Learning and Exploration in Automated Theorem Proving

Moa Johansson
Chalmers University of Technology

Background

- Can machine learning help exploring new theories in formalised mathematics?
- Statistical methods: Efficiently find patterns in large amounts of data.
  - E.g. correlations in proof libraries.
- Symbolic methods: Conceptual hints, make sense of data.
  - E.g. lemma discovery by theory exploration.

Machine Learning in Automated Reasoning

- Premise selection and proof reconstruction.
  - First-order provers.
- Which algorithms used?
  - Simple ones: e.g. naïve Bayes, k-nearest neighbours.
- What features do we learn from?
  - Which symbols occur.
  - How are symbols defined.
  - Common generalisations of terms.
  - Many more…

Theory Exploration

- Input: Functions, constants.
- Term generation:
  - Build type-correct terms.
  - Testing to form conjectures.
- Proof (automated)
  - Trivial proofs: discard.
  - Hard proofs: interesting.
- Output: New lemmas in theory.

Example

Input: `sort`

Generated terms (some):

```
sort(xs)
sort(sort(xs))
xs
```

Test random values, e.g: `xs-->[3,1]`

```
sort[3,1]          [1,3]
sort(sort[3,1])    [1,3]
xs                 [3,1]
```

Extract Conjecture:

```
sort(sort xs)= sort(xs)
```

Theory Exploration Teaser Demo:

Hipster – A theory exploration system for Isabelle/HOL

(more in tomorrows talk)

Learning for Theory Exploration

- Input: Functions, constants.
- Term generation:
  - Build type-correct terms.
  - Testing to form conjectures.
- Proof (automated)
  - Trivial proofs: discard.
  - Hard proofs: interesting.
- Output: New lemmas in theory.

← Which functions to explore together?

← Scalability. ~20-25 symbols OK.

- Don't generate all terms?
- Learn common *term schemas*?
- Common generalisation patterns?

← Want powerful automation

- Learn configurations/ heuristics for proof tactics?

← Learning from proofs found

- How to represent?

Selecting inputs

- Theory exploration is fast.
  - (up to 20-25 symbols).
- Big theory: Split functions into several attempts.
  - Let the user do it?
  - By analogy with known theories?
  - "Common" functions
    - List append, plus…
- Theory exploration in a proof attempt.
  - Find relevant lemmas.
  - Techniques similar to premise selection.

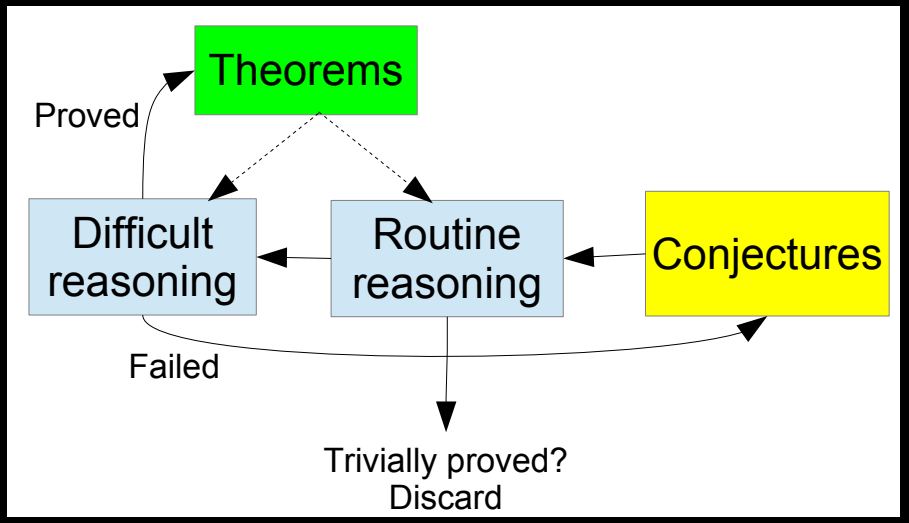
Scalability of Term Generation

- Chris' talk up next!
- Ongoing work: Explore schemas
  - Schematic terms and equations.
  - Fewer instantiations considered.
- Learning common term schemas across theories?
- Speculating common generalisations? (Kalinszyk)
- Lemmas by analogy (LPAR13).

Proofs: Selecting induction schemes

- Routine: simplification
- Difficult: Often induction
  - Structural on datatype?
  - Recursive structure of function?
  - Number of variables?
  - Include all background facts?
  - Order of proof attempts?
- Learn from proofs about
  - Same functions
  - Similar functions
  - Premise selection

Theory Exploration Proof Loop:



Learning from what we found

- Proof output limited.
  - HipSpec: none.
  - Hipster: One line Isabelle/Isar script.
- Ongoing work – minimal proofs:
  - Extract induction scheme.
  - Lemmas used.
- TIP: Benchmarks for inductive provers
  - Augment format with basic proof info.
  - Other proof formats, e.g. OpenTheory.

Summary

- Machine learning and statistical methods:
  - Find patterns in e.g. large proof libraries.
- Theory exploration and symbolic methods:
  - Find new lemmas, concepts.
  - ML techniques can help reducing search.
    - More efficient (faster) proofs.
    - Scalability in term generation.
    - Targeted search for lemmas.