CS 3110

Lecture 23: Formal Verification

Prof. Clarkson

Fall 2014

Today's music: *Hedwig's Theme*
from soundtrack to *Harry Potter and the Sorcerer's Stone*

Review

Current topic:

- How to reason about correctness of code
- Started with informal arguments
- Developed formal logic
- Began mechanizing formal logic in Coq

Today:

- Finish formal logic in Coq—automated proofs
- Mechanically verify correctness of the world's smallest compiler

Question #1

How excited are you about Prelim 2?

A. Excited

B. Super excited

C. Mega excited

D. Ultra excited

E. Super-mega-ultra excited

Prelim 2

- Thursday night
  - Your choice of 5:30-7:00 pm or 7:30-9:00 pm
  - Please arrive 15 minutes early to settle in
  - Three rooms, assigned by NetID (see Piazza)
- Closed book, with one page of notes
  - (8.5x11" two-sided)
- Covers Lecture 12 through Recitation 19, inclusive
  - plus slides 7-10 on "theories" in Lecture 22
  - plus PS4 and PS5
  - minus lecture 17 on "dependent types"
  - minus lecture 20 on "effective OCaml"

Coq

- A functional programming language
- A proof assistant
  - You give tool a theorem
  - You and tool cooperatively find proof
- Implemented in OCaml
- Can produce verified OCaml code

Coq3110.v

- We went through the rest of the file, starting with conjunction

VerifyCompiler.v

- We went through the file.

Wizardry

- If all that Coq seemed like magic, don't worry:
  - I won't ask you to read or write any Coq on exams
  - I might give an optional, bonus PS7 on Coq
- But you're no longer a muggle:
  - You know that formal verification exists
  - You have understanding of how to do it

The Future of 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?

My own use of Coq

- *Authorization logic*
  - Reasoning about security of actions take by agents in a distributed system
  - Formalized a logic in Coq, proved its correctness
  - http://www.cs.cornell.edu/~clarkson/projects/focal/
- *Hyperproperties logic*
  - Reasoning about whether programs leak secret information
  - Work in progress: formalizing a logic in Coq, proving its correctness
  - http://www.cs.cornell.edu/~clarkson/papers/clarkson_hyper_tl.pdf

Please hold still for 1 more minute

WRAP-UP FOR TODAY

Upcoming events

- **Prelim 2 on Thursday**

This is verified.

THIS IS 3110