CS 410/510 Practical Specification and Verification - Winter 2016 - Syllabus

Instructor:
Andrew Tolmach
120-23 FAB
(503) 725-5492
email: tolmach@pdx.edu
Office Hours: W 2-3pm or by appt.

Course web page: web.cecs.pdx.edu/~apt/cs510spec

Description

Software is essential to modern industrial society, embedded in everything from phones to missiles to cars to pace makers. Bugs in software often costs money and sometimes costs lives. Everyone knows that writing correct software is hard, particularly as systems grow larger and more distributed. The goal of this course is to explore how using formally-specified models and automated analysis tools can help make it easier to get software right.

Structure

The course will be structured around practical exercises in applying logic-based tools. We will examine tools that operate at several different levels of specification, such as high-level system structure (e.g. Alloy or Z), explicitly distributed algorithms (e.g. SPIN or TLA), and detailed code behavior (e.g. Dafny or Frama-C).

Since all of these tools are based on mathematical logic, we will begin the course with a rapid review of propositional and predicate logic. Later on, we will also learn the basics of temporal logic.

Classes will be devoted to interactive discussion, problem solving, and tool demonstration. There will be few, if any, conventional lectures. Therefore, it is crucial to do the assigned reading *before* coming to class.

In the final weeks of the course, each student will investigate a specification-based tool of their choice (subject to instructor approval), and give a presentation about it to the class.

Assessment

Course grade will be based on the following elements:

20%	Short weekly exercises
15%	Take-home midterm (roughly week 5)
30%	Common tool projects (3-4 over the term)
35%	Individual tool project

There will be no final exam, but we may use the final exam slot (Tuesday March 15, 5:30-7:20pm) for presentations.

Prerequisites

Students should have a solid understanding of basic discrete math (sets, relations, functions) and logic (boolean connectives, quantifiers). But we will review any necessary logic background at the beginning of the course. Good programming skills in a high-language such as C or Java will also be assumed.

Reading

The required textbook is Huth and Ryan, *Logic in Computer Science, 2nd ed.*, Cambridge Univ. Press, 2004.

There will be additional assigned reading on the various tools we examine, and perhaps also on more theoretical topics.

Schedule

The class meets each Tuesday and Thursday from 4:40-6:30pm. The general plan is to spend roughly one third of the course on logic background, one third on a model checking tool, and one third on a program proof tool. But the detailed schedule will develop dynamically; consult the course web page.

Computing Facilities

You will find it convenient to install the various tools we use on your own computer. If necessary, tools will also be installed on CS department machines.

Individual Work

Unless otherwise specified, all submitted work must represent your own, individual effort. Plagiarism from other students or from the web is considered cheating, which 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.