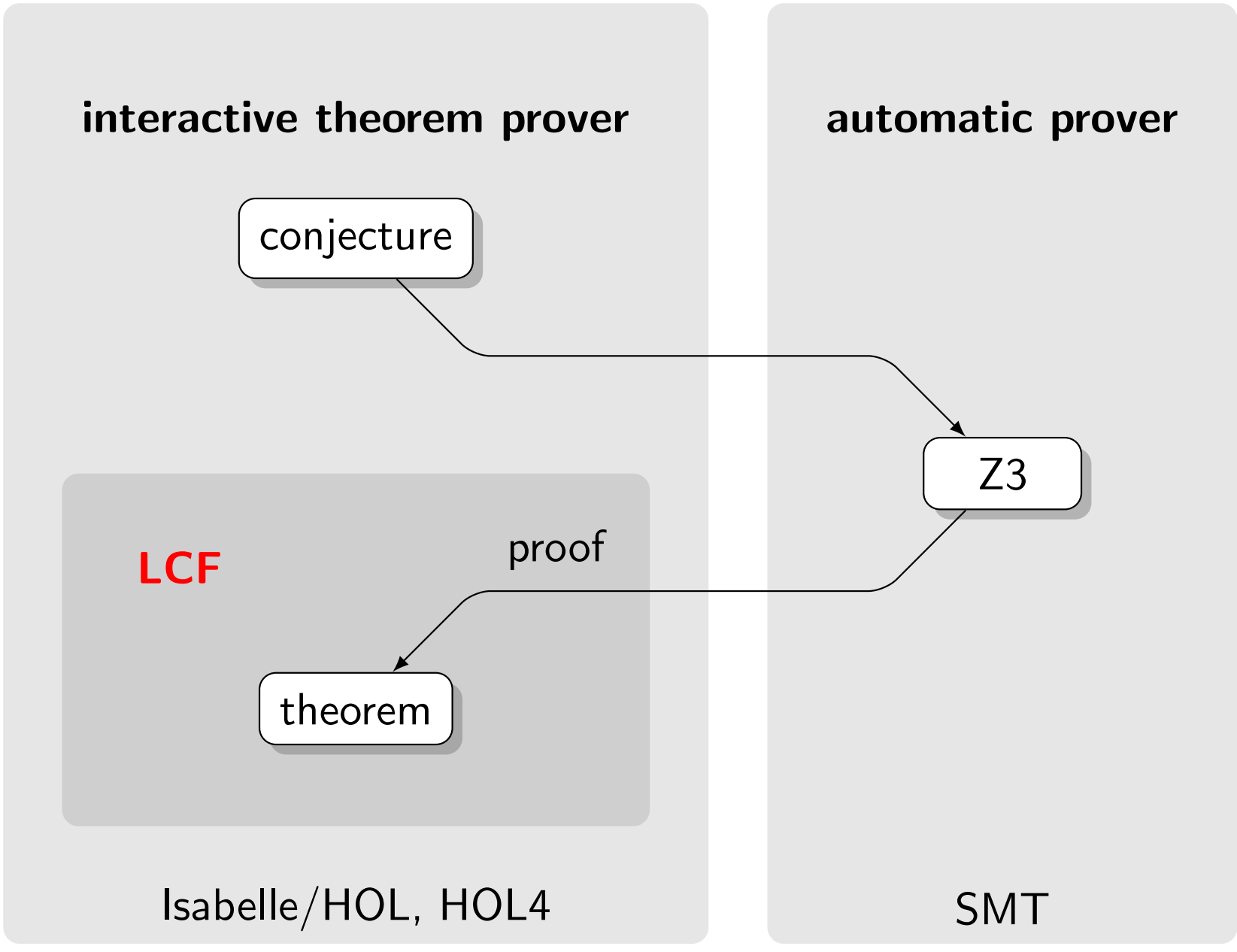

interactive theorem prover
automatic prover
conjecture
leading SMT solver
developed by Microsoft Research
supports proofs
Z3
theorem
SMT

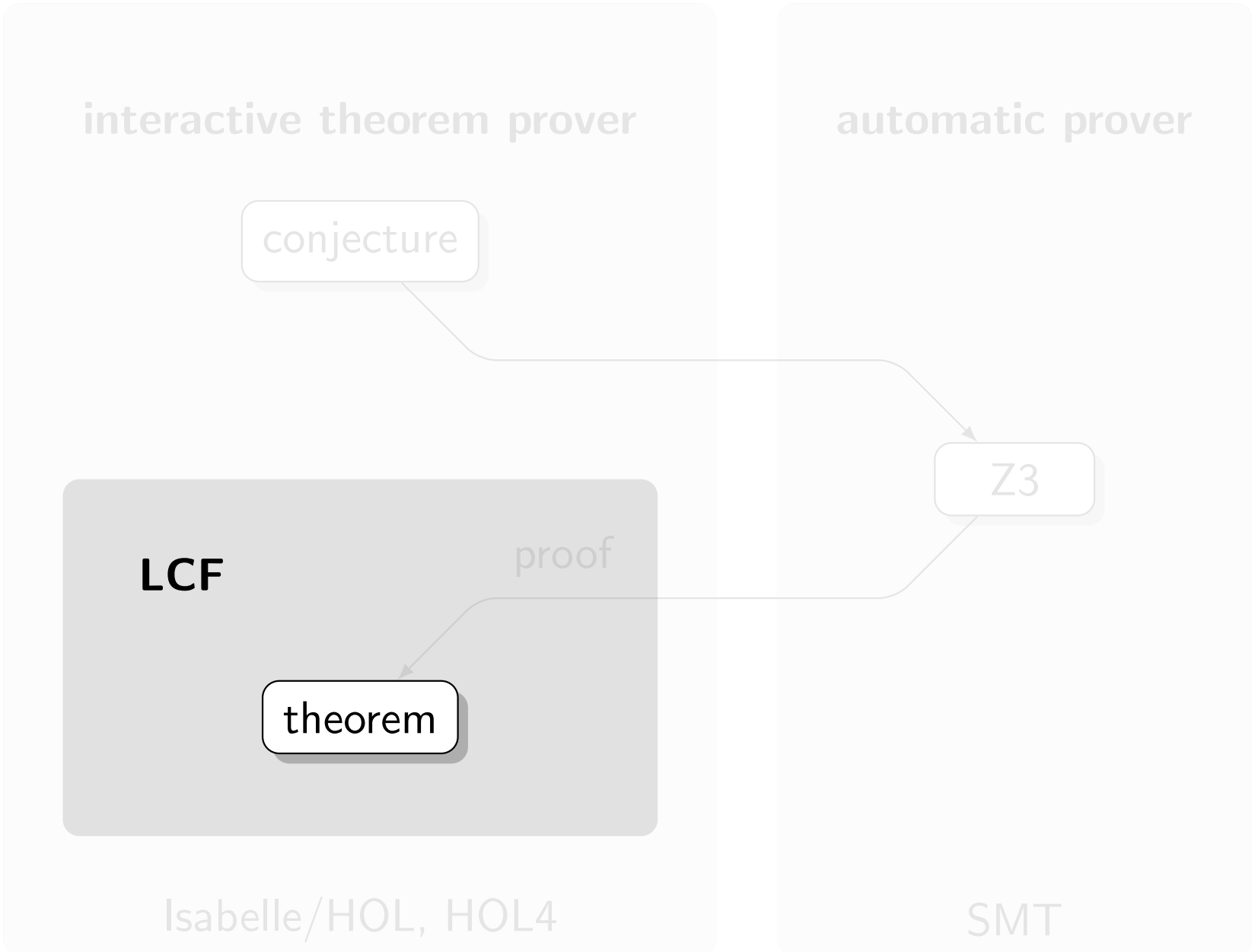

interactive theorem prover
automatic prover
conjecture
leading SMT solver
developed by Microsoft Research
supports proofs
Z3
theorem
SMT

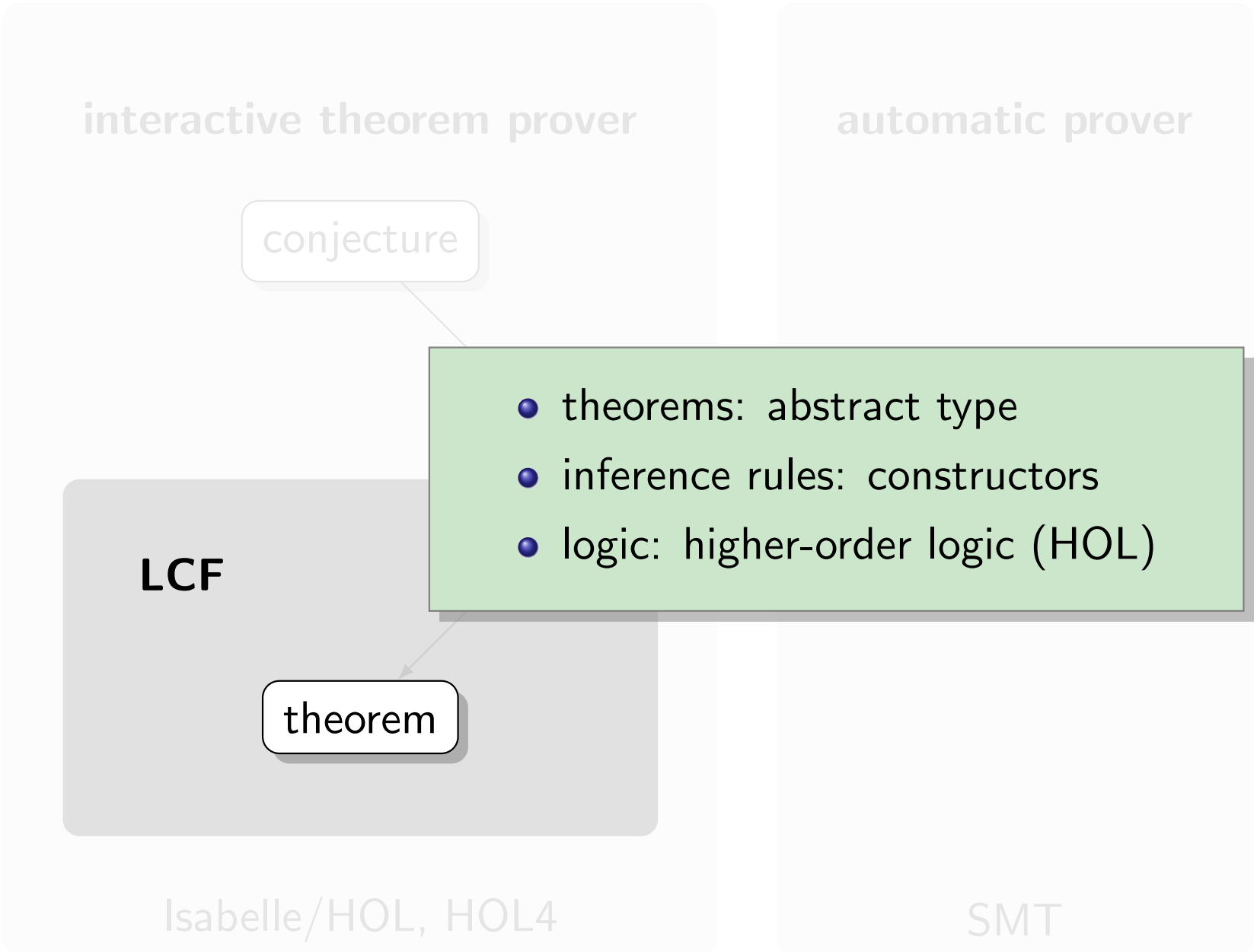

interactive theorem prover
automatic prover
conjecture
leading SMT solver
developed by Microsoft Research
supports proofs
Z3
Virtually all SMT solvers contain bugs!
theorem
SMT

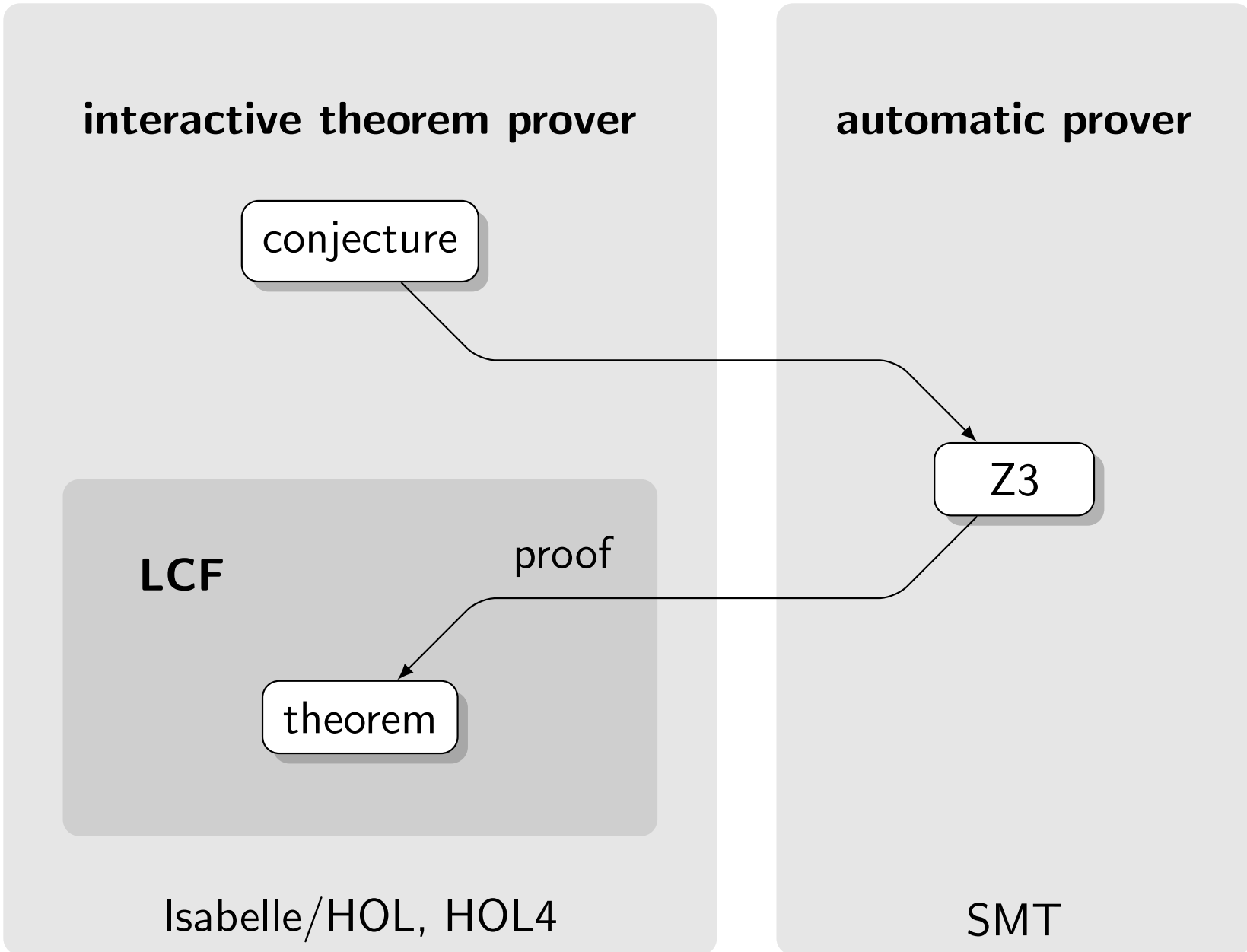

interactive theorem prover
automatic prover
conjecture
Z3
proof
theorem
SMT

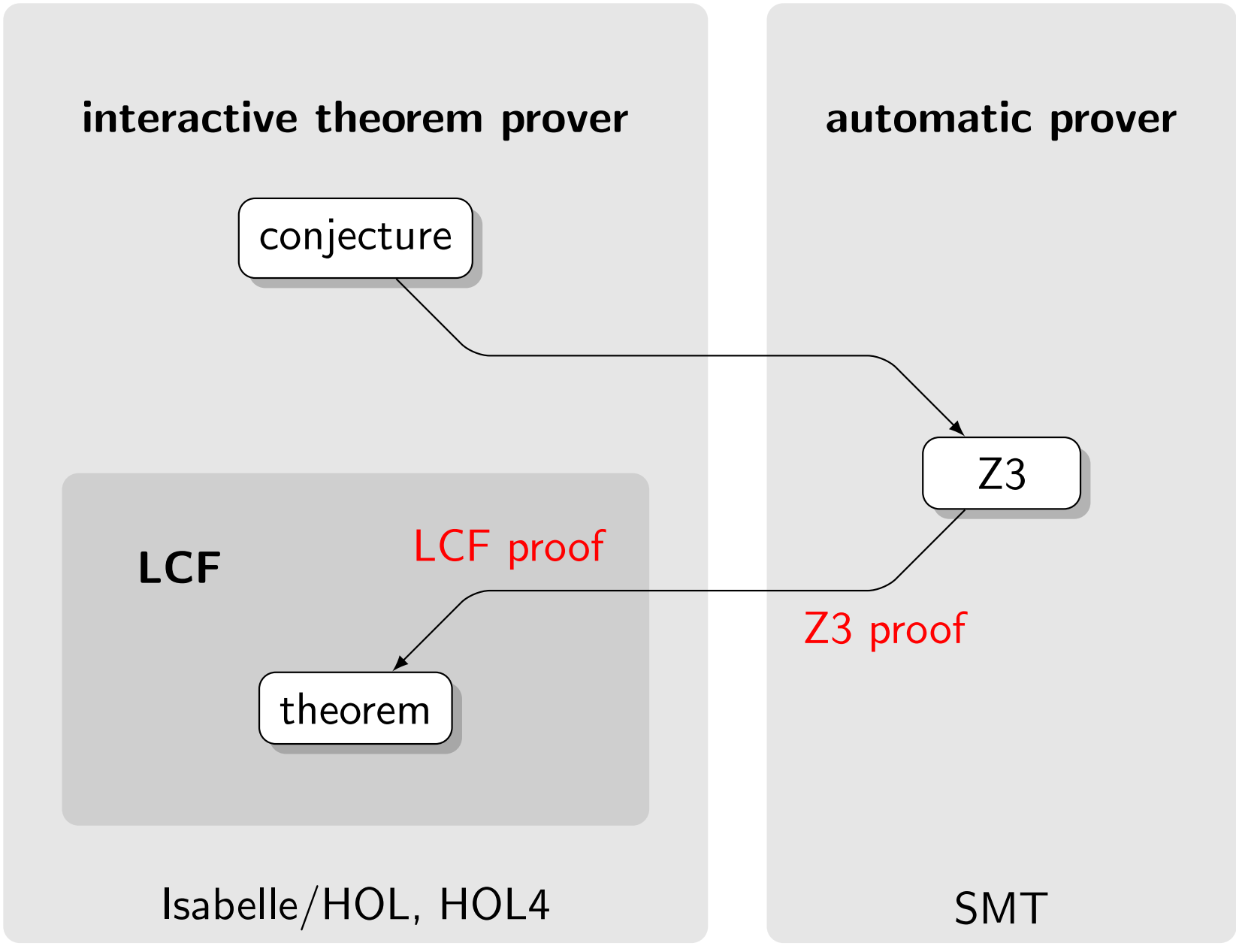

interactive theorem prover
automatic prover
conjecture
Z3
proof
theorem
Isabelle/HOL, HOL4
SMT

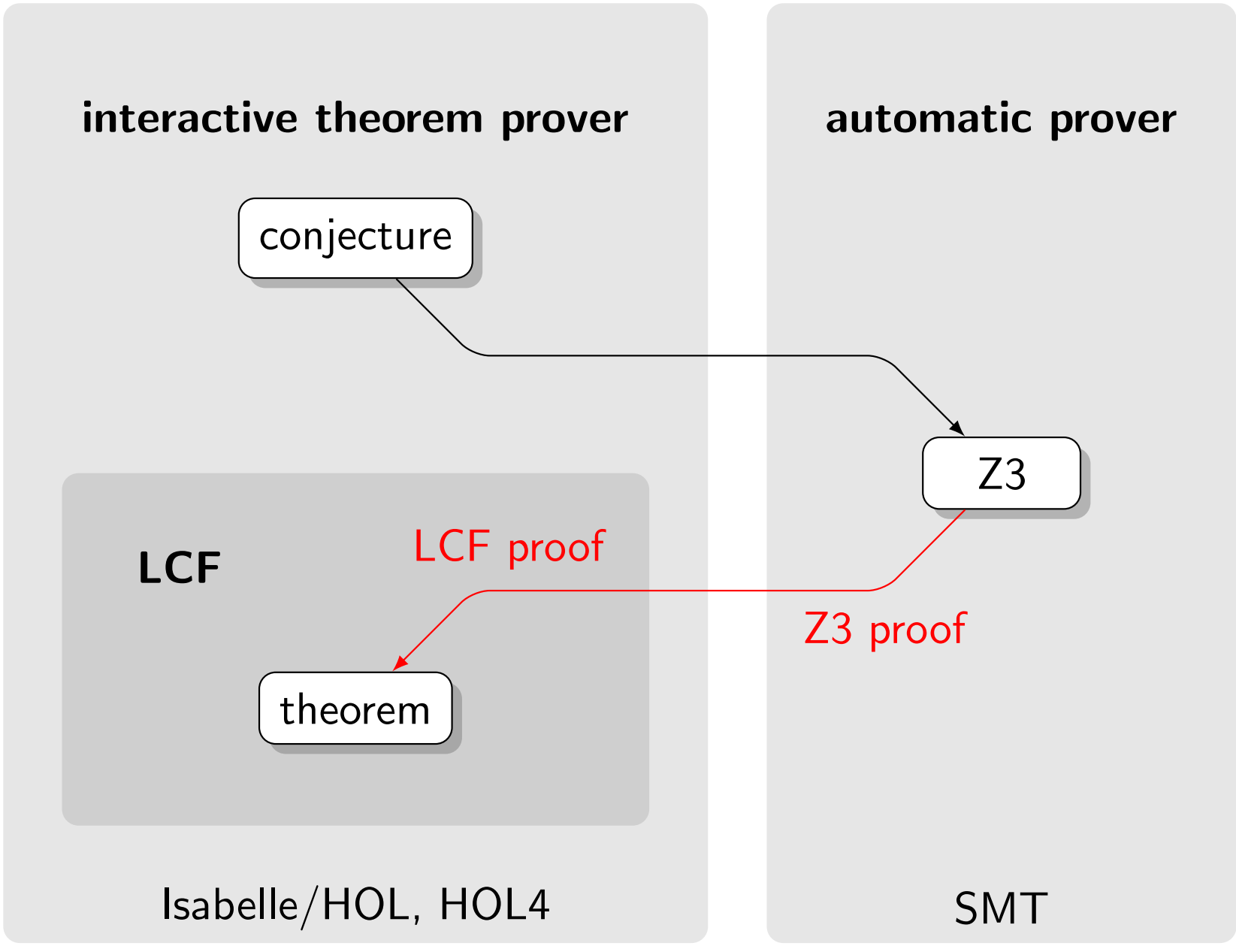

interactive theorem prover
automatic prover
conjecture
Z3
LCF
proof
theorem
Isabelle/HOL, HOL4
SMT

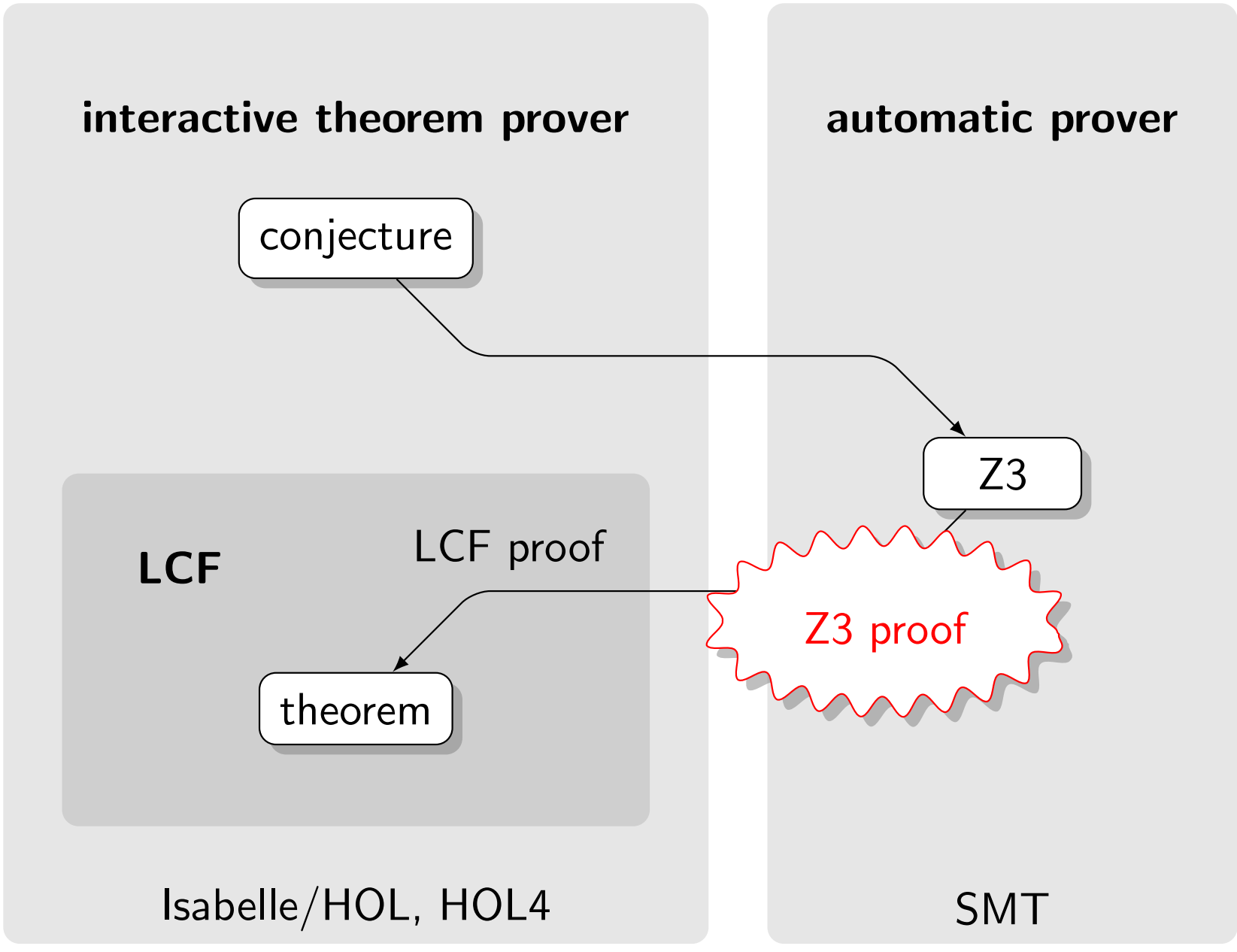

interactive theorem prover
automatic prover
conjecture
Z3
LCF
proof
theorem
Isabelle/HOL, HOL4
SMT


interactive theorem prover
automatic prover
conjecture
theorems: abstract type
inference rules: constructors
logic: higher-order logic (HOL)
LCF
theorem
Isabelle/HOL, HOL4
SMT


interactive theorem prover
automatic prover
conjecture
Z3
LCF
proof
theorem
Isabelle/HOL, HOL4
SMT


interactive theorem prover
automatic prover
conjecture
Z3
LCF
LCF proof
Z3 proof
theorem
Isabelle/HOL, HOL4
SMT


interactive theorem prover
automatic prover
conjecture
Z3
LCF
LCF proof
Z3 proof
theorem
Isabelle/HOL, HOL4
SMT


interactive theorem prover
automatic prover
conjecture
Z3
LCF
LCF proof
Z3 proof
theorem
Isabelle/HOL, HOL4
SMT

# Z3 Proofs

Natural deduction proofs:

## Example

$$\dfrac{\dfrac{}{\neg \mathsf{True} \vdash \neg \mathsf{True}}\ \textsc{asserted} \qquad \dfrac{}{\vdash \neg \mathsf{True} \longleftrightarrow \mathsf{False}}\ \textsc{rewrite}}{\neg \mathsf{True} \vdash \mathsf{False}}\ \textsc{mp}$$

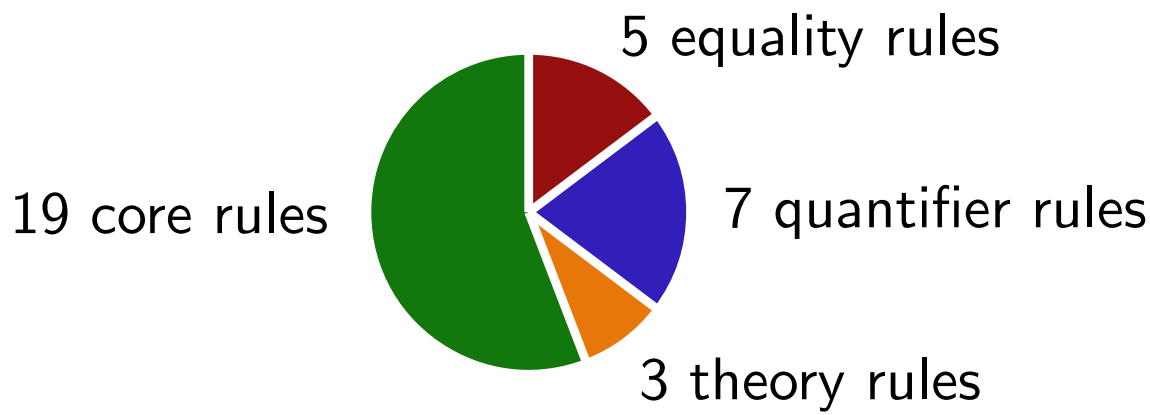
# Z3 Proofs

Natural deduction proofs:

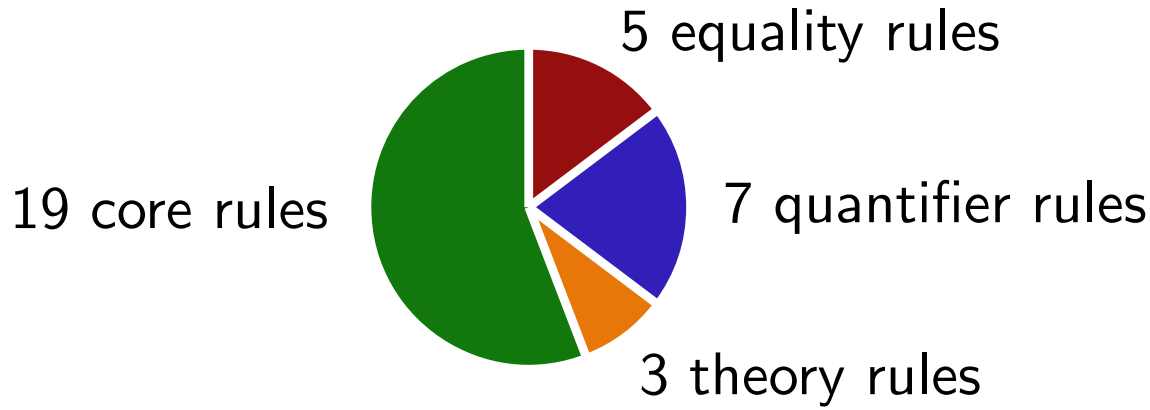
## Example

$$\dfrac{\dfrac{}{\neg\mathsf{True} \vdash \neg\mathsf{True}}\ \textsc{asserted} \qquad \dfrac{}{\vdash \neg\mathsf{True} \longleftrightarrow \mathsf{False}}\ \textsc{rewrite}}{\neg\mathsf{True} \vdash \mathsf{False}}\ \textsc{mp}$$

34 axiom schemata and inference rules:

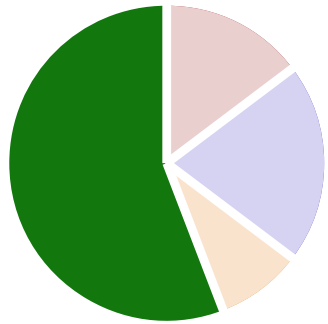


# Z3 Inference Rules



# Z3 Inference Rules

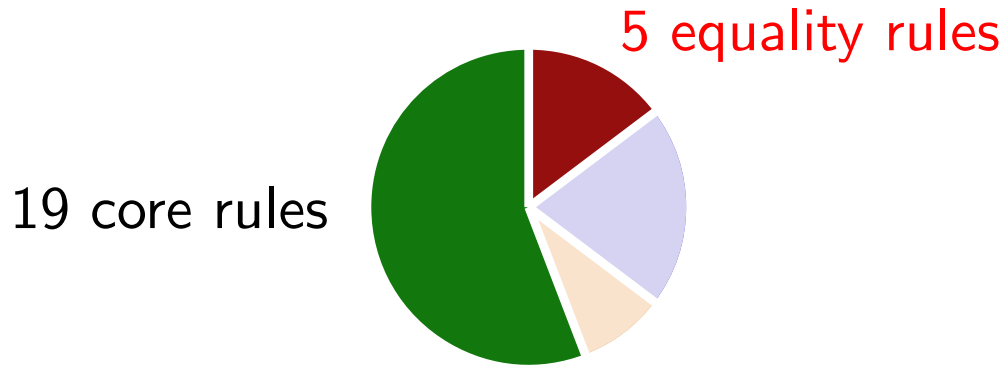
19 core rules



Core rules ASSERTED, MP, UNIT-RESOLUTION, ...

- mostly propositional reasoning

# Z3 Inference Rules

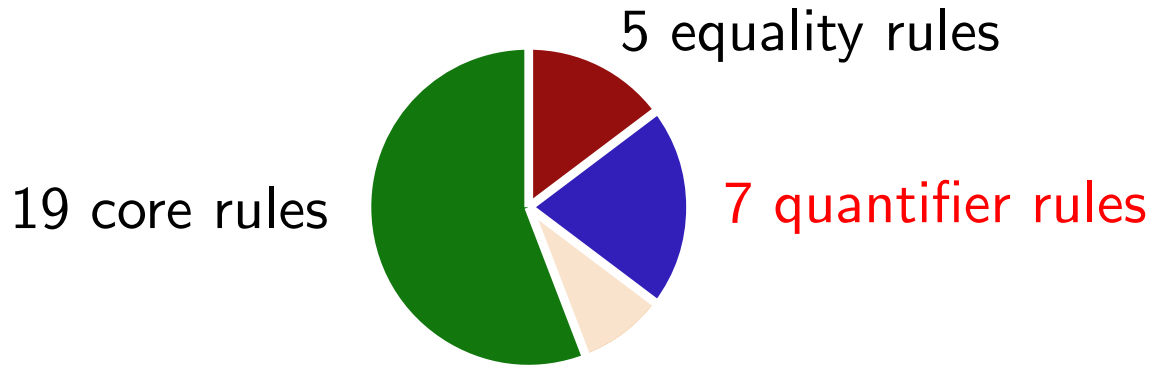



Core rules Asserted, MP, Unit-Resolution, ...

- mostly propositional reasoning

Equality rules Refl, Symm, Trans, Mono, ...

# Z3 Inference Rules



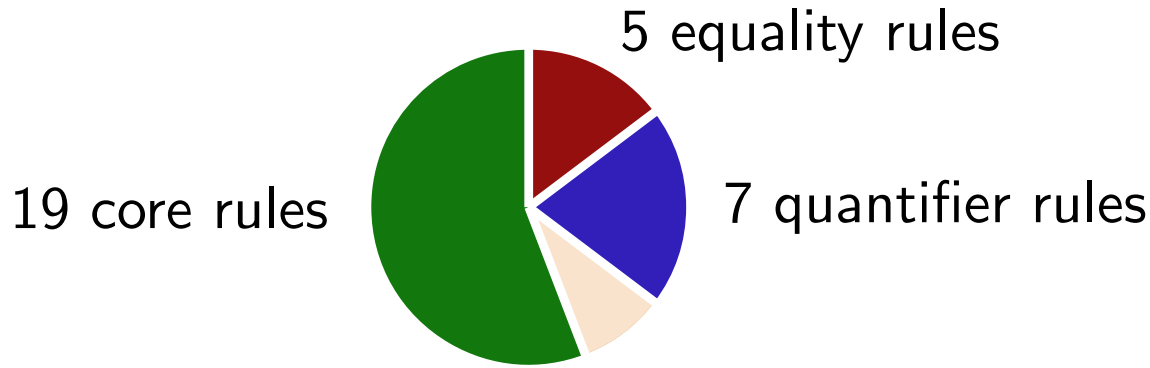

Core rules Asserted, MP, Unit-Resolution, ...

- mostly propositional reasoning

Equality rules Refl, Symm, Trans, Mono, ...

Quantifier rules Quant-Inst, Quant-Intro, ...

# Z3 Inference Rules




Core rules Asserted, MP, Unit-Resolution, ...

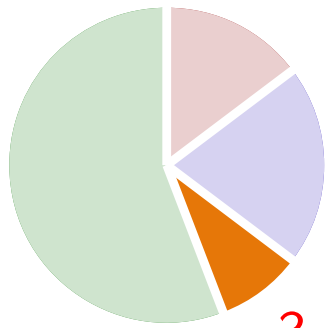
- mostly propositional reasoning

Equality rules Refl, Symm, Trans, Mono, ...

Quantifier rules Quant-Inst, Quant-Intro, ...

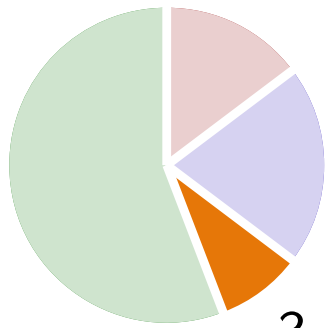
Structure of the proof, simple semantics

# Z3 Inference Rules



3 theory rules

# Z3 Inference Rules

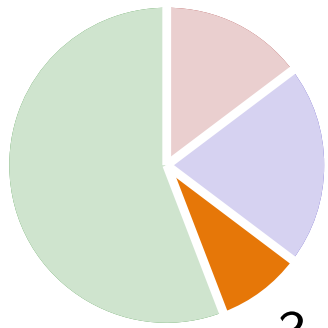



REWRITE: simplification rules

**Example**

$$\frac{}{(P \longrightarrow Q) \longleftrightarrow (Q \vee \neg P)} \qquad \frac{}{0 + x = x}$$

# Z3 Inference Rules

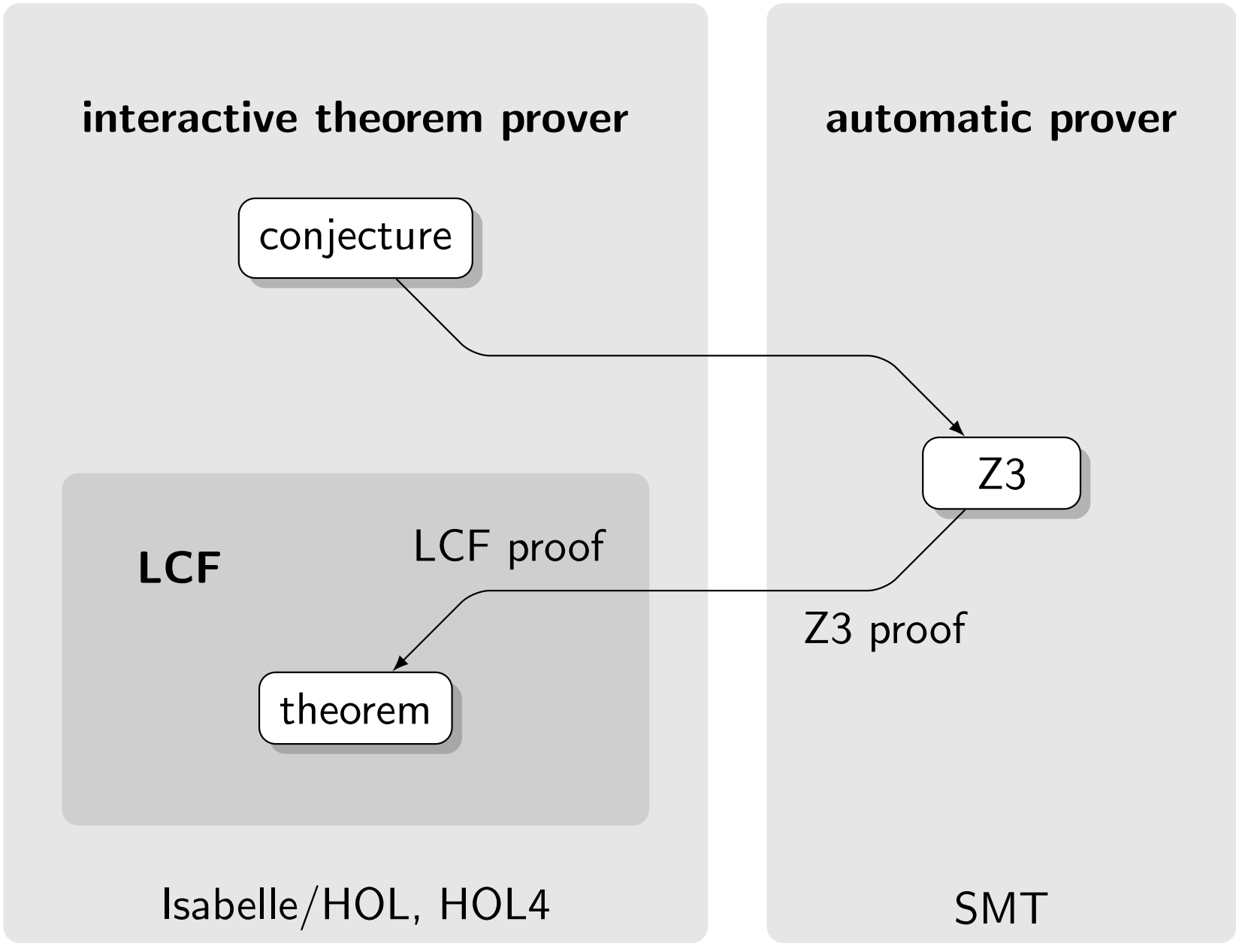


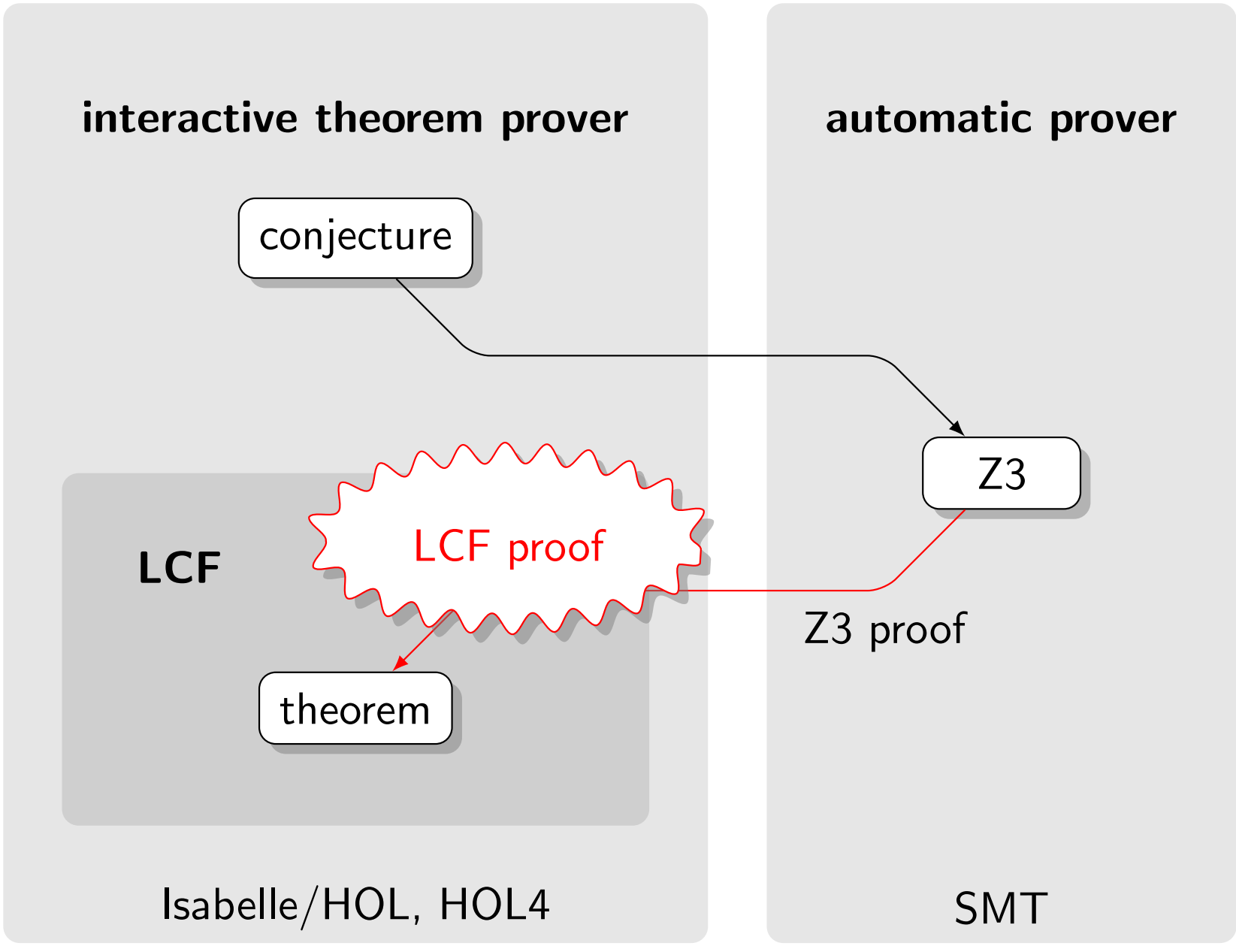
3 theory rules

TH-LEMMA: inconsistency of theory atoms

**Example**

$$\frac{0 < x - 3y \ \wedge \ x - 3y < 0}{\text{False}}$$

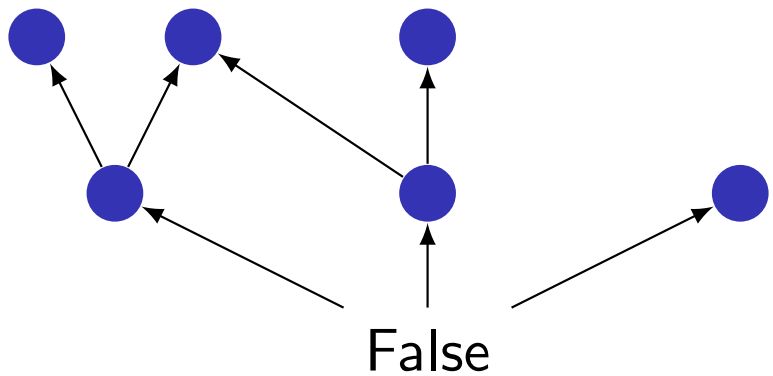

interactive theorem prover
automatic prover
conjecture
Z3
LCF
LCF proof
Z3 proof
theorem
Isabelle/HOL, HOL4
SMT


interactive theorem prover
automatic prover
conjecture
Z3
LCF
LCF proof
Z3 proof
theorem
Isabelle/HOL, HOL4
SMT

# Proof Reconstruction

Overall approach:

- one proof method for every Z3 inference rule
- depth-first traversal of Z3 proof



# First Prototype

Apply existing automated proof tools:

- simp, blast, auto, metis, ...

# First Prototype

Apply existing automated proof tools:

- simp, blast, auto, metis, ...

But: in typical Z3 proofs:

- large terms
- only shallow reasoning required
- automated proof tools get lost (poor performance)

# First Prototype

Apply existing automated proof tools:

- simp, blast, auto, metis, ...

But: in typical Z3 proofs:

- large terms
- only shallow reasoning required
- automated proof tools get lost (poor performance)

Avoiding automated proof tools:

Speed-up of up to 3 orders of magnitude!

# Fast Proof Reconstruction

Reconstruction techniques:

# Fast Proof Reconstruction

Reconstruction techniques:

1. single primitive inference rule or theorem instantiation

# Fast Proof Reconstruction

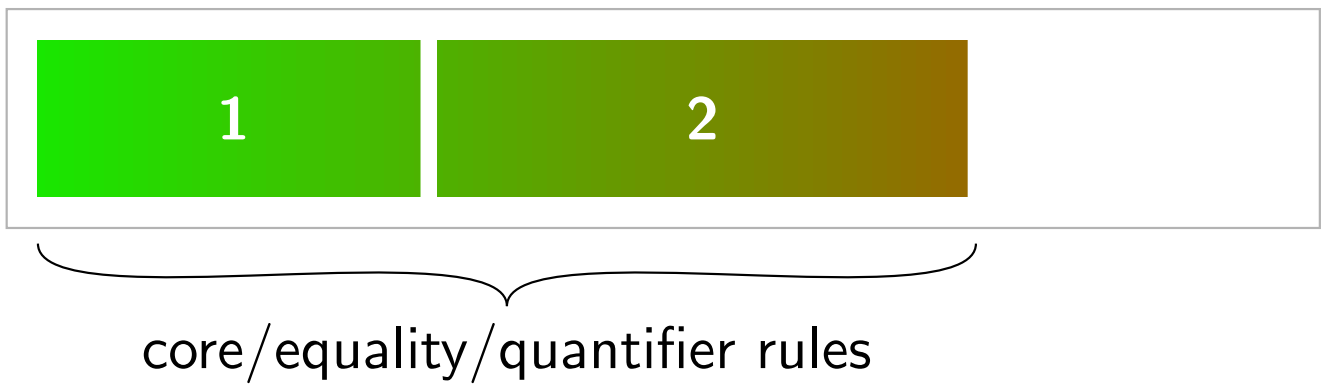
Reconstruction techniques:

1. single primitive inference rule or theorem instantiation
2. specific proof tools

# Fast Proof Reconstruction

Reconstruction techniques:

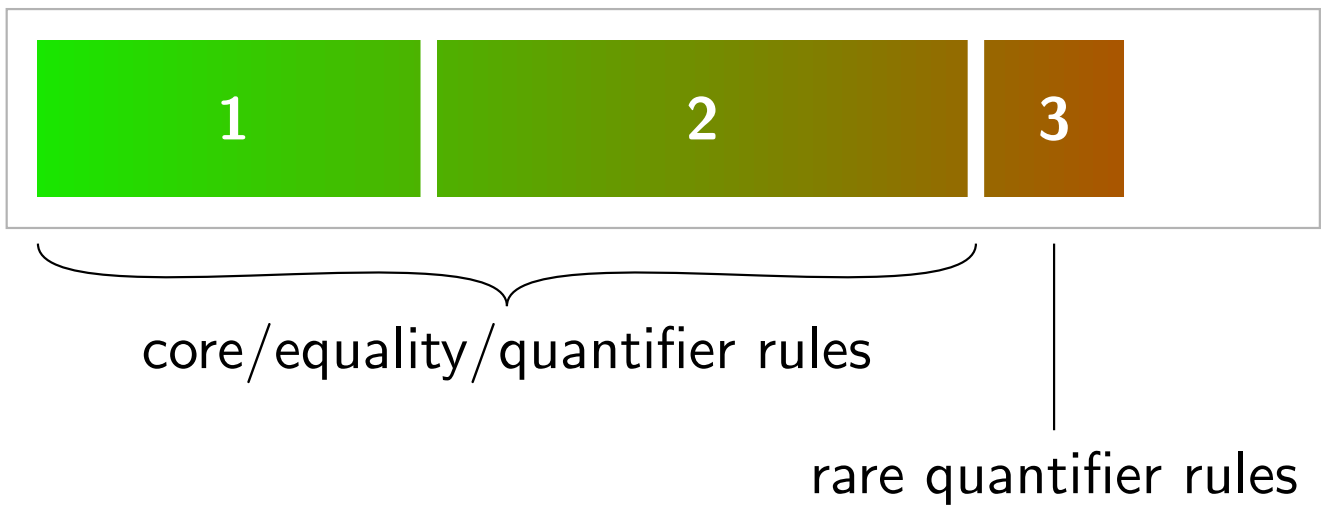
1. single primitive inference rule or theorem instantiation
2. specific proof tools



# Fast Proof Reconstruction

Reconstruction techniques:

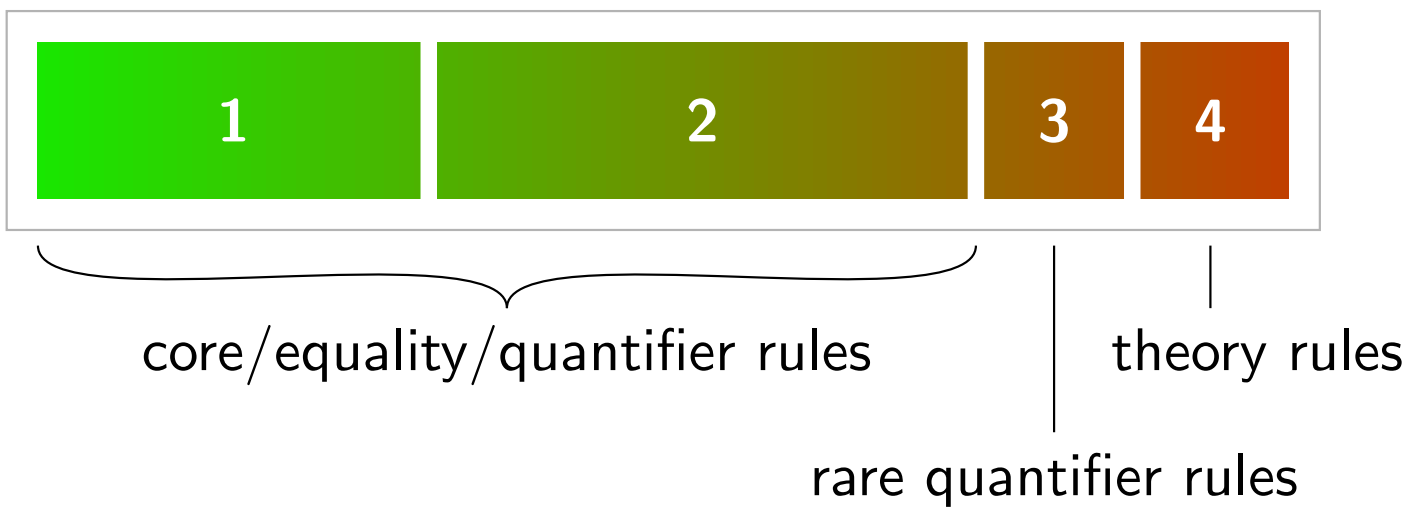
1. single primitive inference rule or theorem instantiation
2. specific proof tools
3. automated proof tools



# Fast Proof Reconstruction

Reconstruction techniques:

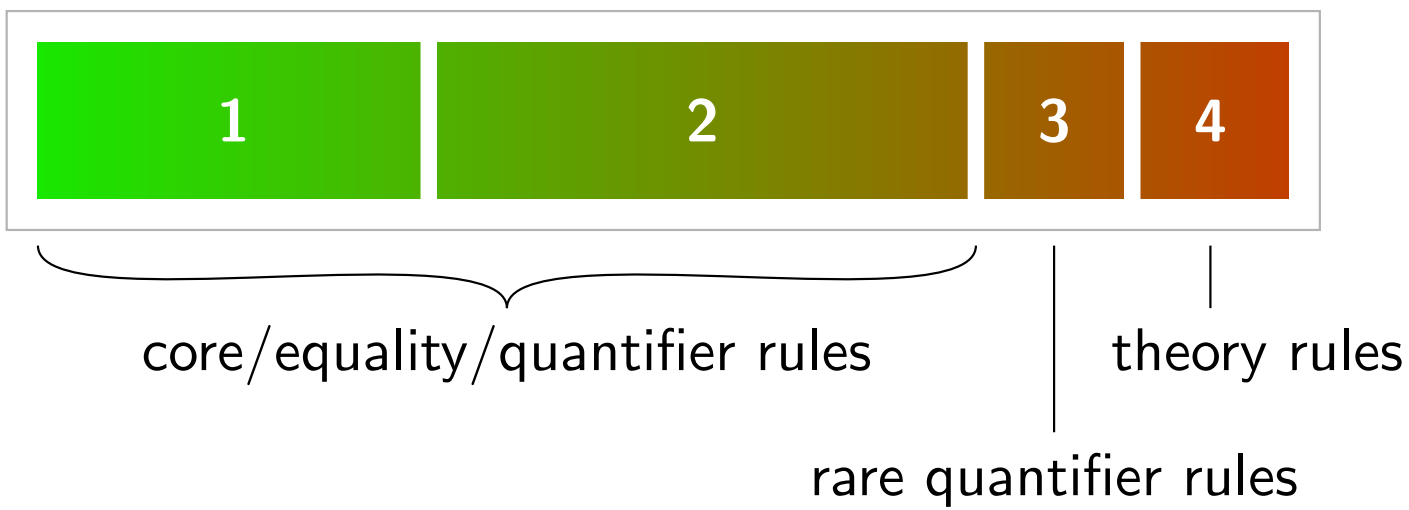
1. single primitive inference rule or theorem instantiation
2. specific proof tools
3. automated proof tools
4. combinations of the above with performance optimizations



# Fast Proof Reconstruction

Reconstruction techniques:

1. single primitive inference rule or theorem instantiation
2. specific proof tools
3. automated proof tools
4. combinations of the above with performance optimizations



# Fast Proof Reconstruction

Reconstruction techniques:

1. single primitive inference rule or theorem instantiation
2. specific proof tools
3. automated proof tools
4. combinations of the above with performance optimizations

REWRITE:

1. schematic theorems
2. conjunction/disjunction-reasoning
3. propositional reasoning
4. linear arithmetic
5. quantifier elimination for integer/real arithmetic

# Fast Proof Reconstruction

Reconstruction techniques:

1. single primitive inference rule or theorem instantiation
2. specific proof tools
3. automated proof tools
4. combinations of the above with performance optimizations

REWRITE:

1. schematic theorems
2. conjunction/disjunction-reasoning
3. propositional reasoning
4. linear arithmetic
5. quantifier elimination for integer/real arithmetic

# Fast Proof Reconstruction

Reconstruction techniques:

1. single primitive inference rule or theorem instantiation
2. specific proof tools
3. automated proof tools
4. combinations of the above with performance optimizations

Rewrite:

1. schematic theorems
2. conjunction/disjunction-reasoning
3. propositional reasoning
4. linear arithmetic
5. quantifier elimination for integer/real arithmetic

# Specific Proof Tools

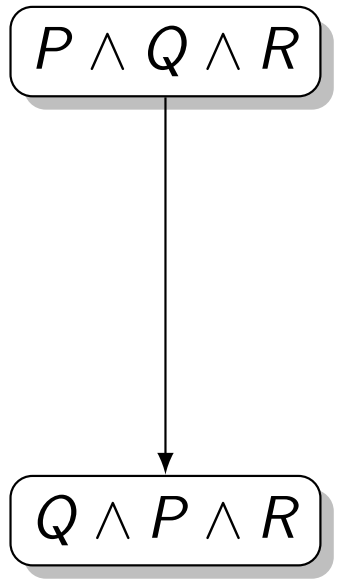
## Example: REWRITE – equivalence of conjunctions

$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$

# Specific Proof Tools

## Example: REWRITE – equivalence of conjunctions

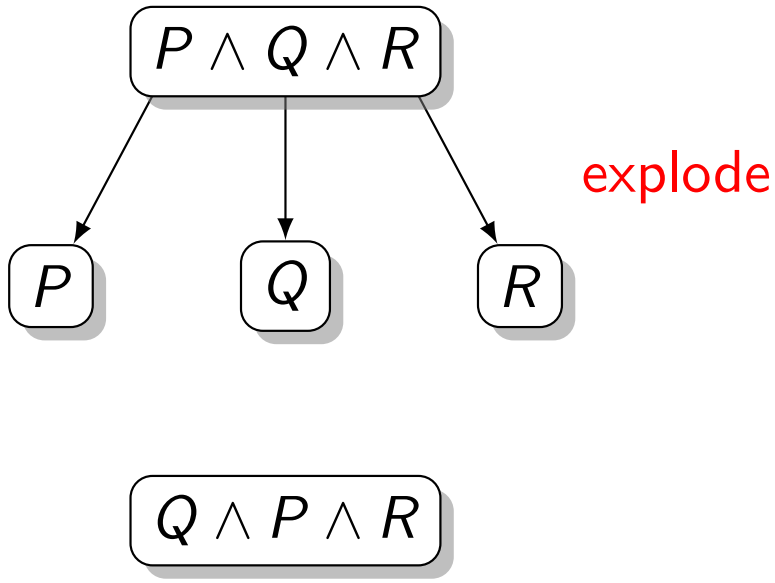
$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$



# Specific Proof Tools

## Example: REWRITE – equivalence of conjunctions

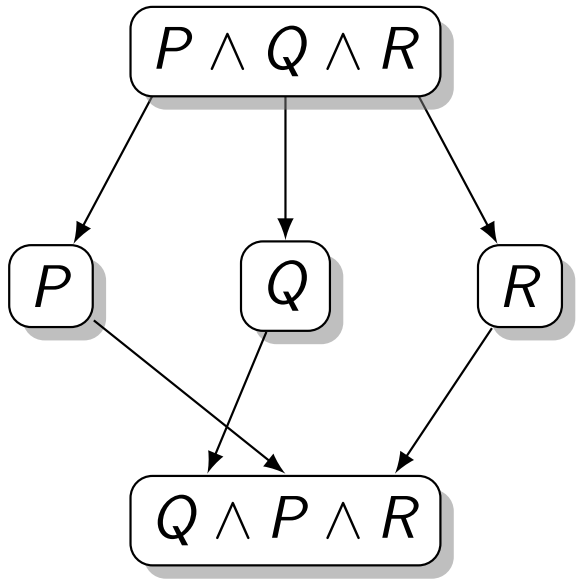
$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$



# Specific Proof Tools

## Example: REWRITE – equivalence of conjunctions

$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$

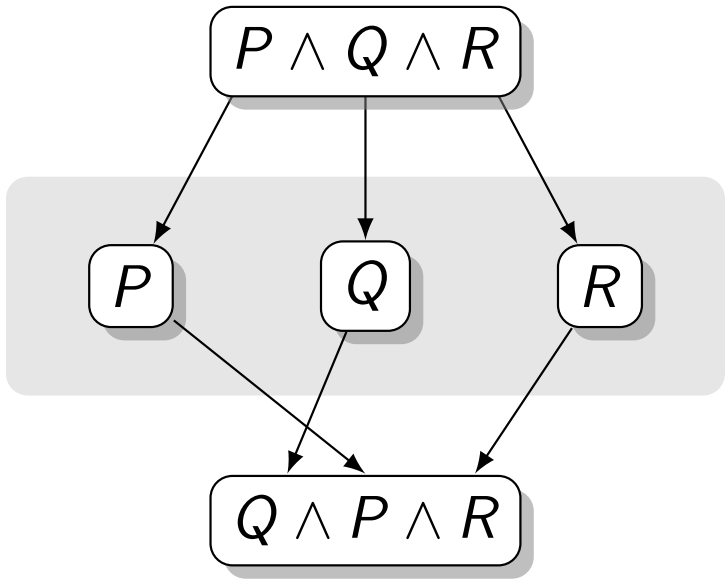



join

# Specific Proof Tools

## Example: REWRITE – equivalence of conjunctions

$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$



# Specific Proof Tools

## Example: REWRITE – equivalence of conjunctions

$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$

## Example: REWRITE – contradiction

$$P \wedge Q \wedge \neg P \longleftrightarrow \text{False}$$

# Specific Proof Tools

## Example: REWRITE – equivalence of conjunctions

$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$

## Example: REWRITE – contradiction

$$P \wedge Q \wedge \neg P \longleftrightarrow \text{False}$$

## Example: NOT-OR-ELIM

$$\frac{\neg(P_1 \vee \ldots \vee P_i \vee \ldots \vee P_n)}{\neg P_i}$$

# Specific Proof Tools

## Example: REWRITE – equivalence of conjunctions

$$P \wedge Q \wedge R \longleftrightarrow Q \wedge P \wedge R$$

## Example: REWRITE – contradiction

$$P \wedge Q \wedge \neg P \longleftrightarrow \text{False}$$

## Example: NOT-OR-ELIM

$$\frac{\neg(P_1 \vee \ldots \vee P_i \vee \ldots \vee P_n)}{\neg P_i}$$

## Example: UNIT-RESOLUTION

$$\frac{P \vee \neg Q \vee R \qquad Q}{P \vee R}$$

# Performance Optimizations

# Performance Optimizations

Schematic theorems:

- instantiation is much faster than proving
- collection of over 200 theorems
- prove 76% of all REWRITE steps

# Performance Optimizations

Schematic theorems:

- instantiation is much faster than proving
- collection of over 200 theorems
- prove 76% of all Rewrite steps

Theorem memoization:

- keep theorems deduced by Rewrite and Th-Lemma
- dynamically enhance set of schematic theorems

# Performance Optimizations

Schematic theorems:

- instantiation is much faster than proving
- collection of over 200 theorems
- prove 76% of all REWRITE steps

Theorem memoization:

- keep theorems deduced by REWRITE and TH-LEMMA
- dynamically enhance set of schematic theorems

Generalization:

- replace complex expressions by fresh variables
- avoids expensive preprocessing in the decision procedures
- enables shallow reasoning, more potential for theorem re-use

# Evaluation

| Logic | Z3 | | | Isabelle | | Rates | |
|---|---|---|---|---|---|---|---|
| | # | Med. Time | Med. Size | # | Med. Time | Succ. | Time-out |
| AUFLIA+p | 187 | 0.03 s | 5 KB | 187 | 0.06 s | 100% | 0% |
| AUFLIA−p | 192 | 0.04 s | 4 KB | 190 | 0.06 s | 98% | 0% |
| AUFLIRA | 189 | 0.02 s | 16 KB | 144 | 0.04 s | 76% | 0% |
| QF_AUFLIA | 92 | 0.02 s | 132 KB | 49 | 2.82 s | 53% | 42% |
| QF_IDL | 40 | 0.27 s | 2 MB | 19 | 3.25 s | 47% | 52% |
| QF_LIA | 100 | 3.74 s | 10 MB | 26 | 45.27 s | 26% | 65% |
| QF_LRA | 88 | 0.15 s | 1 MB | 55 | 23.87 s | 62% | 36% |
| QF_RDL | 52 | 0.70 s | 1 MB | 26 | 19.44 s | 50% | 50% |
| QF_UF | 87 | 0.87 s | 4 MB | 73 | 3.40 s | 83% | 16% |
| QF_UFIDL | 55 | 0.30 s | 2 MB | 8 | 3.59 s | 14% | 85% |
| QF_UFLIA | 91 | 0.02 s | 107 KB | 85 | 2.12 s | 93% | 6% |
| QF_UFLRA | 100 | 0.06 s | 699 KB | 100 | 3.35 s | 100% | 0% |
| Total | 1273 | 3.66 s | 13 MB | 962 | 11.31 s | 75% | 19% |

# Evaluation

| Logic | Z3 | | | Isabelle | | Rates | |
|---|---|---|---|---|---|---|---|
| | # | Med. Time | Med. Size | # | Med. Time | Succ. | Time-out |
| AUFLIA+p | 187 | 0.03 s | 5 KB | 187 | 0.06 s | 100% | 0% |
| AUFLIA−p | 192 | 0.04 s | 4 KB | 190 | 0.06 s | 98% | 0% |
| AUFLIRA | 189 | 0.02 s | 16 KB | 144 | 0.04 s | 76% | 0% |
| QF_AUFLIA | 92 | 0.02 s | 132 KB | 49 | 2.82 s | 53% | 42% |
| QF_IDL | 40 | 0.27 s | 2 MB | 19 | 3.25 s | 47% | 52% |
| QF_LIA | 100 | 3.74 s | 10 MB | 26 | 45.27 s | 26% | 65% |
| QF_LRA | 88 | 0.15 s | 1 MB | 55 | 23.87 s | 62% | 36% |
| QF_RDL | 52 | 0.70 s | 1 MB | 26 | 19.44 s | 50% | 50% |
| QF_UF | 87 | 0.87 s | 4 MB | 73 | 3.40 s | 83% | 16% |
| QF_UFIDL | 55 | 0.30 s | 2 MB | 8 | 3.59 s | 14% | 85% |
| QF_UFLIA | 91 | 0.02 s | 107 KB | 85 | 2.12 s | 93% | 6% |
| QF_UFLRA | 100 | 0.06 s | 699 KB | 100 | 3.35 s | 100% | 0% |
| Total | 1273 | 3.66 s | 13 MB | 962 | 11.31 s | 75% | 19% |

# Evaluation

SMT-LIB benchmarks:

| Z3 | | | Isabelle | | Rates | |
|---|---|---|---|---|---|---|
| # | Median Time | Median Size | # | Median Time | Succ. | T-out |
| 1273 | 3.66 s | 13 MB | 962 | 11.31 s | 75% | 19% |

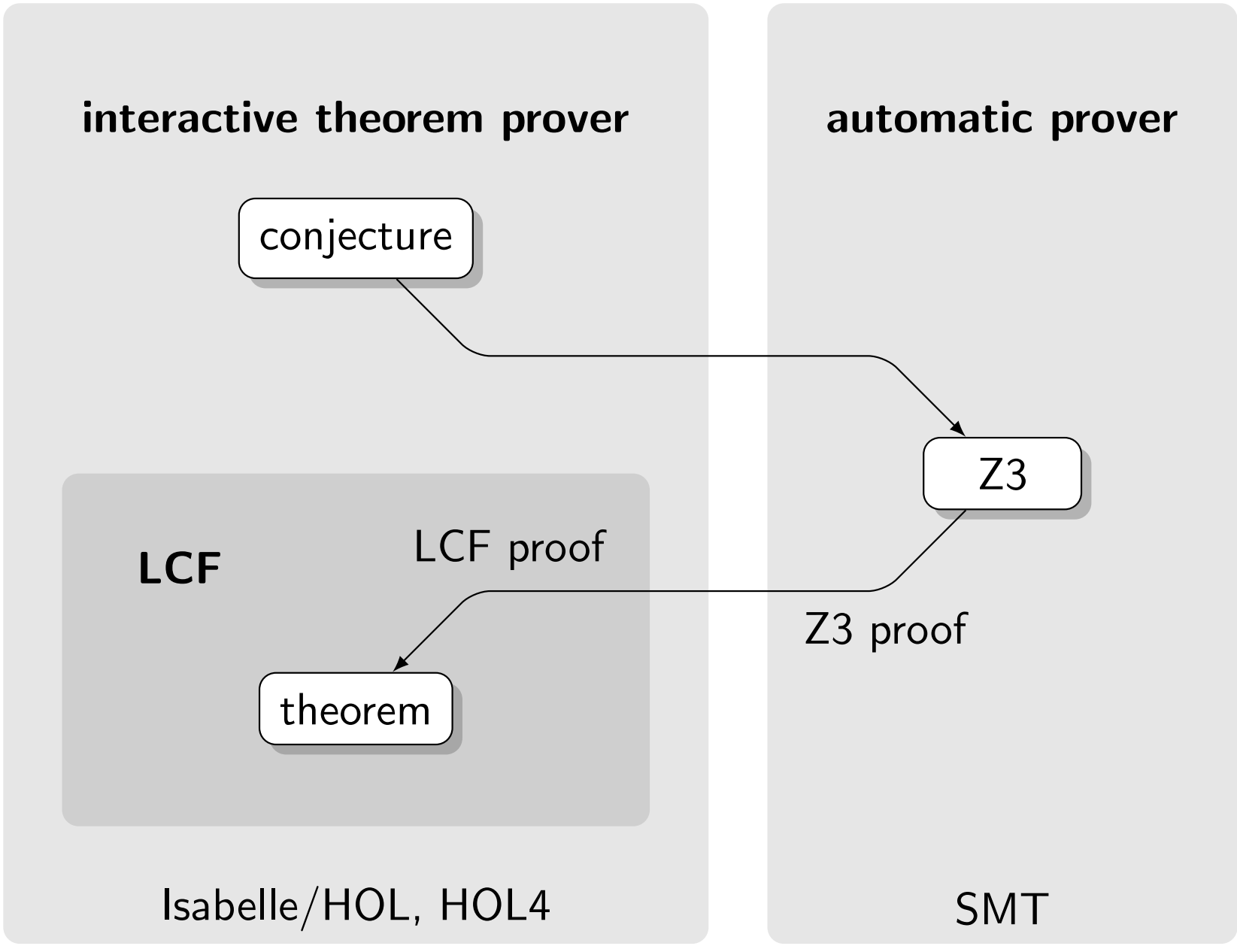
# Evaluation

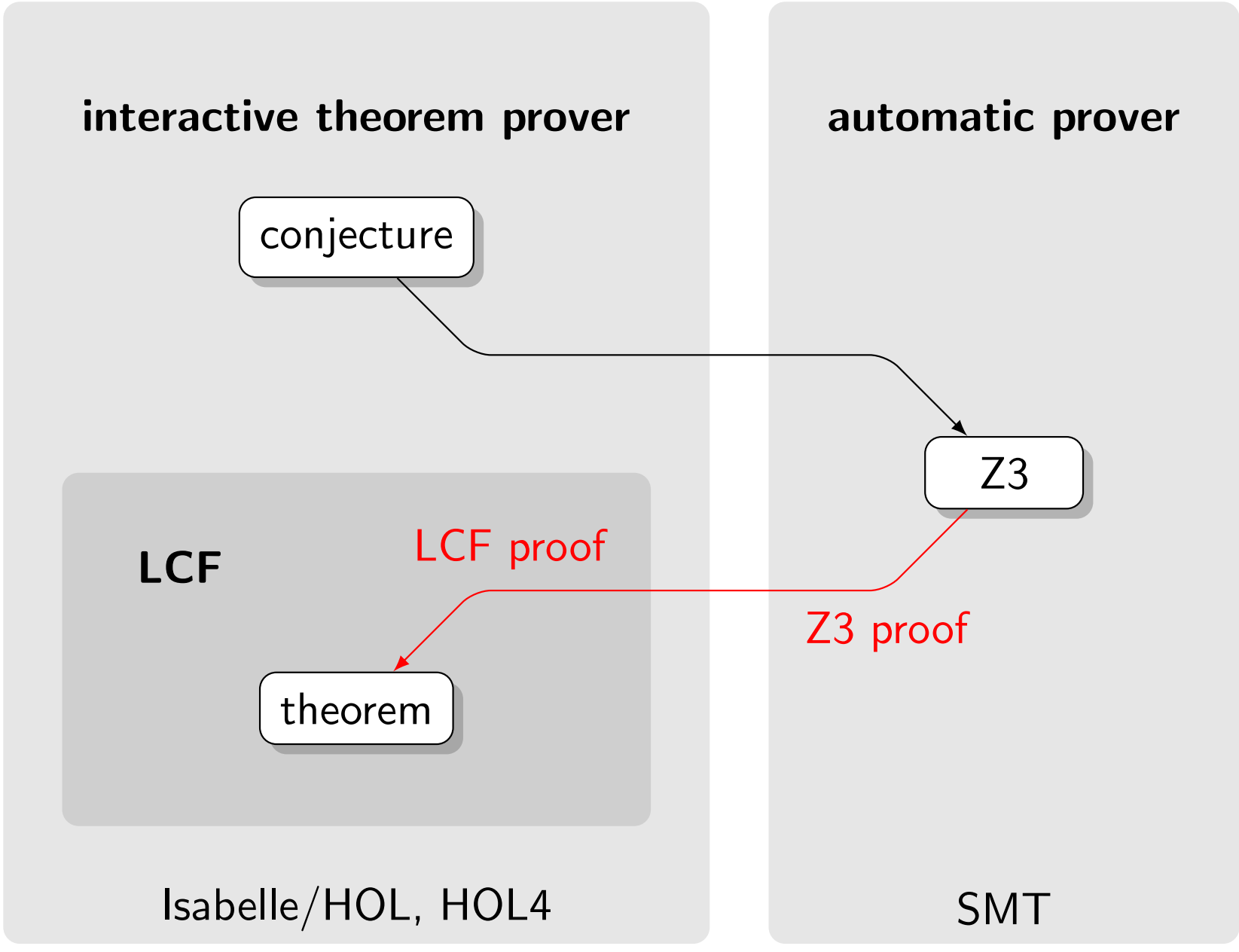
SMT-LIB benchmarks:

| Z3 | | | Isabelle | | Rates | |
|---|---|---|---|---|---|---|
| # | Median Time | Median Size | # | Median Time | Succ. | T-out |
| 1273 | 3.66 s | 13 MB | 962 | 11.31 s | 75% | 19% |

Profiling:

- highly optimized implementation
- bottleneck: theory reasoning requires expensive proof search
- 50% of the runtime is spent on 15% of all Z3 proof steps


interactive theorem prover
automatic prover
conjecture
Z3
LCF
LCF proof
Z3 proof
theorem
Isabelle/HOL, HOL4
SMT


interactive theorem prover
automatic prover
conjecture
Z3
LCF
LCF proof
Z3 proof
theorem
Isabelle/HOL, HOL4
SMT

# Conclusion

LCF-style proof reconstruction for Z3:

- highly optimized
- feasible and efficient
- slower than proof search in Z3
- Z3's proofs could be easier to check (costly theory reasoning)

Benefits for Isabelle/HOL and HOL4:

- powerful automation

Benefits for Z3:

- proof checker
- identification of several bugs

# Some Identified Bugs in Z3

# Some Identified Bugs in Z3

## Proof bug

$$\overline{(\forall\, x\, y\, z.\, f\ x\ y = f\ x\ z \longrightarrow y = z) \longleftrightarrow (\forall\, a\, b.\, ?g\ (f\ b\ a) = a)}$$

# Some Identified Bugs in Z3

## Proof bug

$$\frac{}{(\forall\, x\, y\, z.\, f\ x\ y = f\ x\ z \longrightarrow y = z) \longleftrightarrow (\forall\, a\, b.\, ?g\ (f\ b\ a) = a)}$$

## Documentation problem

$$\frac{s = t \qquad u = t}{s = u}\ \textsc{Trans}$$

# Some Identified Bugs in Z3

## Proof bug

$$\frac{}{(\forall\, x\, y\, z.\, f\ x\ y = f\ x\ z \longrightarrow y = z) \longleftrightarrow (\forall\, a\, b.\, ?g\ (f\ b\ a) = a)}$$

## Documentation problem

$$\frac{s = t \qquad u = t}{s = u}\ \textsc{Trans}$$

## Soundness bug

$$(\forall\, x^T.\, x = c^T)\ \wedge\ a^T \neq b^T$$