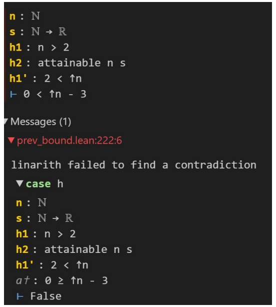
Monthly Community Meeting

Lean FRO
21 November 2023

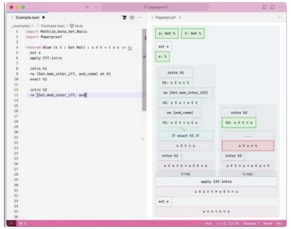
News since last community meeting



```
n : ℕ
s : ℕ → ℝ
h1 : n > 2
h2 : attainable n s
h1' : 2 < ↑n
⊢ 0 < ↑n - 3
```

```
linarith failed to find a contradiction
 ▼case h
 n : ℕ
 s : ℕ → ℝ
 h1 : n > 2
 h2 : attainable n s
 h1' : 2 < ↑n
 a✝ : 0 ≥ ↑n - 3
 ⊢ False
```

- 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.

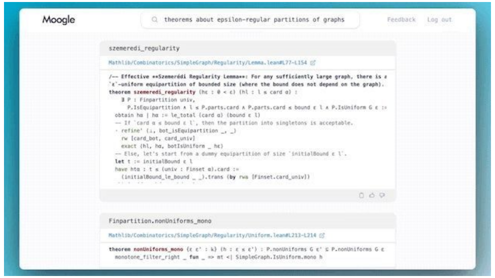


```
import Mathlib.Data.Set.Basic
import Paperproof

theorem blue (s t : Set Nat) : s ∩ t = t ∩ s := by
  ext x
  apply Iff.intro

  intro h1
  rw [Set.mem_inter_iff, and_comm] at h1
  exact h1

  intro h2
  rw [Set.mem_inter_iff, and
```



szemeredi_regularity

Mathlib/Combinatorics/SimpleGraph/Regularity/Lemma.lean#L77-L154

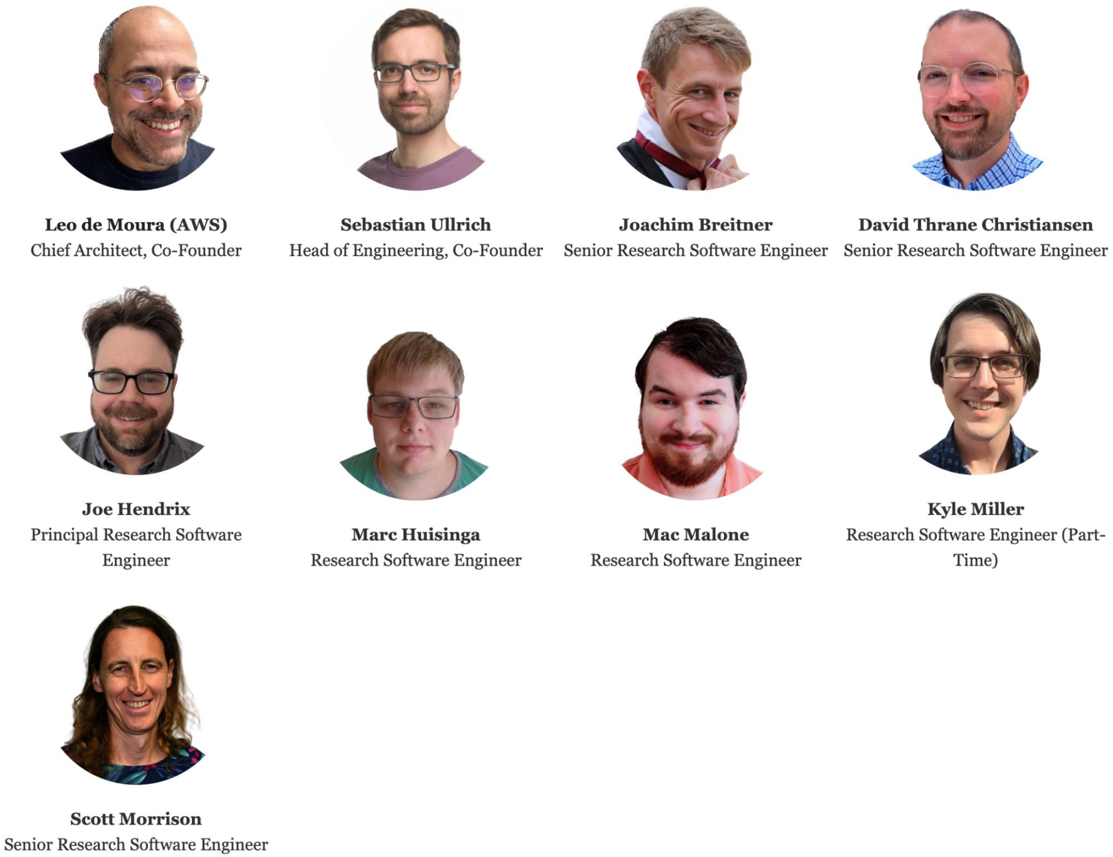
```
/-- Effective **Szemerédi Regularity Lemma**: For any sufficiently large graph, there is ε
`ε`-uniform equipartition of bounded size (where the bound does not depend on the graph).
theorem szemeredi_regularity (hε : 0 < ε) (hl : l ≤ card α) :
    ∃ P : Finpartition univ,
      P.IsEquipartition ∧ l ≤ P.parts.card ∧ P.parts.card ≤ bound ε l ∧ P.IsUniform G ε :=
  obtain hα | hα := le_total (card α) (bound ε l)
  -- If `card α ≤ bound ε l`, then the partition into singletons is acceptable.
  · refine' ⟨⊥, bot_isEquipartition _, _⟩
    rw [card_bot, card_univ]
    exact ⟨hl, hα, botIsUniform _ hε⟩
  -- Else, let's start from a dummy equipartition of size `initialBound ε l`.
  let t := initialBound ε l
  have htα : t ≤ (univ : Finset α).card :=
    (initialBound_le_bound _ _).trans (by rwa [Finset.card_univ])
```

Finpartition.nonUniforms_mono

Mathlib/Combinatorics/SimpleGraph/Regularity/Uniform.lean#L213-L214

```
theorem nonUniforms_mono {ε ε' : 𝕜} (h : ε ≤ ε') : P.nonUniforms G ε' ⊆ P.nonUniforms G ε
  monotone_filter_right _ fun _ => mt <| SimpleGraph.IsUniform.mono h
```

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 Huisinga
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
- Vision
- Mission
- Tracking Progress
- A Laser-Focused First Year
- Deliverables

Vision

Formal mathematics

Software/Hardware verification

Software development

AI for math and code synthesis

Math and CS education

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

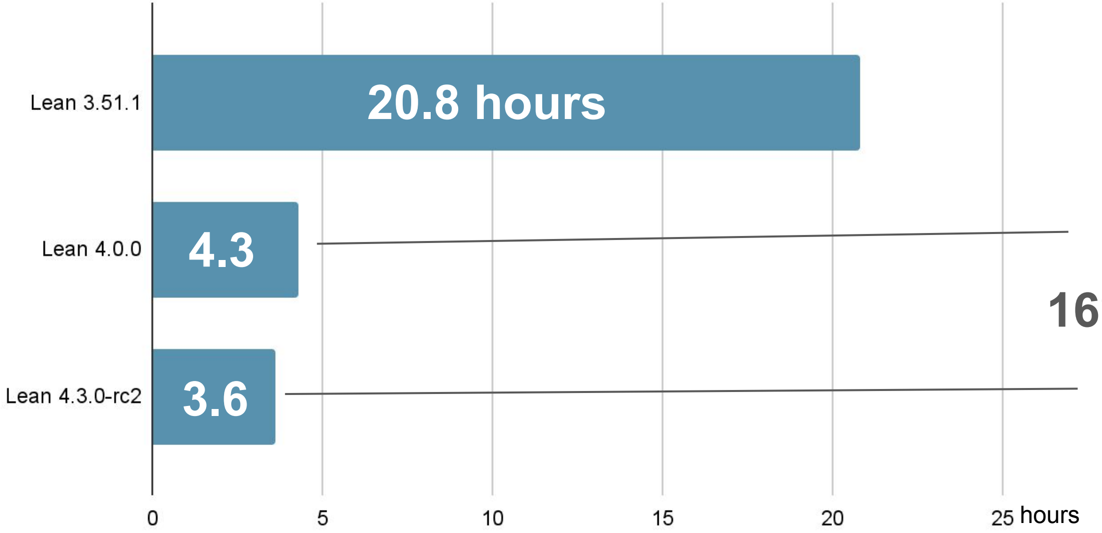
overhead factor

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)



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.

Reference Manual.

Developing Cloud-Build & Reservoir

Robust cloud build support for all Lean packages.

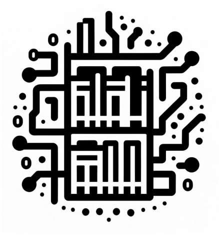
Launch Reservoir: the Lean package repository.

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.

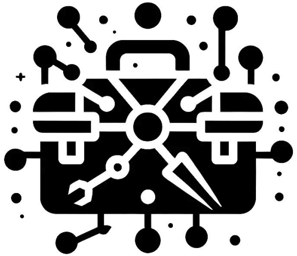
Tooling for Decentralized Work

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.

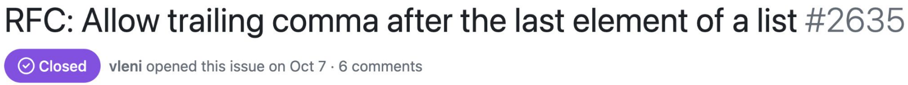
Postponed components: debugger.

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

Closed vleni opened this issue on Oct 7 · 6 comments

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…