Automated Reasoning

Jacques Fleuriot

September 14, 2013

Introduction

Jacques Fleuriot

What is it to Reason?

- Reasoning is a process of deriving new statements (conclusions) from other statements (premises) by argument.
- For reasoning to be correct, this process should generally **preserve truth**. That is, the arguments should be **valid**.
- How can we be sure our arguments are valid?
- Reasoning takes place in many different ways in everyday life:
  - **Word of Authority**: we derive conclusions from a source that we trust; e.g. religion.
  - **Experimental science**: we formulate hypotheses and try to confirm them with experimental evidence.
  - **Sampling**: we analyse many pieces of evidence statistically and identify patterns.
  - **Mathematics**: we derive conclusions based on mathematical *proof*.
- Are any of the above methods **valid?**

What is a Proof? (I)

- For centuries, mathematical proof has been the hallmark of logical validity.
- But there is still a **social aspect** as peers have to be convinced by argument.
- This process is open to **flaws**: e.g. Kempe's proof of the Four Colour Theorem.
- To avoid this, we require that all proofs be broken down to their simplest steps and all hidden premises uncovered.

What is a Formal Proof?

- We can be sure there are no hidden premises by reasoning according to **logical form** alone.

Example

Suppose all men are mortal. Suppose Socrates is a man. Therefore, Socrates is mortal.

- The validity of this proof is independent of the meaning of "men", "mortal" and "Socrates."
- Indeed, even a nonsense substitution gives a valid sentence:

Example

Suppose all borogroves are mimsy. Suppose a mome rath is a borogrove. Therefore, a mome rath is mimsy.

Example

Suppose all Ps are Q. Suppose x is a P. Therefore, x is a Q.

Symbolic Proof

- The modern notion of **symbolic formal proof** was developed in the 20^{th} century by logicians and mathematicians such as Russell, Frege and Hilbert.
- The benefit of formal logic is that it is based on a **pure syntax**: a precisely defined symbolic language with procedures for transforming symbolic statements into other statements, based solely on their **form**.
- **No intuition or interpretation is needed**, merely applications of agreed upon rules to a set of agreed upon formulae.

Symbolic Logic (II)

But!

- Formal proofs are bloated!

> *I find nothing in [formal logic] but shackles. It does not help us at all in the direction of conciseness, far from it; and if it requires 27 equations to establish that 1 is a number, how many will it require to demonstrate a real theorem?*
>
> (Poincaré)

- Can automation help?

Automated Reasoning

- Automated Reasoning (AR) refers to reasoning in a computer using **logic**.
- AR has been an active area of research since the 1950s.
- It uses deductive reasoning to tackle problems such as
  - constructing formal mathematical proofs;
  - verifying programs meet their specifications;
  - modelling human reasoning.

Mathematical Reasoning

Automated mathematical theorem proving is a good test domain. Why?

- Intelligent, often non-trivial activity.
- Circumscribed domain with neat bounds which help control reasoning.
- Mathematics is based around logical proof and — in principle — reducible to formal logic.
- Numerous **applications**
  - the need for formal mathematical reasoning is increasing: need for well-developed theories;
  - e.g. **hardware** and **software verification**.

Understanding mathematical reasoning

- Two main aspects have been of interest

  logical: how should we reason; i.e. what are the valid modes of reasoning? We must find a calculus with rigorous rules.

  psychological: how do we actually reason?
- Both aspects contribute to our understanding
- (Mathematical) Logic:
  - shows how to represent mathematical knowledge and inference;
  - does not tell us how to **guide** the reasoning process.
- Psychological studies:
  - do not provide a detailed and precise recipe for how to reason, but can provide advice and hints or **heuristics**;
  - heuristics are especially valuable in automatic theorem proving — however, finding good heuristics is a hard task.

Automated Theorem Proving

- Many systems: Coq, Isabelle, HOL, PVS, Otter, ...
  - provide a mechanism to formalise proof;
  - user-defined concepts in an **object-logic**;
  - user expresses formal conjectures about concepts.
- Can these systems find proofs **automatically**?
  - In some cases, yes!
  - But sometimes it is too difficult.
- Complicated verification tasks are usually done in an **interactive setting**.

Interactive Proof

- User guides the inference process to prove a conjecture (hopefully!)
- Systems provide:
  - tedious bookkeeping;
  - standard libraries (e.g. lists, complex numbers);
  - guarantee of correct reasoning;
  - varying degrees of automation
    - powerful simplification process;
    - may have decision procedures for decidable theories such as linear arithmetic, propositional logic etc.

What's it like?

- Interactive proof can be difficult but is also very rewarding.
- It combines aspects of programming and mathematics.
- **Difficult** to learn:
  - it is important that you know how to look up and apply theorems;
  - there are often many **tactics** for automation, and it takes time to understand them.
- **Representation** matters!

Limitations (I)

Do you think formalised mathematics is:

complete	can every statement be proved or disproved?
consistent	no statement can be both true and false?
decidable	there exists a terminating procedure to determine the truth or falsity of any statement?

Limitations (II)

- **Gödel's Incompleteness Theorems** showed that, if a formal system can prove certain facts of basic arithmetic, then there are other statements that cannot be proven nor refuted in that system.
- In fact, if such a system is consistent, it cannot prove that it is so.
- Moreover, Church and Turing showed that **first-order logic was undecidable.**
- Do not be disheartened!
- We can still prove many interesting results using logic.

What is a proof? (II)

- **Computerised proofs** are causing **controversy** in the mathematical community
  - proof steps may be in the hundreds of thousands;
  - they are impractical for mathematicians to check by hand;
  - it can be hard to guarantee proofs are not flawed;
  - e.g. Hales' proof of Kepler's Conjecture.
- The acceptance of a computerised proof can rely on
  - formal specifications of concepts and conjectures;
  - **soundness** of the prover used;
  - size of the community using the prover;
  - **surveyability** of the proof.

Isabelle

In this course we will be using the popular interactive theorem prover **Isabelle/HOL**:

- It is based on the simply typed lambda calculus with rank-1 (ML-style) polymorphism.
- It has an extensive **theory library**.
- It supports two styles of proof (procedural and declarative).
- It has a powerful simplifier, classical reasoner, decision procedures for decidable fragments of theories.
- It is widely accepted as a **sound** and **rigorous** system!

Soundness in Isabelle

- Isabelle follows the **LCF approach** to ensure soundness.
- We declare our conjecture as a goal, where we can then:
  - use a known theorem or axiom to prove the goal immediately;
  - use a **tactic** to prove the goal;
  - use a tactic to transform the goal into new subgoals.
- Tactics construct the formal proof in the background.
- Axioms are generally discouraged; definitions are preferred.
- New concepts should be **conservative extensions** of old ones.

Course Contents

- **Logics**: propositional, first-order, aspects of higher-order logics and linear temporal logic.
- **Formalized mathematics**
- **Interactive theorem proving**: introduction to theorem proving with Isabelle/HOL.
- **Model Checking**: theory and algorithms. NuSMV model checker.

Module Outline

- 2 lectures per week: 16.10-17.00 Mon/Thurs.
- 2 coursework assignments and exam
  - Examination: 60%.
  - Coursework: 40% (20% each).
- Lecturers
  - Jacques Fleuriot
    - Office: IF-2.06
    - Email: `jdf@inf.ed.ac.uk`.
  - Paul Jackson
    - Office: IF-4.05
    - Email: `pbj@inf.ed.ac.uk`
- Coursework demonstrators
  - First half of course:
    - Petros Papapanagiotou
    - Email: p.papapanagiotou@sms.ed.ac.uk
  - Second half of course: TBC

Useful Course Material

- AR web pages:
  `http://www.inf.ed.ac.uk/teaching/courses/ar`.
- Lecture slides found on the course website.
- Set course textbooks:
  - M. Huth and M. Ryan. **Logic in Computer Science: Modelling and Reasoning about Systems**, Cambridge University Press, 2$^{\text{nd}}$ Ed. 2004;
  - A. Bundy. **The Computational Modelling of Mathematical Reasoning**, Academic Press, 1983 available on-line at `http://www.inf.ed.ac.uk/teaching/courses/ar/book`.
- Isabelle Cheat Sheet
  http://www.phil.cmu.edu/∼avigad/formal/FormalCheatSheet.pdf
- Other material — recent research papers, technical reports, etc.