

CS 510 Automated Deduction – Fall 2008

Instructor: Andrew Tolmach

120-23 FAB (503) 725-5492

email: apt@cs.pdx.edu

Office Hours: MW 1-2pm & by appt.

Course web page: <http://www.cs.pdx.edu/~apt/cs510coq>

Description

The course will be a hands-on introduction to interactive theorem proving using the Coq proof assistant. A proof assistant doesn't prove things itself; rather, it helps a human construct a proof more easily and reliably. We will begin by learning how to verify properties of simple functional programs, and then progress to proofs about other formal systems, such as programming language semantics. We'll also examine verification of imperative programs in some depth. If time permits, we will take a brief look at fully automated theorem provers, which work without direct human guidance.

Prerequisites

There are no formal course prerequisites, but prior exposure to basic discrete mathematics and formal proof is essential. Undergraduates should have completed CS250, 251 and 311. Prior exposure to functional programming (e.g. CS457/557) will also be very useful.

Readings

There is no published textbook, but for the first part of the course, we will be using a preliminary version of a new textbook/Coq script by Benjamin Pierce. This will be made available on the course web page as we proceed.

Requirements

There will be weekly homework exercises and a take-home midterm and final.

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

The course will require use of the Coq proof assistant. While the precise version we use doesn't matter too much, it will be easiest and safest if we all use the *same* version, namely version 8.1pl3. This is available on the CS linux lab by default and as a package (`coq`) on the CS Solaris systems. You will probably also want to install it yourself on your own machine, particularly if you have a laptop; see the course web page for pointers.

To interact with Coq, an IDE is a must; there are two choices of roughly equal quality. If you are an emacs user, try the ProofGeneral environment (a general-purpose theorem proving IDE that has a Coq binding). Otherwise, you can use CoqIDE, which runs as a stand-alone tool under both unix and Windows. See the course web page for pointers.

Individual Work

It is permitted (even encouraged) for you to work together on homework assignments. However, all homeworks must be prepared and submitted individually; the whole goal of the course is for you to become comfortable using the prover yourself.

Exams must be completed individually without any collaboration. Plagiarism or collaborating 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.

dates *topics*

Sep 30 & Oct 2	Introduction to Coq; Simple functional programs and proofs
Oct 7 & 9	Essential Coq tactics
Oct 14 & 16	Induction in depth; Propositions
Oct 21 & 23	Logical connectives; Relations
Oct 28 & 30	Proofs and computations; Coq automation
Nov. 4 & 6	(take-home midterm due) Case study: Language metatheory
Nov. 11	No Class Veterans Day
Nov. 13	Dependent types in programming and proof
Nov. 18 & 20	Case study: Imperative program verification
Nov. 25	Coq vs. other formal methods
Nov. 27	No Class Thanksgiving Day
Dec. 2 & 4	Case study using automated prover: TBD
Dec. 11	Take-home final exam due