Abstracts

Jeremy Avigad (CMU)

Logic, Foundations, and Dependent Type Theory

Abstract:

First lecture - Logic:
This lecture will explain how logicians think about expressions and their meaning. We’ll survey the mathematical foundation of induction and recursion on syntax, as well as ways of handling bound variables. Then we’ll discuss ways of assigning *semantics*, or meaning, to collections of expressions, including ways of thinking of expressions as computational objects.

Second lecture - Foundations:
This lecture will discuss two standard foundations for mathematics, namely, set theory and simple type theory. We will explore some of the theoretical, philosophical, and practical differences between typed and untyped approaches.

Third lecture - Dependent Type Theory:
This lecture will describe the syntax and semantics of the version of dependent type theory on which the Lean proof assistant is based. We will see, in concrete terms, how the foundation looks from the user’s perspective.

Alexander Bentkamp (Heinrich-Heine-Universität Düsseldorf)

Automated Theorem Proving with Ordered Resolution and Superposition

Abstract: The superposition calculus is an efficient automatic method to find proofs of theorems stated in first-order logic. Superposition provers are used in several interactive theorem provers to help users construct formal proofs more efficiently. The development of the superposition calculus in the 1990s went hand in hand with proving its refutational completeness, a property asserting that the calculus will eventually find a proof of any provable statement. In this series of lectures, I will explain the proof of refutational completeness of ordered resolution, which is the restriction of superposition to first-order logic without equality.
Floris van Doorn (University of Paris-Saclay)

Mathematics in Lean series

Abstract: These lectures are focused on interactive theorem proving in the proof assistant Lean. I
will start by discussing the exciting developments in Lean and the future it may bring. After that,
most of the sessions will be hands-on practice sessions where you get to practice doing mathematics in
Lean yourself. We will start with the basics of Lean, and we will discuss various areas of mathematics
and how to do this in Lean. In each practice session you will get a short introduction on the topic
and then you will prove results on this topic in Lean.

Mathematics in Lean 1:
I will discuss some of the history of interactive theorem proving, the exciting developments of formal-
izing mathematics in the proof assistant Lean, and contemplate about the future that these programs
may hold.

Basics of Lean:
I will go back to the basics of using Lean, and show you how to work with Lean and how to prove
very basic exercises in Lean, focusing on proofs by computation.

Practice sessions in Lean: Basics and Computation:
In this practice session you will get to practice with Lean and do proofs by calculation and finding
and applying results from Lean’s mathematical library

Practice sessions in Lean: Sets, functions, logic:
In this practice session you will learn how to work with the logical connectives (and, or, negation,
implication and equivalence) and quantifiers. We will apply this knowledge to reason about sets and
functions, discussing topics like images, preimages and bijections.

Practice sessions in Lean: Topology
In this practice session we will do topology, and learn how to work in a general way with limits using
filters. We will discuss topological spaces and metric spaces, continuity and separation axioms.

Practice sessions in Lean: Calculus: TBA

Adam Zslot Wagner (Worcester Polytechnic Institute)

Machine Learning and Theorem Proving

Abstract: In this three-part lecture series we will explore the practical integration of machine learn-
ing techniques into the world of research mathematics. In the first lecture, we’ll demonstrate how
reinforcement learning can be employed to identify counterexamples to mathematical conjectures by
framing them as strategic games. Real-world examples will showcase the effectiveness of this approach.
The second lecture introduces saliency analysis, a method for revealing relationships among various
parameters within mathematical structures. We’ll draw inspiration from Davies et al.’s work and show
how this technique can inspire new conjectures and offer insights into seemingly simple problems.
The third lecture takes an experimental approach to transformers in mathematical research. We’ll
discuss potential applications and propose a practical method using Makemore software by Andrej
Karpathy. This lecture will be more interactive, inviting attendees to brainstorm creative ways to
leverage transformers for mathematical insights.