Please note this syllabus will be updated at the start of the fall semester.

CSCI 610: Advanced Program Analysis

Course Description

This course covers advanced techniques for analyzing and verifying software systems; topics include program analysis, automated verification, and software testing.

Class Location and Time

Location: WPH B26
Time: MW 3:30-5:20pm

Instructor

Name: William G. J. Halfond
Office: SAL 330
Office Hours: TBD (will be set during the first week of class)
Email: halfond@usc.edu
Website, Resources, and Q&A: https://piazza.com/usc/fall2023/csci610
Github Organization: https://github.com/CSCI610

Motivation

Software is at the core of many systems that play an important role in our daily activities and provide critical services to end-users. However, significant errors and security vulnerabilities continue to appear in web applications and have, in fact, increased over the past decade. These come at a significant cost. The National Institute of Standards and Technology estimates that software errors cost the US economy over 60 billion dollars annually. Of that, over a third was found to have been preventable if existing techniques for software testing and analysis had been applied. However, ensuring that these systems function correctly and with a high degree of reliability has become more challenging as the systems have grown in complexity. Program analysis and software testing are two related techniques that have become widely used to ensure the quality of software. These techniques allow us to validate, verify, and evaluate the quality of software produced during the software engineering process. Together they offer the potential for software verification techniques that can be fully automated and scale to handle the most complex of software systems.

Objectives

This course introduces students to the techniques of program analysis and software testing. By taking this course, students will gain an understanding of the concepts and theories that underlie these techniques. Students will also learn to design, implement, and leverage program analysis techniques to solve new verification problems.

Please note this syllabus will be updated at the start of the fall semester.

TOPICS TO BE COVERED IN THIS COURSE:

1. Control-flow analysis
2. Control-dependence and dominance
3. Data-flow analysis
4. Program dependence graphs
5. Program slicing
6. Alias analysis
7. Symbolic execution
8. Taint analysis
9. Code coverage criteria
10. Code instrumentation
11. Fault localization
12. Regression testing and analysis

READINGS AND REFERENCE MATERIALS

The following textbooks and papers are classic references for those working in the field of program analysis, software testing, and verification.

1. Compilers: Principles, Techniques and Tools. By Alfred Aho, Jeffrey Ullman, Ravi Sethi, and Monica S. Lam. Second edition, 2006. [CPTL]
2. Software Testing and Analysis. Process, Principles, and Techniques. By Mauro Pezze and Michal Young. Published by John Wiley and Sons, 2008. [STA]
3. Advanced Compiler Design and Implementation. By Steven Muchnick. First edition, 1997.
4. "Visualization of Test Information to Assist Fault Localization." By James A. Jones, Mary Jean Harrold, John Stasko. Proceedings of the 24th International Conference on Software Engineering, Orlando, Florida, USA, May 2002, pp. 467–477. [FL]
5. "A safe, efficient regression test selection technique." By Greg Rothermel and Mary Jean Harrold. In Transactions on Software Engineering and Methodology, April 1997. [REG]
6. "The Program Dependence Graph and Its Use in Optimization ." By Jeanne Ferrante, Karl J. Ottenstein, and Joe. D. Warren. ACM Transactions on Programming Languages and System, July 1987. [PDG]

RECOMMENDED PREPARATION

1. Java programming skills
2. Undergraduate or higher discrete math and algorithms class
3. Undergraduate or higher software engineering class

Class Deliverables and Grading

The grades for the students will be based on completion of several homeworks, a midterm exam, a paper presentation, and a final exam. The breakdown for each of these categories is listed below. A more detailed explanation of the grading for each category is also provided.

- Homeworks - 60% (6 @ 10%)
- Research Paper Presentation – 10%
- Midterm Exam – 12.5%
- Final Exam – 12.5%
- Class Participation – 5%

Homeworks

Students will implement a series of homework assignments. These will focus on designing and implementing lightweight versions of the program analysis techniques presented in class. For each homework students will be provided with a description of the assignment, sample inputs, and sample correct outputs. To grade each homework submission, the instructor will run the implemented analysis on a set of grader programs and evaluate the output for accuracy and analysis speed. Students will have two submission attempts. The first attempt will count for 2/3 of the homework grade, and the second (optional) attempt will count for 1/3 of the homework grade. After the first submission, students will be offered grading feedback that will help them identify and fix mistakes in their submission.

Homework Topics

- HW 1: Control-flow analysis
- HW 2: Reaching Definitions (Data-flow analysis)
- HW 3: Program Slicer
- HW 4: Instrumentation
- HW 5: Symbolic evaluation
- HW 6: Inter-procedural analysis

Exams

There will be one midterm that will cover all material in the class up to the date of the exam. There will be a cumulative final exam for the class. This will cover all concepts and terminology presented in the class lectures.

Class Participation

To achieve full credit for class participation, the student must attend all lectures and participate in all in-class activities.

Research Paper Presentation

Each student will prepare and deliver a presentation of a recently published research paper that deals with the topic of software testing and analysis.

Late Work Policy

Late work will not be accepted without prior approval of the instructor.

Please note this syllabus will be updated at the start of the fall semester.

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Section 11, *Behavior Violating University Standards* https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the *Office of Equity and Diversity* http://equity.usc.edu or to the *Department of Public Safety* http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. *The Center for Women and Men* http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the *American Language Institute* http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. *The Office of Disability Services and Programs* http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, *USC Emergency Information* http://emergency.usc.edu will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.