SOFTWARE FOUNDATIONS

VOLUME 1: LOGICAL FOUNDATIONS

TABLE OF CONTENTS INDEX ROADMAP

POSTSCRIPT

Congratulations: We've made it to the end!

Looking Back

We've covered quite a bit of ground so far. Here's a quick review...

- *Functional programming*:
  - "declarative" programming style (recursion over persistent data structures, rather than looping over mutable arrays or pointer structures)
  - higher-order functions
  - polymorphism

- *Logic*, the mathematical basis for software engineering:

```
          logic                          calculus
--------------------   ~   ----------------------------
software engineering       mechanical/civil engineering
```

  - inductively defined sets and relations
  - inductive proofs
  - proof objects

- *Coq*, an industrial-strength proof assistant
  - functional core language
  - core tactics
  - automation

Looking Forward

If what you've seen so far has whetted your interest, you have two choices for further reading in the *Software Foundations* series:

- *Programming Language Foundations* (volume 2, by a set of authors similar to this book's) covers material that might be found in a graduate course on the theory of programming languages, including Hoare logic, operational semantics, and type systems.
- *Verified Functional Algorithms* (volume 3, by Andrew Appel) builds on the themes of functional programming and program verification in Coq, addressing a range of topics that might be found in a standard data structures course, with an eye to formal verification.

Other sources

Here are some other good places to learn more...

- This book includes some optional chapters covering topics that you may find useful. Take a look at the table of contents and the chapter dependency diagram to find them.
- For questions about Coq, the coq area of Stack Overflow (https://stackoverflow.com/questions/tagged/coq) is an excellent community resource. - Here are some great books on functional programming - Learn You a Haskell for Great Good, by Miran Lipovaca [Lipovaca 2011]. - Real World Haskell, by Bryan O'Sullivan, John Goerzen, and Don Stewart [O'Sullivan 2008] - ...and many other excellent books on Haskell, OCaml, Scheme, Racket, Scala, F sharp, etc., etc. - And some further resources for Coq: - Certified Programming with Dependent Types, by Adam Chlipala [Chlipala 2013]. - Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions, by Yves Bertot and Pierre Casteran [Bertot 2004]. - If you're interested in real-world applications of formal verification to critical software, see the Postscript chapter of _Programming Language Foundations_. - For applications of Coq in building verified systems, the lectures and course materials for the 2017 DeepSpec Summer School are a great resource. https://deepspec.org/event/dsss17/index.html *)