CS 4160
Formal Verification
Prof. Clarkson
Spring 2019
Approaches to validation

• Social
  – Code reviews
  – Extreme/Pair programming

• Methodological
  – Design patterns
  – Test-driven development
  – Version control
  – Bug tracking

• Technological
  – Static analysis ("lint" tools, FindBugs, …)
  – Fuzzers

• Mathematical
  – Sound type systems
  – "Formal" verification

Less formal: Techniques may miss problems in programs

All of these methods should be used!

Even the most formal can still have holes:
• did you prove the right thing?
• do your assumptions match reality?

More formal: eliminate with certainty as many problems as possible.
Verification

• In the 1970s, scaled to about tens of LOC
• Now, research projects scale to real software:
  – **CompCert**: verified C compiler
  – **seL4**: verified microkernel OS
  – **Ynot**: verified DBMS, web services
• In another 40 years?
Automated theorem provers

- theorem
- prover
- proof
- counter-example
- timeout
Automated theorem provers

• **Z3**: Microsoft started shipping with device driver developer's kit in Windows 7

• **ACL2**: used to verify AMD chip compliance with IEEE floating-point specification, as well as parts of the Java virtual machine
Proof assistant

- theorem
- human guidance

assistant

proof
Proof assistants

• **NuPRL** [Prof. Constable]: Formalization of mathematics, distributed protocols, security

• **Coq**: CompCert, Ynot [Dean Morrisett]
Coq

• 1984: Coquand and Huet implement Coq based on calculus of inductive constructions
• 1992: Coq ported to Caml
• Now implemented in OCaml
Coq for program verification

- Coq program
- Coq theorem
- Guidance with tactics
- Verified OCaml program
- Proof of theorem
Coq's full system
Subset of Coq we'll use
CAUTION: HIGHLY ADDICTIVE
LOGISTICS
Prof. Michael Clarkson

- PhD 2010 **Cornell University**
- BS (CS) & BM (piano) 1999 **Miami University**
- Regularly teach: CS 3110 (OCaml), CS 5430 (security)

- AMA: D&D, wine, Gregorian chant
- I like hats
Course website

https://www.cs.cornell.edu/courses/cs4160/2019sp/
Acknowledgment

CS 4160 is based on the online textbook *Software Foundations* and especially on the work of Prof. Benjamin C. Pierce at the University of Pennsylvania and Prof. Andrew Appel at Princeton University in courses they teach.