Learning proofs from refutations

AI4FM – KO meeting
University of Newcastle
21/05/2010 Leo Freitas

Background

- modeling experience in Z, CSP, B, and Circus
- large (300-500 pages model + proofs)
- complex (1000 POs, 5000 proof steps)
- simple, yet tricky proofs (*e.g.*, R^+, finiteness, *etc.*)
- basic understanding of HOL and some of its provers
  - (*i.e.*, HOL-light, PVS, Isabelle, ProofPower)
- basic understanding of other systems
  - (*i.e.*, ACL2, Coq, Simplify, Clean)
- 11 years experience with **Z/Eves**
  - FOL + quantifiers prover tailored for Z schema calculus
  - Z mathematical toolkit, set based representations of terms

Prover families

- LCF - functional HOL
  - sound by construction
  - powerful tactics and heuristics
  - programmable proofs (*e.g.*, ML)
  - usually with steep learning curve
- Type theory - Coq
- FOL - ACL2, **Z/Eves**
- see: * *17 provers of the world – nice feature comparison*
  * *taxonomy of proof systems – search space complexity survey*

correctness vs. debugging

- ascertaining properties vs. finding errors
- varying levels of automation

Cornerstones of theorem proving difficulties

- existential quantifier witnesses
- inductive schemes vs. inductive definitions
- global invariants maintenance (*e.g.*, state)
- dependent type checking (*e.g.*, local invariants)

Problems

- low automation, high repetition
  - necessity of key lemmas in the "right" shape
  - required understanding of proof engine
    (*i.e.*, waterfall, skolemization, generalisation, *etc.*)

Difficulties

- deep vs. shallow embedding
  - easy to express, hard prove/read vs.
  - easier to read/prove, harder express
- inadequate syntax

Better proofs via refuted conjectures

- smaller, resilient more automated proofs (*e.g.*, variable instantiation)
- failed attempts (often) gives good hints, if signs are known
- hints serve for both tactics and lemma layouts
- mostly through (manual) intuition, ingenuity and experience

How to identify/transfer these? – via proof planning?

- strategies for proof modularisation
- need for abstract application-oriented theories
  - specialised heuristics for problem classes

    e.g., Z mathematical toolkit, SMT, decision procedures
- identification of helpful weakening rules
  - simplification without expansion
  - choice between backward and forward reasoning

Finding taxonomies of proof effort

- some obvious theorems are not easy
  - `#(S x T) = #S * #T` - 3 bijections related through multiplication
  - `P({0,1})={{},{0},{1},{0,1}}` - exponential case analysis on the set size
- Polya's inventor's paradox: general easier than specific
- Popper's falsifiability principle: expose where it is likely to fail
- Lakato's refutation praise: to learn as much from refutations as from proofs
- use proof planning strategies(?)
- see examples from GC pilot projects in **Z/Eves**

Conclusions

- design decisions based on industrial use of proof systems
- a lot to learn on proof planning, and other proof systems
- case study on instrumenting RODIN for proof planning
- specification language independent annotation language for proofs, conjectures, tactics, *etc.*
- non-sequential proof tactics (*e.g.*, novel tactic schemes)
- Event-B to HOL, like in Overture for VDM ?
- discussion and questions?