CS 578 Programming Language Semantics – Spring 2010

Instructor: Andrew Tolmach
120-23 FAB (503) 725-5492
email: apt@cs.pdx.edu
Office Hours: MTh 1-2pm & by appt.
Course web page: http://www.cs.pdx.edu/~apt/cs578

Description

This course provides an introduction to the mathematics of program meaning (semantics) using the framework of type systems and typed languages. Topics include big-step and small-step operational semantics for simple imperative and functional languages; Hoare logic; type safety; the lambda-calculus; extensions to records and subtyping; and polymorphism. Simultaneously, the course will provide a hands-on introduction to interactive theorem proving using the Coq proof assistant.

Prerequisites

The theoretical material in the course is self-contained, so there are no specific prerequisites, but a reasonable level of mathematical maturity is desirable. For example, you should be comfortable with proofs by induction.

Since Coq has the flavor of a functional language, previous exposure to a functional language such as Scheme (to the level of CS558), Haskell (as in CS557), or ML is quite desirable.

Readings

We will use the new on-line textbook "Software Foundations" by Benjamin C. Pierce, et al. This book is still under development, and we will be using the latest revision as it becomes available; this version will be posted on the course web page. A recent "frozen" version is available at http://www.cis.upenn.edu/ bcpierce/sf/.

The book "Types and Programming Languages," by Benjamin C. Pierce, MIT Press, 2002 is also strongly recommended as an auxiliary reference.

Requirements

There will be weekly homework assignments, a midterm, and a final exam (possibly take-home). The homework assignments will mostly involve proofs, both fully formal ones using Coq and more informal ones using "pencil and paper."

The course grade will be distributed as follows:

Homework	20%
Midterm	35%
Final	45%

Although it will not be formally assessed, class participation is strongly encouraged, and may affect borderline grades.

Computing Facilities

You'll need to be able run Coq version 8.2. This should soon be available on the CS lab machines, but it is strongly recommended that you install it on your own personal machine; see the course web page for pointers.

Individual Work

It is permitted (even encouraged) for you to work together on homework assignments. However, all homework submissions must be written up (or typed in) individually; an important part of the course is learning how to write down theoretical arguments, even after they are clear in your own mind.

Exams must be completed individually without any collaboration. Cheating on an exam will result in an automatic zero grade and the initiation of disciplinary action at the University level.

Disabilities

If you are a student with a disability in need of academic accommodations, you should register with Disability Services for Students and notify the instructor immediately to arrange for support services.

Tentative Schedule

This schedule is highly subject to change. You should always attempt to do the reading *before* the relevant class meeting.

dates	*SF chapters*	*Topics*
Mar 30 & Apr 1	Basics	Functional programming and reasoning in Coq
Apr 6 & 8	Lists, Poly	Basic data structures
Apr 13 & 15	Ind, Logic	Basic logic
Apr 20 & 22	Imp	Simple imperative programs
Apr 27 & 29	Hoare, MoreHoare	Hoare logic; Axiomatic semantics
May 4 & 6	Smallstep	Small-step operational semantics
May 11 & 13	Stlc	Type systems; simply-type lambda calculus
May 18 & 20	MoreStlc	Extensions to the lambda calculus
May 25 & 27	Subtyping	Subtyping; object-oriented languages
Jun 1 & 3		Polymorphic type systems
Jun 8		**Final Exam slot (10:15-12:05)**