SRL challenge: Extracting proof strategies from exemplar proofs

Gudmund Grov, Katya Komendantskaya, Alan Bundy

Statistical Relational Learning Workshop

30 June 2012

Interactive Theorem Provers

- ...based on higher-order languages/type theory;
- ...provide rich language to express and formalise theorems (in Mathematics) or software/hardware properties (in Verification).
- As opposed to "First-order Provers", will never yeild complete atomation;
- but Machine-Learning may help to data mine, extract, and generalise proof strategies.

Hol/Isabelle, Coq, Agda, Lego, ACL

Main applications:

Software Verification:

- Intel processor bug and solution by ACL2
- Mobile phone OS - Isabelle/HOL

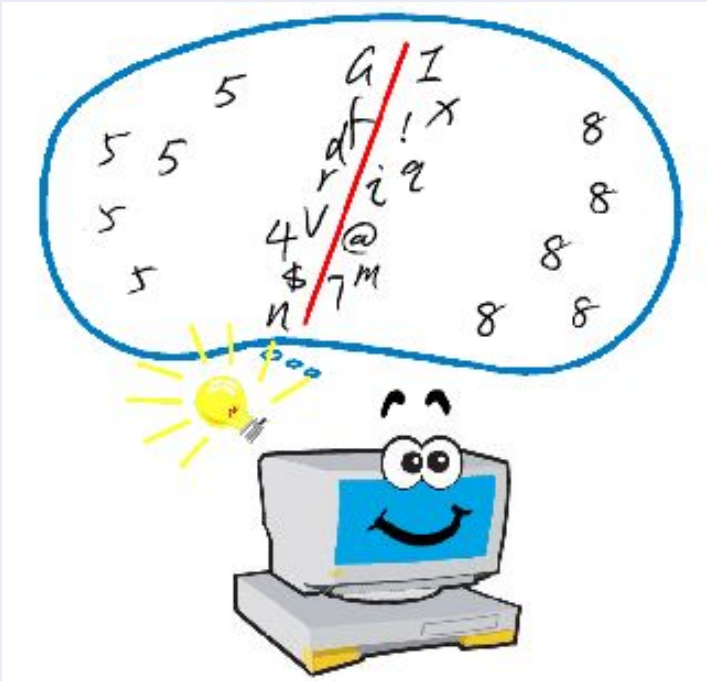
Main applications:

Software Verification:

- Intel processor bug and solution by ACL2
- Mobile phone OS - Isabelle/HOL

Formalisation of Maths

- Four-colour theorem formalised/verified in Coq;
- Prime Number theorem in Isabelle/HOL



Programming languages used for that:

Two styles of verification are common:

- Automated Proving (restricted languages, less of expressivity)
- Interactive Proving (language as rich as you want)

Programming languages used for that:

Two styles of verification are common:

- Automated Proving (restricted languages, less of expressivity)
- Interactive Proving (language as rich as you want) But you have to "interact"!

Coq Demo

Demo Conclusions:

- Not all proofs can be done automatically. That is, automatic tactics fail.
- In bigger industrial proofs there might be thousands of lemmas and theorems needing proofs, with about 5-20% needing programmer's intervention.
- Can statistical methods help us to analyse why these fail? and perhaps even suggest combination of tactics to try instead?

Our "BIG" goal:

- Use machine-learning tools to "learn" the proof patterns;
- Embed them in interactive provers
- Make the process easier for programmers, and hence the overall technlology – faster, cheaper, and more widely accessible.

Projects AI4FM and ML-CAP

Researchers and research groups from Universities of Edinburgh, Dundee, Newcastle, Heriot-Watt.

In the Poster this afternoon:

We will consider three technical approaches to machine-learn proofs:

In the Poster this afternoon:

We will consider three technical approaches to machine-learn proofs:

1. level of tactics (direct interaction)

In the Poster this afternoon:

We will consider three technical approaches to machine-learn proofs:

1. level of tactics (direct interaction)
2. level of term-structures in subgoals (in-direct interaction)

In the Poster this afternoon:

We will consider three technical approaches to machine-learn proofs:

1. level of tactics (direct interaction)
2. level of term-structures in subgoals (in-direct interaction)
3. level of proof-trees (hidden proof strategies)

In the Poster this afternoon:

We will consider three technical approaches to machine-learn proofs:

1. level of tactics (direct interaction)
2. level of term-structures in subgoals (in-direct interaction)
3. level of proof-trees (hidden proof strategies)