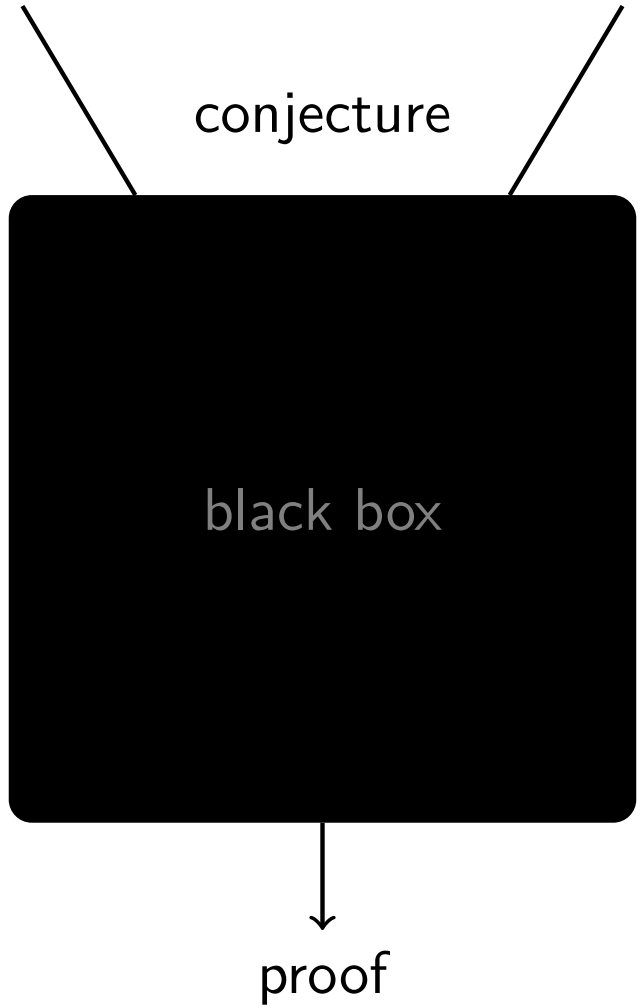
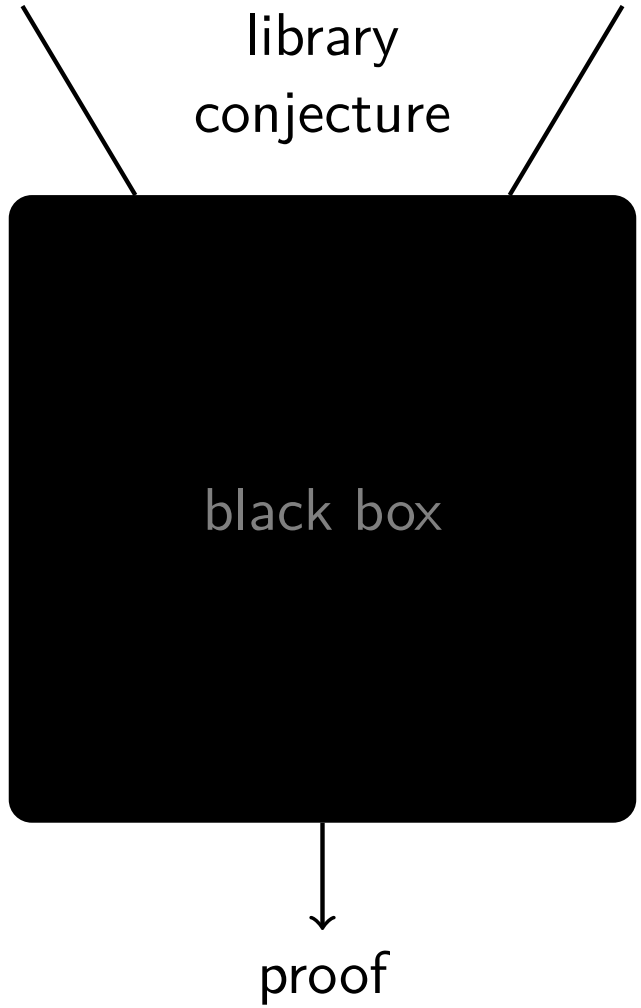
Learning-Assisted Reasoning within Interactive Theorem Provers

Thibault Gauthier

May 17, 2019


conjecture
black box
proof

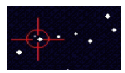

library
conjecture
black box
proof

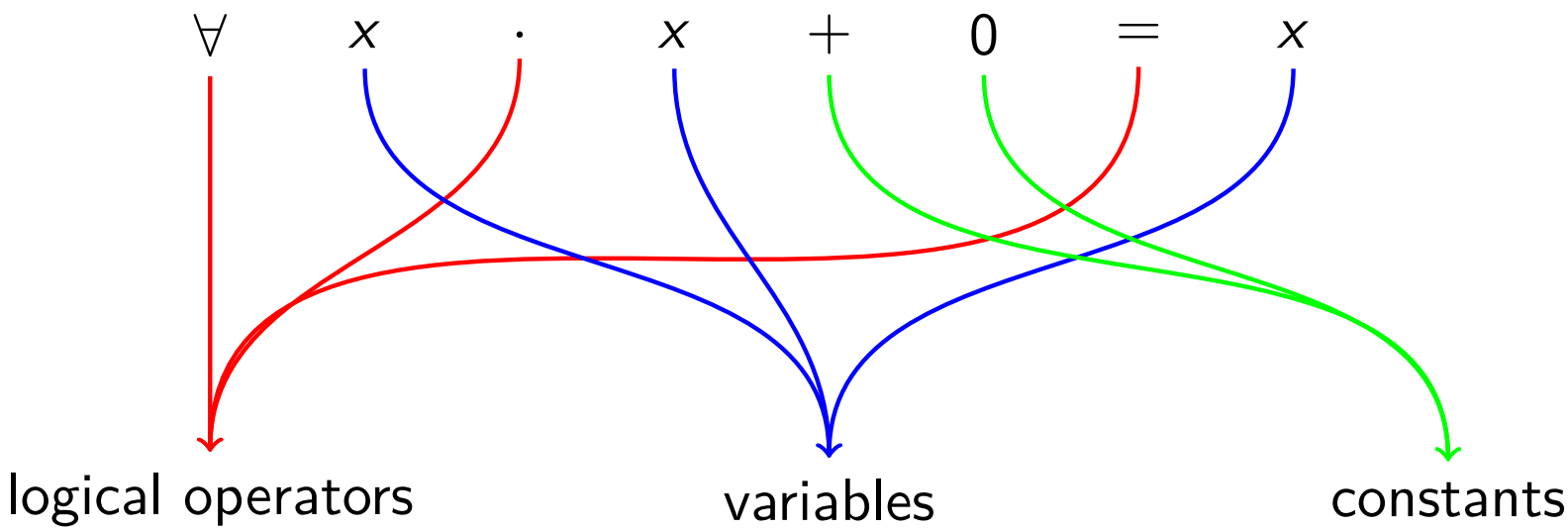
white, spherical, many petals

red, star-shaped, five petals

Object

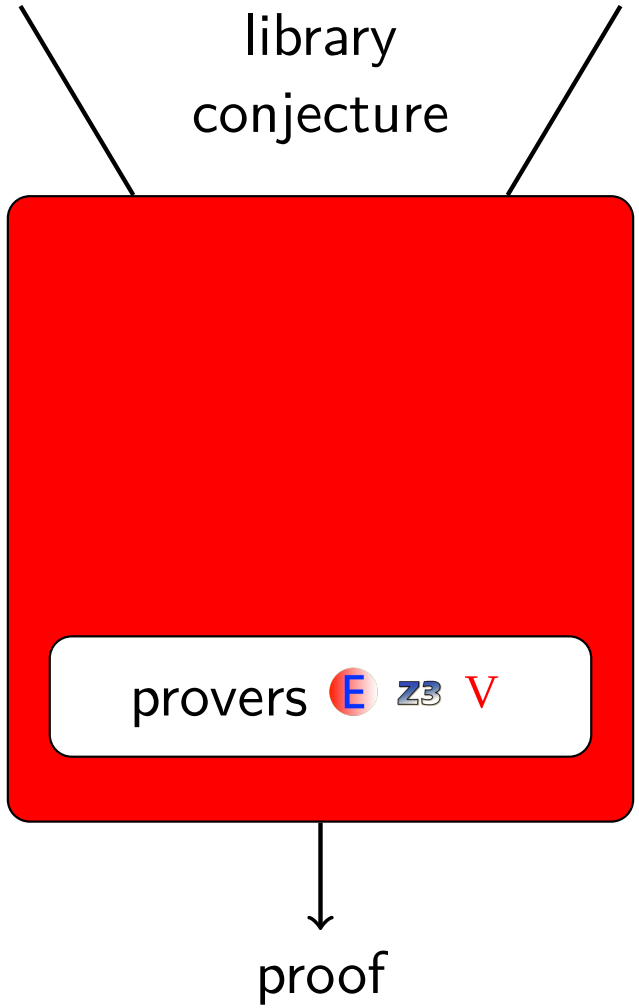
Properties

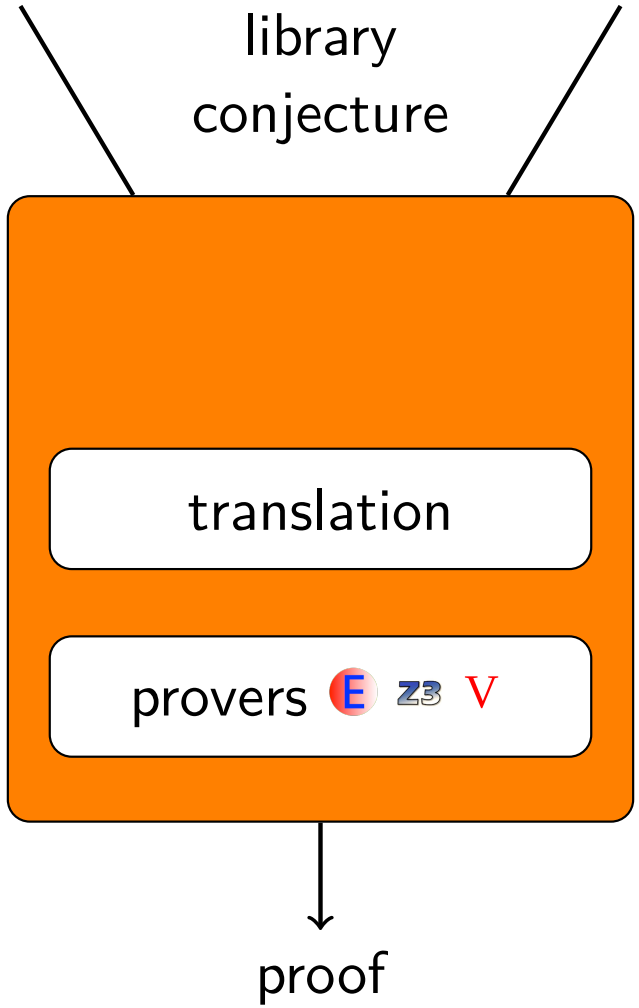
Proof Assistant	Theorems	Constants
Mizar 	51086	9172
Coq 	23320	4841
HOL4 	16476	2247
HOL Light 	16191	820
Isabelle/HOL 	14814	1076
Matita 	1712	629

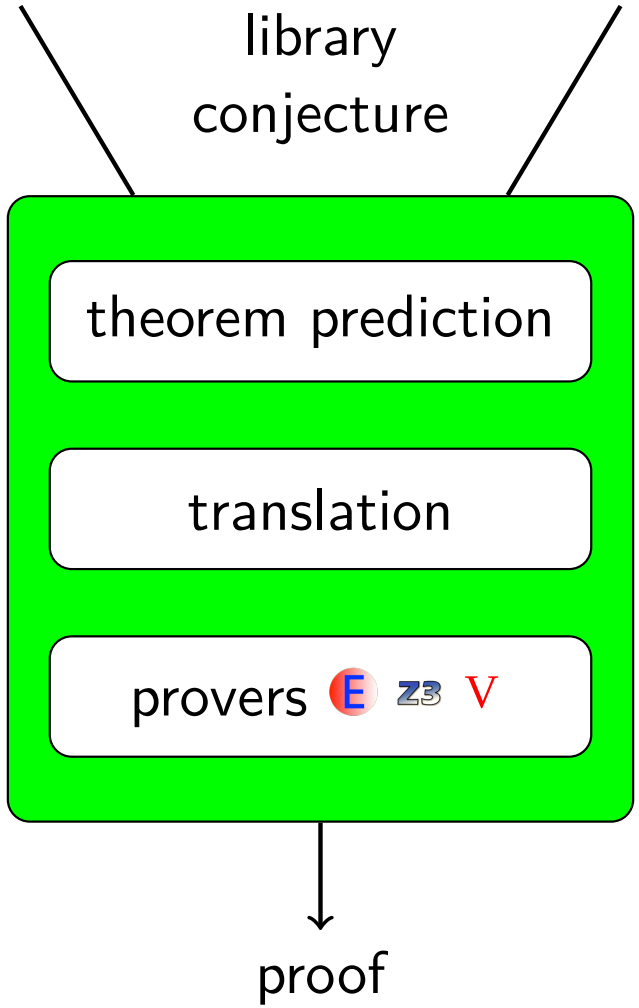

∀ x . x + 0 = x
logical operators
variables
constants


library
conjecture
proof

Demo

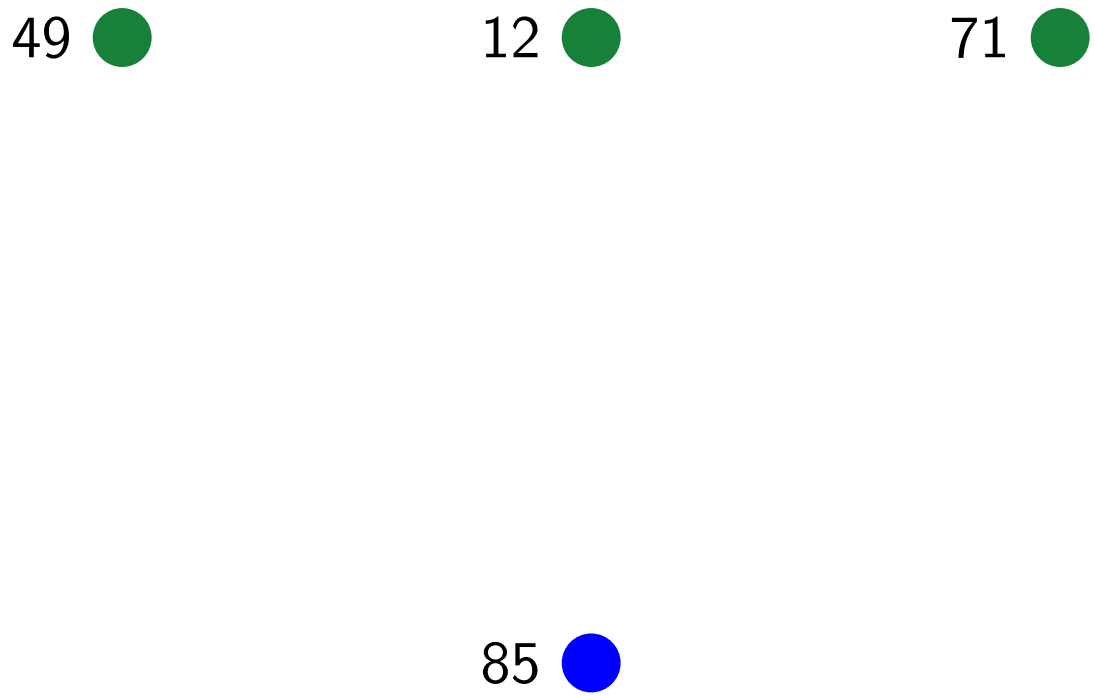

library
conjecture
provers
E
Z3
V
proof

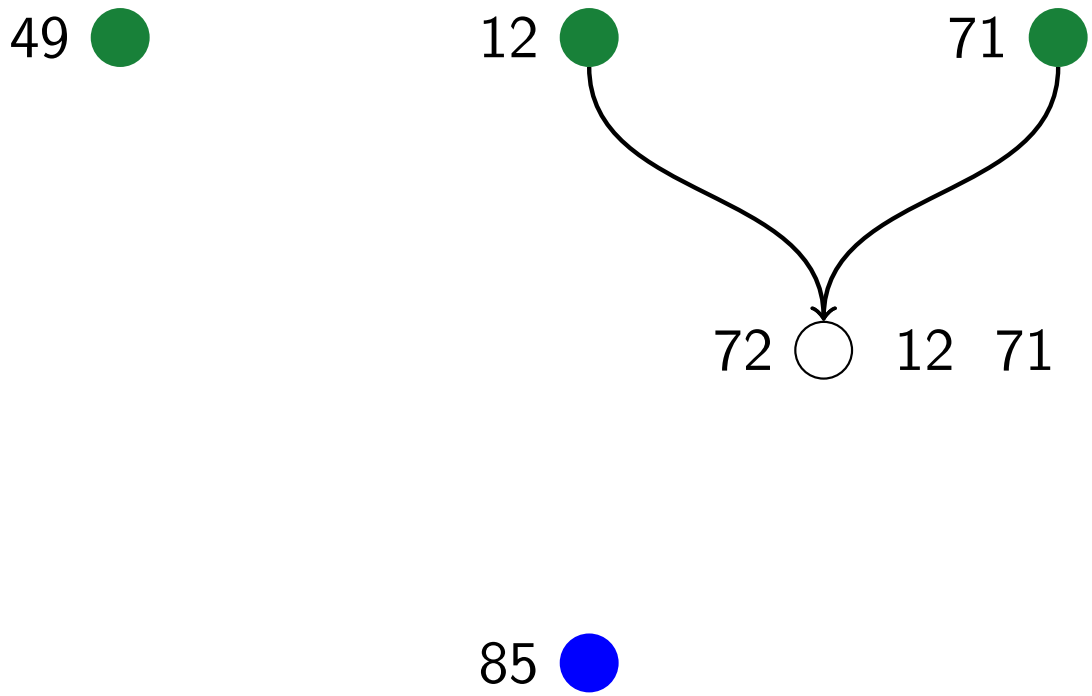

library
conjecture
translation
provers
proof


library
conjecture
theorem prediction
translation
provers E Z3 V
proof

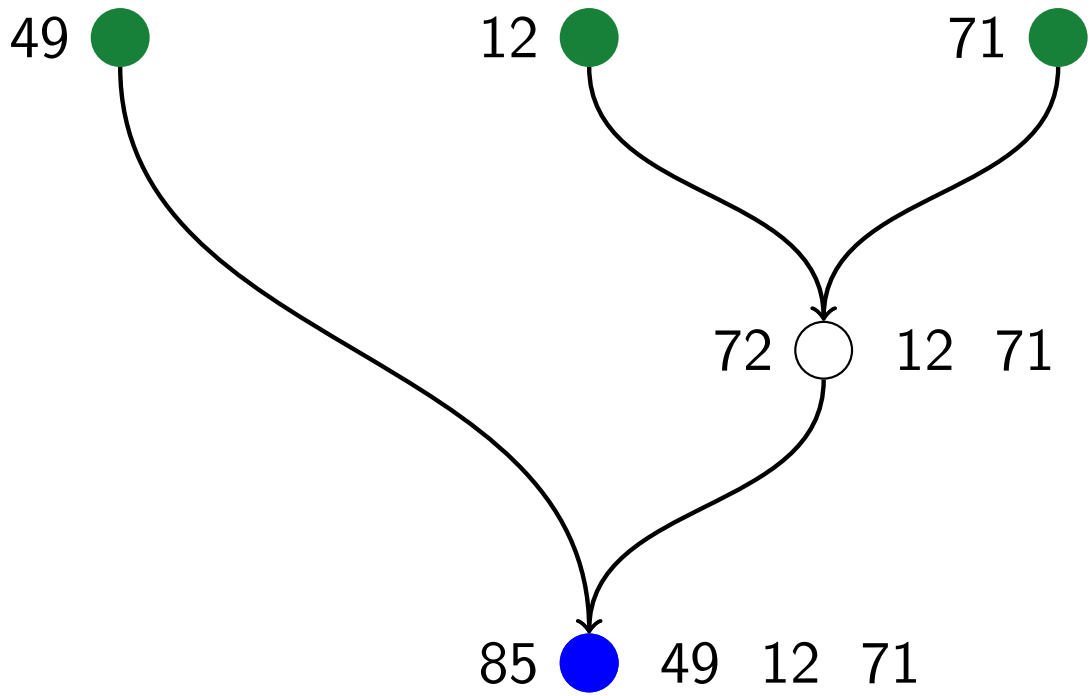
	Formula	Syntactic features
Conjecture	$\forall x, y.\ (x + y) \times (x - y) = x^2 - y^2$	
Library	$\forall x, y, z.\ x \times (y + z) = x \times y + x \times z$	
	$\forall x, y.\ x + y = y + x$	
	$\forall x, y.\ x \times y = y \times x$	
	$e^{i\pi} + 1 = 0$	
	$(x^2)' = 2 \times x$	

	Formula	Syntactic features
Conjecture	$\forall x, y.\ (x + y) \times (x - y) = x^2 - y^2$	$+, \times, \ ^2$
	$\forall x, y, z.\ x \times (y + z) = x \times y + x \times z$	$\times, +$
	$\forall x, y.\ x + y = y + x$	$+$
Library	$\forall x, y.\ x \times y = y \times x$	$\times$
	$e^{i\pi} + 1 = 0$	$e, i, \times, \pi, +, 1, 0$
	$(x^2)' = 2 \times x$	$', 2, \times, \ ^2$

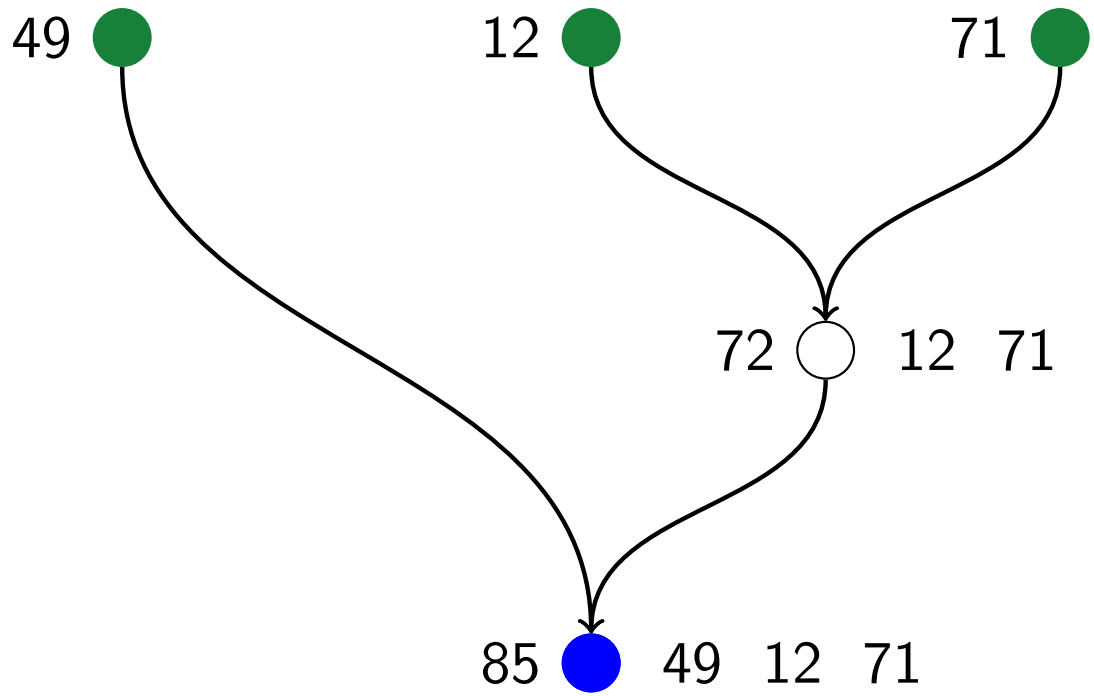

49
12
71
85
conjecture
theorem
→ rule
○ lemma



- conjecture
- theorem
- $\rightarrow$ rule
- ○ lemma



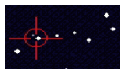
● conjecture
● theorem
→ rule
○ lemma



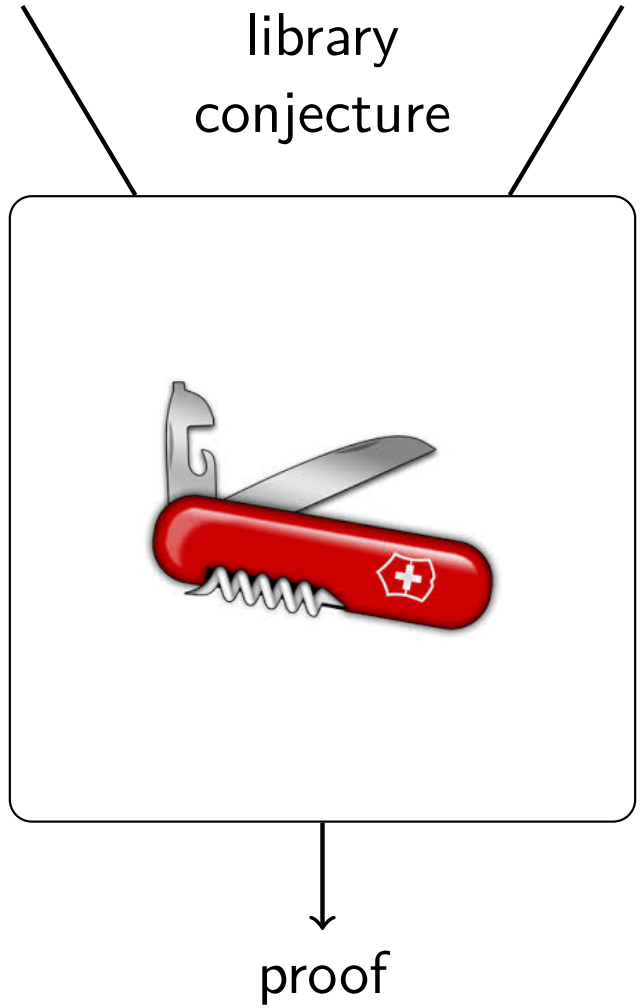
- conjecture
- theorem
- $\rightarrow$ rule
- ○ lemma

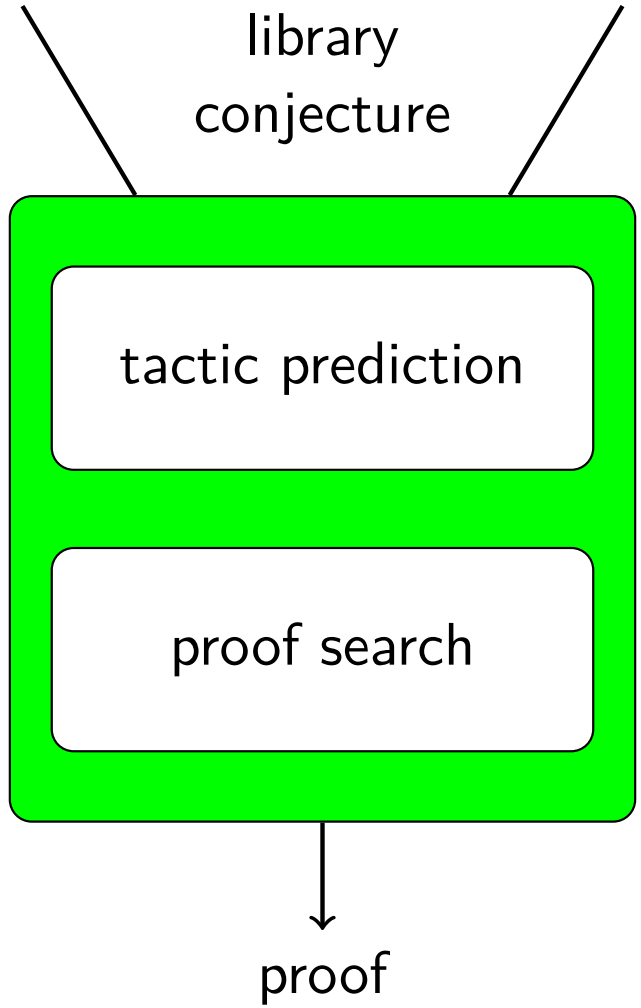
Theorem	Dependencies
85	49 12 71
102	51 45 86 12
...	...

Re-proving

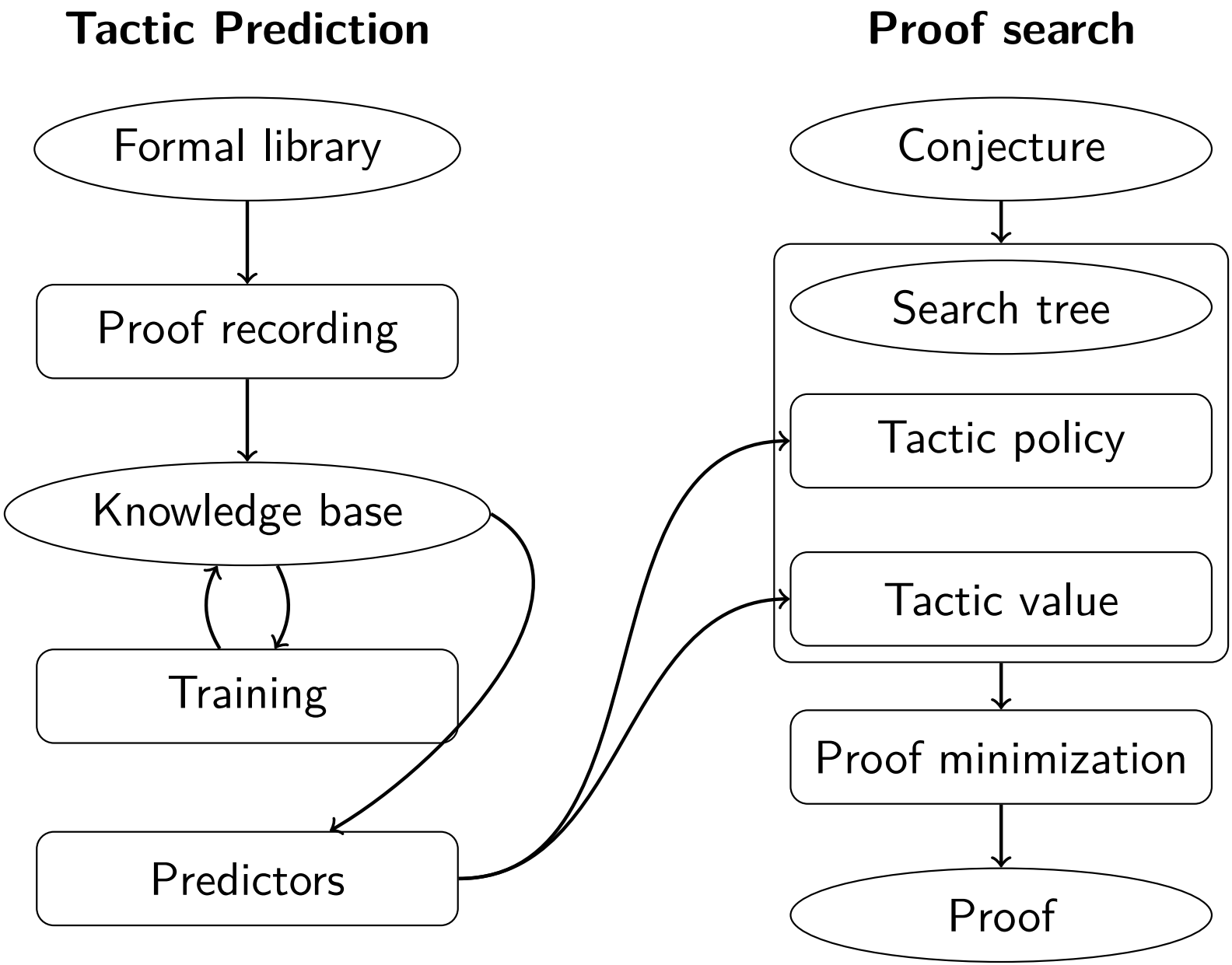
Tested library	Benchmark	Success
	standard library	40%
	judgement day	77%
	flyspeck	39%
	standard library	50%
	standard library	41%

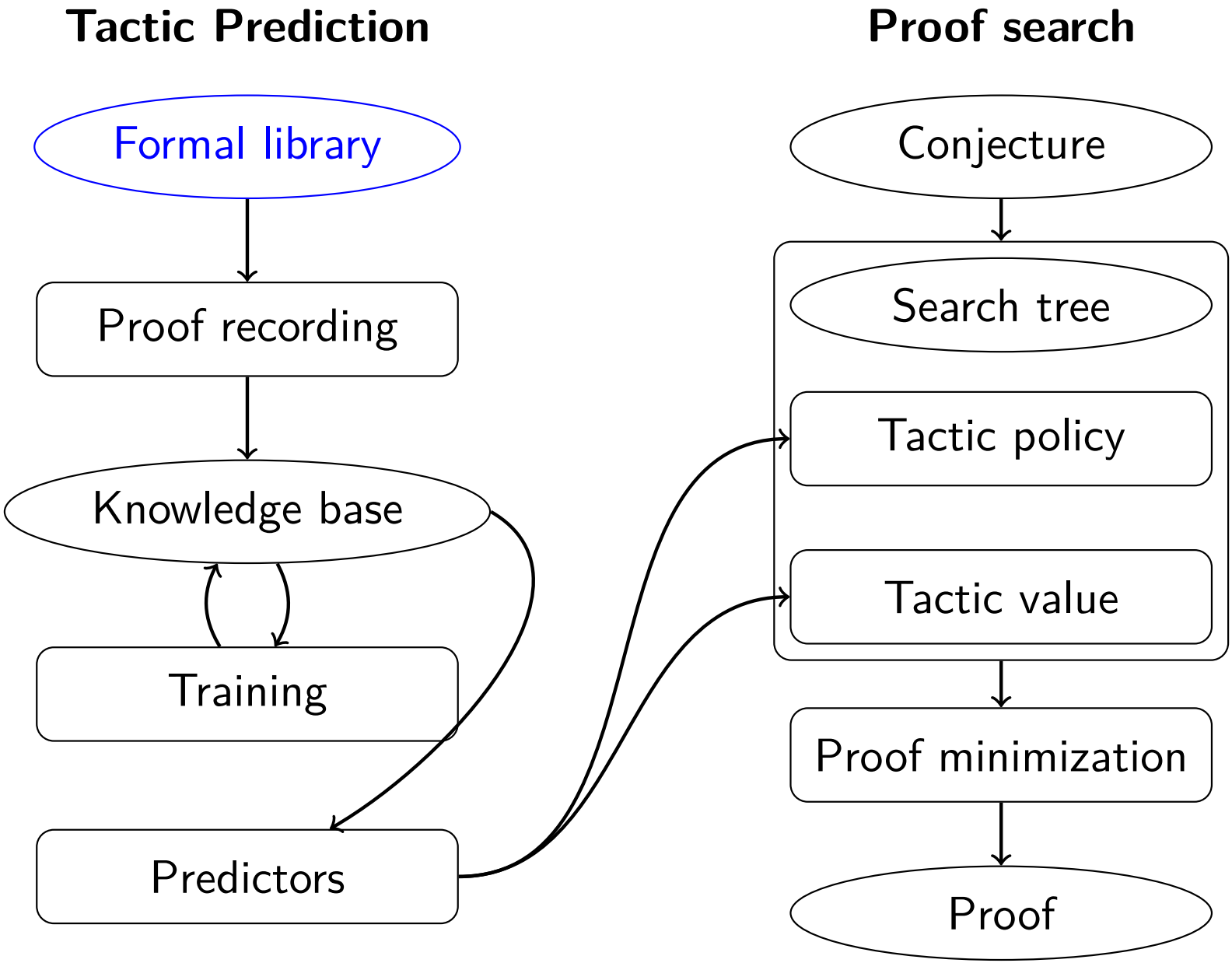
Demo

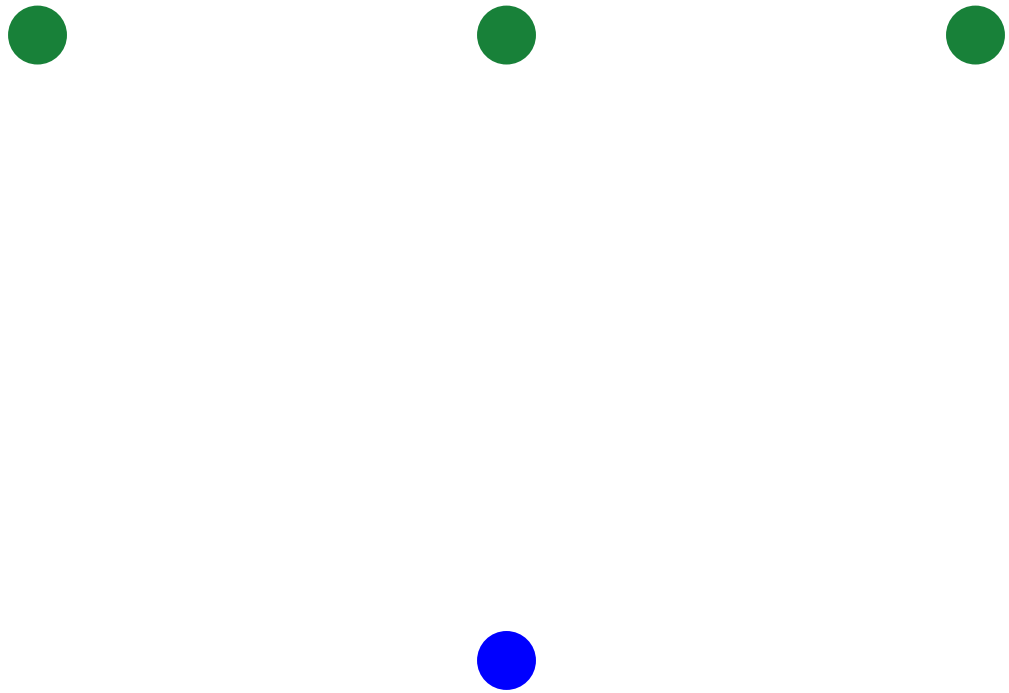

library
conjecture
proof


library
conjecture
tactic prediction
proof search
proof

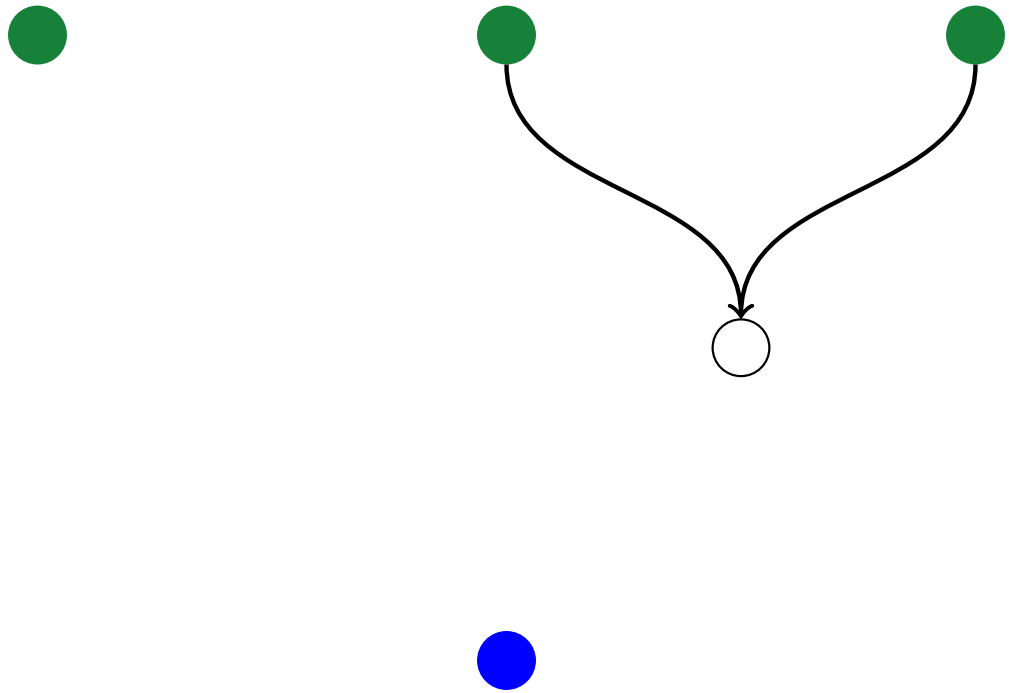
Tactics	Useful for
Solvers	linear system, differential equations
Simplifiers	irreducible fraction, differentiation
Induction	natural numbers, lists, trees


Tactic Prediction
Proof search
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Search tree
Tactic policy
Tactic value
Proof minimization
Proof

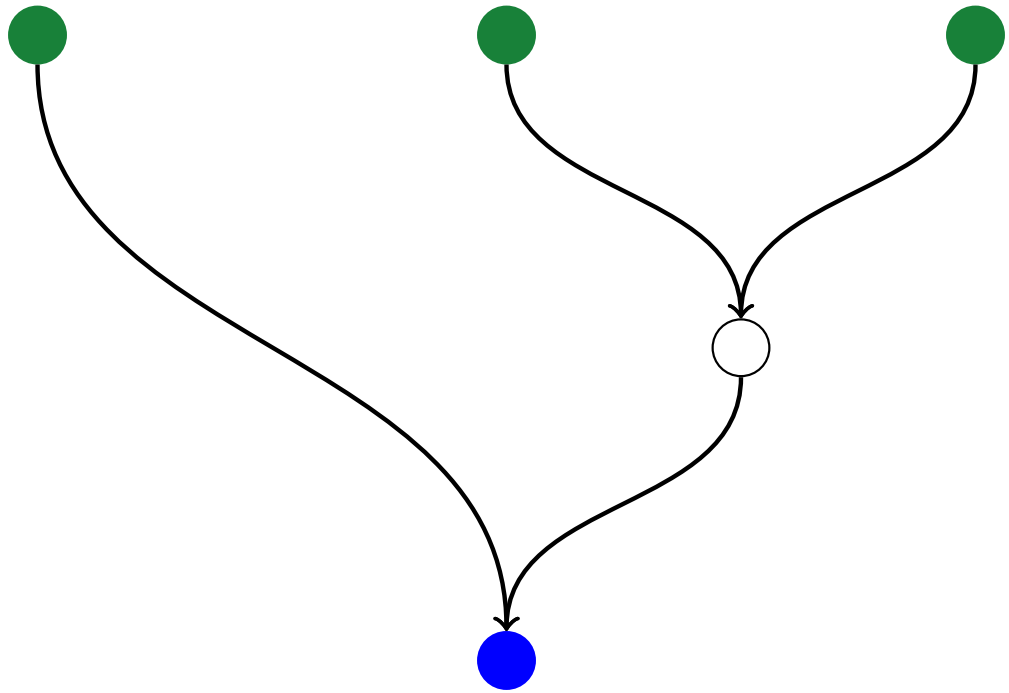

Tactic Prediction
Proof search
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Search tree
Tactic policy
Tactic value
Proof minimization
Proof



● axiom
● conjecture
→ rule
◯ lemma



axiom
conjecture
$\rightarrow$ rule
lemma

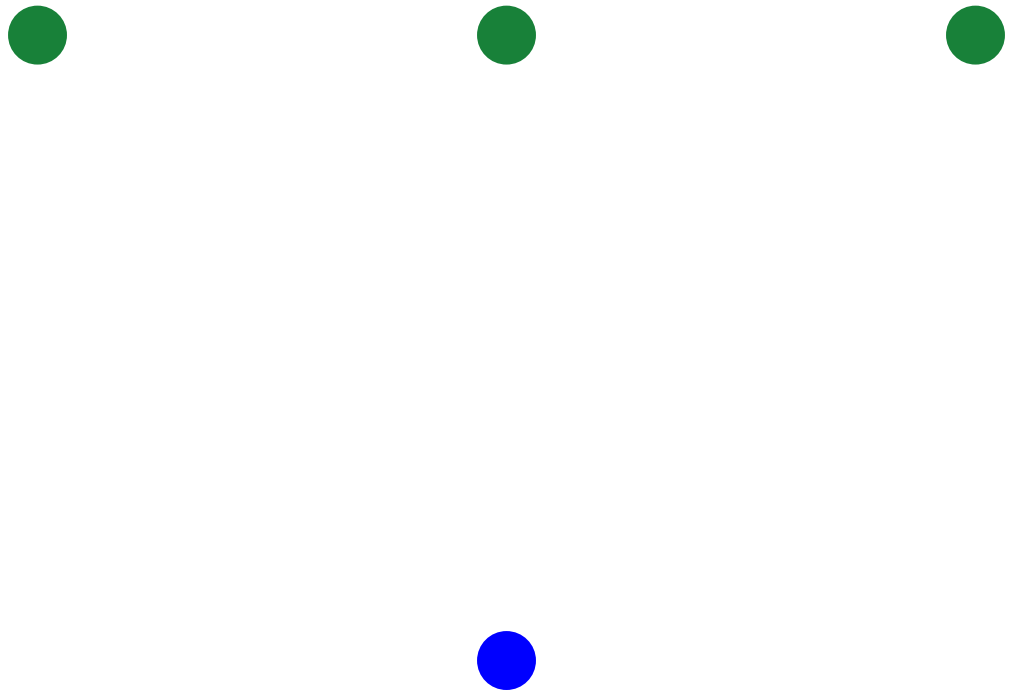


axiom

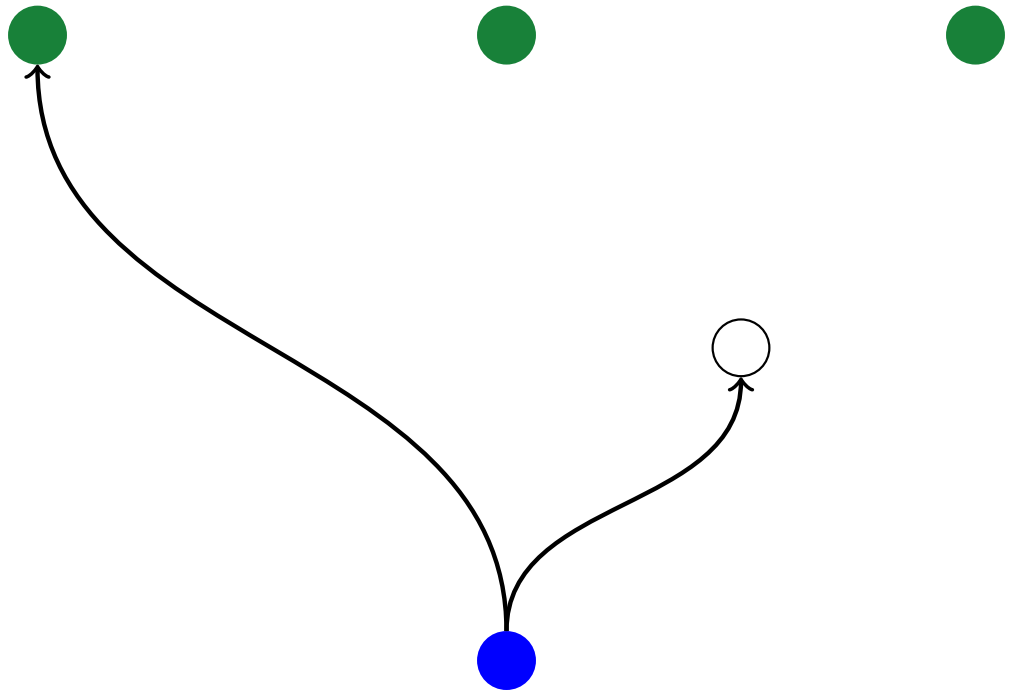
conjecture

$\rightarrow$ rule

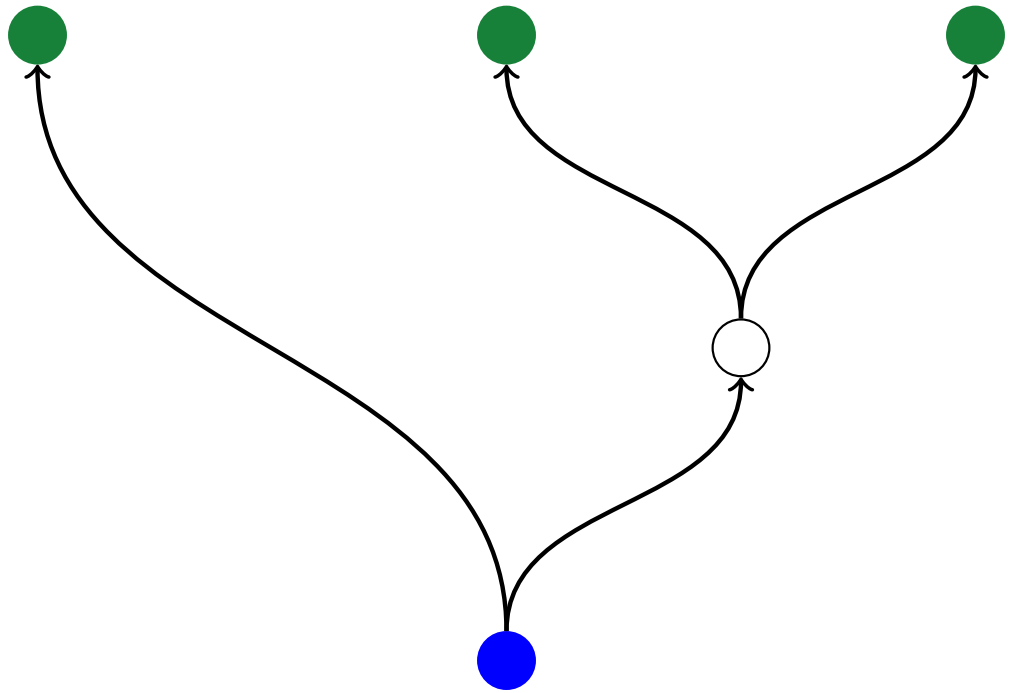
lemma



● axiom
● conjecture
→ tactic
○ goal



axiom
conjecture
$\rightarrow$ tactic
goal



axiom
conjecture
$\rightarrow$ tactic
goal

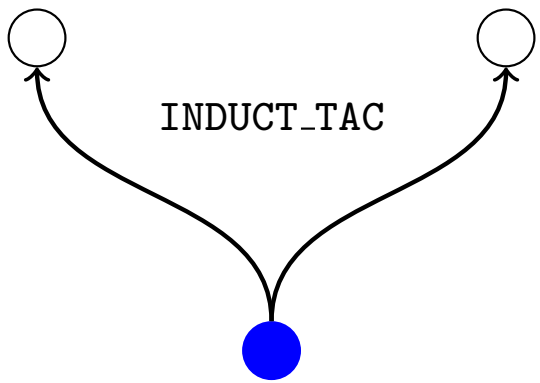
REWRITE_TAC

INDUCT_TAC

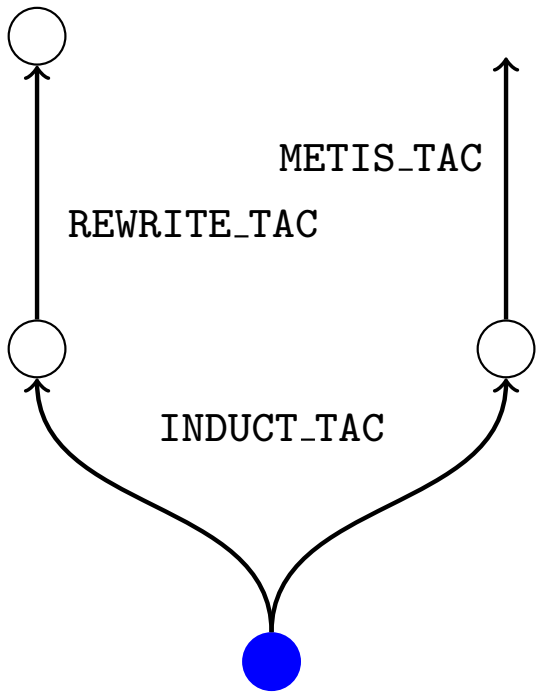
METIS_TAC

THENL tactical composes the effect of tactics.

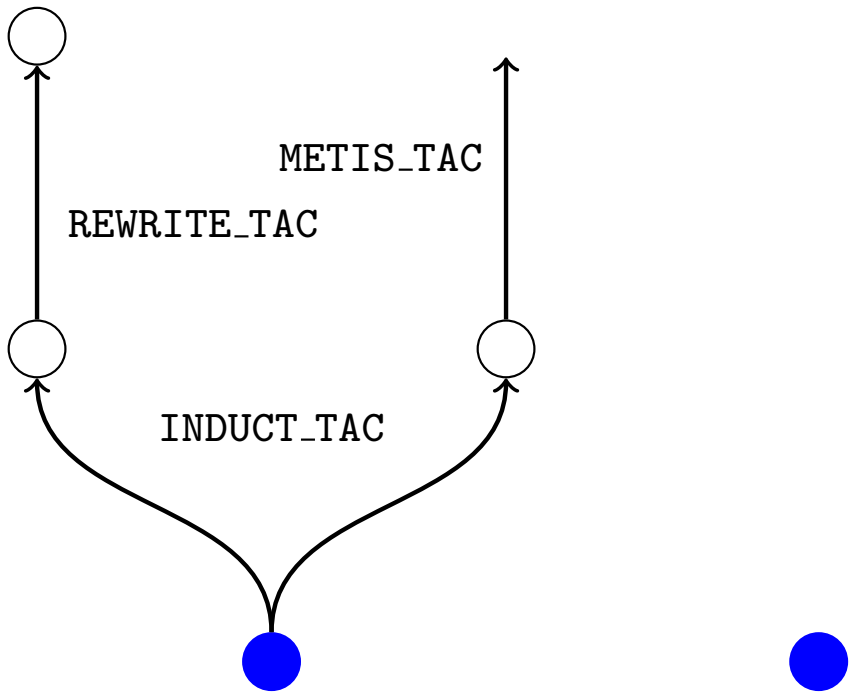
THENL tactical composes the effect of tactics.



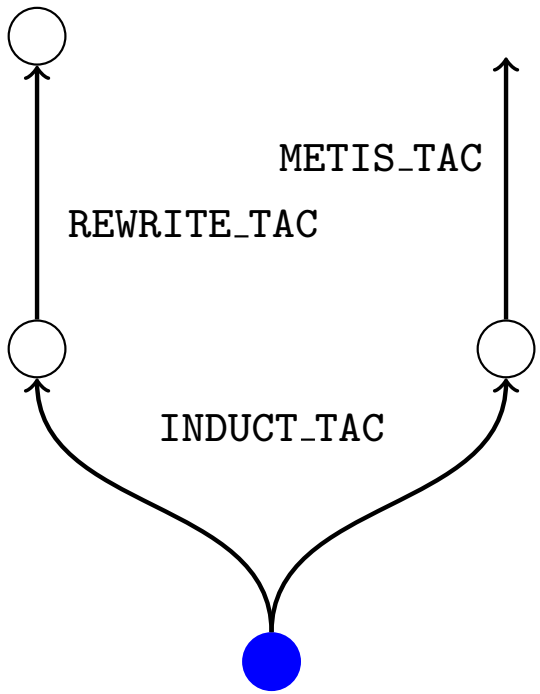
THENL tactical composes the effect of tactics.



THENL tactical composes the effect of tactics.

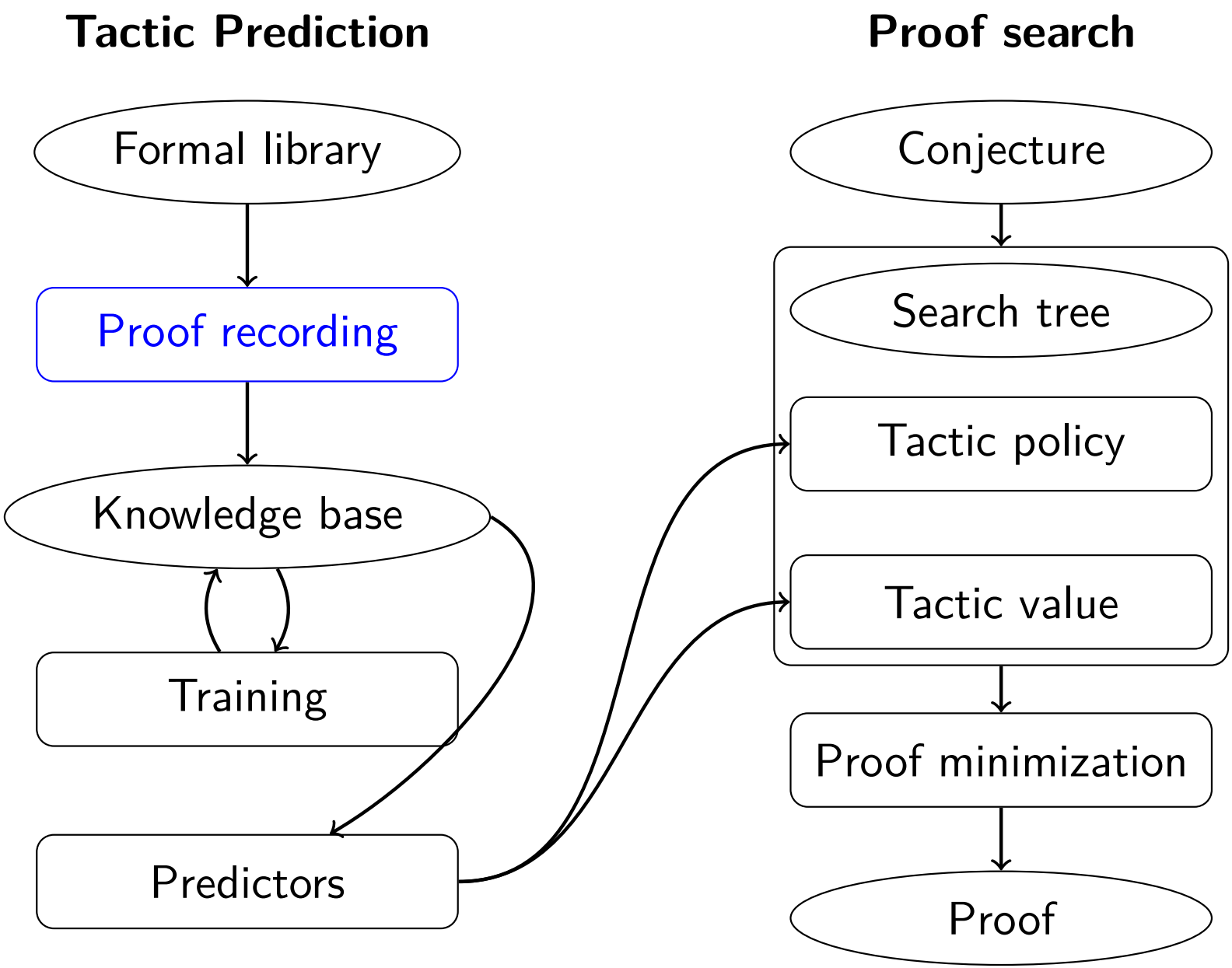


THENL tactical composes the effect of tactics.



```
INDUCT_TAC
THENL
[REWRITE_TAC,METIS_TAC]
```

Demo


Tactic Prediction
Proof search
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Search tree
Tactic policy
Tactic value
Proof minimization
Proof

Proof recording

Original proof:

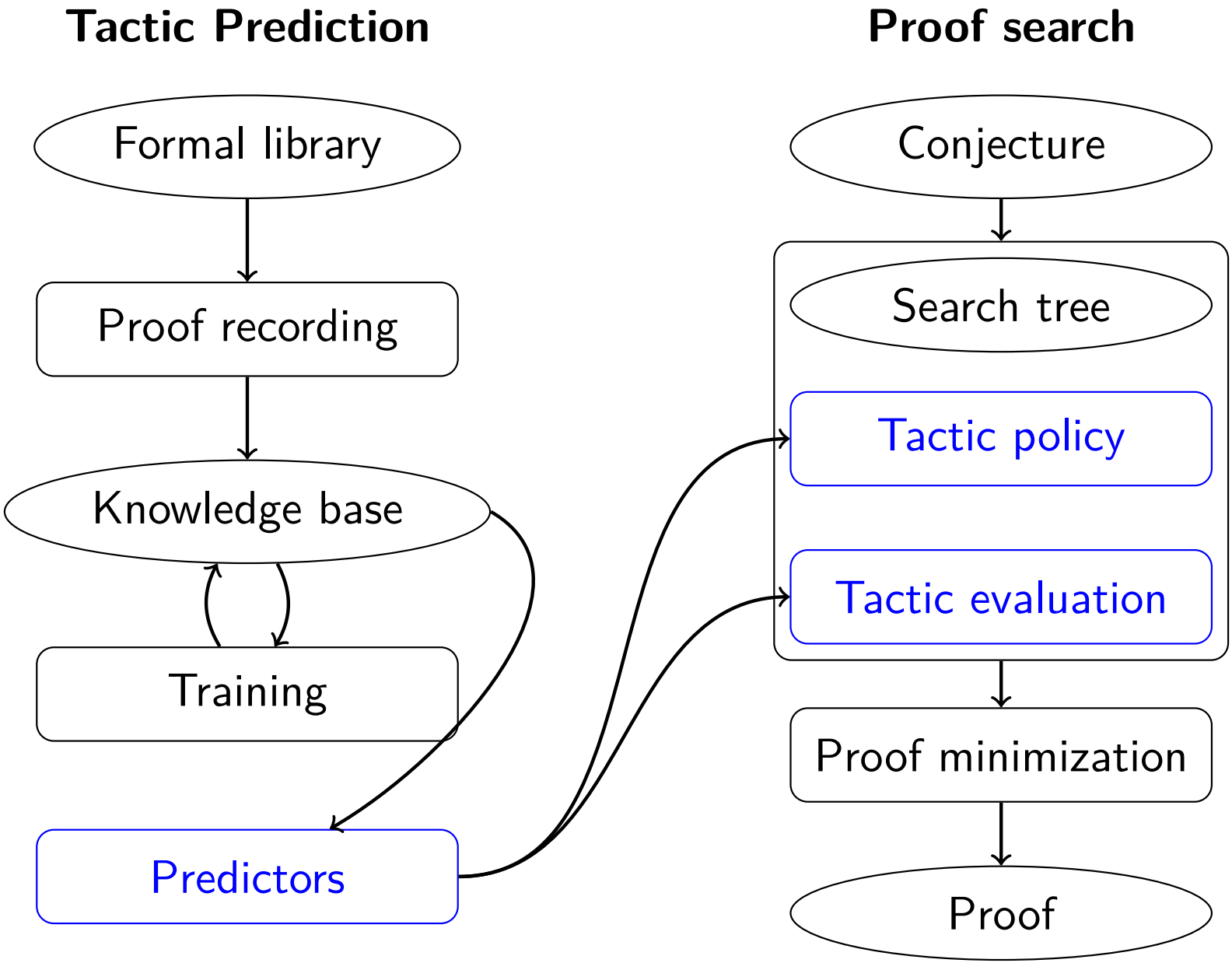
```
INDUCT_TAC THENL [REWRITE_TAC, METIS_TAC]
```

Modified proof:

```
(R numLib.INDUCT_TAC) THENL
  [R boolLib.REWRITE_TAC, R metisLib.METIS_TAC]
```

Database of tactics:

```
R (f n) (f (SUC n)) ⇒ transitive R: INDUCT_TAC
n * m ≤ n * p ⇒ (n = 0) ∨ m ≤ p   : REWRITE_TAC
INJ f U(:num) s ⇒ INFINITE s      : METIS_TAC
...
```


Tactic Prediction
Proof search
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Search tree
Tactic policy
Tactic evaluation
Proof minimization
Proof

Prediction algorithm

Algorithm:
  Nearest neighbor weighted by TF-IDF heuristics

Effect:
  Order goals from the database according to their distance to a target goal.

Remark:
  This is algorithm performs premise selection.
  How do we adapt it to predict tactics?

Policy

Database of tactics is a map from goals to tactics.

```
R (f n) (f (SUC n)) ⇒ transitive R: INDUCT_TAC
n * m ≤ n * p ⇒ (n = 0) ∨ m ≤ p   : REWRITE_TAC
INJ f U(:num) s ⇒ INFINITE s      : METIS_TAC
...
```

An order on goals induces an order on tactics.

New goal appearing during proof search:

```
LENGTH (MAP f l) = LENGTH l
```

Policy for the new goal:

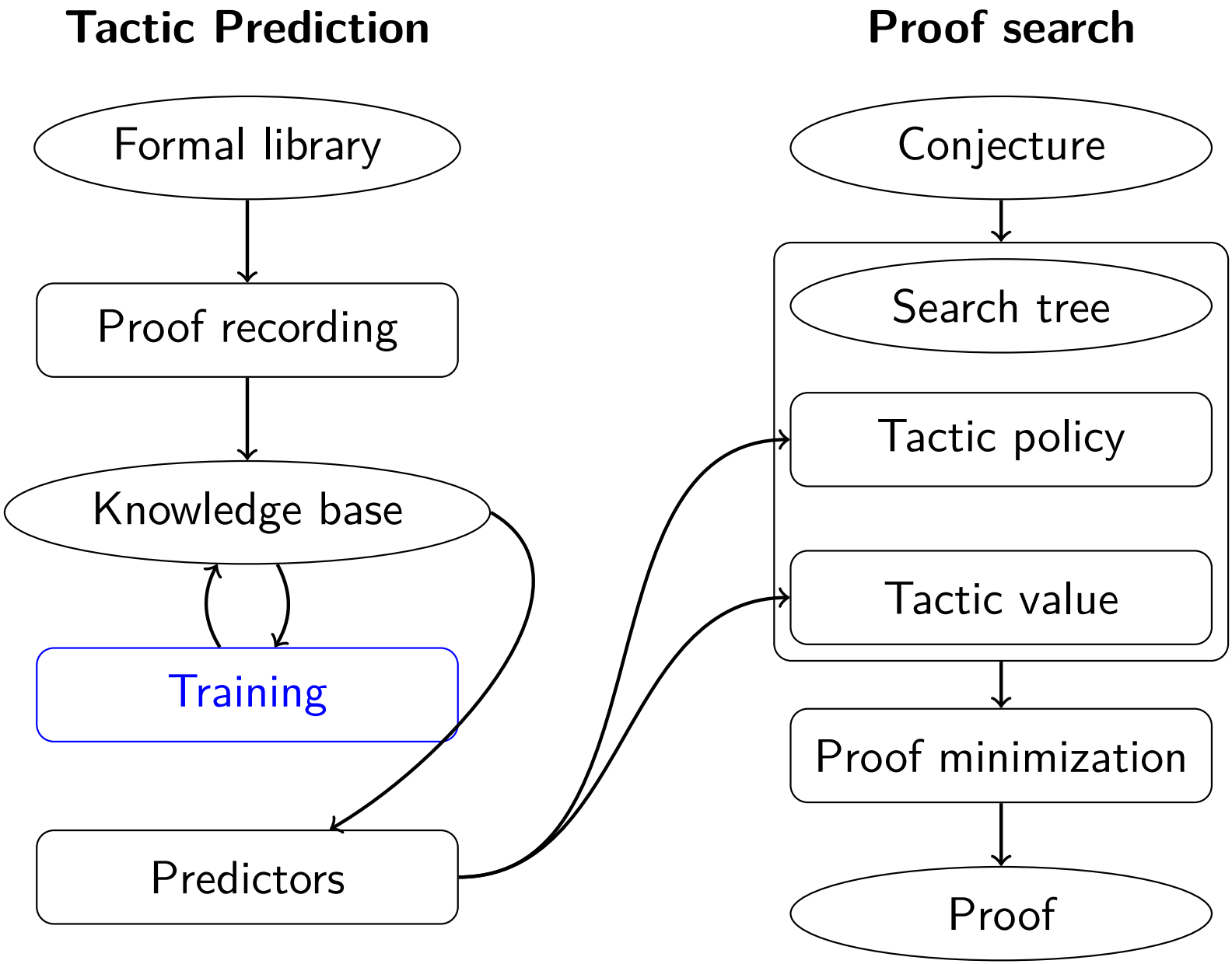
```
Rank Tactic       Policy
1    REWRITE_TAC  0.5
2    METIS_TAC    0.25
...
4    INDUCT_TAC   0.0625
...
```

Value

Database of lists of goals:

- Positive examples: appears in human proofs.
- Negative examples: produced during TacticToe search but do not appear in the final proof.

Plan


Tactic Prediction
Proof search
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Search tree
Tactic policy
Tactic value
Proof minimization
Proof

Training

Improve recorded data to create better predictions during search.

Training: orthogonalization

Issue: Many tactics are doing the same job on a goal g.

Solution: Competition for g where the most popular tactic wins.

Training: orthogonalization

Recorded goal-tactic pair:

```
LENGTH (MAP f l) = LENGTH l: INDUCT_TAC
```

Competition:

	Progress	Coverage
INDUCT_TAC	Yes	136
REWRITE_TAC	No	2567
METIS_TAC	Yes	694

Added to the database:

```
LENGTH (MAP f l) = LENGTH l: METIS_TAC
```

Result: 6 % improvement.

Training: abstraction

Issue: Some theorems are never used inside tactics.

Solution: Abstract all lists of theorems in a tactic
and instantiate them depending on the target goal.

Training: abstraction

Abstraction algorithm:

```
Original      : REWRITE_TAC [T1,T2]
Abstraction   : REWRITE_TAC X
Instantiation: REWRITE_TAC [T67, T1, T43, ..]
```

Question: Dow we keep the original or the abstraction ?

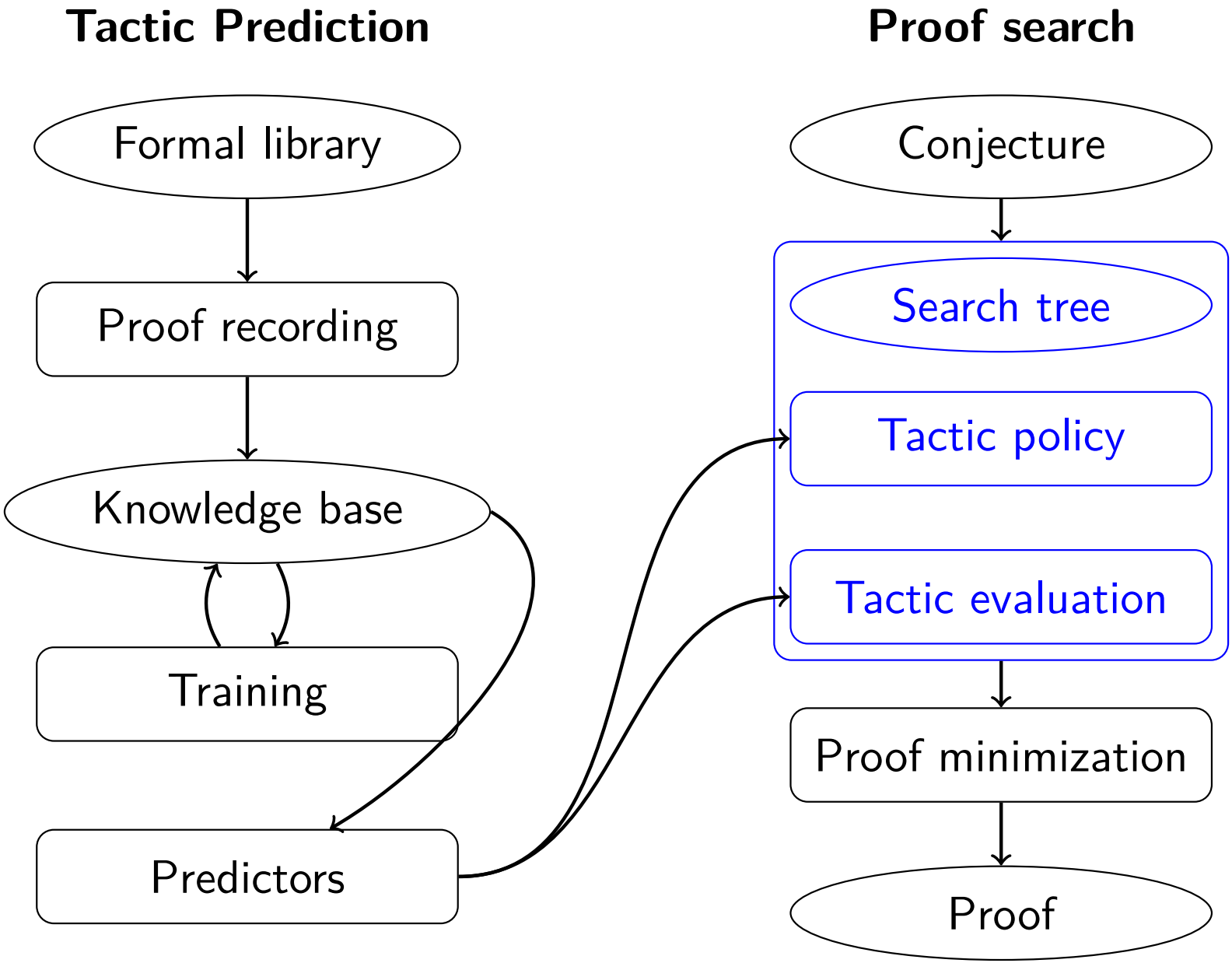
Answer: Let them compete during orthogonalization.

Result: 15% improvement

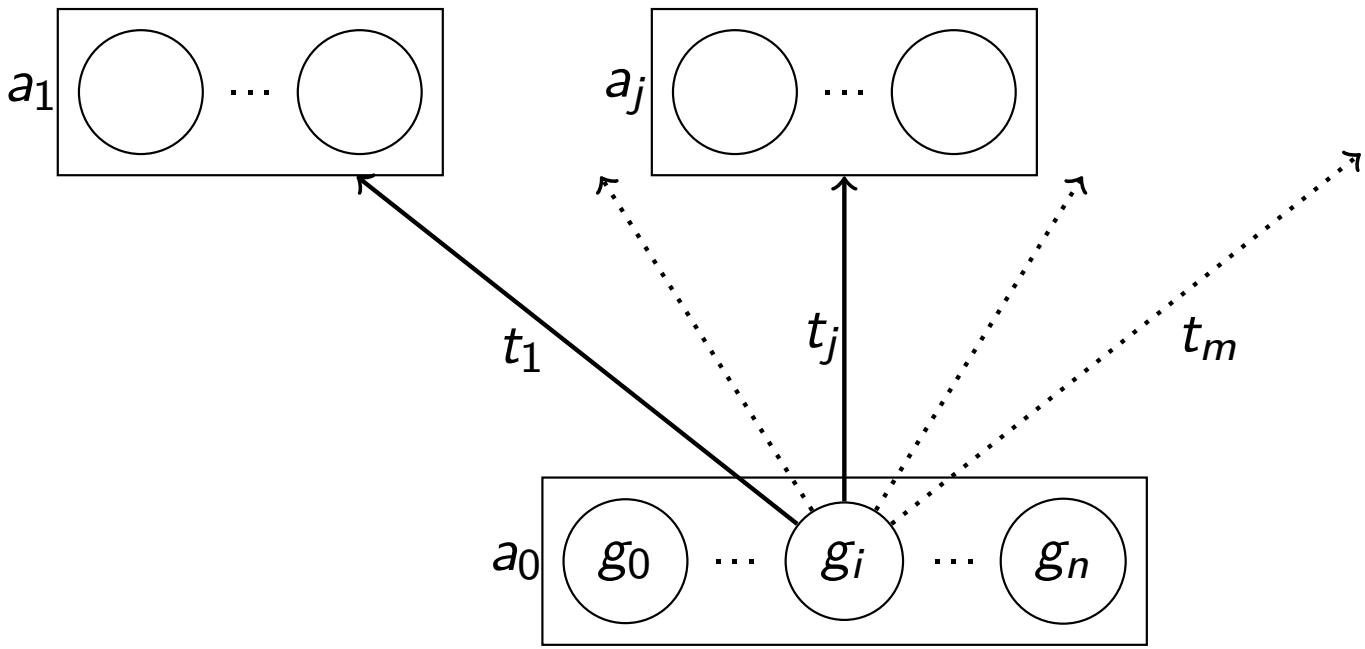
Training: preselection

Issue: Predictions are too slow during proof search.

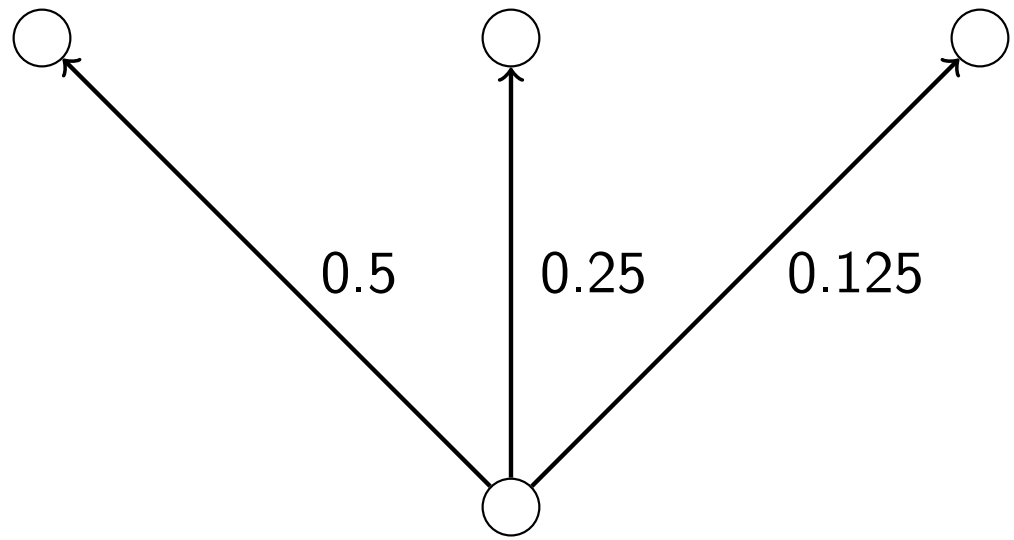
Solution: Preselect 1000 suitable tactics by importing proofs (many tactics) from related goals.


Tactic Prediction
Proof search
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Search tree
Tactic policy
Tactic evaluation
Proof minimization
Proof

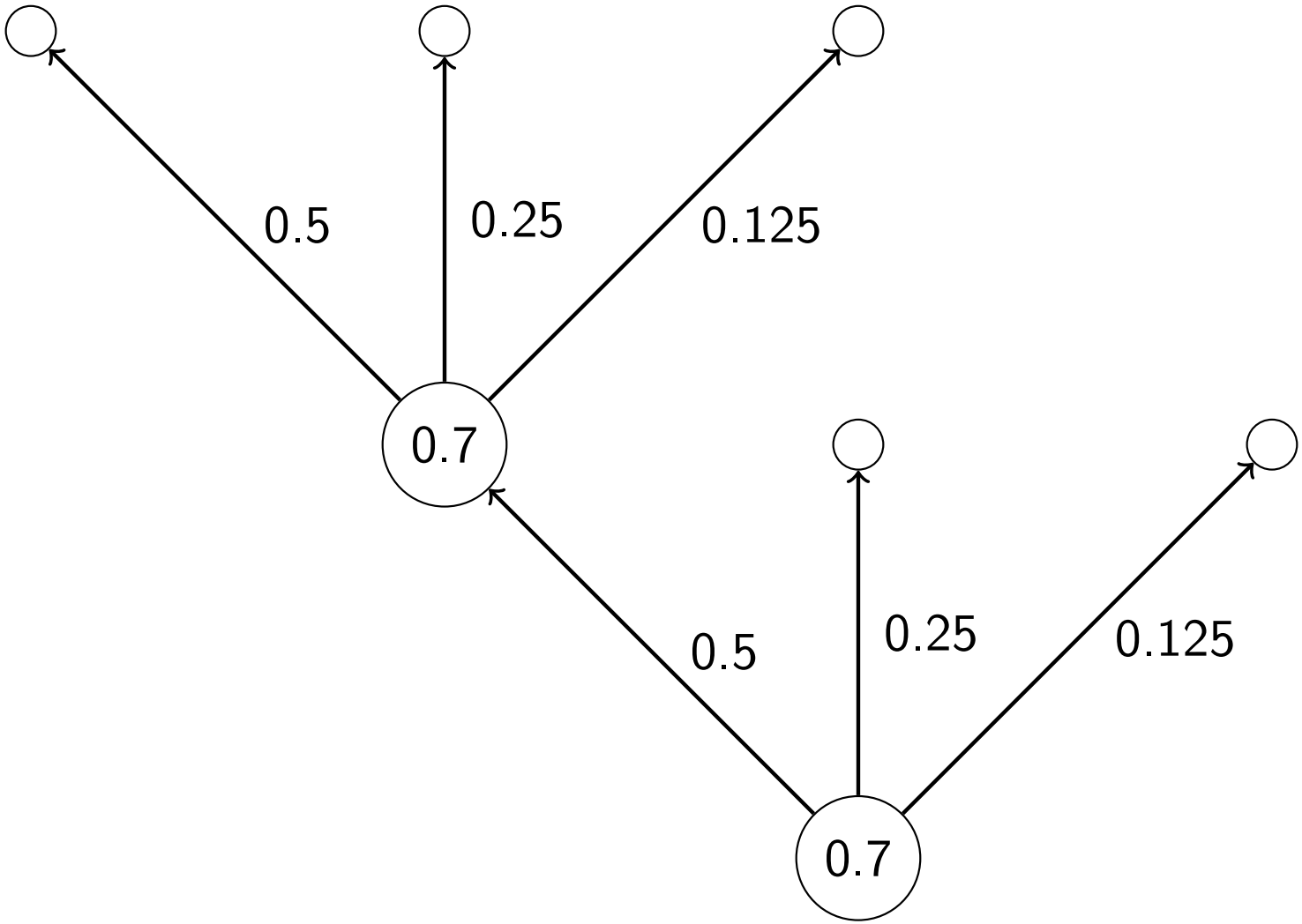
Proof search: search tree



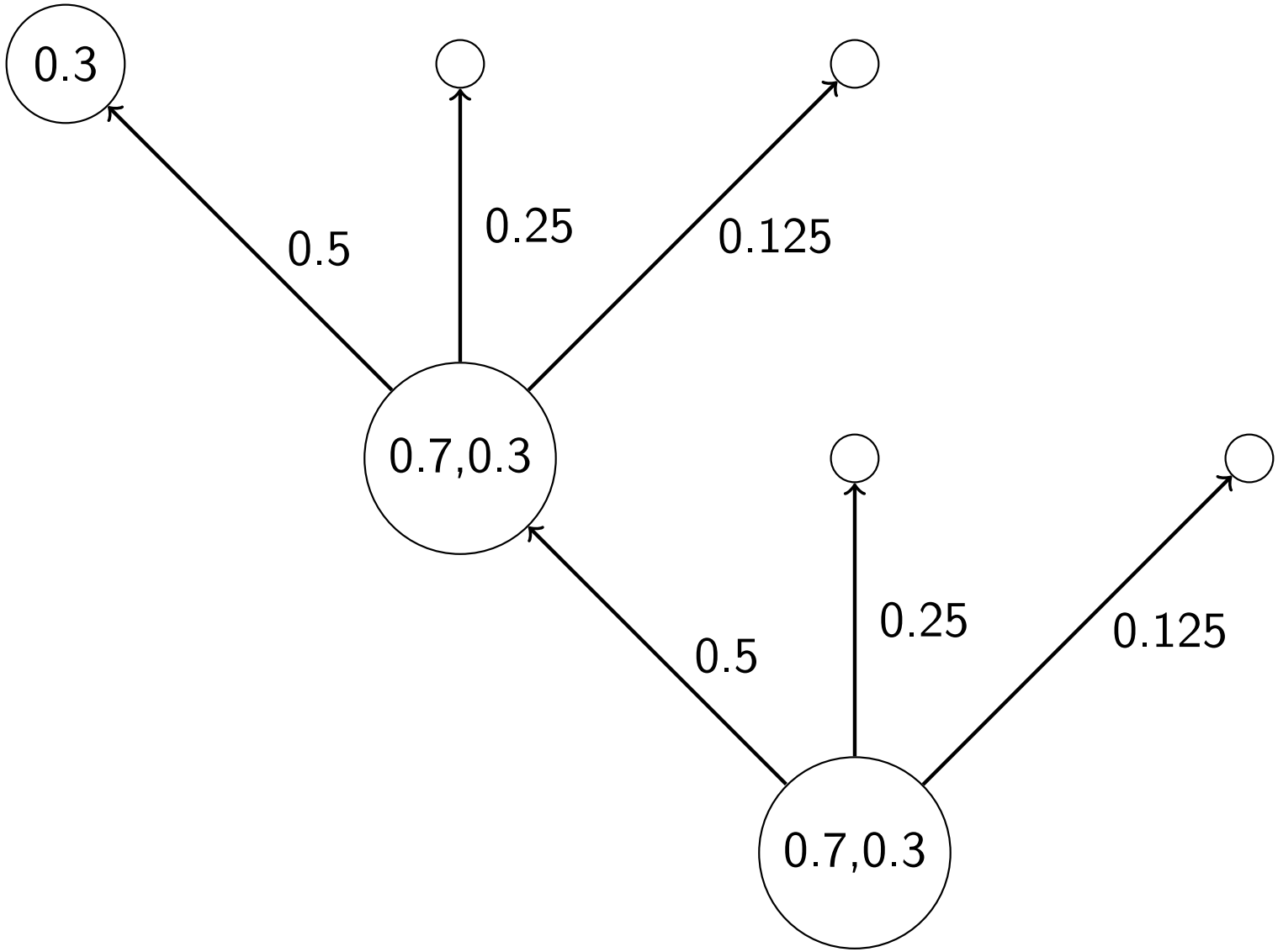
Proof search: advanced tree search



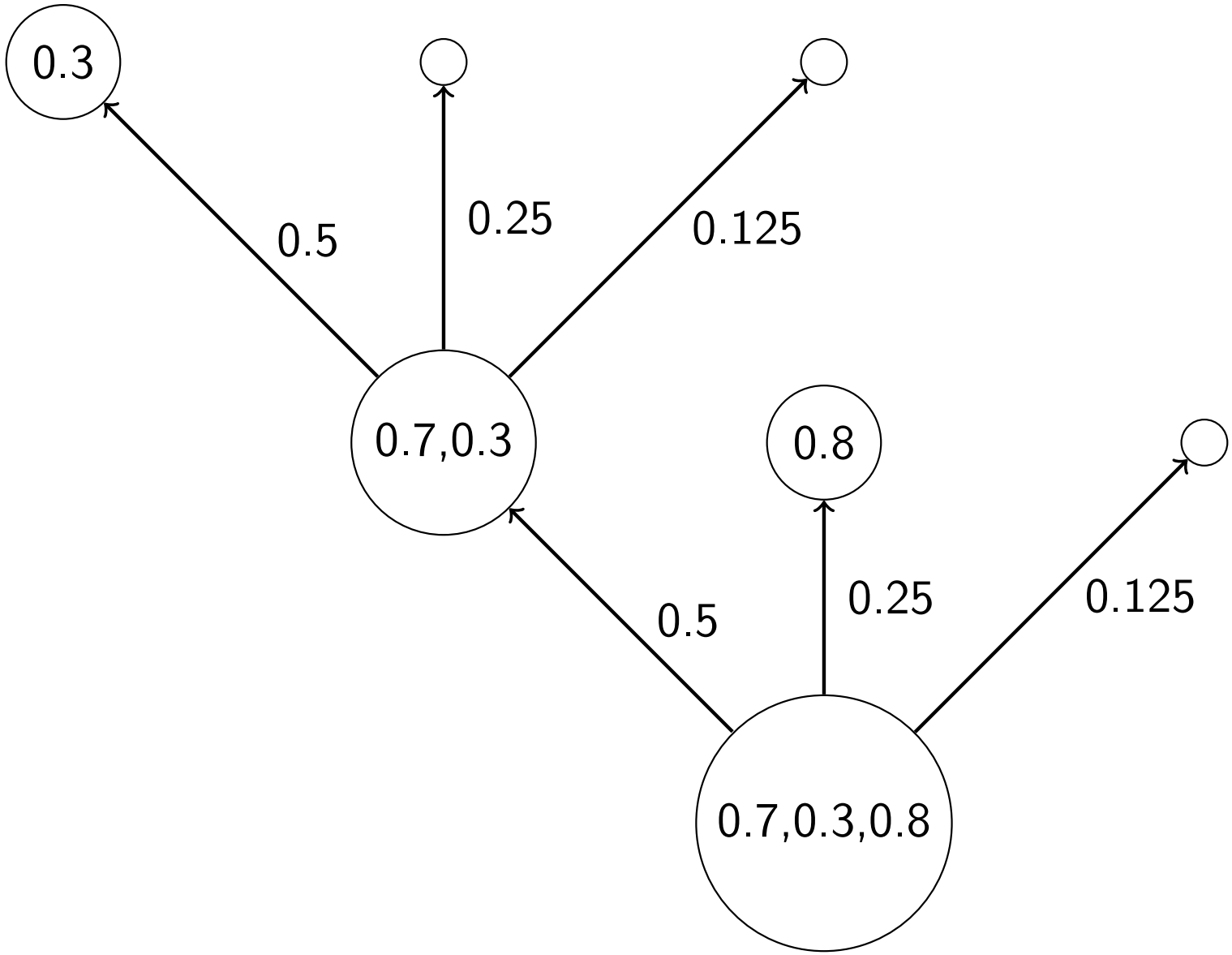
Proof search: advanced tree search



Proof search: advanced tree search



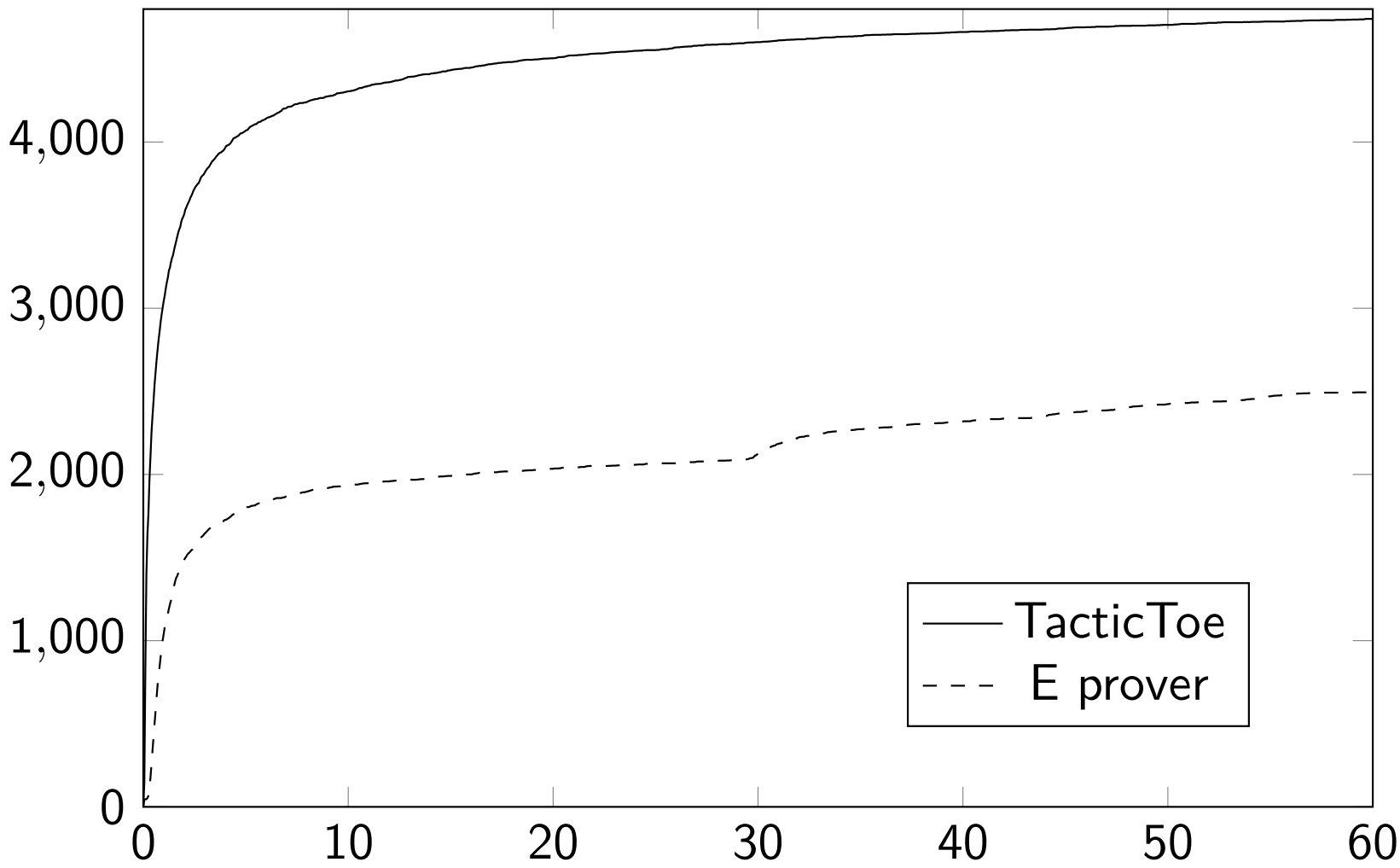
Proof search: advanced tree search



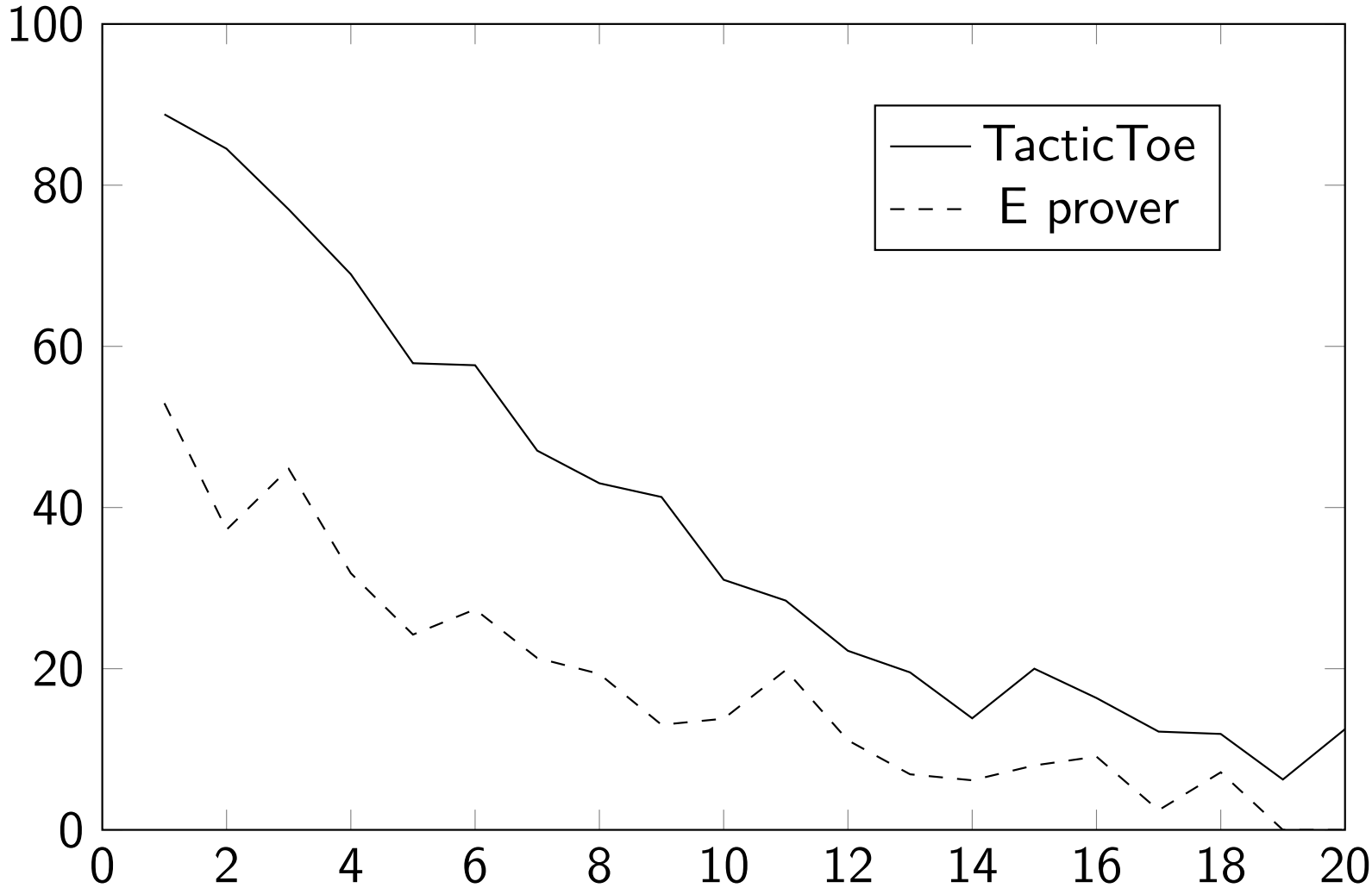
Re-proving

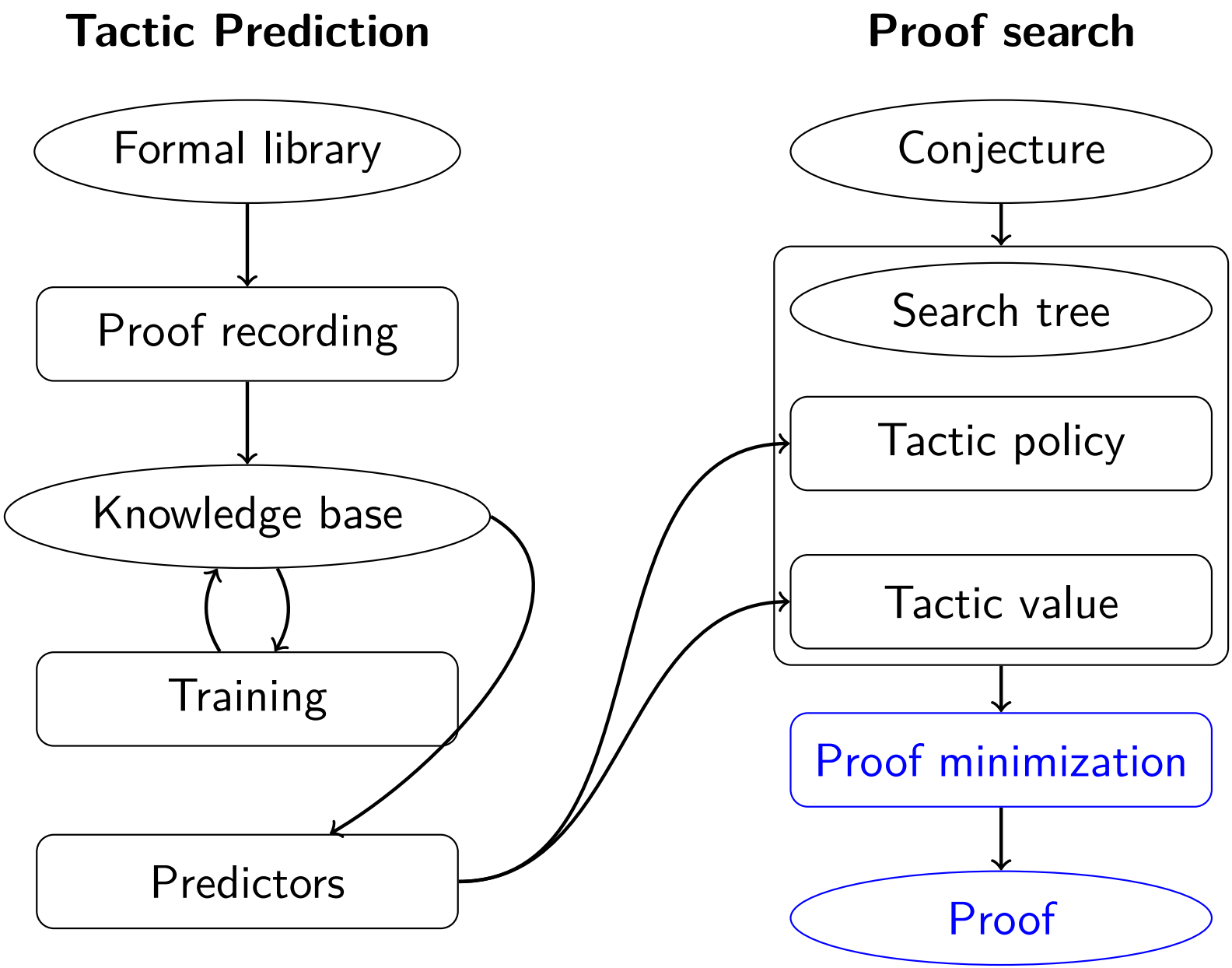
Tested library	Proof automation	Success
		50%
		66%

Re-proving: HOL4 proofs found in less than x seconds



Re-proving: percentage of solved HOL4 proof of size x




Tactic Prediction
Proof search
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Search tree
Tactic policy
Tactic value
Proof minimization
Proof

Before:

```
boolLib.REWRITE_TAC [DB.fetch "list" "EVERY_CONJ",... ]
  THEN
BasicProvers.Induct_on [HolKernel.QUOTE "l"]
  THENL
  [BasicProvers.SRW_TAC [] [],
   simpLib.ASM_SIMP_TAC (BasicProvers.srw_ss ())
   [boolLib.DISJ_IMP_THM, DB.fetch "list" "MAP",
    DB.fetch "list" "CONS_11", boolLib.FORALL_AND_THM]]
```

After:

```
Induct_on `l` THENL
  [SRW_TAC [] [],
   ASM_SIMP_TAC (srw_ss ())
   [DISJ_IMP_THM, FORALL_AND_THM]]
```

Summary: TacticToe learns from human proofs to solve new goals.

Advantages over ATPs (E prover) for ITP (HOL4) users:

- Includes domain specific automation found in the ITP.
- Generated proofs are human-level proofs.
- No translation or reconstruction needed.

Demo