Automated Reasoning and Formal Methods

Alan Bundy

School of Informatics,
University of Edinburgh

AI meets Formal Software Development

Automated Reasoning and Formal Methods

- Formal methods generate proof obligations (POs).
- Proving POs requires mathematical expertise.
  - Which is expensive and time-consuming.
- Can PO proofs be automated?
  - Initial attempts in 1970s were unpromising.
- What has changed in 40 years?
- Are we ready to 'mainstream' formal methods into normal software development processes?
  - Greater industrial demand, driven by increased complexity, concurency, cost of failure, etc.
  - What about the late adopters?

Developments in Automated Reasoning

- Computers are much faster and have more memory.
- We have more decision procedures.
  - Whose worst-case complexity is rarely encountered.
- Automatic provers have improved.
  - Open conjectures have been solved.
- Interactive provers integrate these improvements.
  - Have better interfaces.
  - Have proved significant theorems.
- Machine learning is much improved.

Faster, Bigger Computers

- 42 years of Moore's Law since 1970.
  - 2^{21} increase in transistor density.
- We can quickly search huge search spaces.
  - e.g., million queens problem.
- Has led to improved prover performance.
  - Bigger search spaces; faster search.

Decision Procedures

- SAT solving dates from 1962 with DPLL.
  - Essentially same algorithm still state-of-art.
- Model checking now becoming standard,
  - Especially for hardware.
- Satisfiability Modulo Theory (SMT) solvers extend SAT solvers,
  - with linear arithmetic, arrays, bit-vectors, etc.
  - Coverage includes many POs.
- No human intervention required.
  - Except in problem representation, parameter setting, etc.
- Theorem proving undecidable in general.
  - Although, SMT can also be adapted to undecidable theories.
- Knuth-Bendix completion can invent new decision procedures.

Automatic Provers

- First-order provers usually automatic.
- Significant improvements in speed and search capability.
  - Driven by theory advances, CASC and Moore's Law.
  - Superposition, rewriting, built-in unification, clause indexing, faster algorithms.
- Development of libraries of definitions and lemmas.
- EQP proof that all Robbins algebras are boolean was milestone.
- No human intervention required.
  - Except in problem representation, parameter setting, etc.
- Can't handle higher-order, e.g., induction rules.

Inductive Proof

- Needed for reasoning about repetition, invariants, etc.
- Failure of cut elimination.

$$\frac{\Gamma, Lemma \vdash \phi \quad \Gamma \vdash Lemma}{\Gamma \vdash \phi}$$

  - Need for intermediate lemmas, generalisations, non-standard inductions, etc.
  - Cause of infinite branching rate.
- Practical problem for quite simple inductive theorems.
  - For instance, commutativity of $+$.
- Human intervention usually required.
- Some progress in automatic guessing of lemmas.

Interactive Provers

- Gradual improvement of user interfaces.
  - Finer levels of control.
- Integration of decision procedures, first-order provers, counter-example finders, etc.
- Used to verify large ICT systems.
  - e.g., microprocessors, compilers, etc.
  - And important mathematical theorems: 4 Color Theorem, Kepler's Conjecture, etc.
- However, still requires skilled user and long time frame for non-trival theorems.

Machine Learning

- Machine learning now mature and successful field.
- Small amount of work on data-mining proofs and search spaces for common patterns.
- Can we extract generic proof strategies from manual proofs?
  - Then apply these strategies to automate proof of POs from the same family?
- Can we use this to further reduce the skill level and increase the productivity of PO proof?

Is this Enough?

- Where are we now with the automation of PO proof?
- Can we do better with AI techniques?
- What percentage of PO proofs can we aspire to automate?
- Can we help with modelling process?
- What are the implications for the industrial uptake of formal methods?
- What other barriers to this uptake remain?