Monthly Community Meeting

Lean FRO
21 November 2023
News since last community meeting

- **Released 4.2.0**
- FRO welcomed: Joachim Breitner and Kyle Miller.
- Terence Tao formalized his paper using Lean 4, is already working on the next project.
- Bhavik Mehta: [Formalizing Modern Research Mathematics in Real Time](#).
- At AWS: Formalization of the [Cedar Policy Language in Lean 4](#).
- Lean is now a Strategic Open-Source Project at Amazon.
  - Josh Clune’s LeanSAT is now open-source.
- Evgenia Karunus & Anton Kovsharov: [Paperproof](#).
- Morph.so released a [LLM for Lean](#) and [Moogle.ai](#).
- Many issues fixed.
The Lean FRO: team

Leo de Moura (AWS)
Chief Architect, Co-Founder

Sebastian Ullrich
Head of Engineering, Co-Founder

Joachim Breitner
Senior Research Software Engineer

David Thrane Christiansen
Senior Research Software Engineer

Joe Hendrix
Principal Research Software Engineer

Marc Huisenga
Research Software Engineer

Mac Malone
Research Software Engineer

Kyle Miller
Research Software Engineer (Part-Time)

Scott Morrison
Senior Research Software Engineer
The Lean FRO: Roadmap

- [https://lean-fro.org/about/roadmap](https://lean-fro.org/about/roadmap)
- Vision
- Mission
- Tracking Progress
- A Laser-Focused First Year
- Deliverables
Vision

Cultivating a Diverse, Collaborative Ecosystem

Aiming to Make Lean Essential in Research, Development, and Education
Mission

Enhancing and Expanding Lean's Capabilities

- Scalability
- Usability
- Proof Automation
- Documentation
- Broad Application in Diverse Fields

Achieving Self-Sustainability and Enduring Growth
Tracking progress

# users
# courses
# packages
overhead factor
build time

Measuring Lean’s Adoption and Impact
Quick update: scalability improvements in Lean

Single-core compilation time / 1 million lines of code (Mathlib on AMD Ryzen 9 7950X3D)

- Lean 3.51.1: 20.8 hours
- Lean 4.0.0: 4.3 hours
- Lean 4.3.0-rc2: 3.6 hours

16% speedup in 4 months

Multi-core compilation time: 10.6 minutes (38% speedup in 4 months, it was 17.1 minutes)
A Laser-Focused First Year

We recognize that there are several obstacles to the wider adoption of Lean.

Addressing these challenges is our foremost priority in the first year.
Addressing Paper-Cuts in Lean

Focus on identifying and resolving small yet impactful issues.

Better error messages, performance, tooling.
Enhancing User-Experience and Documentation

- Refining the user interface.
- Documentation authoring tools.
Developing Cloud-Build & Reservoir

Robust cloud build support for all Lean packages.

Launch Reservoir: the Lean package repository.
Developing and Integrating Proof Automation

Advancing Lean’s proof automation capabilities is another key focus.

We aim to make the process of developing and verifying proofs more efficient.
Standard Library

We envision a standard library with specifications, proofs and tactics for its main components.

The Lean standard library will also serve as the bedrock for many other packages and projects.
Cathedral vs Bazaar: Open-Source Development Models

**Cathedral**
- Centralized control
- Few contributors with high expertise
- Long release intervals
- More structured and planned development

**Bazaar**
- Decentralized and open contribution
- Many contributors of varying skill levels
- Short release intervals, early and frequent releases
- Organic development process
Decentralized Development

Empowering **Decentralization** through **Lean's Extensibility**.

A decentralized model fosters **ownership** and **diverse contributions**.

Many proof automation packages: **Aesop**, **Duper**, **Lean-Auto**, **LeanSAT**, **Lean-SMT**.

Other successful packages: **ProofWidgets** and **Paperproof**.
Tooling for Decentralized Work

We acknowledge we need better tooling.

Reservoir is only part of it.

We also need tools for **breaking unnecessary dependencies**.

A **Module System** for specifying clear interfaces between modules.
Deliverables

Our roadmap and all deliverables are available at http://lean-fro.org/about/roadmap.html

Other components not covered here: website, code generator, etc.

Postponed components: debugger.

If you are interested in contributing to Lean, please check it out.
RFC, PR, and external contributions

60 open & 25 closed.

Latest successful RFC and PR: RFC: Allow trailing comma after the last element of a list #2635

Several open RFCs are being actively developed.

We are laser-focused on our first year goals.

A few thumbs up ≠ strong community support.

We pay attention to RFCs tagged as Mathlib High Priority.

We are adding more automation to streamline the process.
Our Vision for the Standard Library…