MATH0109 Theorem Proving in Lean

Year:	2024–2025
Code:	MATH0109
Level:	6 (UG)
Normal student groups:	UG: Year 3 Mathematics degrees or Year 3 MEng Mathematical Computation
Value:	Half unit (= 7.5 ECTS credits)
Term:	2
Structure:	3 hours per week. Assessed coursework and in-class tests
Assessment:	60% in-class tests, 40% coursework
Normal Pre-requisites:	Familiarity with the notion of proving theorems (for example, at least one of MATH0034, MATH0051, MATH0052 or MATH0053). Good coding skills are also desirable.
Lecturers:	Dr R Hill & Dr J Talbot

Course Description and Objectives

Lean is a programming language that can be used to interactively prove mathematical theorems. More precisely, when one types a proof into Lean, the computer checks that the proof is correct. Furthermore, when one has typed part of a proof, the computer determines at any stage what we have proved so far, and what is still to be proved. Lean also has various high level commands which can be used to prove simple statements automatically.

One very appealing feature of Lean for mathematicians is the existence of a large and growing library of formalised mathematics:
`https://leanprover-community.github.io/mathlib-overview.html`

The ultimate goal of this course is to provide students with a level of proficiency in Lean that will allow them to formalise (i.e. prove in Lean) theorems of undergraduate level pure mathematics.

Recommended Texts

Mathematics in Lean, Jeremy Avigad, Kevin Buzzard, Robert Y. Lewis, and Patrick Massot.
`https://leanprover-community.github.io/mathematics_in_lean/`

Theorem Proving in Lean, Jeremy Avigad, Leonardo de Moura, and Soonho Kong
`https://leanprover.github.io/theorem_proving_in_lean/introduction.html`

Detailed Syllabus

The course will be taught entirely through workshops during which students will learn by doing.

We will focus on *tactic mode* proofs, and in particular on methods for automating proofs. We will *not* emphasise the theory of dependent types.

0. Basic introduction to type theory as required to use Lean, e.g. types, terms, `Prop`, equality (syntactic, definitional, propositional).

1. Introduction to Lean tactics.

2. Functions and Propositional Logic.

3. Sets and subtypes.

4. Inductive types.
5. Structures and classes.
6. Formalisation of selected theorems from real analysis, algebra, number theory, combinatorics and topology.