CS 494 – Provably Correct Programming

William Mansky

Software works badly

NASA engineers finally came to the conclusion that there were too many files on the file system
https://en.wikipedia.org/wiki/Spirit_(rover)

possible for the stack to grow large enough to overwrite data that could cause unintended acceleration
https://en.wikipedia.org/wiki/2009%E2%80%932011_Toyota_vehicle_recalls

Knight Capital Says Trading Glitch Cost It $440 Million

BY NATHANIEL POPPER AUGUST 2, 2012 9:07 AM 356

Runaway Trades Spread Turmoil Across Wall St.

https://dealbook.nytimes.com/2012/08/02/knight-capital-says-trading-mishap-cost-it-440-million/

Can we change the way we program?

STOP DOING JAVASCRIPT

- DOCUMENTS WERE NOT SUPPOSED TO BE TURING-COMPLETE
- YEARS OF WEB TECHNOLOGY yet NO REAL-WORLD USE FOUND for doing more than HTML FORMS
- Wanted to do more anyway for a laugh? We had a tool for that: It was called “ADOBE FLASH”
- “Yes please connect the Redux thunk to a Suspense. Please instantiate WebAssembly streaming.” - Statements dreamed up by the utterly Deranged

LOOK at what Web Developers have been demanding your Respect for all this time with all the computers & compilers we built for them

(This is REAL Javascript, done by REAL Web Developers)

“Hello I would like `undefined` apples please”

They have played us for absolute fools

Can we change the way we program?

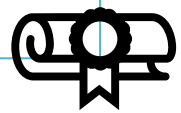
- Programs in safer languages (Rust, OCaml, …) still have bugs

- Most programmers don’t get to choose what language they write in!
  - Need to maintain/interoperate with existing code
  - We mostly write in the languages we know

Can we change the way we program?

- Prove programs correct!

```
i = 1;
while(i <= n){
   r = r * i;
   i++;
}
```



This course:

How can we write these proofs?

How can we write programs so they're easier to prove?

- This program will:
  - never overflow its stack
  - never dereference a null pointer
  - never call a function without meeting its preconditions
  - always return the right result!

Questions?

Top

Welcome to the Course!

- This is CS 494, Provably Correct Programming
- I'm glad you're here!
- Meets TR 2:00-3:15 PM
- You can attend:
  - in person, in TBH 180C (**after the first two weeks**)
  - online live, through Echo360 on Blackboard
  - online asynchronously, by watching recorded lectures on Echo360
- Office hours Monday 12-1 and Thursday 11-12, and by appointment, in SEO 1331 and on Blackboard Collaborate
  - Office hours are great for homework help, or just to say hi!

Course Information

- Professor: William Mansky (he/him) (mansky1@uic.edu)
- Prerequisites: CS 301 (logic and proofs)
- Website: https://www.cs.uic.edu/~mansky/teaching/cs494sf/sp22/
- Anonymous in-class questions: https://pollev.com/wmansky771
- In-person lectures will be streamed and recorded via Echo360 on Blackboard
- Discussion board on Piazza, assignments via Gradescope (entry code ERYG7D)

Asking questions

- In class: raise your hand (or type in BBCollab chat) anytime
- You can ask questions anonymously with PollEverywhere (https://pollev.com/wmansky771)
- On Piazza
  - Can ask/answer anonymously
  - Can post privately to instructors
  - Can answer other students' questions
- In office hours, Monday 12-1 and Thursday 11-12

- If you have a question, someone else probably has the same question!

Grading

- Exercises: 25%
- Assignments: 50%
- Project: 25%

Exercises

- In each class, we'll work through some example problems/proofs
- Submit via Gradescope
- Due at the start of the next class
- You get credit as long as you make some progress on the problem

Assignments

- Programming/proving assignments
- Submit via Gradescope
- Due at 2 PM on the due date
- You can discuss strategy with other students, but don't look at each other's code!
- Cite your sources (websites, other students, stackoverflow, etc.)
- You'll get most of the credit for attempting a problem, even if you don't finish it – do what you can, and we'll work through tricky ones in class after the deadline

Questions?

Top

Getting Started with Proofs

- You already know how to write programs: languages, compilers, IDEs, etc.
- How do we write proofs? The same way: with digital tools!

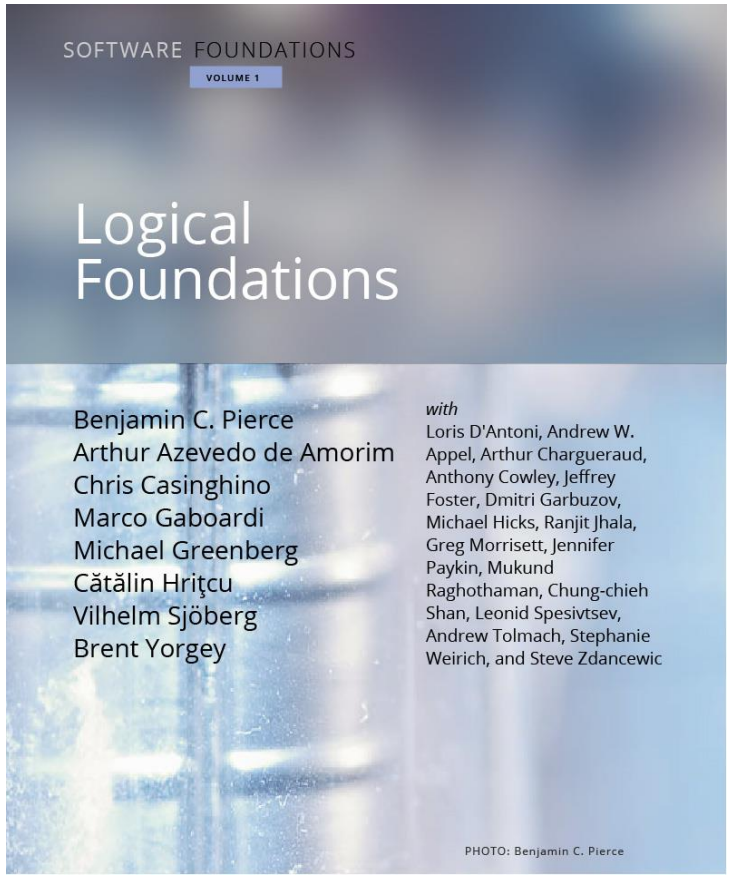
- Tool #1: The Coq proof assistant (https://coq.inria.fr/)
- For the first ~4 weeks, we'll learn how to use it to write guaranteed-correct mathematical proofs
- After that, we'll apply those techniques to programs!

Interactive Theorem Provers

- In the theorem prover, we can:

1. Write **definitions**, in a math-like programming language
2. Write **proofs** about those definitions, using logic “tactics”
3. See the **proof state** at each point in a proof (what do we know? what do we still need to show?)
4. Automatically **check** that each step of our proofs is valid

Logical Foundations



- Online textbook, each chapter is a file that can be run in Coq
- Contents:
  - Introduction to Coq
  - Basic logic and functional programming
  - More advanced logic, mostly induction
  - How to describe the behavior of programs

https://softwarefoundations.cis.upenn.edu/lf-current/index.html

Getting Started with Coq

- Available online at https://coq.inria.fr/
- You can download installers for Windows and Mac from the website
- Coq file extension is .v
- If it matters, we'll use version 8.13.2

- Two main IDEs: CoqIDE (ships with Coq) and Visual Studio Code (VSCoq extension)

Today's Exercise

1. Download and install Coq (https://github.com/coq/platform/releases/tag/2021.09.0, or from the download links at https://coq.inria.fr/)
2. Download and unpack the textbook (https://softwarefoundations.cis.upenn.edu/lf-current/index.html)
   - It's a .tgz file, so you may need to install 7-zip (https://www.7-zip.org/) to unpack it
3. Run `make` in the textbook's folder to compile the textbook. If you don't have a command line with `make`, you'll need to set one up: I use Cygwin (https://cygwin.com/install.html)
4. If you finish, run demo.v, then submit it for Exercise 1/11 on Gradescope. If you haven't finished, submit a description of where you're stuck instead.

If you get stuck at any point, say so in chat, on https://pollev.com/wmansky771, or on Piazza

Questions?

Top