

LEAN: The future of Mathematics

Ignacio Muñoz Jiménez

Università di Genova

2024

Quick Introduction

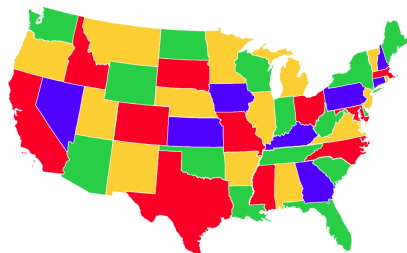
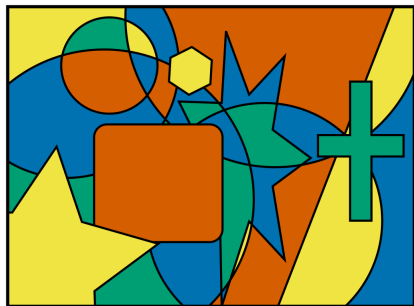
Topic: computers in mathematical research.

LEAN: proof verifier / proof assistant.

Contents

- 1 What is a proof verifier?
- 2 LEAN. How does it work?
- 3 Try it yourself!
- 4 Examples and projects.
- 5 The future of mathematics.

Four Color Theorem



Proved by Kenneth Appel and Wolfgang Haken (1997).

Checking 633 configurations using a computer.

AUTHOR



VERIFIER



AUTHOR

VERIFIER

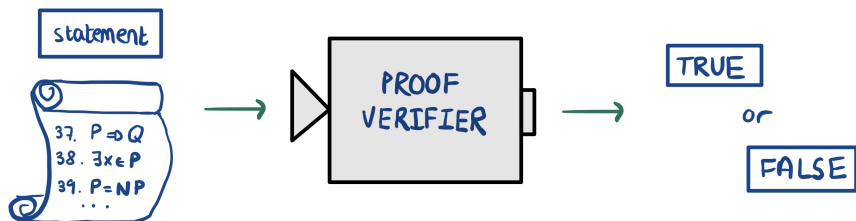


AUTHOR

VERIFIER



What is a proof verifier?



Remarks

- 1 The proof must be written in a **LANGUAGE** computers can understand.
- 2 The computers must be able to check logic implications. This include the choice of axioms.

What is NOT a proof verifier?

It does NOT prove the theorems for you!!

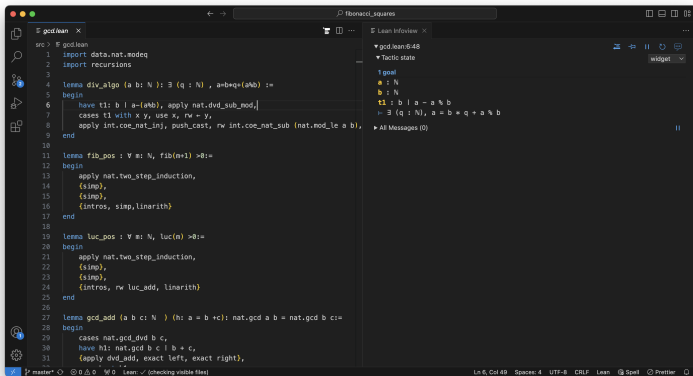
What is NOT a proof verifier?

It does NOT prove the theorems for you!!

for now

What is LEAN?

- LEAN is a proof verifier (proof assistant).
- Developed in 2013 by Leonardo de Moura at Microsoft Research (now Amazon Web Services).
- It is open sourced.



```
src > f gcd.lean
1  import data.nat.modEq
2  import recursions
3
4  lemma div_algo (a b : N) : ∃ (q : N), a+bq=(a%b) :=
5  begin
6    have t1: b | a-(a%b), apply nat.dvd_sub_mod,
7    cases t1 with x y, use x, rw - y,
8    apply int.coe_nat_inj, push_cast, rw int.coe_nat_sub (nat.mod_le a b),
9    end
10
11 lemma fib_pos : ∀ n : N, fib(n+1) > 0 :=
12 begin
13   apply nat.two_step_induction,
14   (simp),
15   (simp),
16   (intros, simp, linarith)
17 end
18
19 lemma luc_pos : ∀ n : N, luc(n) > 0 :=
20 begin
21   apply nat.two_step_induction,
22   (simp),
23   (simp),
24   (intros, rw luc_add, linarith)
25 end
26
27 lemma gcd_add (a b c : N) (h: a = b + c) : nat.gcd a b = nat.gcd b c :=
28 begin
29   cases nat.gcd_dvd b c,
30   have h1: nat.gcd b c | b + c,
31   (apply dvd_add, exact left, exact right),
32   --
33 end
```

Logic exercise

How does it work?

Logic exercise

Let P, Q be propositions.

Lemma 1. (*Logic Exercise*) $P \rightarrow (Q \rightarrow P)$.

Proof. Assume P is True (Hypothesis P). Then we have to prove $Q \rightarrow P$.

Assume P is True (Hypothesis Q). Then we have to prove P .

We know by Hypothesis P that P is True. This finishes the proof. \square

```
variables (P Q : Prop)

lemma logic_exercise: P->Q->P

begin

  intro hP,

  intro hQ,

  exact hP,

end
```

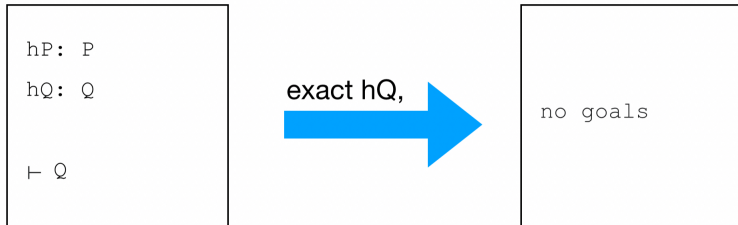
Example 1

Some basic *tactics*

exact (+hypothesis)

Provides the exact proof of the goal.

Example:

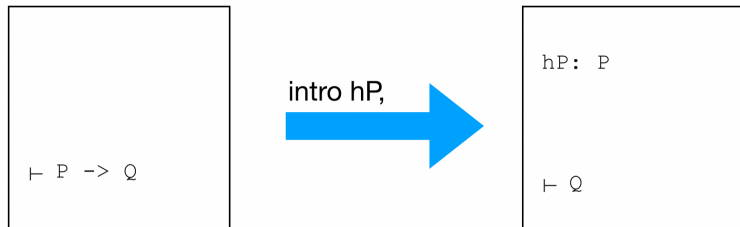


Some basic *tactics*

intro (+hypothesis' name)

Given a goal $P \rightarrow Q$, it introduces P as an hypothesis and turns the goal into Q .

Example:

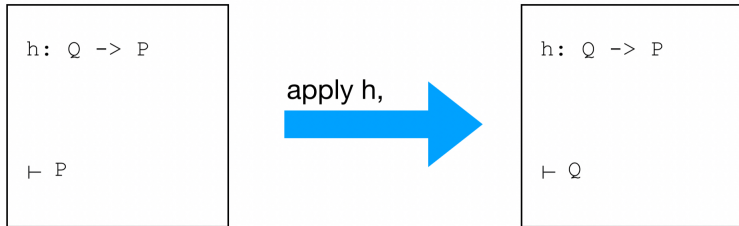


Some basic *tactics*

apply (+hypothesis)

Given a goal $P \rightarrow Q$ and a hypothesis $Q \rightarrow P$, turns the goal into P .

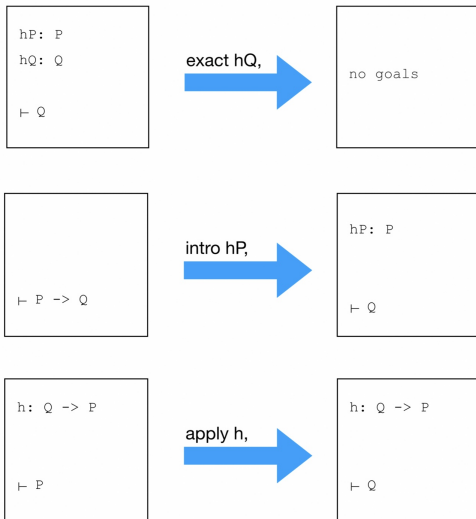
Example:



Try it yourself!



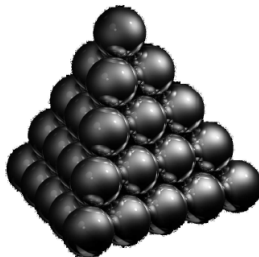
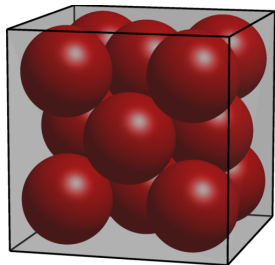
<https://shorturl.at/bikrG>



Example 2

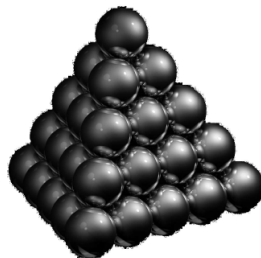
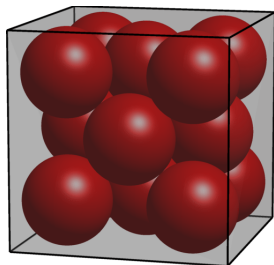
People and Projects

Kepler Conjecture. Sphere Packaging in 3 dimensions. Minimizing a function of 150 variables.



People and Projects

Kepler Conjecture. Sphere Packaging in 3 dimensions. Minimizing a function of 150 variables.



Kevin Buzzard (ICL) → Xena's Project.
Formalize the Imperial's Undergraduate Curriculum.

The library

MATHLIB

- Unified library of mathematics.
- Nov. 2023: +127.000 theorems, +70.000 definitions.



```
Documentation
Mathlib.NumberTheory.Padics.PadicNumbers

General documentation
Index
foundational types

Library
  ▶ Aesop (file)
  ▶ Archive (file)
  ▶ Counterexamples (file)
  ▶ ImportGraph
  ▶ Init (file)
  ▶ Lake (file)
  ▶ Lean (file)
  ▼ Mathlib (file)
    ▶ Algebra
    ▶ AlgebraicGeometry
    ▶ AlgebraicTopology
    ▶ Analysis
    ▶ CategoryTheory
    ▶ Combinatorics
    ▶ Computability
    ▶ Condensed
    ▶ Control
    ▶ Data
    ▶ Deprecated
    ▶ Dynamics

def Padic
  (p : ℕ) [fact (Nat.Prime p)] :
  Type
  The p-adic numbers  $\mathbb{Q}_p$  are the Cauchy completion of  $\mathbb{Q}$  with respect to the p-adic norm.
  ▶ Equations
  ▶ Instances For

def «term@[_]»
  :
  Lean.Parser.Descr
  notation for p-adic rationals
  ▶ Equations

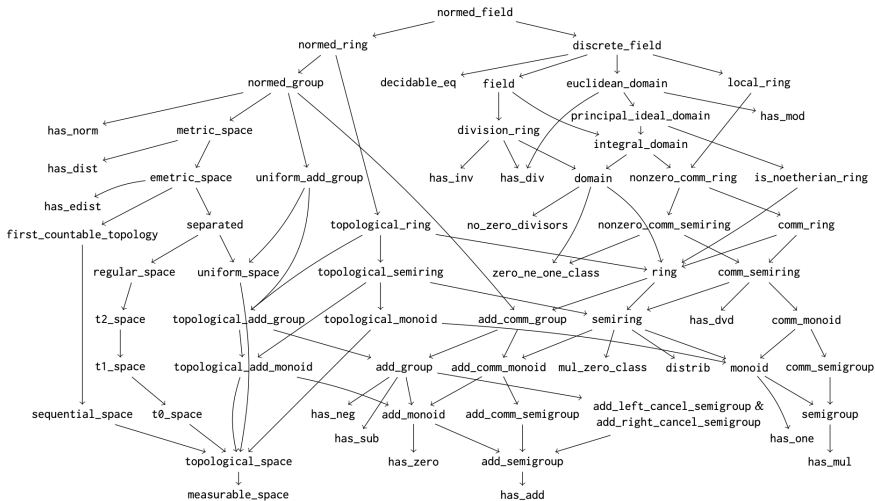
instance Padic.field
  (p : ℕ) [fact (Nat.Prime p)] :
  Field  $\mathbb{Q}_p$ 
  ▶ Equations

instance Padic.instInhabitedPadic
  (p : ℕ) [fact (Nat.Prime p)] :
  Inhabited  $\mathbb{Q}_p$ 
  ▶ Equations

instance Padic.instCommRingPadic
  source
  return to top
  source
  ▶ Imports
  ▶ Imported by
  PadicSeq
  PadicSeq.stationary
  PadicSeq.stationaryPoint
  PadicSeq.stationaryPoint.spec
  PadicSeq.norm
  PadicSeq.norm_zero_iff
  PadicSeq.equiv_zero_of_val_eq_of_eqiv_zero
  PadicSeq.norm_nonzero_of_not_eqiv_zero
  PadicSeq.norm_eq_norm_app_of_no_nzero
  PadicSeq.not_limZero_const_of_nonzero
  PadicSeq.not_eqiv_zero_const_of_nazero
  PadicSeq.norm_norSeq
  PadicSeq.lim_index_left
  PadicSeq.lim_index_right
  PadicSeq.vsubstition
  PadicSeq.norm_eq_pow_val
```

Compare to Wikipedia or The Stacks project.

Dependency graph

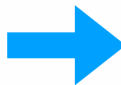


The future

Formalize difficult
known proofs



Formalize all
known proofs



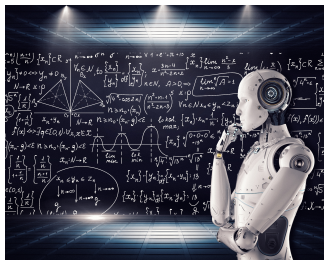
Formalize unknown
proofs?

Use LEAN to generate new knowledge. Proof assistant.

- 1 LEAN + AI.
- 2 LEAN on publishing articles.
- 3 LEAN on education.

The future

Artificial Intelligence



- Solves very easy proofs.
- Uncover hidden connections.
- Propose proof strategies.

The future

Research and scientific articles

Partly inspired by this work, software tools are now being developed to automatically convert formal proofs (in languages such as Lean) into human-readable interactive proofs.

Theorem 1.1. Let X be a topological space and Y a regular topological space. Let f be a dense subset of X . Let $f: X \rightarrow Y$ be a function. Assume that for all closures of X , f is continuous at x within A . Then f is continuous.

Proof. Let V be an open subset of Y . The aim is to prove that for all closed neighborhood W of $f(x)$, there exists a neighborhood U of x such that $f(U) \subseteq W$. We use the following argument:

Let V be a neighborhood of $f(x)$. Using our assumption that f is regular and our assumption that f is continuous at x within A , we obtain that f is a neighborhood of $f(x)$ in V . We will show that f is a neighborhood of $f(x)$ in V by showing that V is a neighborhood of $f(x)$ in V . We assume that V is a neighborhood of $f(x)$. Using our assumption that f is regular and our assumption that f is continuous at x within A , we obtain that f is a neighborhood of $f(x)$ in V . We will show that V is a neighborhood of $f(x)$ in V by showing that V is a neighborhood of $f(x)$ in V . Let the statement P be: the set U such that $f(U) \subseteq W$ is a neighborhood of x within A .

Consider the following:

- Using our assumption that f is regular and our assumption that f is continuous at x within A , we obtain that f is a neighborhood of $f(x)$ in V . We will show that V is a neighborhood of $f(x)$ in V by showing that V is a neighborhood of $f(x)$ in V .
- Using our assumption that f is regular and our assumption that f is continuous at x within A , we obtain that f is a neighborhood of $f(x)$ in V .
- Using our assumption that f is regular and our assumption that f is continuous at x within A , we obtain that f is a neighborhood of $f(x)$ in V .

This completes the proof.

(Example from a talk of Patrick Massot, 2023)

Terence Tao Machine assisted proofs

Terence Tao, "Machine Assisted Proof"



Subscribirse

2,8 K Compartir Clip ...

Education: Xena's Project: LEAN as a game.

To learn more

More info about LEAN:

- Lean Webpage: <https://lean-lang.org>
- Lean Comunity / Mathlib:
<https://leanprover-community.github.io>
- The Xena Project:
<https://www.ma.imperial.ac.uk/~buzzard/xena/>

Where to learn LEAN:

- Natural Number Game: <https://adam.math.hhu.de>
- More references:
<https://leanprover-community.github.io/learn.html>

Videos about LEAN:

- Terence Tao, "Machine Assisted Proof":
<https://www.youtube.com/watch?v=AayZuuDDKP0>
- Microsoft research:
<https://www.microsoft.com/en-us/research/project/lean/>