COMP 4161: Advanced Topics in Software Verification

NICTA Advanced Course, Session 2/2014

- Home
- Course Outline
- Lectures
- Assignments
- Exam
- Reading

Course Outline

Staff

Lecturer: Toby Murray
Email: toby.murray at nicta.com.au
Phone: +61 2 8306 0567
Office: Level 6, Building L5
Consultations: by appointment

Lecturer: June Andronick
Email: june.andronick at nicta.com.au
Phone: +61 2 8306 0581
Office: Level 4, Building L5
Consultations: by appointment

Course Goal

To educate students in advanced topics in software verification. Topics include higher order logic, natural deduction, lambda calculus, term rewriting, data types and recursive functions, induction principles, calculational reasoning, mathematical proofs, decision procedures for a variety of logical domains, and proofs about programs.

Parallel Teaching

None

Course Prerequisites

Experience with (first-order) logic or functional programming is required. The course is intended for 4th year or post graduate students. Second and third year students can participate with permission of the lecturer.

Course Exclusions

None

Constituents

- Lecture: the lecture will cover the following main topics: higher order logic, natural deduction, lambda calculus, term rewriting, data types and recursive functions, induction principles, mathematical proofs, and proofs about programs.
- Tutorials: there are no tutorials for this lecture
- Assignments: there will be three assignments. Penalty for late submission of assignments will be 4% (of the worth of the assignment) subtracted from the raw mark per day of being late. In other words, earned marks will be lost. For example, assume an assignment worth 25 marks is marked as 20, but had been submitted two days late. The late penalty will be 2 marks, resulting in a mark of 18 being awarded. No assignments will be accepted later than one week after the deadline.

Preliminary Course Schedule

Week	Topic
1	Introduction and Admin, Lambda Calculus
2	Typed Lambda Calculus, Proofs in Isabelle, Natural Deduction
3	HOL
4	Term Rewriting
5	Induction
6	Recursive Datatypes and Primitive Recursion
7	General Recursion
8	Hoare Logic
9	C verification
10	Writing Automated Proof Methods
11	Isar: Structured Proofs
12	Typeclasses and Locales

Course philosophy and teaching strategies

The learning focus in this course is primarily on lectures and assignments. The first two assignments are intended to give early feedback and to test your preparedness for the final exam. While marks are assigned to the assignments, their primary purpose is to give you concrete tasks with deadlines to help you structure your learning.

Assessment

- Assignments: There will be three written assignments. Assignment 1 will be due in week 3, assignment 2 in week 8, and assignment 3 in week 11.
- Exam: The final exam will be a take-home exam with Isabelle/HOL proofs and questions on the lecture material.
- Final Mark: the class mark consists of the assignments (each 1/3). The arithmetic mean of the class mark and exam mark is used to determine the final mark. To pass the course, a minimum of 40% is necessary in each component. If both the class mark and the exam mark are greater or equal to 40%, the final mark will be `(class mark + exam mark) / 2` otherwise `minimum ((class mark + exam mark) / 2, 44)`

Text and Reference Books

See the page on reading material.

Continual course improvement

Feedback from the last evaluation of this course was positive and we intend to maintain the same style and content. One issue raised by students was the lack of textbook. We are considering writing one.

Further information

- Students enrolled in this course are expected to attend all classes.
- Plagiarism policy.
- The use of School of Computer Science and Engineering computing laboratories is subject to rules described in the Yellow Form, which you acknowledge (electronic) receipt of when you receive your computing account. The Yellow Form also outlines what to do in case illness or misadventure that affects your assessment, and supplementary examinations procedures within the School of Computer Science and Engineering.
- UNSW Occupational Health and Safety policies and responsibilities.
- Equity and Diversity.