

Lean for the Curious Mathematician 2023

Introduction

Alexander Bentkamp

What Is Lean?

- Proof assistant (=interactive theorem prover)
 - Allows users to specify theorems and proofs in a formal language
 - Automates smaller proof steps; user only needs to provide the main steps
 - Verifies proofs down to the axioms of its logical foundation
- Programming Language
 - Definitions made in Lean are often executable and can be compiled into computer programs
 - This is useful to automate proofs

Why Proof Assistants for Mathematics?



Verification



Understanding



Creation



Collaboration

Verification



- Examples of success stories:
 - Kepler conjecture
 - Liquid tensor experiment
 - Erdős-Graham conjecture about unit fractions
 - and many more (see also LftCM colloquium)

- Peer reviewer's dream:
 - Only needs to check that theorem and definitions make sense

Understanding



- Proof authors must usually chose an appropriate level of detail
- What if the reader could choose?
- Demo by Patrick Massot and Kyle Miller:
`https://www.imo.universite-paris-saclay.fr/~patrick.massot/Examples/ContinuousFrom.html`

Creation

- A formal proof can help with questions like:
 - Does tweaking Definition 1 break anything?
 - Are there any unused assumptions in my lemma?
- Currently, formalization in the creation process is typically too time consuming



Collaboration



- A formal proof can help with questions like:
 - Did I interpret my colleague's theorem correctly?
 - Can I slightly generalize my colleagues proof?
- Lean's mathematical library (Mathlib): Allows hundreds of people to contribute to a large consistent library of mathematics

Another Good Reason to Use Proof Assistants

- It's fun!
- But: like a good puzzle, Lean can also be frustrating...

This Tutorial

- Get your hands dirty
- Tutors are here to help
- Lectures (you don't need to stick to the pace)
- Projects
- Main goal: Get people together to have fun with Lean

Additional Ressources

- **Natural Number Game:**
`https://adam.math.hhu.de/`
- **Mathematics in Lean:**
`https://leanprover-community.github.io/mathematics_in_lean/`
- **Zulip**
`https://leanprover.zulipchat.com/`
- **Lean community website:**
`https://leanprover-community.github.io/`