Automated Reasoning

Lecture 1: Introduction

Jacques Fleuriot
jdf@inf.ed.ac.uk

What is it to Reason?

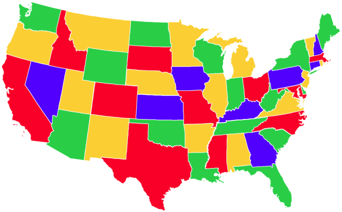
- Reasoning is a process of deriving new statements (conclusions) from other statements (premises) by argument.
- For reasoning to be correct, this process should generally **preserve truth**. That is, the arguments should be **valid**.
- How can we be sure our arguments are valid?
- Reasoning takes place in many different ways in everyday life:
  - **Word of Authority**: derive conclusions from a trusted source.
  - **Experimental science**: formulate hypotheses and try to confirm or falsify them by experiment.
  - **Sampling**: analyse evidence statistically to identify patterns.
  - **Mathematics**: we derive conclusions based on deductive *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.

> *A proof is a repeatable experiment in persuasion*
>
> — Jim Horning[1]

- This process is open to **flaws**: e.g., Kempe's acclaimed 1879 "proof" of the Four Colour Theorem, etc.



[1] https://en.wikipedia.org/wiki/Jim_Horning

What is a Formal Proof?

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

Example

Suppose all humans are mortal. Suppose Socrates is human. Therefore, Socrates is mortal.

- The validity of this proof is independent of the meaning of "human", "mortal" and "Socrates".
- 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.[2]

Example

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

---

[2] https://en.wikipedia.org/wiki/Mimsy_Were_the_Borogoves

Symbolic Logic

- The modern notion of **symbolic proof** was developed in the late-19th and 20th century by logicians and mathematicians such as Bertrand Russell, Gottlob Frege, David Hilbert, Kurt Gödel, Alfred Tarski, Julia Robinson, ...
- 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.
- Traditionally viewed as part of Artificial Intelligence (AI $\neq$ Machine Learning!).
- It uses deductive reasoning to tackle problems such as
  - constructing formal mathematical proofs;
  - verifying that programs meet their specifications;
  - modelling human reasoning.

Mathematical Reasoning

Mechanical mathematical theorem proving is an exciting field. Why?

- Intelligent, often non-trivial activity.
- Circumscribed domain with bounds that 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**;
  - e.g. research mathematics, where formal proofs are starting to be accepted.

Understanding mathematical reasoning

- Two main aspects have been of interest
  - **Logical**: how should we reason; what are the valid modes of reasoning?
  - **Psychological**: how do we 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 — but finding good ones is a hard task.

Mechanical Theorem Proving

- Many systems: Isabelle, Coq, HOL Light, PVS, Vampire, E, …
  - 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., arithmetic, lists, real analysis);
  - guarantee of correct reasoning;
  - varying degrees of automation:
    - powerful simplification procedures;
    - may have decision procedures for decidable theories such as linear arithmetic, propositional logic, etc.;
    - call fully-automatic first-order theorem provers on (sub-)goals and incorporating their output e.g. Isabelle's sledgehammer.

What is it like?

- Interactive proof can be challenging, but also rewarding.
- It combines aspects of **programming** and **mathematics**.
- Large-scale interactive theorem proving is relatively new and unexplored:
    - Many potential application areas are under-explored
    - Not at all clear what The Right Thing To Do is in many situations
    - New ideas are needed all the time
    - This is what makes it **exciting!**
- What we do know: **Representation** matters!

$\sqrt{2}$ is irrational in Isabelle

```
theorem sqrt_prime_irrational:
  assumes "prime (p::nat)"
  shows "sqrt p ∉ ℚ"
proof
  from ‹prime p› have p: "1 < p" by (simp add: prime_nat_def)
  assume "sqrt p ∈ ℚ"
  then obtain m n :: nat where
      n: "n ≠ 0" and sqrt_rat: "¦sqrt p¦ = m / n"
    and gcd: "gcd m n = 1" by (rule Rats_abs_nat_div_natE)
  from n and sqrt_rat have "m = ¦sqrt p¦ * n" by simp
  then have "m² = (sqrt p)² * n²"
    by (auto simp add: power2_eq_square)
  also have "(sqrt p)² = p" by simp
  also have "... * n² = p * n²" by simp
  finally have eq: "m² = p * n²" ..
  then have "p dvd m²" ..
  with ‹prime p› have dvd_m: "p dvd m" by (rule prime_dvd_power_nat)
  then obtain k where "m = p * k" ..
  with eq have "p * n² = p² * k²" by (auto simp add: power2_eq_square ac_simps)
  with p have "n² = p * k²" by (simp add: power2_eq_square)
  then have "p dvd n²" ..
  with ‹prime p› have "p dvd n" by (rule prime_dvd_power_nat)
  with dvd_m have "p dvd gcd m n" by (rule gcd_greatest_nat)
  with gcd have "p dvd 1" by simp
  then have "p ≤ 1" by (simp add: dvd_imp_le)
  with p show False by simp
qed


corollary sqrt_2_not_rat: "sqrt 2 ∉ ℚ"
  using sqrt_prime_irrational[of 2] by simp
```

Limitations (I)

Do you think formalised mathematics is:

1. **Complete**: can every statement be proved or disproved?
2. **Consistent**: no statement can be both true and false?
3. **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 or 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** is **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's proof of the Kepler 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;
  - (for specialists) the kind of logic used.

Isabelle

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

- It is based on the simply typed λ-calculus with rank-1 (ML-style) polymorphism.
- It has an extensive **theory library**.
- It supports two styles of proof: procedural ('apply'-style) and declarative (structured).
- It has a powerful simplifier, classical reasoner, decision procedures for decidable fragments of theories.
- It can call automatic first-order theorem provers.
- 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, and then we can:
  - use a known theorem or axiom to prove the goal;
  - 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 (in brief)

- **Logics**: first-order, aspects of higher-order logic.
- **Reasoning**: unification, rewriting, natural deduction.
- **Interactive theorem proving**: introduction to theorem proving with Isabelle/HOL.
  - Representation: definitions, locales etc.
  - Proofs: procedural and structured (Isar) proofs.
- **Formalised mathematics**.

Module Outline

- 2 lectures per week 14:10–15:00:
  - Tuesday: 1.02, 21 Buccleuch Place, Central Campus
  - Thursday: G.02 - Classroom 2, High School Yards Teaching Centre, Central Campus
- 7 tutorials (starting Week 3)
- Lab sessions (drop-in):
  - Mondays 09:00–11:00 (starting Week 3, to be confirmed)
  - 4.12, Appleton Tower
- 1 assignment and 1 exam:
  - Examination: 60%
  - Coursework: 40% (so this is a non-trivial part of the course)
- Lecturer:
  - Jacques Fleuriot
  - Office: IF 2.15
- TA:
  - Imogen Morris
  - Email: `s1402592@sms.ed.ac.uk`

Useful Course Material

- AR web pages: `http://www.inf.ed.ac.uk/teaching/courses/ar`.
- Lecture slides are on the course website.
- Recommended course textbooks:
  - T. Nipkow and G. Klein. ***Concrete Semantics with Isabelle/HOL***, Springer, 2014.
  - M. Huth and M. Ryan. ***Logic in Computer Science: Modelling and Reasoning about Systems***, Cambridge University Press, 2^{nd} Ed. 2004.
  - J. Harrison. ***Handbook of Practical Logic and Automated Reasoning***, Cambridge University Press, 2009.
  - 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`.
- Other material — recent research papers, technical reports, etc. will be added to the AR webpage.
- Class discussion forum (open for registration): `http://piazza.com/ed.ac.uk/fall2019/infr09042`.