

Formalization of Proofs Using Lean

Rida Hamadani

LMAP, UPPA

February 11, 2026

Outline

- 1 Introduction
- 2 Tactic Mode
- 3 Calc Mode

Introduction

So far, we have only told lean that certain things objects exist, and maybe evaluated some calculations. Now it is time to state propositions about these objects, and formally proof these statements. This boils down to constructing a term of a proposition type.

Proof Modes

This are different modes of proving statements in lean, think of each mode as a DSL inside of Lean. You have been constructing terms as expressions, this is caeled "term mode". Today we will look at "tactic mode" and "calc mode".

Outline

- 1 Introduction
- 2 Tactic Mode
- 3 Calc Mode

Tactic Mode

- What are tactics?
- Basic tactics.
- Cases and induction.
- Proof automation.

What are tactics?

Tactics are commands that changes your goal state.

Tactic Mode

- What are tactics?
- Basic tactics.
- Cases and induction.
- Proof automation.

Basic Tactics

- `rfl` : reflexivity.
- `exact` : term mode inside of tactic mode.
- `decide` : evaluation.
- `intro` : assumes the premise.
- `rw` : rewriting equalities or equivalences.
- `ext` : extensionality.
- `apply` : performs backward reasoning.
- `refine` : specifies the structure of the proof.
- `obtain` : construct a term from an existential hypothesis.

Tactic Mode

- What are tactics?
- Basic tactics.
- Cases and induction.
- Proof automation.

Cases & Induction

- `by_cases` : to be followed by a statement. It then creates two subgoals, one assuming said statement, and the other assuming its negation.
- `cases` : takes an OR statement, then creates two subgoals each assuming one side of the OR.
- `induction` : performs structural induction. It creates as many subgoals as there are constructors on the structure.

Tactic Mode

- What are tactics?
- Basic tactics.
- Cases and induction.
- Proof automation.

Automation tactics

- `simp` : rw on steroids. It goes through a tree of preconfigured theorems and try to simplify the goal as much as possible.
- `simp_all` : like `simp` but also has access to the hypotheses.
- `tauto` : finishing tactic. Tries to solve the goal by breaking down all statements as much as possible using de Morgan's law.
- `aesop` : breaks down assumptions too but has the ability to call `simp_all` in between.
- `omega` : integers.
- `grind` : SMT-like tactic.

Outline

- 1 Introduction
- 2 Tactic Mode
- 3 Calc Mode

What is calc mode?

Let's say we have a transitive relation, and we want to prove that two objects satisfy this relation. Calc mode allows us to do so by constructing a sequence of elements that satisfy this relation starting with the first object and ending at the second object.