Lecture - 1

Touseef Haider

May 12, 2025

1.1. What is an Interactive Theorem Prover? And Lean's Role?

Imagine you're working on a complex math problem. You write down a step, and then pause to ask, "Is this step correct?" An **interactive theorem prover (ITP)** is like a meticulous assistant that checks your logic step-by-step.

- You state a mathematical claim (theorem or proposition).
- You prove it using logical rules and definitions.
- The ITP checks every step to ensure sound reasoning.
- It's called **interactive** because you are in constant dialogue with the software.

Lean is a modern interactive theorem prover used for:

- **Formalizing mathematics**: Creating verified mathematical definitions and proofs.
- **Software verification**: Ensuring programs meet specifications.
- **Education**: Teaching logic and precision in mathematical reasoning.

Think of it as a video game where the objective is to build a sound proof, and Lean is the game engine validating your moves!

1.2. The Basic Workflow: Editor, Lean, and the Infoview

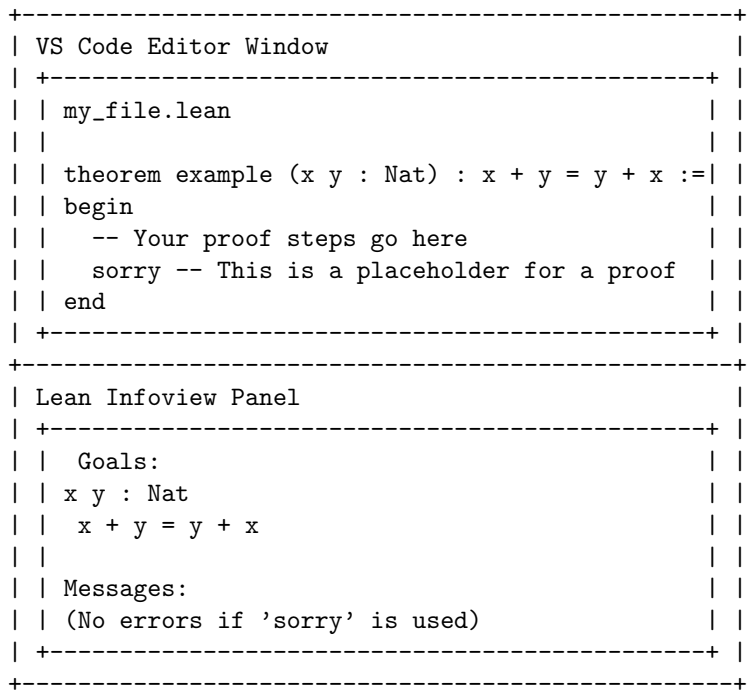
You interact with Lean using an editor—most commonly **Visual Studio Code (VS Code)**—with the Lean 4 extension:

1. **Writing Code**: Code is typed in '.lean' files:

```
def hello := "Hello, Lean!"
#check hello
```

2. **Lean Processing**: Lean checks your code as you type or save it.
3. **Infoview Panel**: Provides continuous feedback:
   - **Goal State**: What is left to prove?
   - **Hypotheses**: What assumptions are available?
   - **Messages**: Info, warnings, or errors.

Conceptual Illustration:



1.3. A Simple Example: `#check` and Lean's Response

Use `#check` to ask Lean the type of an expression:

```
#check Nat
#check 1 + 2
#check Nat → Bool -- Function type from Nat to Bool
```

Lean's Responses:

- `Nat : Type` – Nat is a type.
- `1 + 2 : Nat` – Result is a natural number.
- `Nat → Bool : Type` – Type of functions from Nat to Bool.

Example Error:

```
#check 1 + "hello"
```

Lean's Infoview might show:

```
error: type mismatch
  "hello"
has type
  String
but is expected to have type
  Nat
```

This is Lean's feedback loop in action: instant diagnostics help you write correct code and proofs!