Boosting Automated Reasoning by Mining Existing Proofs

Thomas Gransden

Department of Computer Science
University of Leicester
tg75@student.le.ac.uk

Interactive Theorem Proving is Difficult

Interactive Theorem Proving is Difficult

- User Driven

Interactive Theorem Proving is Difficult

- User Driven
- Expert Required

Interactive Theorem Proving is Difficult

- User Driven
- Expert Required
- Large amounts of knowledge

Interactive Theorem Proving is Difficult

- User Driven
- Expert Required
- Large amounts of knowledge
- Time Consuming

Interactive Theorem Proving is Difficult

- User Driven
- Expert Required
- Large amounts of knowledge
- Time Consuming

A Large Scale Verification:

L4.Verified

25-30 years combined effort

200,000 lines of Isabelle code

Interactive Theorem Proving is Difficult

- User Driven
- Expert Required
- Large amounts of knowledge
- Time Consuming

A Large Scale Verification:

L4.Verified

25-30 years combined effort

200,000 lines of Isabelle code

Problem:

Finding a suitable sequence of proof steps is hard!

Proof Automation

Proof Automation

- Much sought after property
  - Reduces Human Intervention
  - Benefits in many fields

Proof Automation

- Much sought after property
  - Reduces Human Intervention
  - Benefits in many fields
- Very active research area

Proof Automation

- Much sought after property
  - Reduces Human Intervention
  - Benefits in many fields
- Very active research area
  - International Tournaments!

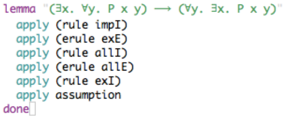
Proof Automation

- Much sought after property
  - Reduces Human Intervention
  - Benefits in many fields
- Very active research area
  - International Tournaments!
- Restricted by underlying logic

Proof Automation

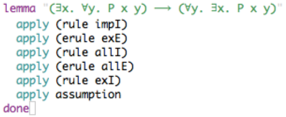
- Much sought after property
  - Reduces Human Intervention
  - Benefits in many fields
- Very active research area
  - International Tournaments!
- Restricted by underlying logic
  - Expressivity vs Automation Tradeoff

Proof Libraries



```
lemma "(∃x. ∀y. P x y) ⟶ (∀y. ∃x. P x y)"
  apply (rule impI)
  apply (erule exE)
  apply (rule allI)
  apply (erule allE)
  apply (rule exI)
  apply assumption
done
```

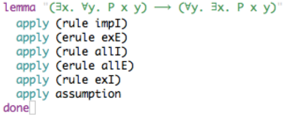
Proof Libraries



```
lemma "(∃x. ∀y. P x y) ⟶ (∀y. ∃x. P x y)"
  apply (rule impI)
  apply (erule exE)
  apply (rule allI)
  apply (erule allE)
  apply (rule exI)
  apply assumption
done
```

- Examples of successful proofs

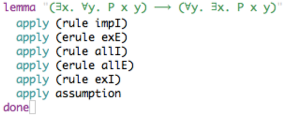
Proof Libraries



```
lemma "(∃x. ∀y. P x y) ⟶ (∀y. ∃x. P x y)"
  apply (rule impI)
  apply (erule exE)
  apply (rule allI)
  apply (erule allE)
  apply (rule exI)
  apply assumption
done
```

- Examples of successful proofs
- Provided by an expert

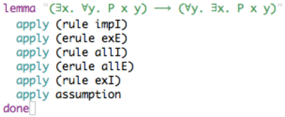
Proof Libraries



```
lemma "(∃x. ∀y. P x y) ⟶ (∀y. ∃x. P x y)"
  apply (rule impI)
  apply (erule exE)
  apply (rule allI)
  apply (erule allE)
  apply (rule exI)
  apply assumption
done
```

- Examples of successful proofs
- Provided by an expert
- Variety of complexities/domains

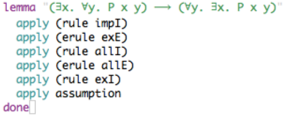
Proof Libraries



```
lemma "(∃x. ∀y. P x y) ⟶ (∀y. ∃x. P x y)"
  apply (rule impI)
  apply (erule exE)
  apply (rule allI)
  apply (erule allE)
  apply (rule exI)
  apply assumption
done
```

- Examples of successful proofs
- Provided by an expert
- Variety of complexities/domains
- Specified as **proof steps**

Proof Libraries



```
lemma "(∃x. ∀y. P x y) ⟶ (∀y. ∃x. P x y)"
  apply (rule impI)
  apply (erule exE)
  apply (rule allI)
  apply (erule allE)
  apply (rule exI)
  apply assumption
done
```

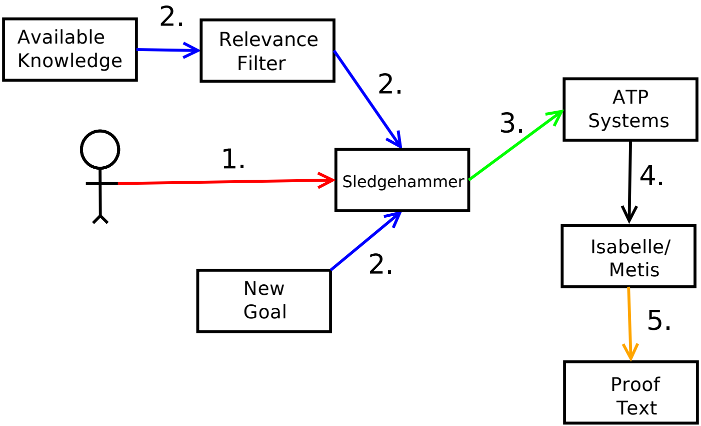
- Examples of successful proofs
- Provided by an expert
- Variety of complexities/domains
- Specified as **proof steps**

Idea:

Can we use this information to automate new proofs?

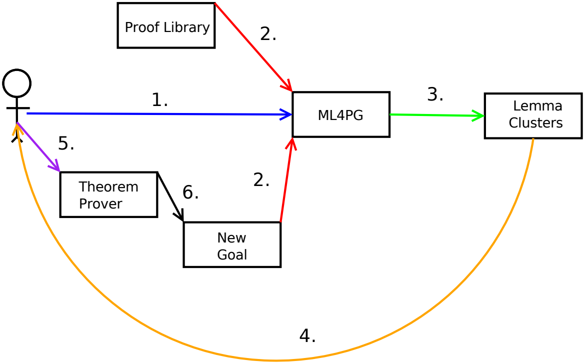
Increasing Automation in ITP's - Link ATP's and ITP's

Increasing Automation in ITP's - Link ATP's and ITP's



Increasing Automation in ITP's - Proof Hints

Increasing Automation in ITP's - Proof Hints



Tactic Mining Terminology

Useful Sequences - Sequences of proof steps that could prove useful in proving some new goal

Tactic Mining Terminology

Useful Sequences - Sequences of proof steps that could prove useful in proving some new goal

Tactic - A function that is applied to a proof state

Tactic Mining Terminology

Useful Sequences - Sequences of proof steps that could prove useful in proving some new goal

Tactic - A function that is applied to a proof state

Tactic Mining - Automatically forming tactics from large libraries of existing proofs

Tactic Mining Terminology

Useful Sequences - Sequences of proof steps that could prove useful in proving some new goal

Tactic - A function that is applied to a proof state

Tactic Mining - Automatically forming tactics from large libraries of existing proofs

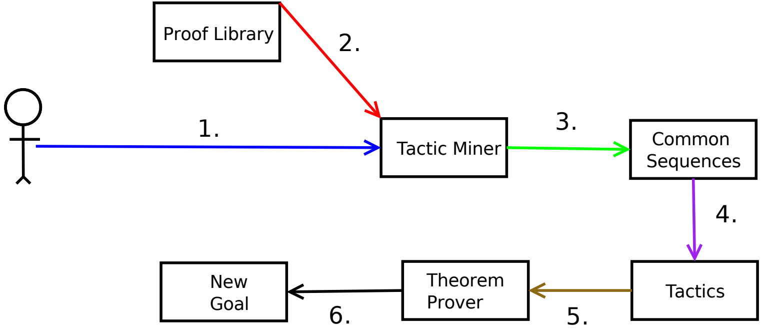
Sequence 1:	**Sequence 2:**	**Tactic:**
rule impI assumption	rule conjI assumption	(rule impI OR rule conjI) THEN assumption

Previous Tactic Mining Work

Carried out by Hazel Duncan at Edinburgh.

Previous Tactic Mining Work

Carried out by Hazel Duncan at Edinburgh.



Critique of Duncan’s approach

There are some limitations of Duncan’s work:

Critique of Duncan’s approach

There are some limitations of Duncan’s work:

- Moderatley effective on test set

Critique of Duncan's approach

There are some limitations of Duncan's work:

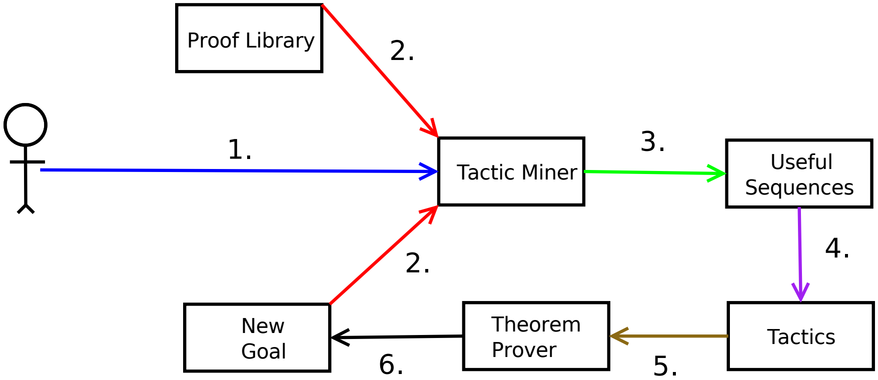
- Moderatley effective on test set
- No subgoal information

Critique of Duncan's approach

There are some limitations of Duncan's work:

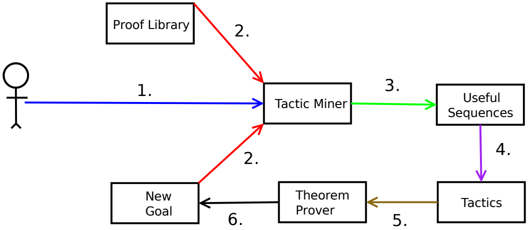
- Moderatley effective on test set
- No subgoal information
- Inefficent tactic application

My Tactic Mining Approach



1. How can we deal with complex Higher Order Languages?

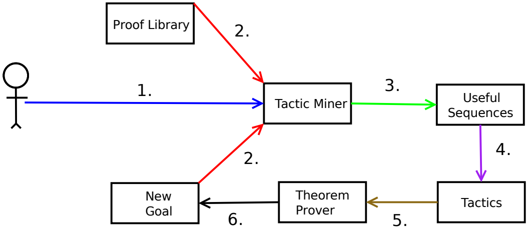
Variable instantiations and proof directives



1. How can we deal with complex Higher Order Languages?

Variable instantiations and proof directives

One sequence of steps solves many proofs and vice versa

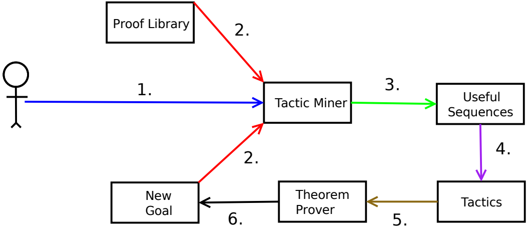


1. How can we deal with complex Higher Order Languages?

Variable instantiations and proof directives

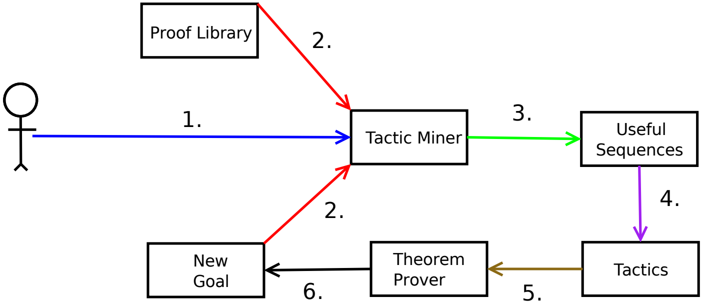
One sequence of steps solves many proofs and vice versa

Different proof styles



2. Which Data Mining Techniques can help?

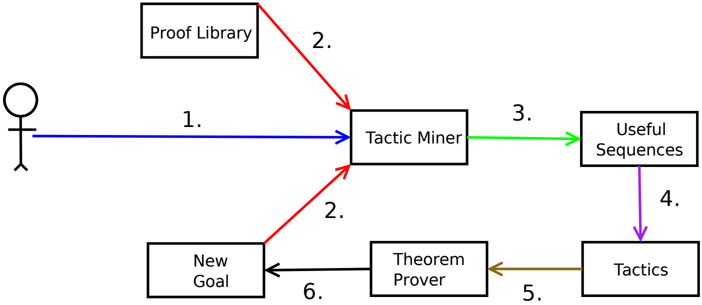
An open research question



2. Which Data Mining Techniques can help?

An open research question

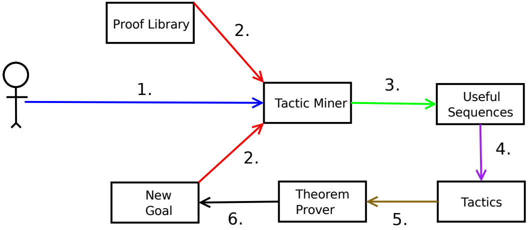
Two tasks: Finding the patterns and generalising into tactics



3. How will the theorem prover and tactic miner communicate?

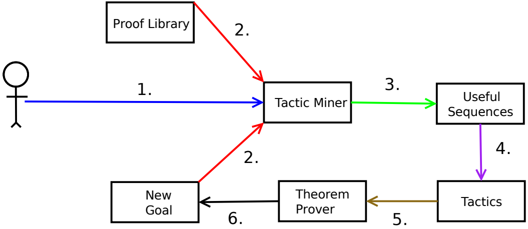
We require two methods of communication to be defined:

- Theorem Prover to Tactic Miner
- Tactic Miner to Theorem Prover



4. How can we make use of negative information?

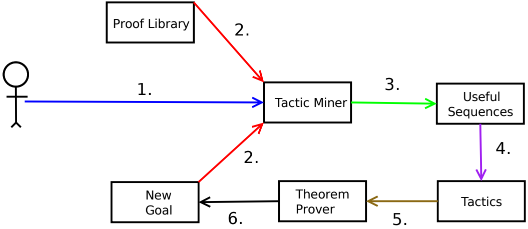
Leverage negative information from:



4. How can we make use of negative information?

Leverage negative information from:

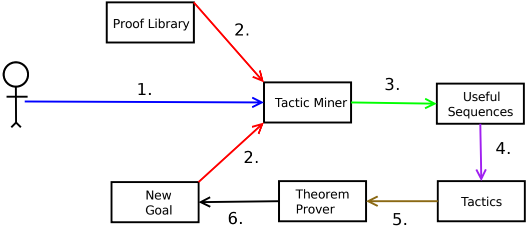
- User inputs



4. How can we make use of negative information?

Leverage negative information from:

- User inputs
- Failed traces from existing automated tools

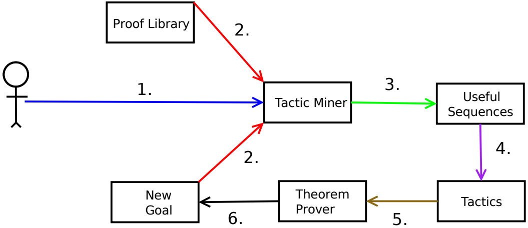


4. How can we make use of negative information?

Leverage negative information from:

- User inputs
- Failed traces from existing automated tools

Would enable a supervised learning approach.



Current Work

I am currently at the following stage with my work:

Current Work

I am currently at the following stage with my work:

- Data Extraction from Isabelle

Current Work

I am currently at the following stage with my work:

- Data Extraction from Isabelle
- Considering learning techniques

Any Questions?

Please feel free to ask me any questions, either now or at any point during the workshop!