Computational Proof Assistants

Kevin Liu

Canadian Undergraduate Mathematics Conference, July 2024

Table of Contents

Motivation

- Writing correct software is hard!
- Theorem provers can ensure mathematical correctness of code (Compcert)
- Formal verification is becoming widely used in industry for critical tasks (Microsoft, Intel)
- Can formally prove many results in mathematics (4-color theorem)
- Computation of BB(5) (last week)!

Coq

- Coq is an interactive proof assistant for formal verification
- Developed in 1984 by INRIA (France)
- Includes a programming language (Gallina) and can check proofs for correctness
- Can “extract” Coq proofs into OCaml or Haskell scripts

Example Coq Programs

See VSCode!

The Curry-Howard Isomorphism

Theorem

- *Types are propositions*
- *Programs and proofs are the same*

Examples

- $$A \implies B \equiv f : A \to B$$
- $$A \wedge B \equiv (A, B)$$

Remark

- Direct link between computation and logic
- Proofs can be run!

Another example

Example

Consider the function

```
    def map (x: A, y: A -> B):
        return y(x)
```

which has type $map : A \to (A \to B) \to B$. Thus, if we have evidence for A and evidence for $A \implies B$, then we have evidence for B. (Modus ponens) How can we know this works? Type checking!

References and Further Readings

- Coq Website
- Coq GitHub
- Software Foundations
- Coq in a Hurry
- Curry Howard for Dummies
- CS 3110 Textbook @ Cornell
- Computerphile video