exlean — beginners teaching Lean in Exeter

Gihan Marasingha, Omar Harhara, James Arthur

University of Exeter

Lean Together 2021

How I learned to stop worrying and love Lean

- Background: analytic number theory. Encountered Haskell in 2009.
- Projects with school children since 2014 using Haskell. Huffman compression algorithm, Dijkstra's algorithm, tautology checker.
- Noticed that Haskell's type system is an excellent model for mathematical structures and helps students avoid type errors.
- Introduced more formal logic into my introductory proof course at the University of Exeter.
- Started dipping into type theory, reading Type Theory and Formal Proof (Nederpelt and Geuvers, 2014).
- Discovered Logic and Proof (J. Avigad, R. Lewis, F. van Doorn) in August 2019.
- Formalised in Lean a decision procedure for D. Hofstadtder's MIU formal system, from his book *Gödel, Escher, Bach*.

Issues with traditional math(s) education

- High school maths is a matter of applying algorithms to calculate numerical or algebraic quantities.
- Most new maths undergraduates have little experience of *reasoning*.
- Without resource-heavy tutoring (e.g. the Oxbridge model), the proof-correction feedback loop is
  - too slow and
  - rarely converges to a correct proof.
- As a result, students lack confidence in their proofs or (worse) are confident in their incorrect proofs.

A first course in pure mathematics

- Autumn Term: propositional and predicate logic, natural numbers, sets, functions, complex numbers (!), real numbers, sequences, series.
- Winter Term: primes, modular arithmetic, Euclid's algorithm, fundamental theorem of arithmetic, examples of groups, subgroups, Lagrange's theorem, group homomorphisms and isomorphisms, vector subspaces of $\mathbb{R}^n$, span, linear independence.
- Cohort: $\sim$260 first-year undergraduates. Mainly taking 'straight' maths, but a sizeable minority taking joint honours and maths with . . . programmes.

Preparing for Lean

- I rewrote my course to refer to types not sets.
- ‘Blackboard’ proofs were made more formal:
    - every logical step justified by an introduction or elimination rule,
    - discussion of shifting goal(s) and context in the course of a proof,
    - greater emphasis on parsing statements.
- Planned (online) Lean sessions together with
- ‘proof skills’ sessions run by senior undergraduates James Arthur and Omar Harhara.
- Discussions via Discord.

Example

Classical reasoning

In this section, we prove one result in three different ways: using proof by cases, proof by contrapositive, and proof by contradiction. In addition, we will use the double negation theorem.

7. Let P and Q be propositions. Prove $\neg(P \land Q) \to \neg P \lor \neg Q$ by completing the following proof sketch. You may *not* use De Morgan's law (of which this result is a constituent part)! This is an exercise in proof **by cases**.

Proof. Assume $h_1 : \neg(P \land Q)$. By implication introduction, it suffices to prove $\neg P \lor \neg Q$. Using proof by cases, it suffices to prove (a) $P \to (\neg P \lor \neg Q)$ and (b) $\neg P \to (\neg P \lor \neg Q)$.

(a) Assume $h_2 : P$. By implication introduction, it suffices to prove $\neg P \lor \neg Q$.

(b) .

$\square$

Lean logistics

- CoCalc project for each student, supported by a grant from the University of Exeter Education Incubator.
- A set of 40 Lean problem sheets, shared to students via CoCalc.
- Draft lecture notes in the style of Mathematics in Lean.
- Pros of CoCalc: no installation. You can enter and edit student projects live.
- Cons of CoCalc: slower than a local installation. Prone to crashing when busy.

Lean at High School

- 6-week project with students at the Exeter Mathematics School.
- Started with Natural Number Game
- Continued to prove ‘algebra of limits’ results for sequences of real numbers.

What we learned

- The ‘marmite effect’. Some loved Lean, others didn’t.
- Kevin Buzzard’s Natural Number Game helped to make things more manageable.
- Better to teach a topic informally first then in Lean later.
- Students have adopted the language of goals, introduction and elimination rules.
- Lean forces students to confront extensionality for functions, sets, etc.
- Gapped proofs (Lean or otherwise) help students in constructing arguments.
- Tactic-mode Lean can be *too* helpful. One *can* write a proof without understanding it.
- Conversely, Lean facilitates chunking of proofs via `sorry`.