CS252r: Program Analysis

Now with static and dynamic !!!

Fall 2015
CS252r

• Survey of program analysis concepts
  • Program analysis: automatic reasoning about programs
  • Foundations, formalism, techniques, applications, implementations

• Aims:
  • Understand relative strengths and weaknesses of different analyses
  • Understand some current challenges in program analysis

• Research project:
  • Advance the state-of-the-art in program analysis; OR
  • Implement/apply state-of-the-art program analysis techniques

• Prereq: CS 152, CS 153, or equivalent
Class meetings

- Fairly informal
- Meet twice weekly
- Combination of lectures, papers, and workshops
  - First 8 classes will be background lectures
    - May include additional/relevant/recommended reading
  - Research papers
    - Mostly recent research, some classic papers.
  - Research workshops
    - 4-5 classes where we collaboratively workshop a research idea
- Expect to present/lead discussion once (maybe twice) during semester
Assessment

• Auditors welcome
• Class participation
  • Presentation/discussion
• Project
  • Dig deep into one or more aspects of material covered in class
  • Many possibilities: lit survey, reproducing results, implementation of analysis, developing new analysis, performance/scalability study, applying an analysis to a domain you are interested in…
• More details and project suggestions coming later
Topics/syllabus

• See web page
Research project

• Primary course assessment
• Goal: develop a deep understanding in one or more of the areas studied in this course, and, ideally, to conduct original research.
• You may (and are even encouraged) to work in groups (up to 3 members).
• Weekly meetings with me (from Sept 14th)
• Sept 29: Project proposals due
• Dec 3: In-class presentations
• Dec 9: Projects due
**Static and Dynamic Analyses**

**What?**
- Static analyses
  - Analyze programs without actually running them
- Dynamic analyses
  - Analyze a program execution

**Why?**
- Static analyses
  - Reject program
    - e.g., insecure, may have bugs, ...
  - Rewrite program
    - To be secure, bug-free, ...
  - Find problems
  - Understand program
    - Performance, requirements on external environment
  - ???
- Dynamic analyses
  - Reject execution
    - e.g., insecure, stop bad behavior before catastrophe
  - Modify execution
    - Prevent insecurities, avoid bugs, ...
  - Find problems
  - Understand program
    - Possible behaviors of this (and maybe other) executions
  - ???

*For this lecture: assume that dynamic analysis is about analyzing execution, not about monitoring/modifying a deployed system.*
Static and Dynamic Analyses

**Static analyses**
- Approximate all possible executions of program
  - Type checking
  - Abstract interpretation
  - Data-flow analysis
  - Model checking
  - Hoare logic
  - ...

- Approximate some executions of program
  - Symbolic execution
  - Unsound data-flow analysis
  - Model checking
  - ...

**Dynamic analyses**
- Analyze single execution
  - Execution monitoring
    - Program rewriting
    - Interpreter/virtual machine
    - Interposition
  - Modify execution
    - Adversarial scheduler
    - Fault injection
    - ???

- Analyze single execution as representative of set of executions
  - Symbolic (concolic?) execution
  - Thread interleavings
  - ...

How?
Some themes...

• What is the interaction between static and dynamic analysis?
  • (or maybe I mean static analysis and dynamic intervention)

• Gaps between analysis and real programs
  • Precision
  • Soundness
  • Completeness

• ... and how to bridge them