Course Proposal:

Proof Writing and Proof Checking with a Computer
by Sergey Cherkis

How can you verify that a proof you wrote is actually correct? What is a proof to begin with? In this course you will experience first hand the meaning of the Curry–Howard isomorphism between proofs and computer programs (theorems and declarations of types, proof checking and compilation). You will be writing traditional proofs together with their corresponding formal proofs (in Lean 4 computer language) that are checked by a computer. By the end of the course, writing formal proofs will be nearly as natural for you. In the last week you will also be able to compare directly your newly acquired proof-writing abilities to those of the current state AI.

Prerequisites:

No prerequisites for graduate students. Math323 or equivalent for undergraduate students.

Texts:

- Jeremy Avigad and Patrick Massot, "Mathematics in Lean," Github https://github.com/leanprover-community/mathematics_in_lean
- Jeremy Avigad, Leonardo de Moura, and Soonho Kong, "Theorem Proving in Lean," Github https://lean-lang.org/theorem_proving_in_lean

Approximate Schedule

- Computers and Theorem Proving.
- Simple Type theory, function abstraction.
- Propositional Logic. Propositions as Types.
- Classical Logic.
- Meaning of Quantifiers.
- Algebraic Structures.
- Using logic in writing proofs in Lean.
- Sets and Functions (in type theory).
- Number Theory.
- Working with Structures (and Inductive Types).
- Hierarchies.
- Topology.

Learning Outcomes

By the end of this course the students are expected to be able to

- write formal proofs in Lean 4.
- have a grasp of functional programming and the Curry–Howard correspondence.
- have facility in exploring existing Matlib libraries of formal proofs and definitions to develop formal proofs in any mathematical domain.