Formal Proof and Verification (CSCI 1951X)

Robert Y. Lewis

1 Basic Info

- Fall 2021
- MW 3:00-4:20pm
- Location: TBD
- Instructor: Dr. Robert Y. Lewis (robert_lewis@brown.edu)

2 Brief Description

Proof assistants are tools that are used to check the correctness of programs. Unlike tools like model checkers and SAT solvers, proof assistants are highly interactive. Machine-checked formal proofs lead to trustworthy programs and fully specified reliable mathematics.

This course introduces students to the theory and use of proof assistants, using the system Lean. We will use Lean to verify properties of functional programs and theorems from pure mathematics. We will learn the theory of deductive reasoning and the logic that these tools are based on.

3 Extended Description

When we learn to program, we often think in terms of implementing instructions: a program is a list of steps the computer should take, in order to do what we want it to do. But there are important ideas missing from this picture. How do we *specify* precisely what we want the computer to do? And how do we *verify* that our instructions match this specification?

Languages have been developed that offer a unified framework for specification, implementation, and verification. This course is about the theory and use of these languages, focusing on one called Lean, a new language developed at Microsoft Research.

Lean is known as a "proof assistant." The kind of verification seen in most proof assistants is not probabilistic, as in writing large test suites. It is not refutational, as commonly seen with model checkers and similar tools. Nor is it always automated. Part of the programmer's task is to build a *derivation*, or proof, that formally guarantees their program meets their specification.

Because of this focus on proof, a system like Lean blurs the line between programming and mathematics. One can write down and prove mathematical theorems in the same way one writes down and verifies program specifications.

This course will cover a variety of topics related to formal proof and verification. We will look at the theory of deductive reasoning, and how it is realized in the type theory of Lean; paradigms from functional programming that make verification easier; automatic generation of proofs; and applications of these tools in pure mathematics. But most of all, we will learn to *use* Lean to write trustworthy, verified functional programs. This course puts the theory of deductive reasoning into practice.

This course is based off of a course called Logical Verification, taught at the Vrije Universiteit Amsterdam: `https://lean-forward.github.io/logical-verification/2021/index.html` This is the first time it is being taught at Brown. As a student in this course, you'll be a bit of a guinea pig, but you'll also have the opportunity to help design things for future generations of students!

4 Course Objectives

The main objective of this course is to learn how to specify and verify properties of programs and mathematical objects in type theory. You will learn to use the Lean proof assistant to implement functional programs, state their important properties, and verify that these properties hold. At the same time, you will learn the theory underlying these reasoning systems: what deductive proofs consist of, and why they are trustworthy.

By the end of this course, you should be able to:

- Write executable functional programs in Lean.
- Specify and verify properties of these functional programs.
- Write and prove mathematical statements in Lean.
- Write *metaprograms*: tactics that automatically search for proofs.
- Explain the semantics of functional programs.
- Distinguish *proof objects* (derivations, deductions) from other kinds of verifications or tests for program correctness

5 Prereqs

We recommend students to have taken either CSCI 1710 Logic for Systems or a proof-based mathematics course. Basic familiarity with functional programming (e.g. Haskell, ML) is helpful but not required.

6 Textbooks

We will follow *The Hitchhiker's Guide to Logical Verification* by Jasmin Blanchette, et al:

`https://github.com/blanchette/logical_verification_2021/raw/main/hitchhikers_guide.pdf`

7 Assignments

- There are 11 homework assignments, due approximately weekly. Your lowest score will be dropped from your final grade.
- A final exam will cover all material from the semester.
- A final "formalization project" will give students the chance to implement and verify something of their own interest in Lean. We will make project ideas available in advance, and students are encouraged to consult with the instructor.