CS 59200: Advanced Topics in Types and Programming Languages

Adv Topics in Types and PL

Instructor: Ben Delaware
TTH 12:00pm-1:15pm
GRIS 133

Overview:
Over the last several decades, the programming languages and formal methods communities have made significant progress toward ensuring that a program is correct with respect to some specification of its intended behavior. These communities have developed a number of techniques to reason about programs, to the point that these techniques have been adopted to verify a wide variety of real-world systems, from device drivers at Microsoft, to operating systems and hypervisors running on autonomous vehicles. Unfortunately, despite their remarkable potential for increasing the software quality, many of these techniques have not seen wider adoption. The aim of this seminar is to study the theory underpinning state-of-the art tools for reasoning about programs, understand their limitations, analyze the barriers to applying these techniques, and to investigate potential remedies.

Structure:
This seminar course will be a complementary mixture of lectures, paper discussions, and tool demonstrations. Lectures will provide an introduction to important concepts, while the paper discussions will provide an in-depth look at the state-of-the-art application of those concepts. Each paper will have a designated facilitator responsible for leading the discussion. To ensure a lively discussion, students will be responsible reading assigned papers at sufficient level to summarize the the research problem, the proposed solution, the relationship to existing work, and the evaluation of any claimed contributions. Tool demonstrations will similarly have a designated facilitator that is responsible for developing a demonstration of a tool that highlights both its strengths and weakness.

Evaluation:
Students will develop (in concert with the instructor) and work on an open-ended project over course of the semester. Ideally this will be an application of program reasoning to their personal interests. Students are welcome to pair up for more ambitious projects. The project will include a short final report and presentation during the last week of class.

Final grades will be assigned following

Class Project	40%
Facilitating Discussion	30%
Participation	30%

Prerequisites:
As an advanced topic course, we assume that students already have a basic understanding of programming language theory. The students must have taken either CS456 (undergraduate PL), CS565 (graduate PL), or an equivalent courses. Project experience and good programming skills are a must, as the course project is an important part of this class.

Topics*:

- Dependent Types
  - Theory: Calculus of Constructions

  - Theory: Pure Type Systems
  - Practice: Refinement types (Liquid Haskell)
  - Practice: Tactic-based metaprogramming (Ltac / Reflective proof automation in Lean)

- Modal Type systems
  - Theory: S4, Linear Lambda Calculus, Session Types
  - Practice: Ownership system (Rust)
  - Practice: Linear Haskell
  - Practice: Stack allocated values in OCaml

- Program Logics:
  - Theory: Separation Logic
  - Practice: Automatic Invariant Inference (Houdini / Data-Driven Approaches)
  - Practice: Interactive Invariant Inference (Ivy)

- Security Type Systems
  - Theory: Noninterference
  - Theory: Hyperproperties
  - Practice: Secure Distributed Programs (Viaduct)
  - Practice: Oblivious Algebraic Datatypes

- Deductive Program Synthesis
  - Theory: Refinement Calculi
  - Practice: Interactive Synthesis using Dependent Types (Fiat)

* This list of topics is tentative, and will be adjusted based on the interests of seminar participants