Program Analysis:

Introduction and Basics (Part 2)

Course page:

http://software-lab.org/teaching/winter2020/pa/

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart

Winter 2020/2021

Plan for Today

- **Introduction**
  - What the course is about
  - Why it is interesting
  - How it can help you
- **Organization** 
  - Lectures, exercises, course project
  - Final exam
- **Foundations**
  - Grammars, ASTs, CFGs, etc.

Organization

- Lectures
- Exercises
- Course project
- Final exam

Organization

Grading:

- **Lectures**
- **Exercises** → **10%**
- **Course project** → **40%**
- **Final exam** → **50%**

Lectures

- 10 topics
- Each topic: 1 or 2 weeks
- Videos available via YouTube
- Recommended week(s) to watch

Exercises

- **3 exercises**
- **Pen and paper**
- **Timeline**
  - Published on day X
  - Submission due on X + 7 days
  - Discussion on X + 10 days
- **Individual work: No collaboration or sharing of solutions**

Course Project

- **Design, implement, and evaluate a program analysis based on an existing framework**
  - Data flow analysis of JavaScript code
  - Based on Google Closure compiler
- **Individual, independent project**
  - Mentor available for questions

Course Project: Timeline

- **Published on November 27**
- **Three progress meetings**
- **Due on February 12**
  - Implementation and results
  - Report
- **Presentation: February 15 to 19**

Course Project: Timeline

Grading:
(% of overall grade)

- **Published on November 27**
- **Three progress meetings** — **10%**
- **Due on February 12** — **10%**
  - Implementation and results — **10%**
  - Report — **10%**
- **Presentation: February 15 to 19**

Academic Integrity

- Work you submit must be your own
- Unauthorized group efforts and any form of plagiarism are considered academic dishonesty and will be punished
- Allowed to discuss the problem with your peers, but not to reuse any part of an existing solution

Content

Introduction and basics

Operational semantics

Data flow analysis

Call graphs

Random test generation/fuzzing

Symbolic and concolic execution

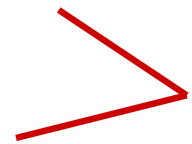
Information flow analysis

Slicing

Path profiling

Analyzing concurrent programs

Content



Foundations

Introduction and basics

Operational semantics

Data flow analysis

Call graphs

Random test generation/fuzzing

Symbolic and concolic execution

Information flow analysis

Slicing

Path profiling

Analyzing concurrent programs

Content

Introduction and basics

Operational semantics

Data flow analysis

Call graphs

Random test generation/fuzzing

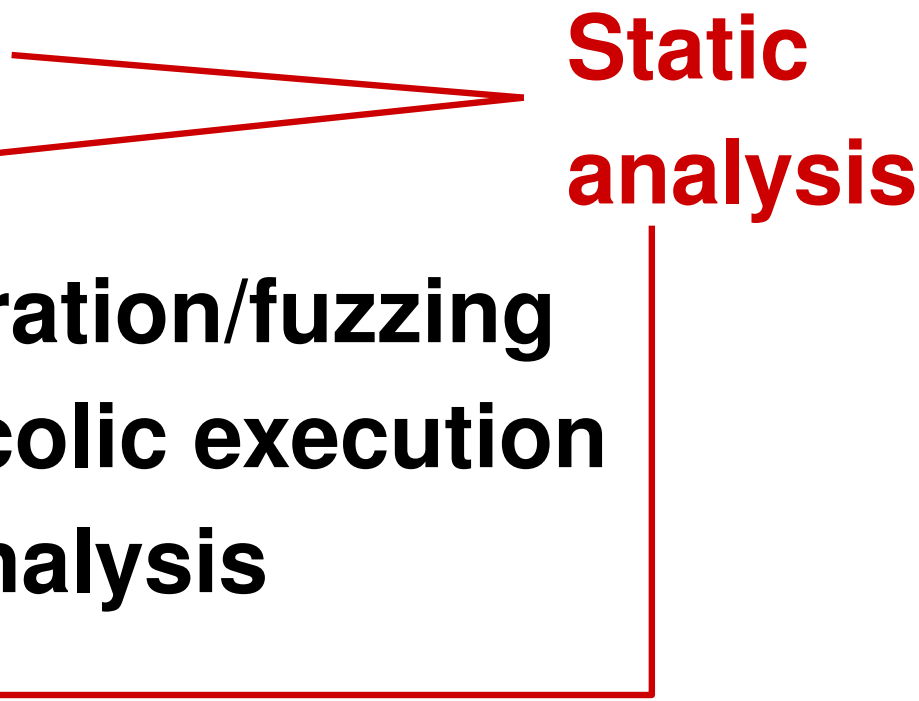
Symbolic and concolic execution

Information flow analysis

Slicing

Path profiling

Analyzing concurrent programs



Content

Introduction and basics

Operational semantics

Data flow analysis

Call graphs

Random test generation/fuzzing

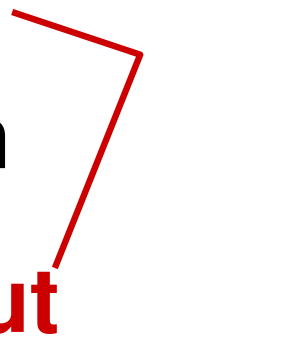
Symbolic and concolic execution

Information flow analysis

Slicing

Path profiling

Analyzing concurrent programs



Content

Introduction and basics

Operational semantics

Data flow analysis

Call graphs

Random test generation/fuzzing

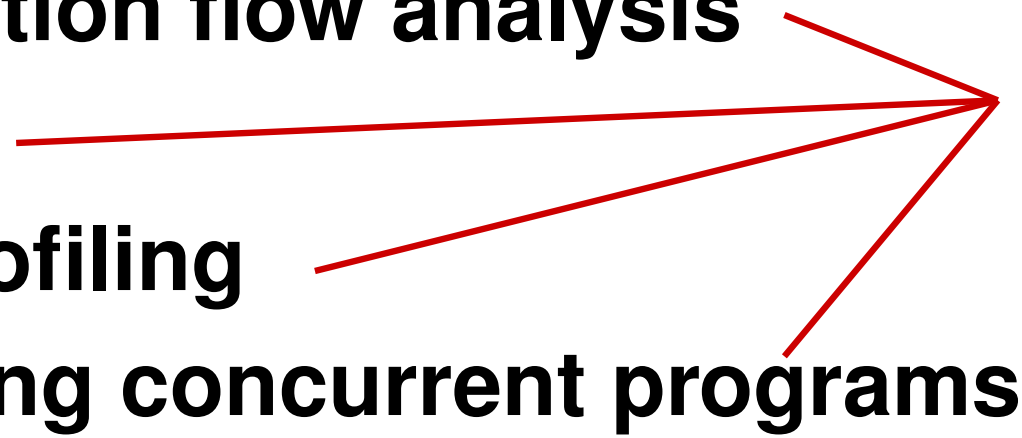
Symbolic and concolic execution

Information flow analysis

Slicing

Path profiling

Analyzing concurrent programs



Content

Introduction and basics

Operational semantics

Data flow analysis

Call graphs

Random test generation/fuzzing

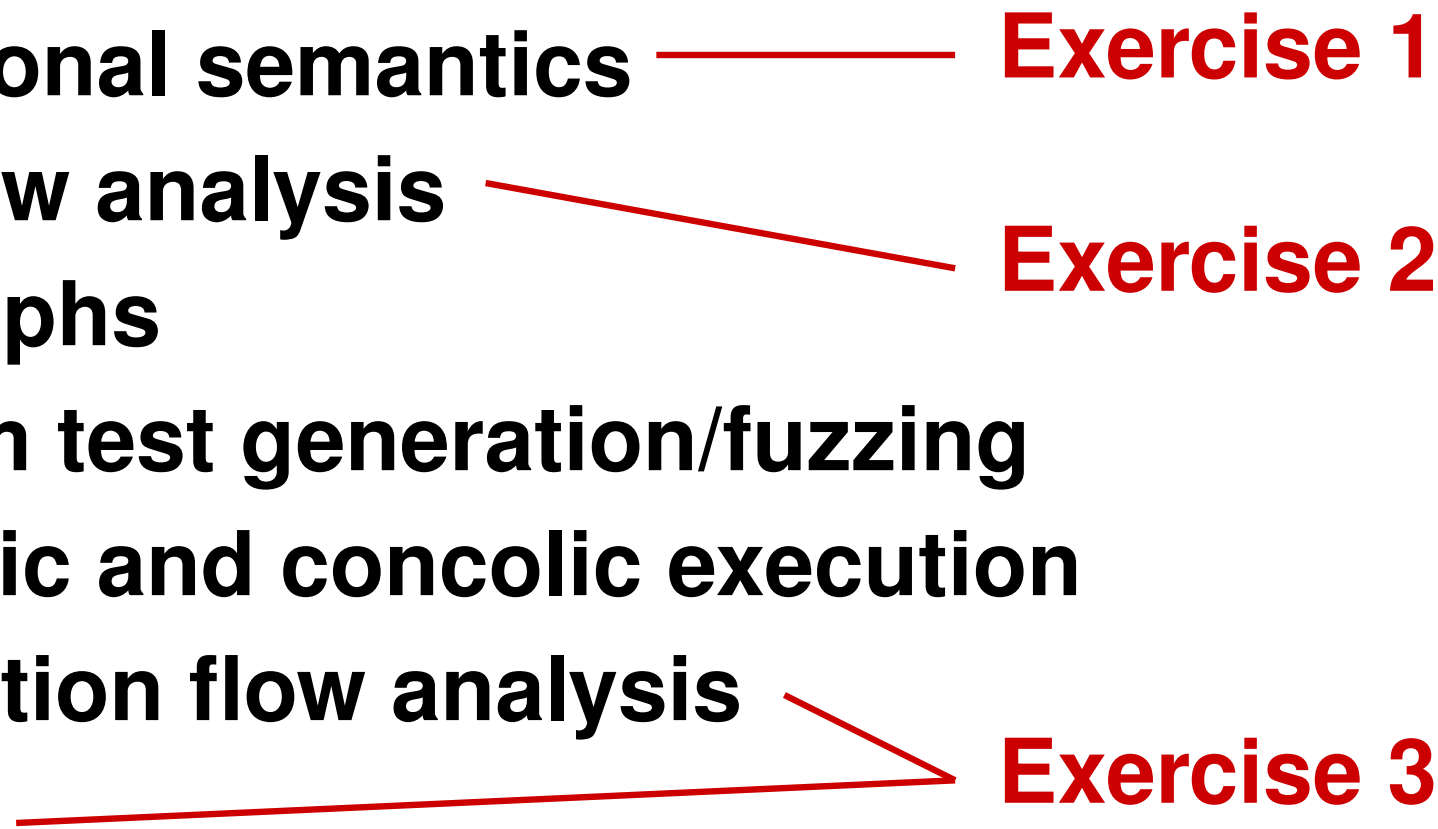
Symbolic and concolic execution

Information flow analysis

Slicing

Path profiling

Analyzing concurrent programs



Content

Introduction and basics

Operational semantics

Data flow analysis

Call graphs

Random test generation/fuzzing

Symbolic and concolic execution

Information flow analysis

Slicing

Path profiling

Analyzing concurrent programs

Course project

Learning Material

There is no script or single book that covers everything

- Slides and hand-written nodes:
  Available after lecture
- Pointers to papers, book chapters, and web resources

Programming Language

Most concepts taught in this course:

Language-independent

Most examples: JavaScript

- Very popular: client-side web applications, but also for server, mobile, and desktop applications
- Various interesting research challenges

Course project: Java and JavaScript

- Analysis written in Java
- Analysis of JavaScript code

Schedule

- **Asynchronous activities**
  - Lectures
  - Work on exercises and course project
- **Synchronous activities**
  - Discussion of exercises
  - Progress meetings
  - Project presentations
- **Strict deadlines**
  - Submission of exercises and course project

Ilias

Platform for questions, discussions, and sharing additional material

- Please register for the course
- Use it for all questions related to the course
- Messages sent to all students go via Ilias

Link to Ilias course on

software-lab.org/teaching/winter2020/pa/

A Friendly Warning

This is not going to be an easy course!

- Do the exercises
- Work regularly on the course project

A Friendly Warning

This is not going to be an easy course!

- Do the exercises
- Work regularly on the course project

... but the effort is worth it!