ECE750-T05: Static Analysis for Software Engineering
Fall 2008

Patrick Lam

September 12, 2008

Brief Overview

Code review is a key technique for ensuring software quality. However, human reviewers have limitations: limited time, limited attention spans, and a limited understanding of the implications of a software modification (due to interactions between parts of a potentially vast codebase).

Computers can successfully carry out many tasks at which humans fail. A goal of my research is to find classes of properties which are amenable to automatic verification, using static analysis techniques (which trace their roots to optimizing compilers). Many challenges exist. The most notorious problems include the undecidability of the halting problem (which we overcome using approximations) and the need to deal with unknown input values. More recently, large codebases and plugin-based software architectures pose additional challenges to static analysis.

Nevertheless, a number of static analysis techniques for software engineering have recently been proposed and even deployed in commercial development environments. Microsoft's Static Driver Verifier is the most prominent example of a static analysis technique that has escaped the laboratory.

In this seminar course, we will first briefly explore the strengths and limitations of static analysis techniques. These techniques traditionally come from the compiler research community. The bulk of this course, however, will consist of a discussion of current research papers in the field of software verification using static analysis techniques.

General Information

Course Web Page: `http://patricklam.ca/sase`
Lectures: Fridays, 8:30-11:20, EIT 3151

Instructor:

Prof. Patrick Lam
Office: DC2534
Office Hours: Thursday 2:00PM-3:00PM, or by appointment
Email: `p.lam@ece.uwaterloo.ca`

Course Description

Objectives

- Understand the strengths and limitations of static analysis techniques.
- Gain a familiarity with the research literature on static analysis for software verification.

- Carry out a moderately-sized research project which implements a static analysis and evaluates its efficacy at solving a software engineering problem.

Topics (a guideline). Overview of techniques used in static analysis, including dataflow analysis, pointer analysis, model checking, and theorem proving. Abstract interpretation. Type systems. Lightweight specification languages. Applications to software engineering.

Reference Material

Since this course is a seminar course based on published articles, your primary source of information will be the articles that we will discuss during the class. A list of these articles will appear on the course webpage.

In the first lecture or two, I will give a crash course on compiler representations and dataflow analysis.

Evaluation

You will be expected to present two papers to the class and to complete a course project, which includes a short presentation and may be done in teams. Due to departmental regulations, there will also be a final examination.

Here is the breakdown of marks:

20%	Paper presentations
10%	Questions during paper presentations
20%	Final exam
40%	Project write-up
10%	Project presentation
-5%	Missed proposal or midterm report deadlines.

Presentation. You will, collectively, present all of the papers that we are discussing in this class. Please let me know which papers you have chosen by the end of the second week of classes. The presentations will be an hour long. Be prepared to answer questions on the papers. Also, I will expect one student to prepare a set of questions or clarifications for each paper, to be discussed after the paper presentation.

Course project. The course project gives you an opportunity to work on a particular application of static analysis techniques to software engineering issues. I expect that most projects will include an implementation component. To help you stay on track, I will expect a project proposal by the third week of class and a mid-term project update (1 page) by the end of October. (Start early!) Please come see me to talk about possible projects!

Note that I will not assign a grade to the proposal or midterm report. However, if you do not hand in a proposal on time, I will deduct 5% from your course mark.

Final examination. The open-notes final examination will ask about your ability to summarize the key points of some of the papers that we'll discuss.

Grading. I will assign letter grades for the various components of the course and convert them to numbers as follows.

A	95%
B	80%
C	60%

I will also use plusses and minusses as appropriate.