Big Proof SIG: Formal Methods in Education

Jeremy Avigad

Department of Philosophy and
Department of Mathematical Sciences
Carnegie Mellon University

July 24, 2017

Goals

Goals of today's meeting:

Share expertise and resources regarding the use of formal methods in education.

This is open-ended: we can specialize the topic in a number of ways.

Topics

Formal methods:

- interactive theorem proving
- automated reasoning
- any kinds of formal methods

Audience:

- the public at large
- undergraduates
- graduate students
- advanced researchers

Topics

Audience:

- mathematicians
- computer scientists
- logicians and type theorists

Reasons it is important to reach mathematicians:

- formal methods can benefit mathematics
- mathematics can benefit formal methods

Topics

Extent:

- We can teach classes about formal methods.
- We can incorporate formal methods in other classes.

Desiderata

We need to share:

- textbooks (it is nice when they are freely available online)
- provers and tools
- experience and expertise

Resources

Almost every theorem prover has:

- reference manuals
- tutorial introductions

Resources

Teaching interactive theorem proving to computer science students:

- Benjamin Pierce at al., Software Foundations
  `https://softwarefoundations.cis.upenn.edu`
- Adam Chlipala, *Certified Programming with Dependendent Types*
  `http://adam.chlipala.net/cpdt/`
- Adam Chlipala, *Formal Reasoning About Programs*
  `http://adam.chlipala.net/frap/`
- Tobias Nipkow and Gerwin Klein, *Concrete Semantics*
  `http://www.concrete-semantics.org/`

Resources

Teaching type theory to computer science students:

- Peter Dybjer, "Types for Programs and Proofs"
  `http://www.cse.chalmers.se/edu/year/2013/course/DAT140_Types/`
- Femke van Raamsdonk, "Type Theory and Coq"
  `http://www.cs.ru.nl/~freek/courses/tt-2016/`
- Avigad, "Interactive and Automated Theorem Proving"
  (Using the Lean tutorials, `http://leanprover.github.io`)

Resources

Here is a link to J. Strother Moore's class, "A Formal Model of the Java Virtual Machine:"

`https://www.cs.utexas.edu/users/moore/classes/cs378-jvm/index.html`

It is based on the Moore method (no relation to J):

`https://en.wikipedia.org/wiki/Moore_method`

Resources

What about teaching mathematical reasoning to beginning undergraduates?

- Daniel Velleman, "How to Prove It," and his *Proof Designer*: `https://app.cs.amherst.edu/~djvelleman/pd/pd.html`
- Avigad, Lewis, van Doorn, "Logic and Proof" `https://leanprover.github.io/logic_and_proof/`

Logic and Proof

Goals of the course:

- Teach students to write ordinary mathematical proof.
- Teach students how to use symbolic logic (to make assertions, prove assertions, and specify properties).
- Teach students how to use an interactive theorem prover (Lean).

Logic and Proof

Mathematical topics: sets, relations (order, equivalence relations), functions, induction, combinatorics, probability, elementary analysis (the real numbers and limits), axioms of set theory

Logic topics: propositional logic, natural deduction, first-order logic, truth assignments and models (informally), higher-order quantifiers

Lean exercises: e.g. propositional and first-order logic, set-theoretic identities, showing that the composition of surjective functions is surjective, or proving the commutativity of multiplication by induction.

Each strand was standard. The main novely was in combining them.

Logic and Proof

Observations:

- Make it clear that there are three distinct languages:
  - ordinary mathematics
  - symbolic logic
  - formal proof languages

  Be up front about the differences.
- If you do, students will not get them confused.
- The parallel developments support their understanding.
- Students liked the course.