MPhil in Advanced Computer Science
Interactive Formal Verification (L21)

Leader:	Lawrence Paulson (course lecturer)
Timing:	Lent Term
Prerequisites:	familiarity with elementary logic, functional programming and operational semantics
Structure:	12 Lectures and 4 Practical Classes

AIMS

This module introduces students to interactive theorem proving using Isabelle. It includes techniques for specifying formal models of software and hardware systems and for deriving properties of these models.

SYLLABUS

1. Introduction to interactive theorem provers and higher-order logic.
2. Theories. Declaring recursive datatypes and functions.
3. Proofs. Simplification heuristics.
4. Advanced Recursion, Induction and Simplification. Ackermann's function.
5. Predicate Logic in Isabelle.
6. Structured proofs.
7. Set-theoretic primitives, notation and reasoning methods.
8. Inductive definitions and proofs involving them.
9. Operational semantics: definitions and proofs of typical properties.
10. Structured proofs revisited: Induction.
11. Modelling Case study I: hardware verification.
12. Modelling Case study II: the Mutilated Chess Board.

OBJECTIVES

On completion of this module students should

- possess basic skills in the use of Isabelle
- be able to specify inductive definitions and perform proofs by induction
- be able to express a variety of specifications in higher-order logic
- be able to write structured proofs of nontrivial results.

PRACTICAL WORK

Four supervised practical sessions will allow students to develop skills.

COURSEWORK AND ASSESSMENT

Each student must undertake two small verification projects, delivering a practical write-up accompanied by an Isabelle theory file. These will be started during the practical sessions but will probably be completed on the student's own time. These projects will assess the extent to which each candidate has absorbed the syllabus and develop practical skills. The lecturer will set and mark the assessments. The mark will be reported as a percentage.

RECOMMENDED READING

In order of decreasing priority. The first title should suffice for most purposes. All of these manuals are visible in the documentation sidebar of an Isabelle/jEdit session, and also at `http://www.cl.cam.ac.uk/research/hvg/Isabelle/documentation.html`

- Tobias Nipkow. *Programming and Proving in Isabelle/HOL.*
- Tobias Nipkow, L. C. Paulson and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic.*
- Alexander Krauss. *Defining Recursive Functions in Isabelle/HOL.*

Last updated: 17/12/2015