

The LEAN Proof Assistant (Interactive Theorem Prover)

CAS CS 320: Principles of Programming Languages

Thursday, April 25, 2024

Administrivia

- Project 2 due yesterday, **Apr 24**.
- Project 3 posted and due Monday, **Apr 29**.
- Final exam review on Tuesday, **Apr 30**.
- Final exam on Wednesday, **May 8**, 3:00–5:00 pm in STO 50.

Interactive theorem provers? Theorem
provers? Proof assistants? *What are they?*

Interactive theorem provers? Theorem provers? Proof assistants? *What are they?*

There are two general categories of automated tools to help *prove mathematical facts* - i.e. to rigorously establish formal statements according to laws of formal reasoning:

Interactive theorem provers? Theorem provers? Proof assistants? *What are they?*

There are two general categories of automated tools to help *prove mathematical facts* - i.e. to rigorously establish formal statements according to laws of formal reasoning:

- ♦ *proof assistants*, a.k.a. interactive theorem provers
- ♦ *theorem provers*

Interactive theorem provers? Theorem provers? Proof assistants? *What are they?*

There are two general categories of automated tools to help *prove mathematical facts* - i.e. to rigorously establish formal statements according to laws of formal reasoning:

- ♦ *proof assistants*, a.k.a. interactive theorem provers
- ♦ *theorem provers*

Examples of proof assistants:

Isabelle, Coq (web interface), Lean (web interface).

Examples of theorem provers:

Vampire, Z3 (web interface).

Experiments for enjoyment with Z3:

1. Go to "Z3 Playground" (a web interface for running Z3 in the browser).
2. The window to write Z3 scripts is in the right column, try some of the "canned examples" from the text in the left column.

Experiments for enjoyment with Lean:

1. Go to "Lean Playground" (a web interface for running Lean in the browser).
2. The window to write Lean scripts is on the left, try some of the "canned examples" – click on ★Examples at the top bar towards the right to display several options.

Experiments for enjoyment with Lean:

Lean has a component (just like *Coq* and *Isabelle*) supporting functional programming, i.e. one can define functions in functional programming style and then prove properties about the functions thus defined as programs.

Click [here](#) for functional programming in Lean.

Facts You May Want To Prove With LEAN:

- Irrationality of square-root of 2.
- Denumerability of the rational numbers.
- Pythagorean theorem.
- Infinitude of prime numbers.
- Finite sum of inverse squares.
- Infinite sum (Basel problem) of inverse squares.
- Law of quadratic reciprocity.

(THIS PAGE INTENTIONALLY LEFT BLANK)