A puzzle ...

Suppose that you want to run some JavaScript programs—all the cool kids are doing it.

However, you are deathly afraid that your mom/significant other/boss will see the “curse” word 'b00' appear on your screen as the result of running the program.

What do you do?

---

JavaScript

Josh - close the lid - don’t run any programs
SAFE but limiting "one"
Paul - look at you child - calm review Manual
Depends Paul’s paranoia, this one (Paul’s)
Kira - testing the program by as much possible is "private"
the answer, depends on “paranoia” Manual
human error, impossible to be eliminated.

---

1. Josh - Stack Analyz
2. Build a language tool that just reads the source code.
3. Verifier: program -> bool

---

How do I write this verifier?
1. JavaScript
2. always return NO
3. Paul - No if I see console.log
   SAFE (caut.)
4. NO if console.log('b00') UNSAFE
   var s = 'b00'
   console.log(s) var p = console.log(p('b00'))
Introductions: Your guide this semester

- Office hours: TR 8:30-9:30, and by appointment in ECOT 621

Introductions: Another guide

Arlen Cox

Today

- Some context and motivation
- Program analysis in a nutshell
- Goals for this course
- Requirements and grading
- Course summary

- Convince you that program analysis is cool and exciting!
Meta-Level Information

• Please interrupt at any time!
• It's completely ok to say:
  - I don't understand. Please say it another way.
  - Slow down! Wait, I want to read that!
• Discussion, not lecture

No Useless Memorization

• I will not waste your time with useless memorization
• This course will cover complex subjects
• I will teach their details to help you understand them the first time
• But you will never have to memorize anything low-level
• Rather, learn to apply broad concepts

Discussion: What is Program Analysis?

• Static Program Analysis?
  - Analyze code while running it
• Dynamic Program Analysis?
  - "What is program analysis?"

Is Program Analysis Useful?

• How is it used?
Big Motivation: Software Quality

Software errors cost a lot

~$60 billion annually (~0.5% of US GDP)
- 2002 National Institute of Standards and Technology report

- total annual revenue of Microsoft

- 10x annual budget of NSF

But there’s hope in program analysis

Because program analysis can eliminate entire classes of bugs

For example,
- Reading from a closed file: read( ) ✗
- Reacquiring a locked lock: acquire( ) ✗

How?
- Systematically examine the program
- Simulate running program on “all inputs”
- ”Automated code review”
Program analysis by example: Checking for double acquires
Simulate running program on "all inputs"

// x now points to an unlocked lock
acquire(x);
... code ...

Program analysis by example: Checking for double acquires
Simulate running program on "all inputs"

// x now points to an unlocked lock in a linked list
acquire(x);
... code ...

Uh oh!

- Rice's Theorem: “Any non-trivial property of programs is undecidable”

Abstract!

... code ...
// x now points to an unlocked lock in a linked list
acquire(x);
... code ...

Abstract Interpretation
[Cousot and Cousot 1977]

Abstractions

- Examples?

\[
\mathbb{Z} \xrightarrow{\text{int}} \{-, 0, +\}
\]

\[
\mathbb{Z} \xrightarrow{\text{odd}} \{\text{odd, even}\}
\]

\[
\mathbb{Z} \xrightarrow{\text{even}} \{\text{even}\}
\]

\[
\mathbb{Z} \xrightarrow{\text{less}} \{\text{less}, \geq 42\}
\]

\[
\mathbb{Z} \xrightarrow{\text{leq}} \{\text{leq}, \geq 42\}
\]

Should we go home then?
Goals

**Goal 1**
Learn to build program analyses

**Goal 2**
Grok the theory behind correct program analyses

**Goal 3**
Understand current PL research (POPL, PLDI, OOPSLA, TOPLAS, ...)

- abstract semantics
- abstract domain
- abstraction function for sets
- concretization function
Most Important Goal

Have Lots of Fun!

Requirements

Project-Based 7000 Graduate Course

- Expectations of You

Prerequisites

- "Programming experience"
  - building analyses in OCaml
  - ideal: undergraduate compilers (e.g., CSCI 4555)

- "Mathematical maturity"
  - basic logic
  - ideal: programming language semantics (e.g., CSCI 5535)

- Not strict but strongly advised. If you do not satisfy these, please see me.

Assignments

- Reading and participation (each meeting)
- Homworks
- (Possible) Presentation of Papers
- Final Semester project

Reading and Participation

- ~2 papers/book chapter, each meeting
  - Spark class discussion, post/bring questions
- Online discussion
  - Post ≥1 substantive comment, question, or answer for each lecture
  - Due before the next meeting
What is “substantive”?

- May be less than a standard blog post but more than a tweet.
- Some examples:
  - Questions
  - Thoughtful answers
  - Clarification of some point
  - What you think is the main point in the reading set.
  - An idea of how some work could be improved
  - Comments on a related web resource related
- Intent: take a moment to reflect on the day’s reading/discussion (not to go scour the web)

Homework

- Homework
  - Guided practice
  - Minimal (2-3 2-week assignments)
  - Collaborate with peers (but acknowledge!)

Semester Project

- Implement an advanced abstract domain and apply your analysis
- Write a ~5-8 page paper (conference-like)
- Give a ~15-20 minute presentation
- Reachable Goal: Publishable after 1 more semester of work.

Course At-A-Glance

- Part I: Program Analysis Foundations
  - Interleave implementation and theory
  - Textbook:
    - Other notes and papers
- Part II: Research Applications

Possible Special Topics

- Software model checking
- Shape analysis
- Symbolic/concolic execution
- What do you want to hear about?
Next Week: Model Checking

- Verify properties or find bugs in software
- Take an important program (e.g., a device driver)
- Merge it with a property (e.g., no deadlocks)
- Transform the result into a boolean program
- Use a model checker to exhaustively explore the resulting state space
  - Result 1: program provably satisfies property
  - Result 2: program violates property "right here on line 92,376!"

Remember

- Join the Piazza and introduce yourself
- Write a few sentences on why you are taking this course