Learning to Prove with Tactics

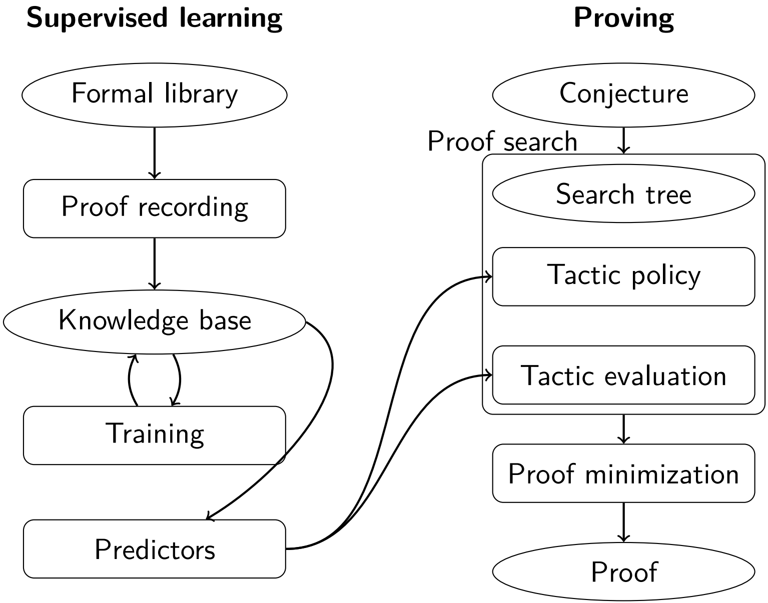
Thibault Gauthier, Cezary Kaliszyk, Josef Urban,
Ramana Kumar, Michael Norrish

March 28, 2018

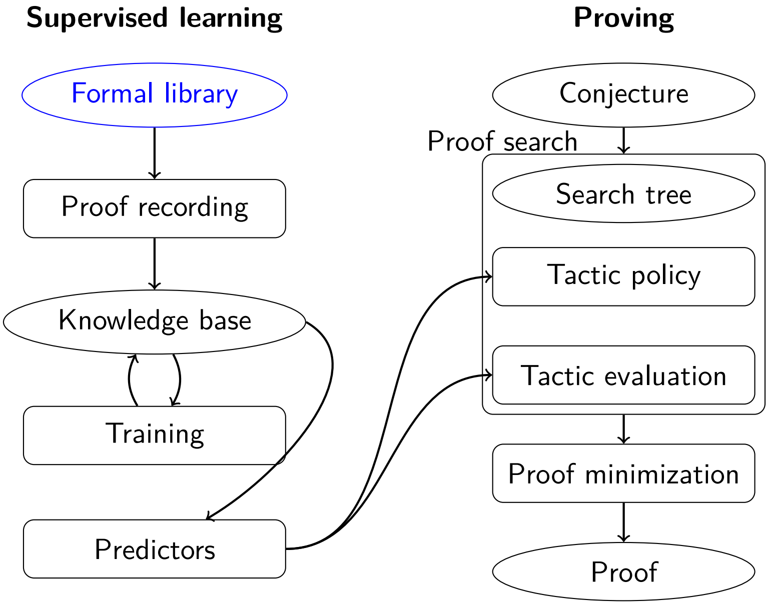
Problem

Can we formally prove mathematical formulas automatically?

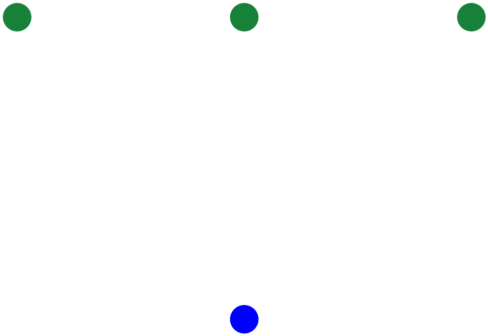
Our solution



Our solution

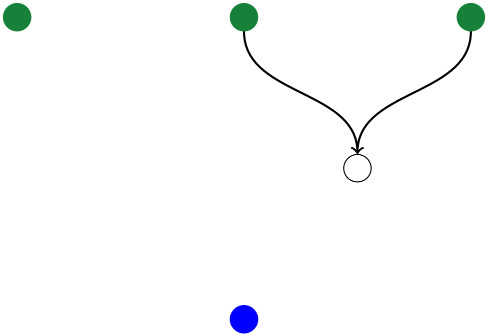


Formal library: reasoning with inference rules



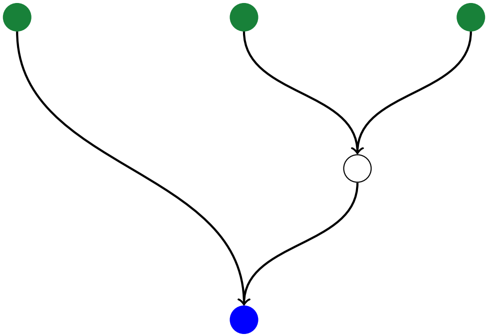
● axiom
● conjecture
→ rule
○ lemma

Formal library: reasoning with inference rules



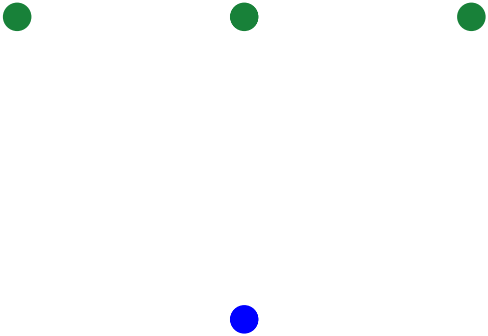
- axiom
- conjecture
- $\rightarrow$ rule
- lemma

Formal library: reasoning with inference rules



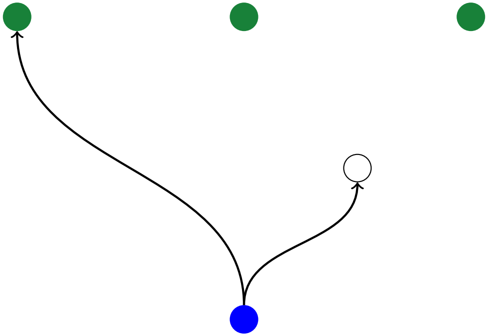
- axiom
- conjecture
- → rule
- lemma

Formal library: reasoning with tactics



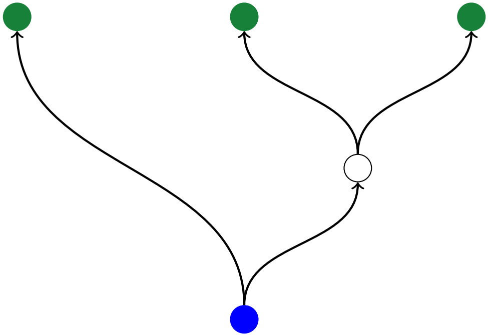
- axiom
- conjecture
- → tactic
- goal

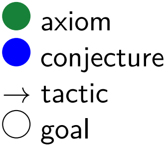
Formal library: reasoning with tactics



- axiom
- conjecture
- → tactic
- goal

Formal library: reasoning with tactics





Formal library: tactics

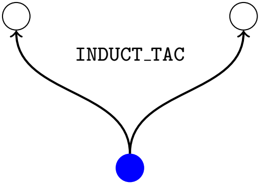
```
REWRITE_TAC

INDUCT_TAC

METIS_TAC
```

Formal library: composing tactics

`THENL` tactical composes the effect of tactics.

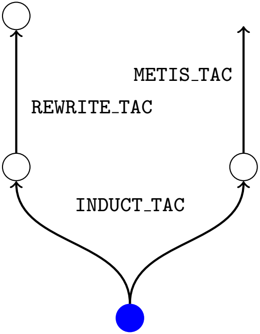
Formal library: composing tactics

THENL tactical composes the effect of tactics.



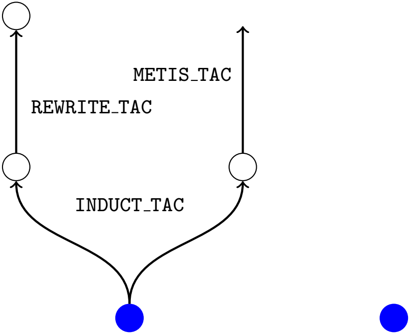
Formal library: composing tactics

THENL tactical composes the effect of tactics.



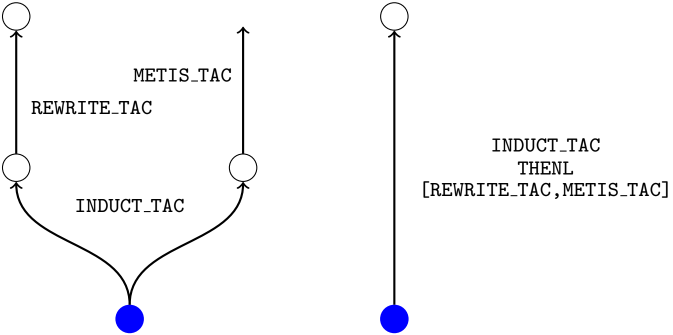
Formal library: composing tactics

THENL tactical composes the effect of tactics.



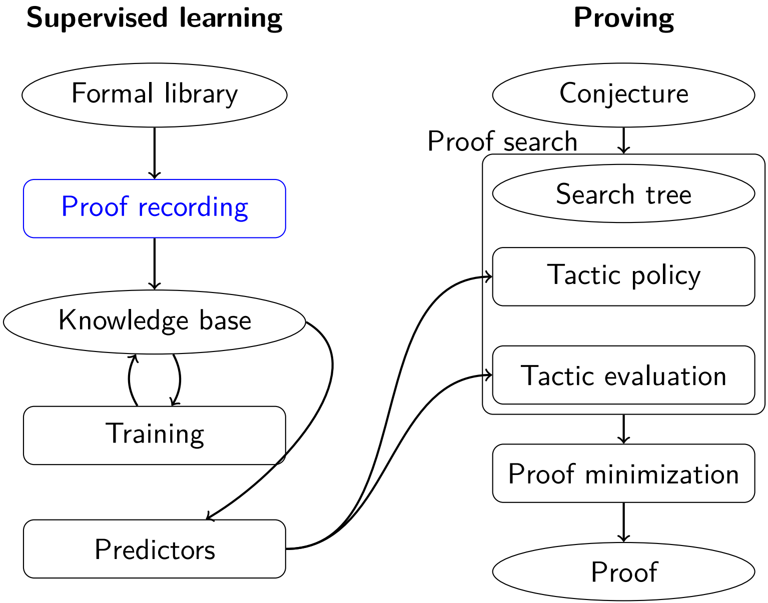
Formal library: composing tactics

THENL tactical composes the effect of tactics.



Demo

Plan



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

Recording

Original proof:

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

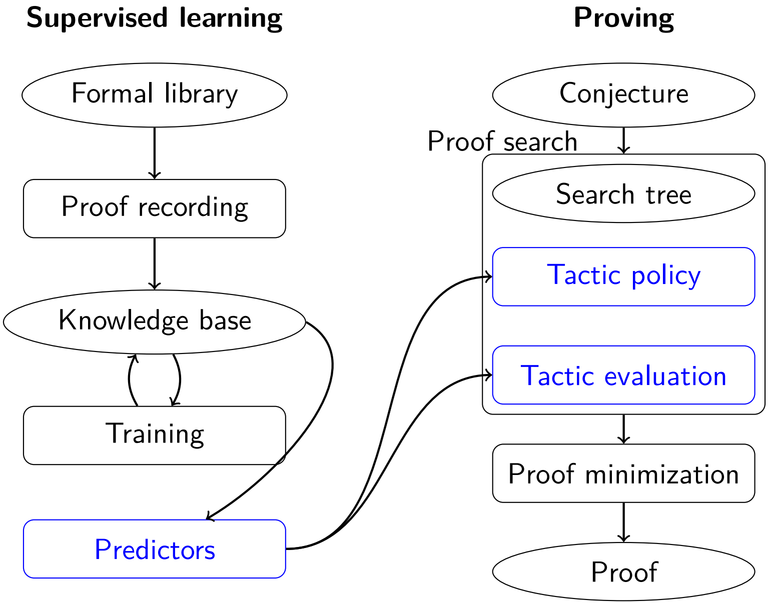
Modified proof:

```
(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
...
```

Plan


Supervised learning
Proving
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Proof search
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: choosing a tactic

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 function: provability of a list of goals

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.

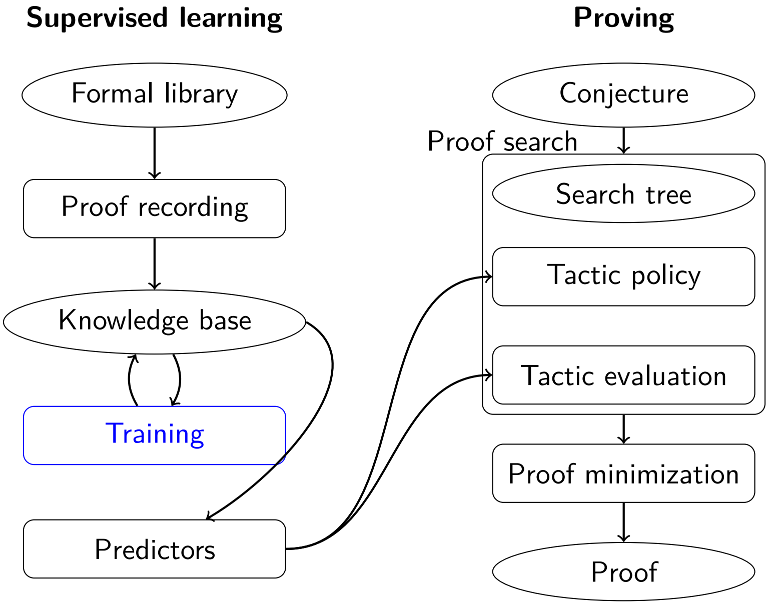
Value function:

Percentage of positives in the 10 closest lists of goals of a target list of goals.

Future work:

Estimate the number of steps needed to prove a list of goals.

Plan



Supervised learning
Proving
Formal library
Proof recording
Knowledge base
Training
Predictors
Conjecture
Proof search
Search tree
Tactic policy
Tactic evaluation
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.

Training: preselection

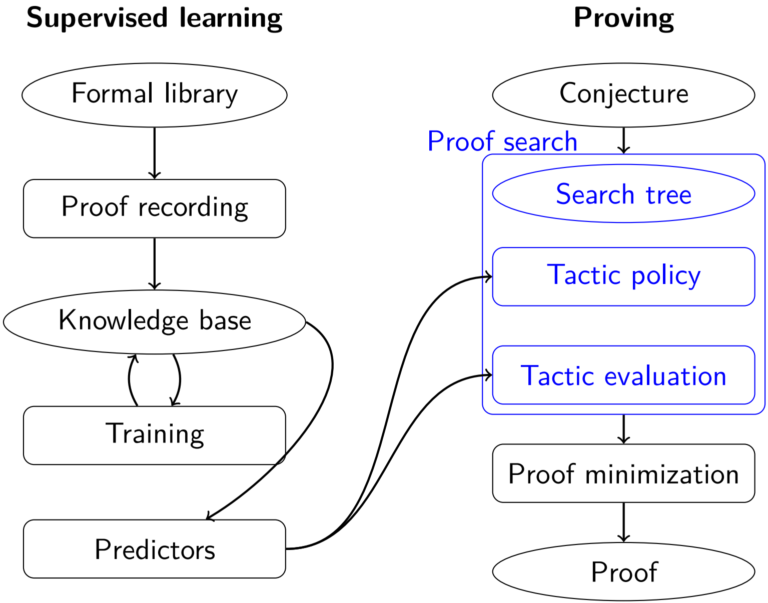
Issue: Predictions are too slow during proof search.

Solution: Preselect 1000 suitable tactics using dependencies.

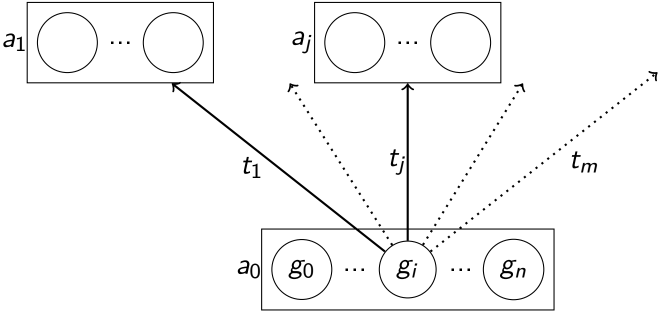
Dependency: Appear in the same proof.

Result: 15% improvement.

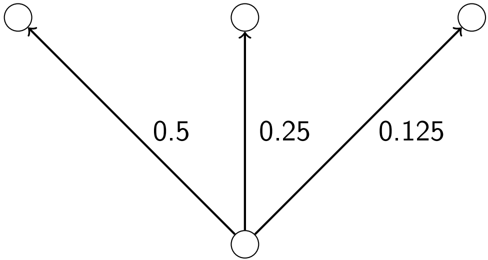
Plan


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

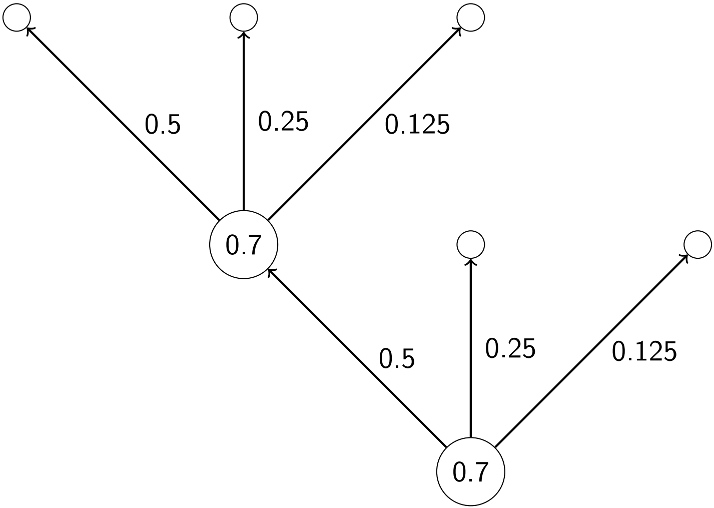
Proof search: search tree



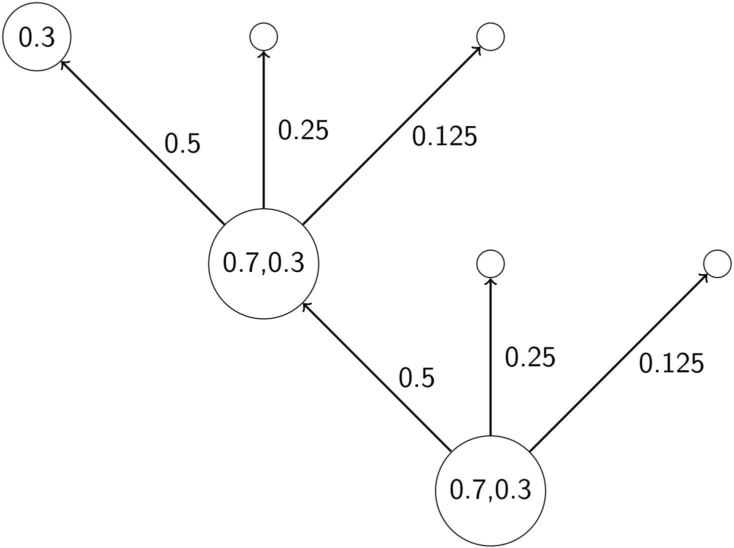
Proof search: advanced tree search



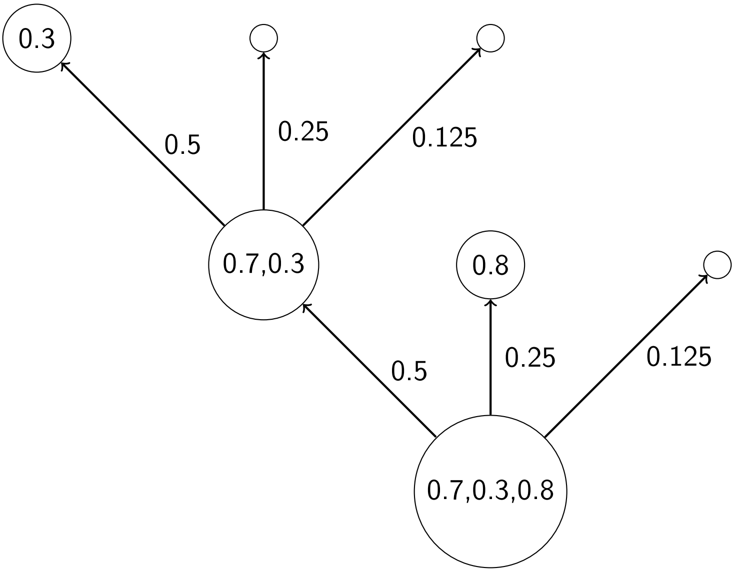
Proof search: advanced tree search



Proof search: advanced tree search



Proof search: advanced tree search



Re-proving: results

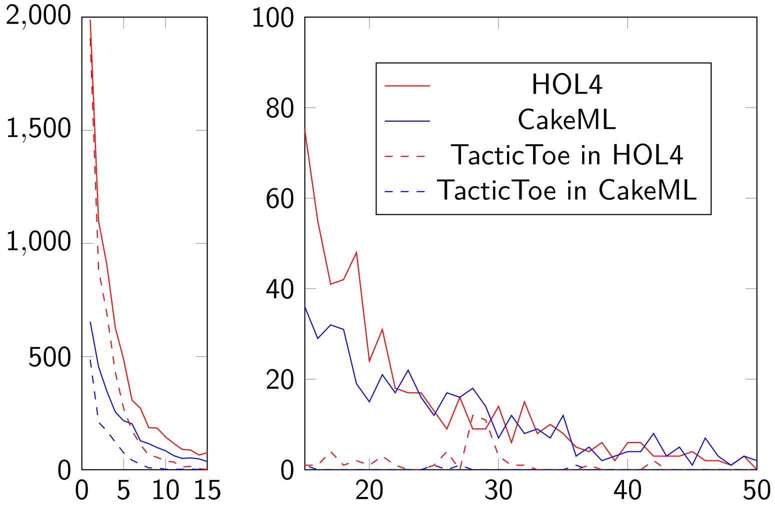
Evaluation is "fair". (not totally if you ask Freek)

Only previous proofs are available for training.

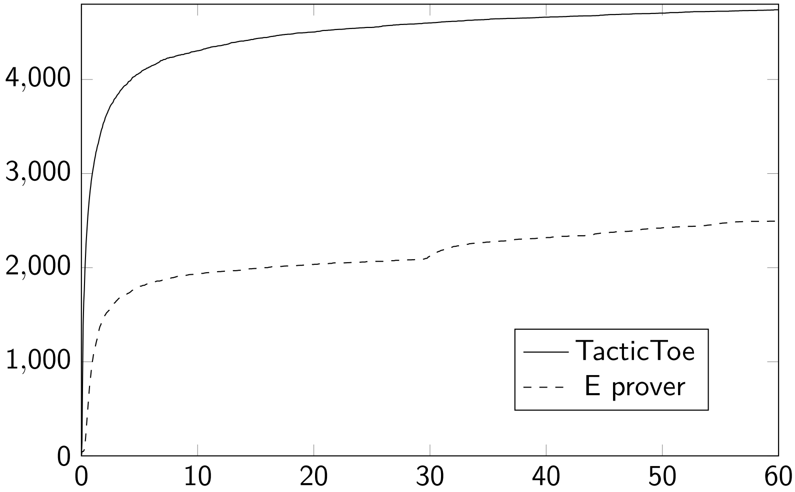
TacticToe does not call external provers.

	HOL4: 7164, 60s	CakeML: 3329, 15s
E prover	2472 (34.5%)	
TacticToe	4760 (66.4%)	1161 (34.9%)
Total	4946 (69.0%)	

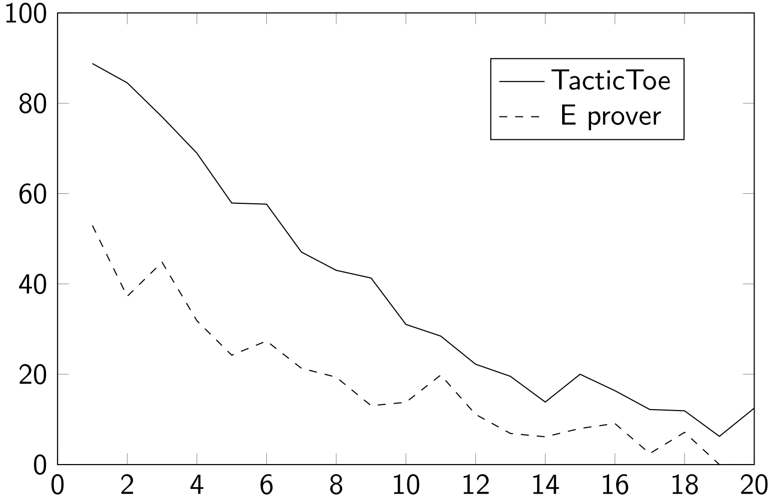
Re-proving: proofs of size x



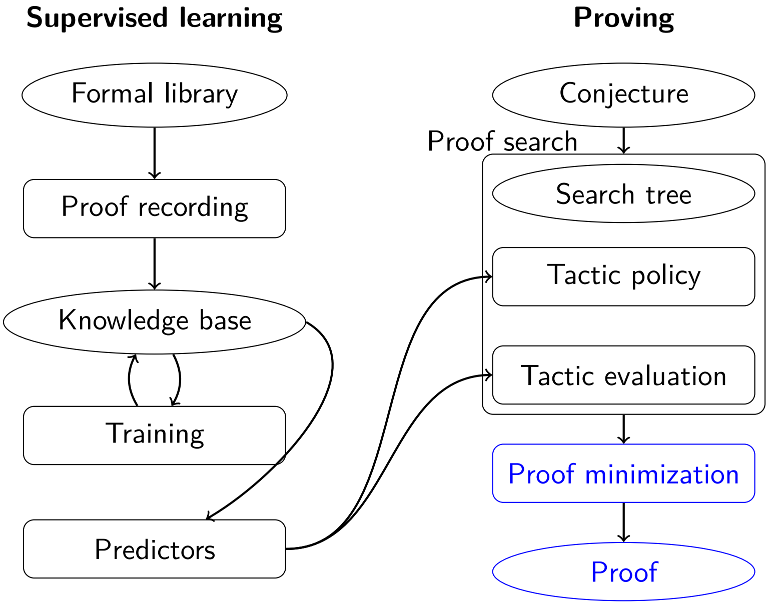
Re-proving: HOL4 proofs found in less than x seconds



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



Plan


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

Minimization and embellishment

Raw proof:

```
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]]
```

Processed proof:

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

Conclusion

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.

Future work

Enlarge the action space:
parameter synthesis, sequence of tactics, forward proofs.

Train tactics by evaluating input/output pairs.

Conjecture intermediate lemmas.