Interactive Theorem Proving with Lean

Dr.-Ing. Sebastian Ullrich
Head of Engineering, Lean FRO

January 30, 2025

About me

2010–2016 MSc in CS at KIT

- 2016 Master’s thesis on Lean+Rust at CMU (Pittsburgh, USA)

2017–2023 PhD in CS at the Programming Paradigms group, KIT

- 2018 Internship at Microsoft Research (Redmond, USA), design draft of Lean 4

2023–now Co-founder of the Lean Focused Research Organization together with Leonardo de Moura (AWS)

- 18 people worldwide, 3 in Munich

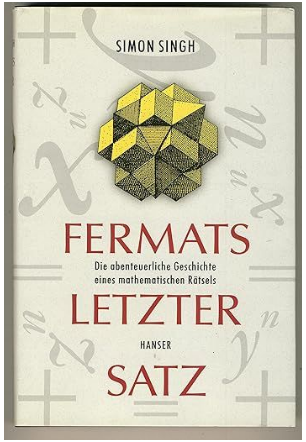
Why Prove?

Unverified Mathematics: Mistakes in proofs or logical gaps that go unnoticed

Unverified Software: Bugs, vulnerabilities, and failures in critical systems

Unverified AI: Hallucinations, incorrect outputs, and unreliable reasoning steps

The Lean project started in 2013 with the goal of addressing challenges in software verification. Today, it has gained popularity in both mathematics and AI.



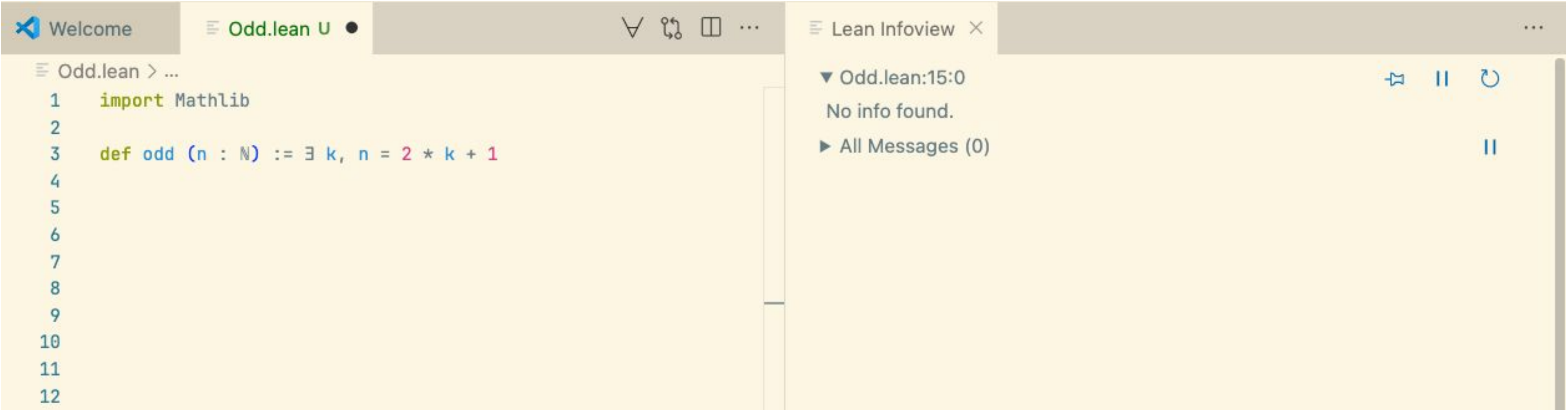
Lean is an open-source **programming language** and **proof assistant** that is transforming how we approach mathematics, software verification, and AI

Lean provides **machine-checkable proofs**

Lean addresses the “**trust bottleneck**”

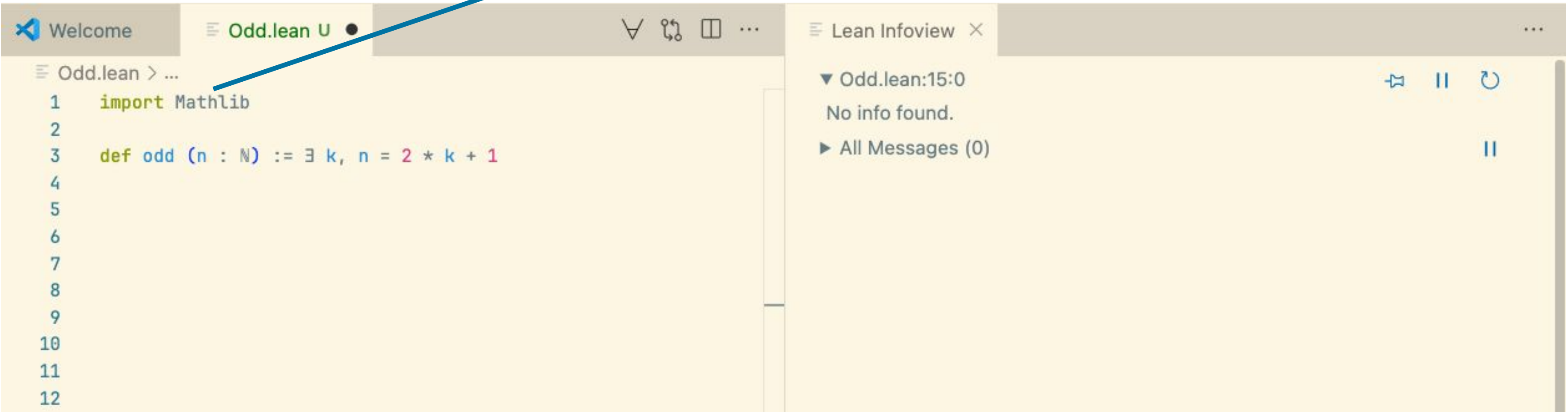
Lean opens up new possibilities for **collaboration**

A small example

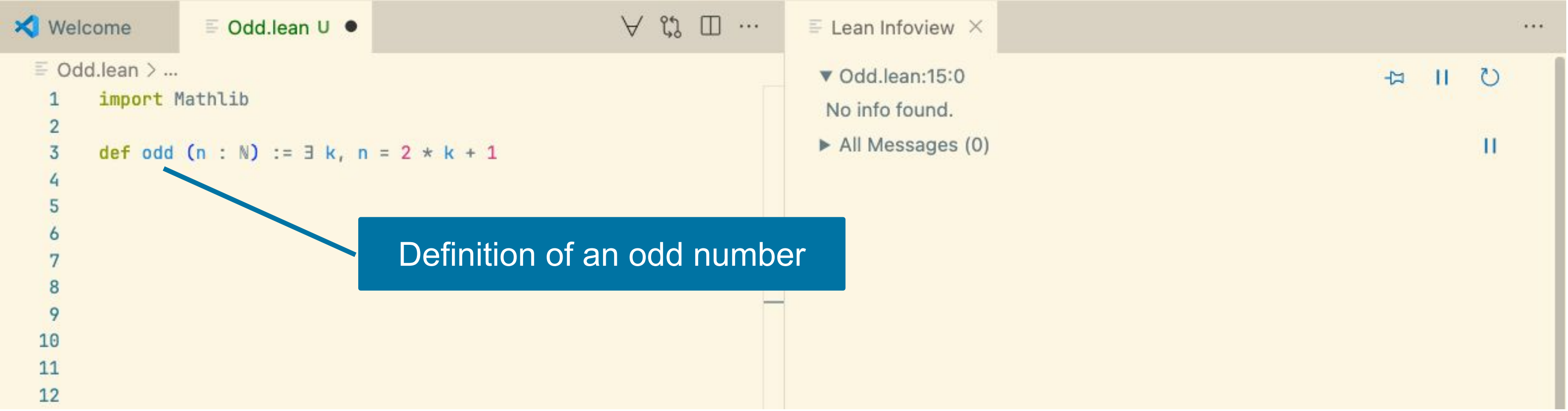


A small example

Mathlib is the Lean Mathematical library



A small example



Our first theorem

Welcome | Odd.lean U

Odd.lean

```lean
import Mathlib

def odd (n : ℕ) := ∃ k, n = 2 * k + 1

theorem five_is_odd : odd 5 := by
  use 2
  done
```

Lean Infoview

▼ Odd.lean:15:0

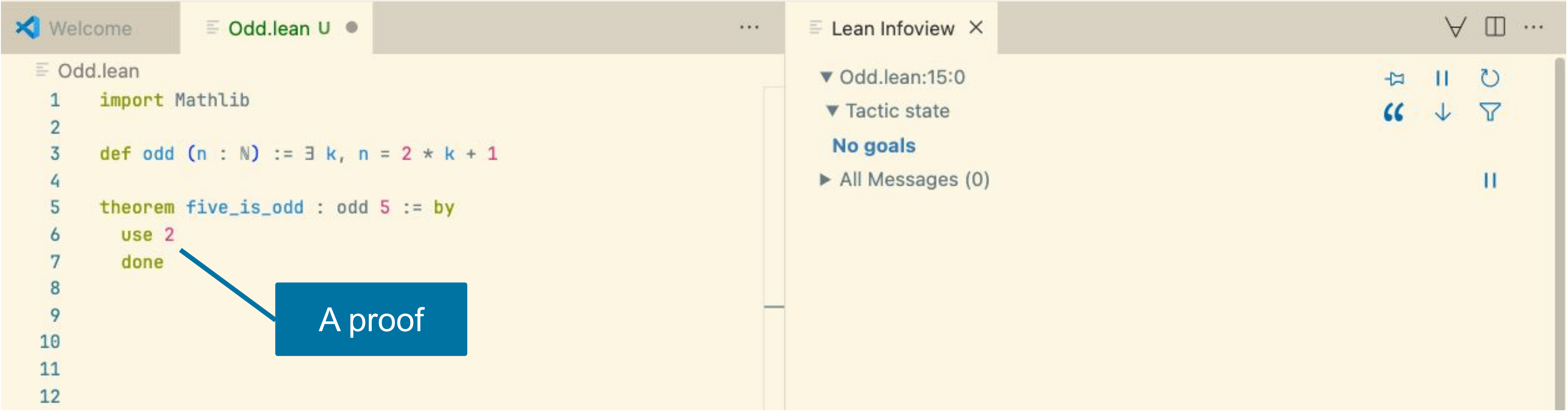
▼ Tactic state

No goals

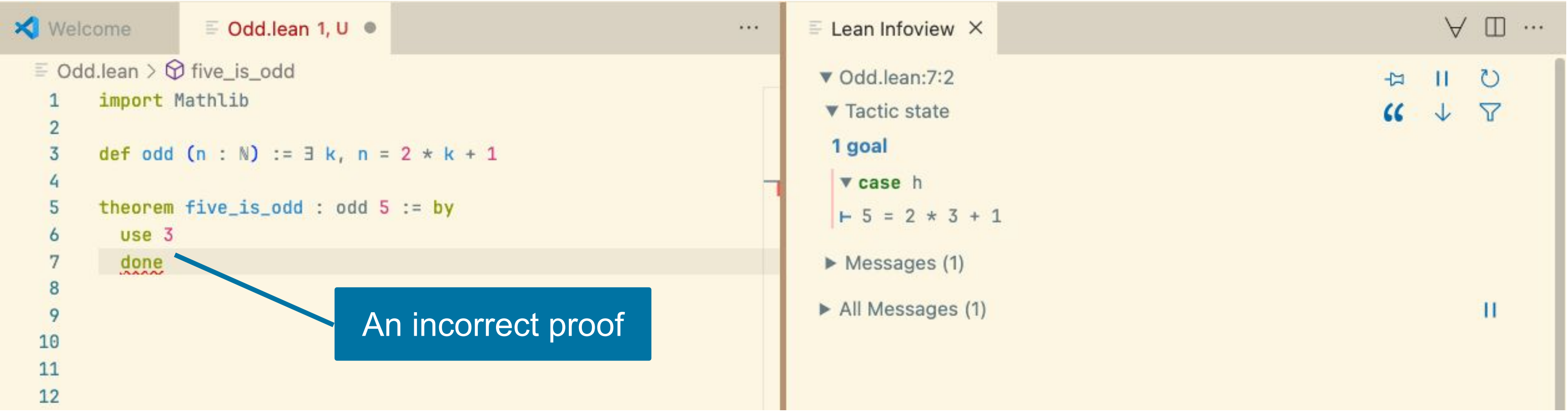
▶ All Messages (0)

Theorem statement, i.e., the claim being made

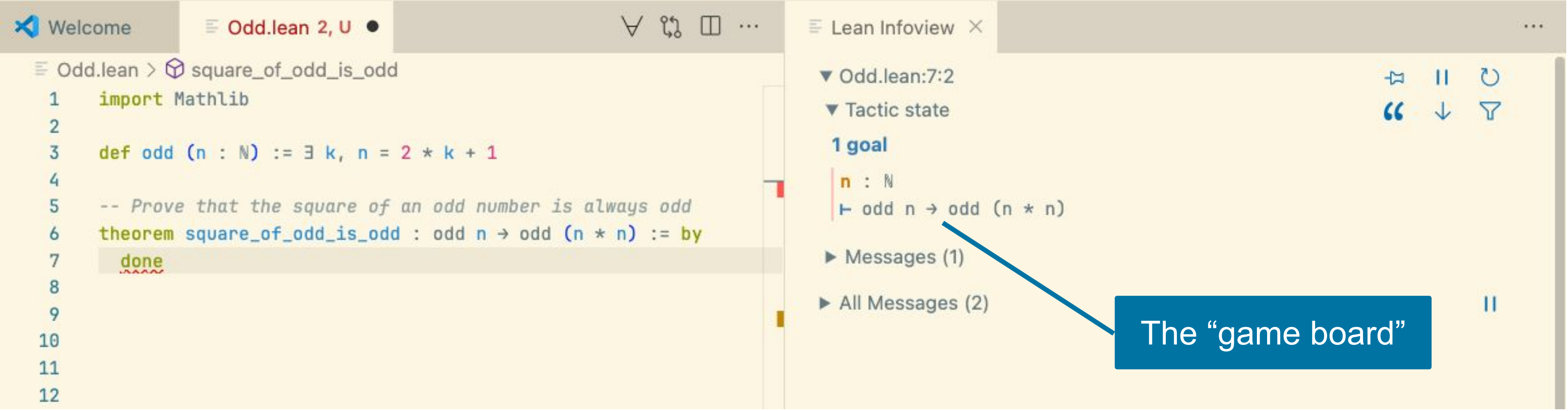
Our first theorem



Our first theorem

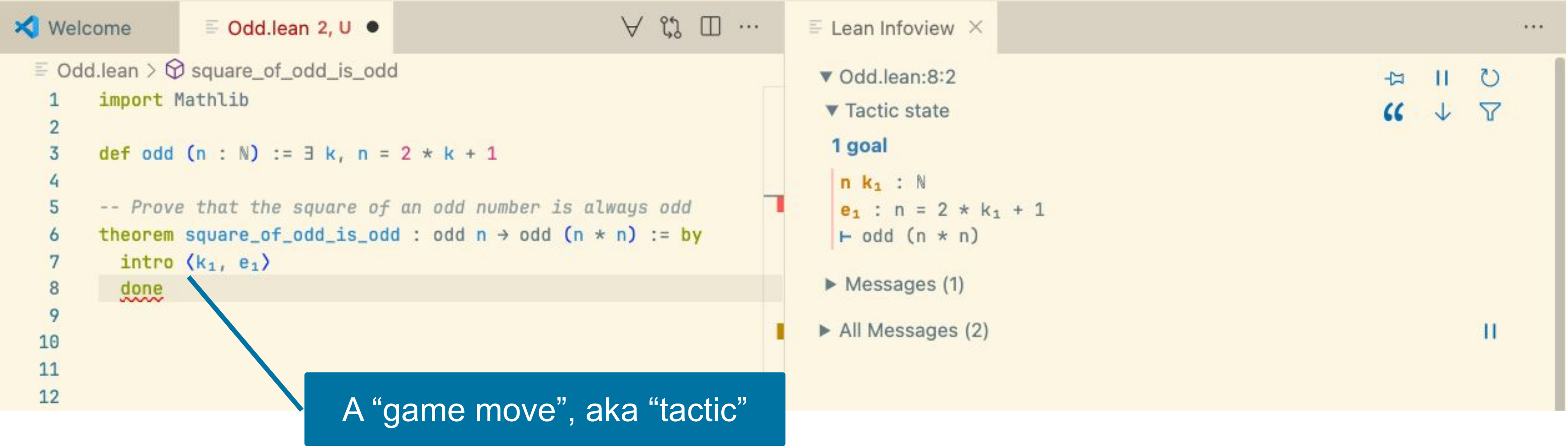


Theorem proving in Lean is an interactive game

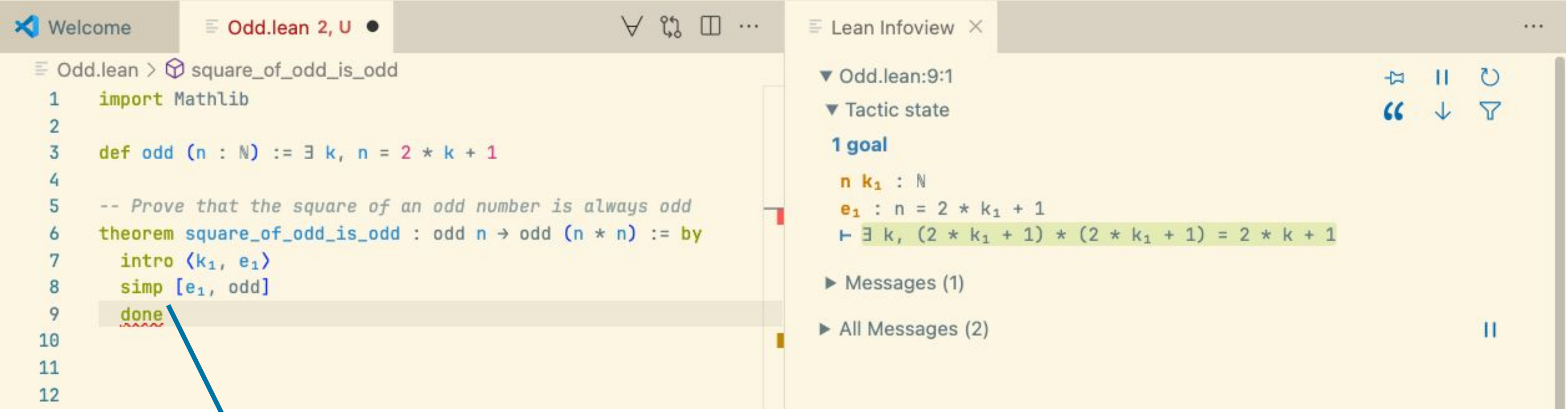


“You have written my favorite computer game” – Kevin Buzzard (Imperial College London)

Theorem proving in Lean is an interactive game

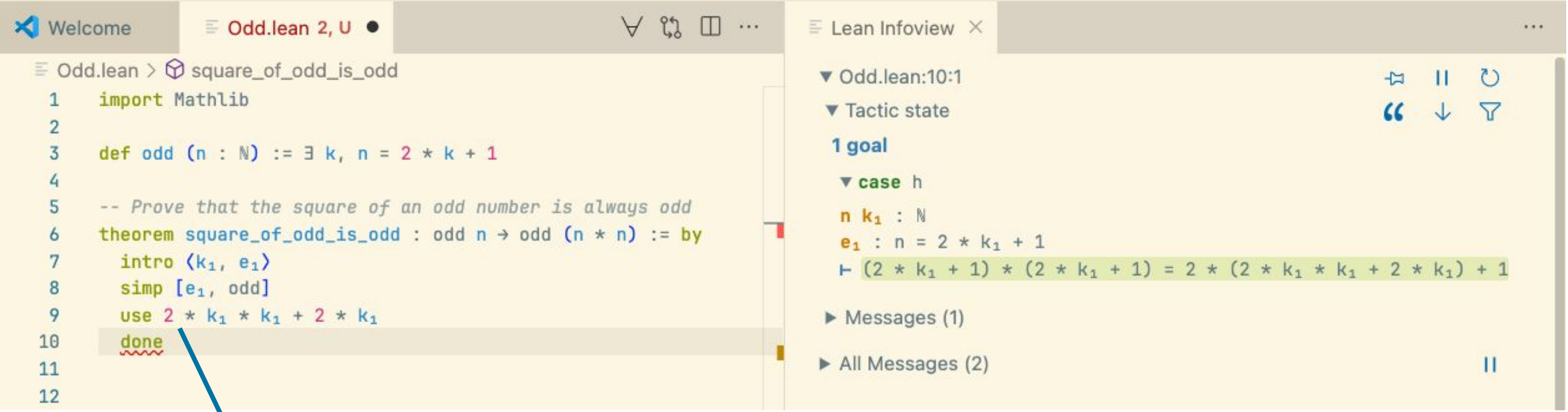


Theorem proving in Lean is an interactive game



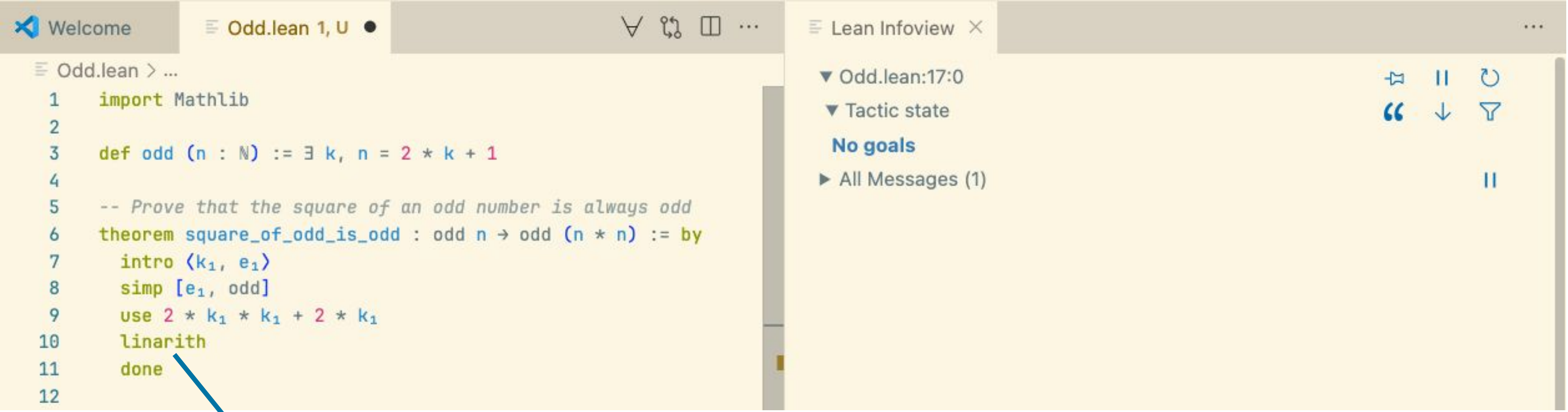
The "game move" `simp`, the simplifier, is one of the most popular moves in our game

Theorem proving in Lean is an interactive game



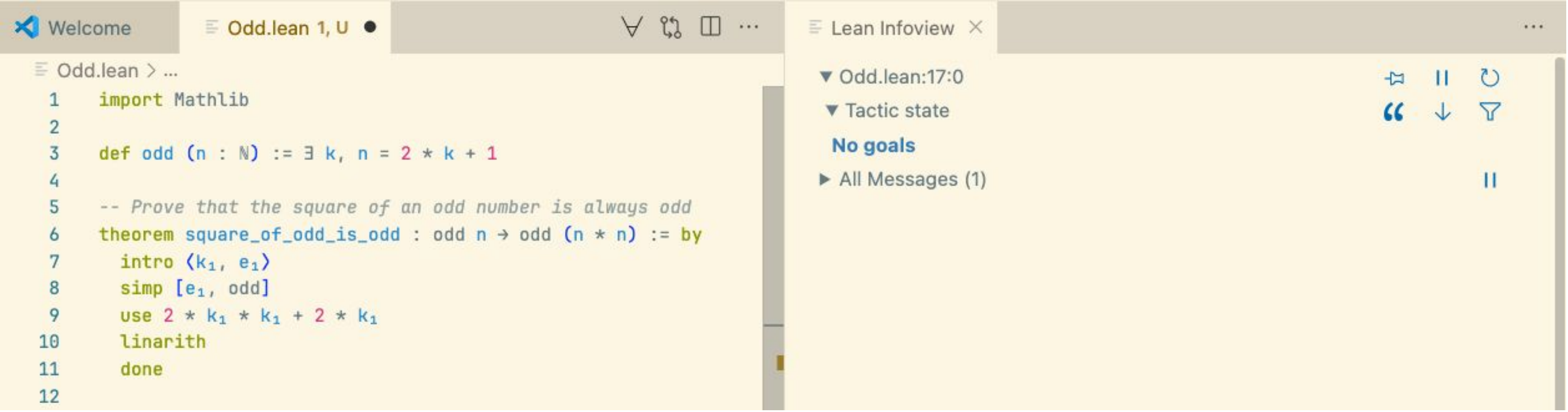
The “game move” `use` is the standard way of proving statements about existentials

Theorem proving in Lean is an interactive game



We complete this level using `linarith`, the linear arithmetic, move

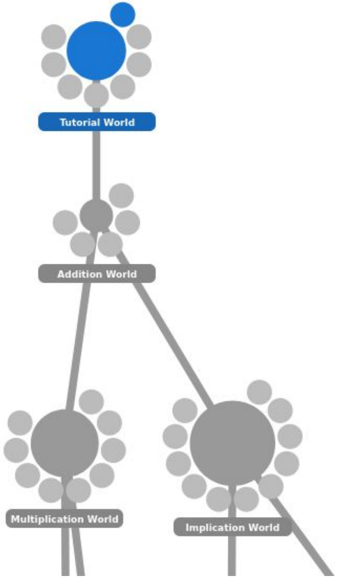
Theorem proving in Lean is an interactive and addictive game

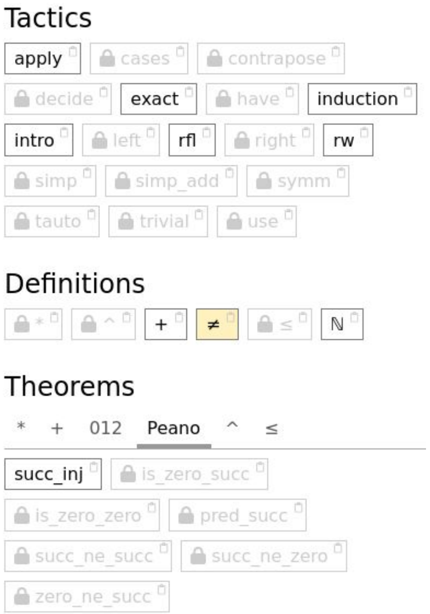


"You can do 14 hours a day in it and not get tired and feel kind of high the whole day.

You're constantly getting positive reinforcement" – Amelia Livingston (University College London)

Trying it for yourself: The Natural Number Game





adam.math.hhu.de

Mathematics

Software

AI

Mathematics

Mathlib github.com/leanprover-community/mathlib4

The Lean Mathematical Library supports a wide range of projects

It is an open-source **collaborative project** with over 500 contributors and 1.5M LoC

“I’m investing time now so that somebody in the future can have that amazing experience” –

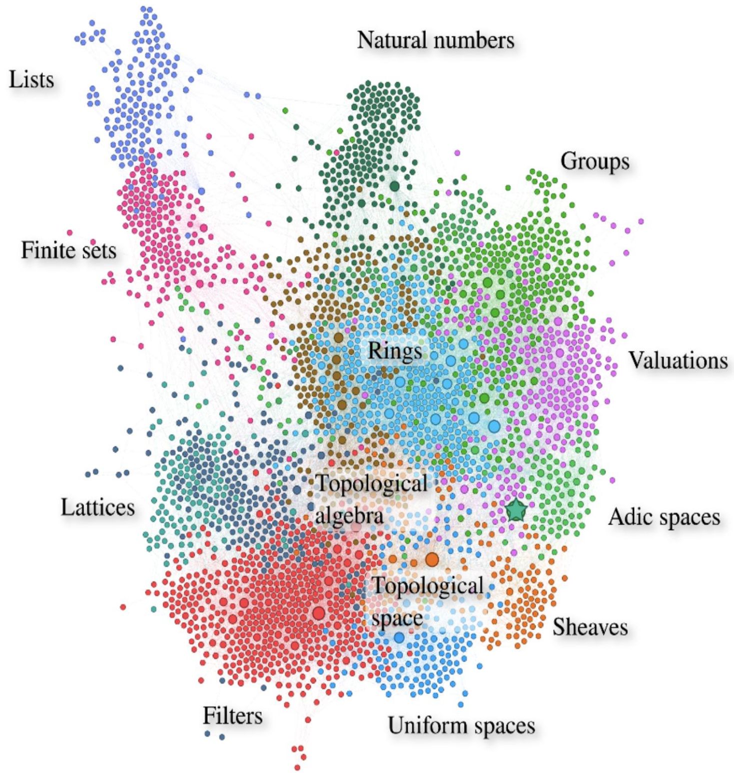
Heather Macbeth (Fordham University)

KEVIN HARTNETT SCIENCE OCT 11, 2020 8:00 AM

The Effort to Build the Mathematical Library of the Future

A community of mathematicians is using software called Lean to build a new digital repository. They hope it represents where their field is headed next.

Mathlib


Natural numbers
Lists
Groups
Finite sets
Rings
Valuations
Lattices
Topological algebra
Adic spaces
Topological space
Sheaves
Filters
Uniform spaces

The Perfectoid Spaces Project

Kevin Buzzard, Patrick Massot, Johan Commelin

Goal: Demonstrate that we can **define complex mathematical objects** in Lean.

They translated Peter Scholze's definition into a form a computer can understand.

It not only achieved its goals but also demonstrated to the math community that **formal objects can be visualized and inspected with computer assistance**.

Math is now **data** that can be **processed**, **transformed**, and **inspected** in various ways.

The Perfectoid Spaces Project

mathoverflow

Home

What are "perfectoid spaces"?

72

Here is a completely different kind of answer to this question.

A *perfectoid space* is a term of type `PerfectoidSpace` in the Lean theorem prover.

Here's a quote from the source code:

```
structure perfectoid_ring (R : Type) [Huber_ring R] extends Tate_ri
(complete  : is_complete_hausdorff R)
(uniform   : is_uniform R)
(ramified  : ∃ ϖ : pseudo_uniformizer R, ϖ^p ∣ p in R°)
(Frobenius : surjective (Frob R°/p))

/-
CLVRS ("complete locally valued ringed space") is a category
```

The Challenge

In November of 2020, Peter Scholze posits the Liquid Tensor Experiment (LTE) challenge

> *"I spent much of 2019 obsessed with the proof of this theorem, almost getting crazy over it. In the end, we were able to get an argument pinned down on paper, but I think nobody else has dared to look at the details of this, and so I still have some small lingering doubts"*

The First Victory

Johan Commelin led a team with several members of the **Lean community and announced the formalization of the crucial intermediate lemma** that Scholze was unsure about, with only minor corrections, in **May 2021**.

> *“[T]his was precisely the kind of oversight I was worried about when I asked for the formal verification. [...] The proof walks a fine line, so if some argument needs constants that are quite a bit different from what I claimed, it might have collapsed”* – Peter Scholze


nature

Explore content · Journal information · Publish with us · Subscribe

nature > news > article

NEWS | 18 June 2021

Mathematicians welcome computer-assisted proof in ‘grand unification’ theory

Success

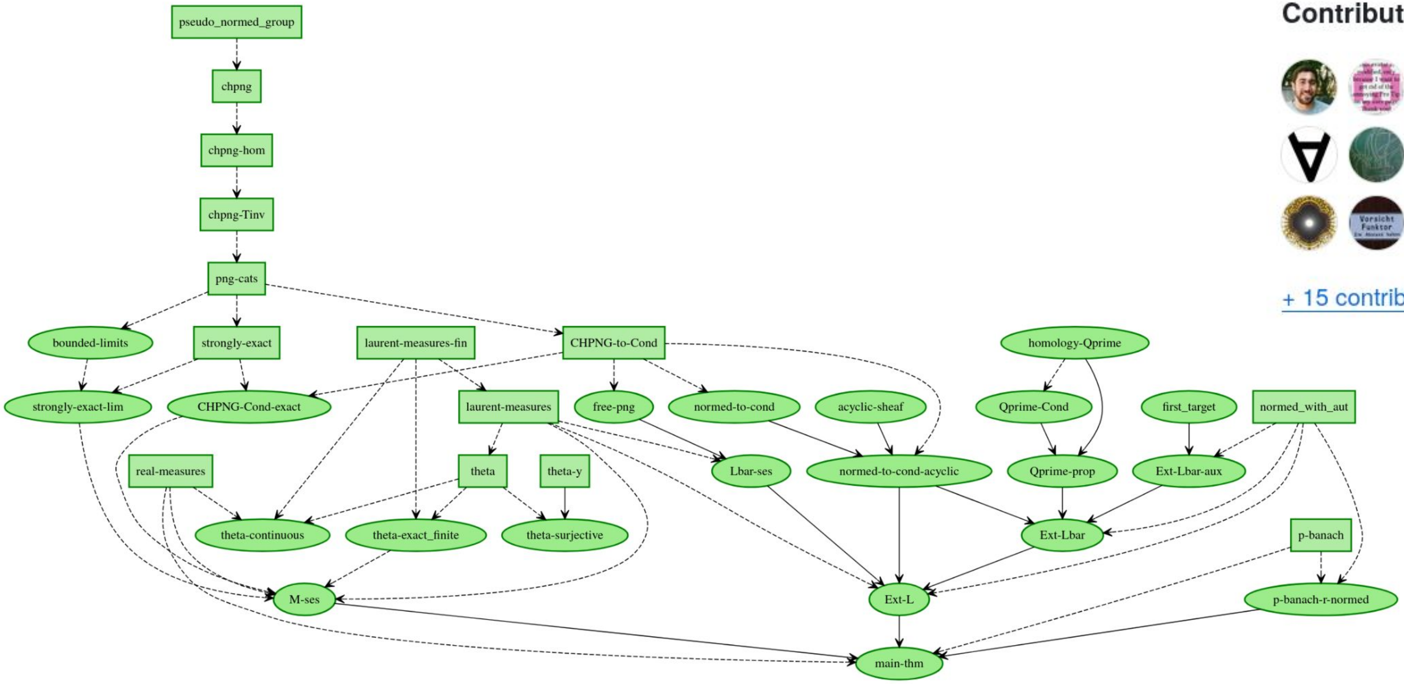
The full challenge was completed in July 2022.

The team not only verified the proof but also simplified it.
Moreover, they did this without fully understanding the entire proof.

Johan, the project lead, reported that he could only see two steps ahead. **Lean was a guide**.

> *"The Lean Proof Assistant was really that: an assistant in navigating through the thick jungle that this proof is. Really, one key problem I had when I was trying to find this proof was that I was essentially unable to keep all the objects in my RAM, and I think the same problem occurs when trying to read the proof"* – Peter Scholze

Crowd-Source Mathematics



Contributors 29

+ 15 contributors

Only the Beginning

Independence of the Continuum Hypothesis, Han and van Doorn, 2021

Sphere Eversion, Massot, Nash, and van Doorn, 2020-2022

Fermat’s Last Theorem for regular primes, Brasca et al., 2021-2023

Unit Fractions, Bloom and Mehta, 2022

Consistency of Quine's New Foundations, Wilshaw and Dillies, 2022-2024

Polynomial Freiman-Ruzsa Conjecture, Tao and Dillies, 2023

Prime Number Theorem And Beyond, Kontorovich and Tao, 2024-ongoing

Carleson Project, van Doorn, 2024-ongoing

Equational Theories Project, Tao, Monticone, and Srinivas, 2024-ongoing

Fermat’s Last Theorem (FLT), Buzzard, 2024-ongoing, community estimates it will take 1M+ LoC

Summary

Machine-checkable proofs enable a new level of **collaboration** in mathematics

It is not just about proving but also understanding complex objects and proofs, getting new insights, and navigating through the “thick jungles” that are **beyond our cognitive abilities**

Software

Lean in Software Verification: The Story of SampCert

Lean is a programming language, and is used in **many software verification projects**

You can write code and reason about it simultaneously

You can prove that your code has the properties you expect

"Testing can show the presence of bugs, but not their absence" – E. Dijkstra

Differential Privacy

A mathematical framework that ensures the **privacy of individuals** in a dataset by adding controlled **random noise** to the data

Discrete sampling algorithms, like the *Discrete Gaussian Sampler*, are used to add carefully calibrated noise to data

What may go wrong if a buggy sampler is used?

- **Privacy Violations**: leakage of sensitive information
- **Incorrect Results**: distorted analysis results

SampCert

A project led by **Jean-Baptiste Tristan** at AWS

An **open-source** Lean library of formally **verified differential privacy primitives**

Tristan's implementation is not only verified, but it is also **twice as fast as the previous one**

He managed to implement **aggressive optimizations** because Lean served as a guide, ensuring that **no bugs** were introduced

SampCert would not exist without Mathlib

SampCert is software, but its verification relies heavily on Mathlib

The verification of code addressing practical problems in data privacy depends on the formalization of mathematical concepts, from **Fourier analysis** to **number theory** and **topology**

Many more open-source projects

Cedar, a policy language and evaluation engine

LNSym, a symbolic simulator for Armv8 native-code programs: cryptographic machine-code programs

TenCert, a tensor compiler, verified StableHLO and NKI

NFA2SAT, a verified string solver

Many more at the **Lean Project Registry**: reservoir.lean-lang.org

Summary

Machine-checkable proofs enable you to **code, refactor, and optimize without fear**

AI

Lean Enables Verified AI for Mathematics and Code

LLMs are powerful tools, but they are prone to **hallucinations**

In Math, a **small mistake can invalidate the whole proof**

Imagine manually checking an AI-generated proof with the size and complexity of FLT

- The informal proof is **over 200 pages**
- Buzzard estimates a formal proof will require more than **1M LoC** on top of Mathlib

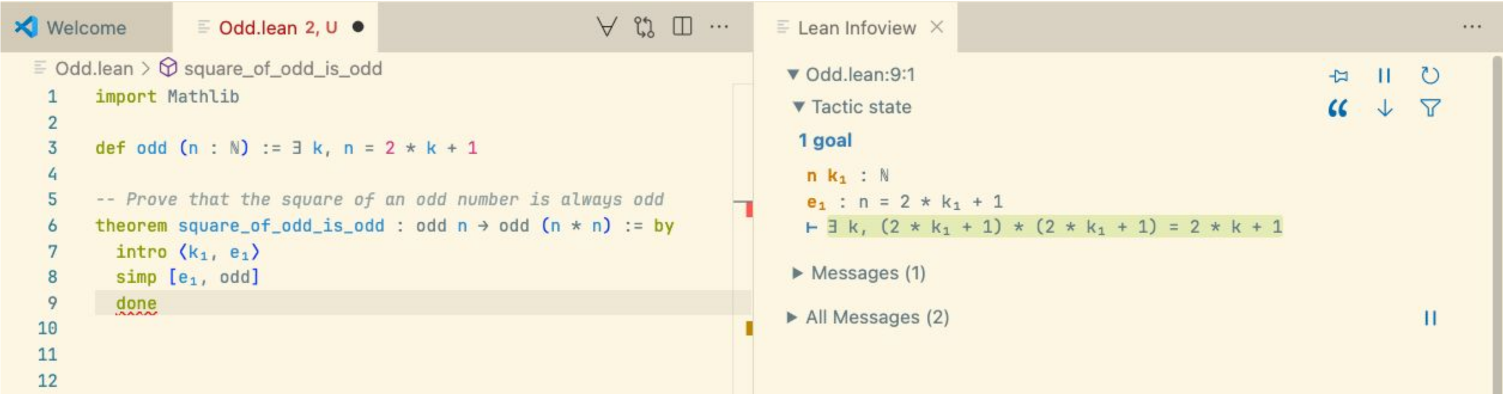
Machine-checkable proofs are the antidote to hallucinations

Synthetic Data Generation

LLMs require **vast amounts of data** for training

Lean mathematical libraries provide valuable, **correct-by-construction training data**

Tools like **lean-training-data** by **Kim Morrison** extract data that includes the “game board” before and after each “move”



AI Proof Assistants

Several groups are developing AI that suggests the **next move**(s) in Lean's interactive proof game

LeanDojo is an open-source project from Caltech, and everything (model, datasets, code) is open

OpenAI and **Meta AI** have also developed AI assistants for Lean

AI Proof Assistants

DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search

Huajian Xin*, Z.Z. Ren*, Junxiao Song*, Zhihong Shao*, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang
Xuan Lu, Qiushi Du, Wenjun Gao, Qihao Zhu, Dejian Yang, Zhibin Gou, Z.F. Wu, Fuli Luo, Chong Ruan

DeepSeek-AI

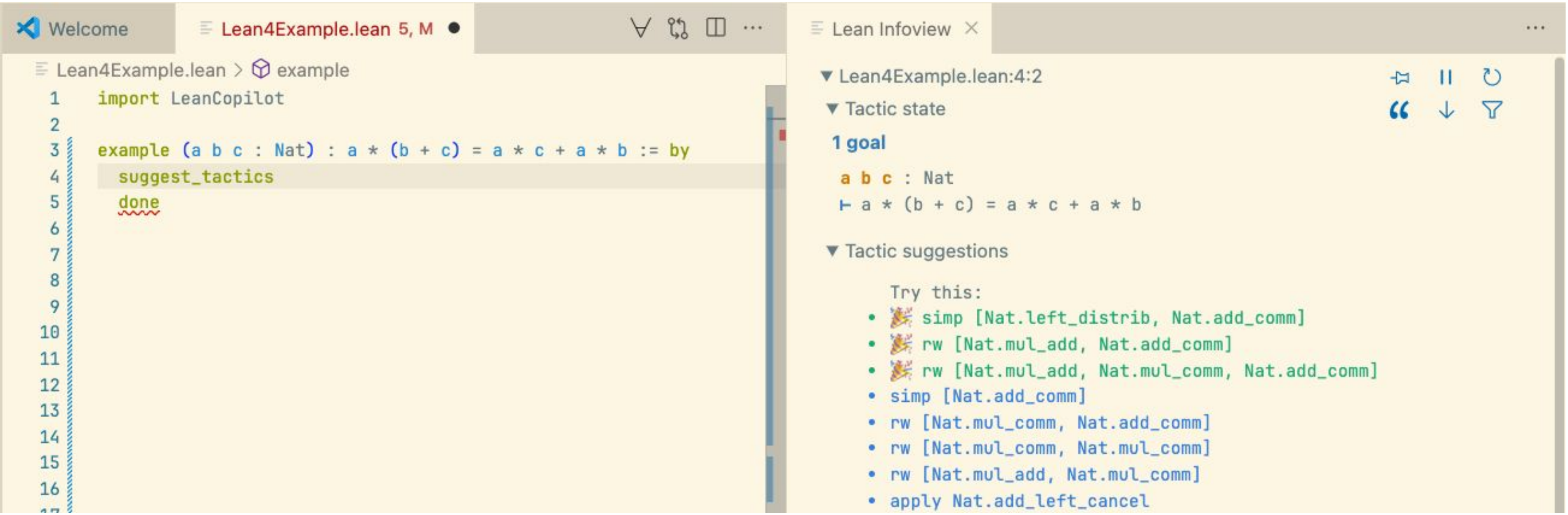
https://github.com/deepseek-ai/DeepSeek-Prover-V1.5

Abstract

We introduce DeepSeek-Prover-V1.5, an open-source language model designed for theorem proving in Lean 4, which enhances DeepSeek-Prover-V1 by optimizing both training and infer-

AI Proof Assistants

LeanCopilot is part of the LeanDojo project at Caltech. It uses the move (aka tactic) suggestion feature available in the Lean IDE.



The New York Times

Artificial Intelligence › | A.I.'s Math Problem | A.I. Training Data Disappears | Microsoft's Risk-Taker | Fine Print Changes | Quiz: Fake or Real Images?

Move Over, Mathematicians, Here Comes AlphaProof

A.I. is getting good at math — and might soon make a worthy collaborator for humans.

Share full article 47

Ringing the gong at Google Deepmind's London headquarters, a ritual to celebrate each A.I. milestone, including its recent triumph of reasoning at the International Mathematical Olympiad. Google Deepmind

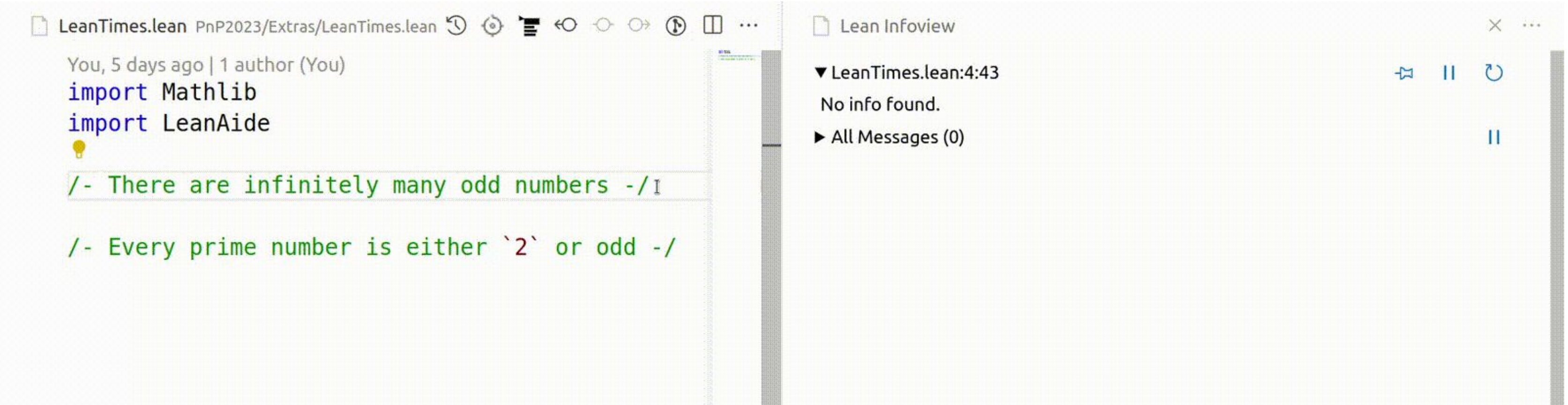
and expressed in code. It used theorem prover and proof-assistant software called Lean, which guarantees that if the system says a proof is correct, then it is indeed correct. "We can exactly check that the proof is correct or not," Dr. Hubert said. "Every step is guaranteed to be logically sound."

Auto-formalization

The process of converting natural language into a formal language like Lean

It is much **easier to learn to read Lean than to write it**

LeanAide is one of the auto-formalization tools available for Lean



Specification-Oriented Programming

What if we could describe complex systems in plain language, and AI turned them into formal, provable code?

Specification-Oriented Programming

What if we could describe complex systems in plain language, and AI turned them into formal, provable code?

- You describe what you want in natural language or pseudo-code.
- AI auto-formalizes it in Lean.
- You review the result and collaborate until it matches your intent.
- AI synthesizes efficient, machine-checkable code and proofs.

Summary

Machine-checkable proofs enable **AI that does not hallucinate**

LLMs enable **auto-formalization**

Lean can generate **synthetic correct by-construction datasets**

Machine learning opens doors to **new proof search engines**

Wrap-Up

Lean enables decentralized collaboration

Lean is Extensible

Users extend Lean using Lean itself

Lean is implemented in Lean

You can make it your own

You can create your own moves

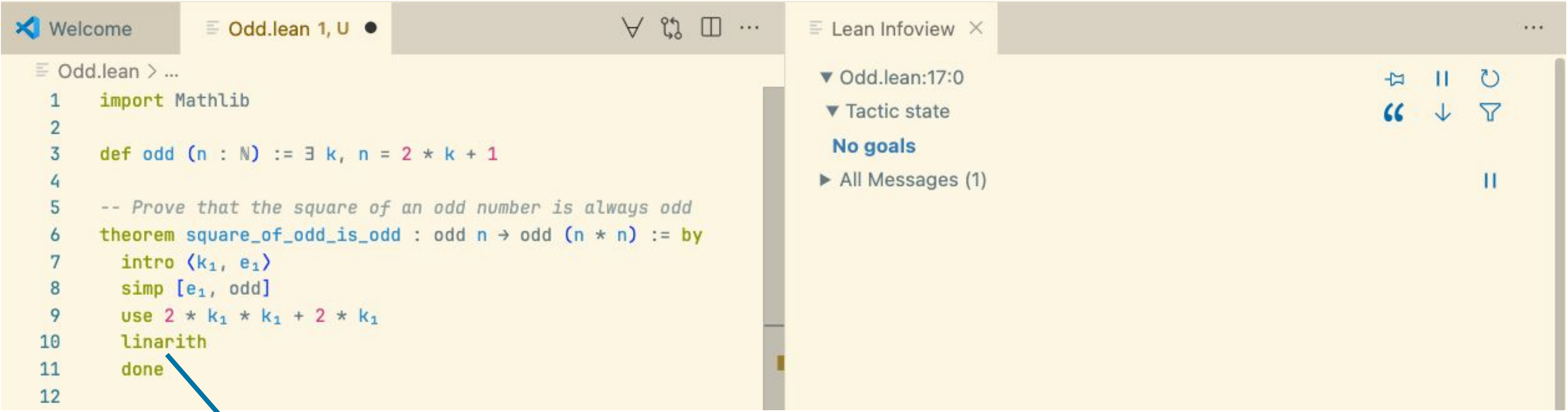
Machine-Checkable Proofs

You don’t need to trust me to use my proofs

You don’t need to trust my automation to use it

Code without fear

Lean is a game where we can implement your own moves

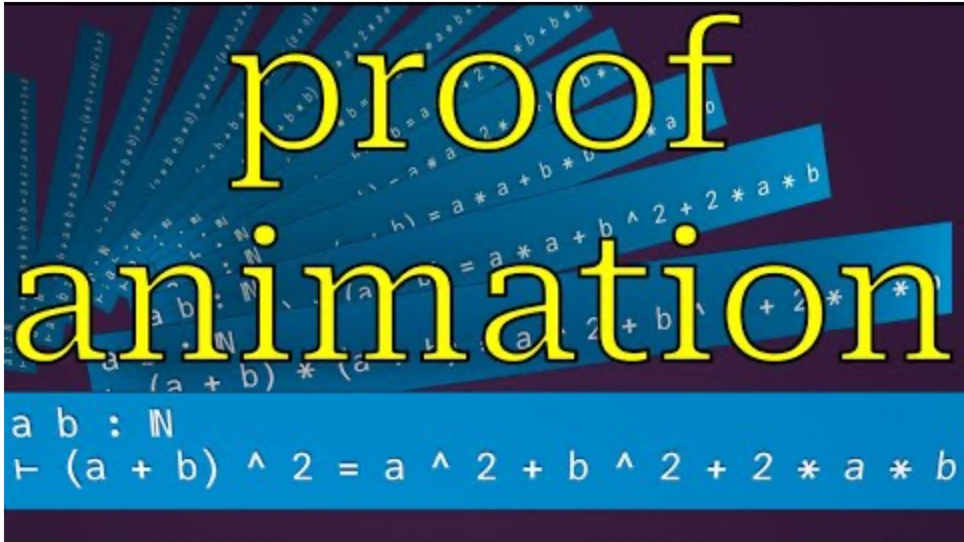


The `linarith` “move” was implemented by the Mathlib community in Lean!

You can use Lean to introspect its internal data

The tool **lean-training-data** is implemented in Lean itself. **It is a Lean package**.

A similar approach can be used to automatically generate proof animations



Focused Research Organizations

A new type of nonprofit startup for science developed by Convergent Research

Lean FRO

The Lean Focused Research Organization (FRO) is a non-profit dedicated to Lean's development

Founded in **August 2023**, the organization has 18 members

Its mission is to enhance critical areas: **scalability**, **usability**, **documentation**, and **proof automation**

We want to reach **self-sustainability by 2028** and become the **Lean Foundation**

Philanthropic support is gratefully acknowledged from the **Simons Foundation**, the **Alfred P. Sloan Foundation**, **Richard Merkin**, and **Founders Pledge**

Conclusion

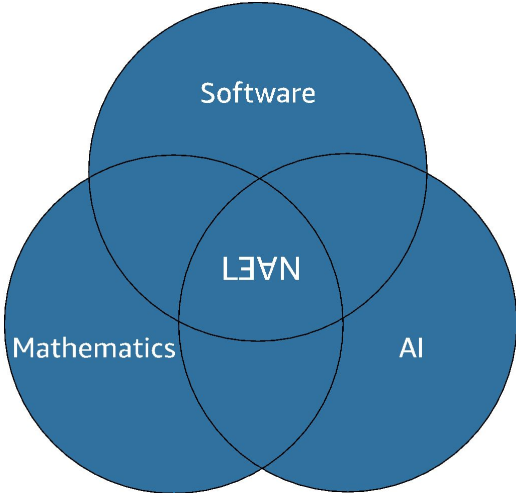
Lean is an **efficient programming language** and **proof assistant**

Machine-checkable proofs eliminate the trust bottleneck

Lean enables **decentralized collaboration**

Lean is very **extensible**

The Mathlib community is changing how math is done



It is not just about proving but also understanding complex objects and proofs, getting new insights, and navigating through the “thick jungles” that are beyond our cognitive abilities

Thank You

lean-lang.org
lean-fro.org
Community chat: leanprover.zulipchat.com
Mastodon: @leanprover@functional.cafe