Version of 17 December 2016

Course Information

A project-centered class covering various topics in programming language implementation.

Course structure for Spring 2017

Graduate students: register for CRN 42531. Undergraduate students: the course is also listed as CS 491; register for CRN 42530.

Course topics this semester are intermediate representations, code transformations, and machine code generation in the compilation of functional programming languages. In particular, we will study representations of control flow. We will examine how higher-order functions can be efficiently implemented. In addition to their use in compilation, the transformations we will study can be viewed as programming techniques, and we will look at examples of their use.

Another area of interest is the definition, implementation, and validation of programming languages via abstract machines.

Format: Lectures; research paper presentations.

Prerequisites

CS357, CS558, CS556 or equivalent experience.

Students should be familiar with several high-level programming languages, so that they can appreciate the purpose and the tasks of a compiler.

Students should be experienced programmers able to develop large programming projects quickly, keep up with deadlines, and work in teams.

Assumed background:

- functional programming in general, and Scheme, ML, or Haskell in particular
- understanding recursive data types, recursive functions to compute over them, and structural induction to prove things about them
- familiarity with computer organization and architecture, operating systems, machine language and assembly language programming, and the C programming language

Lectures

Tuesdays and Thursdays 12:30-1:45 in Mechanical Engineering 220

Instructor

Darko Stefanovic, office ECE 137B, phone 2776561, email darko — office hours Mondays 11:00-12:00

Teaching assistant

None

Grading

The grade will be determined as follows:

- Programming projects 70%
- Oral presentations 30%

You are expected to attend class regularly, read the assigned reading before class, give occasional oral presentations on research papers or team projects, and participate in class discussion.

Programming assignment hand-in policy

Programming assignments are to be submitted on-line. Detailed instructions will be provided with each assignment.

Sample topics (selection varies from year to year)

- Introduction to compilation
- The structure of compilers
- Front-end design
- Back-end design
- Common issues in the compilation of functional languages
- Abstract machines for lambda calculi
- Representing and analyzing control flow
- Higher-order functions and their implementation
- Parametric polymorphism and its implementation
- Validation of programs and compilers

UNM statement of compliance with ADA

Qualified students with disabilities needing appropriate academic adjustments should contact the instructor as soon as possible to ensure their needs are met in a timely manner. Handouts are available in alternative accessible formats upon request.