

Lean Project Update

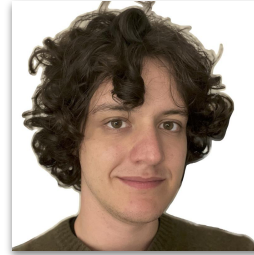
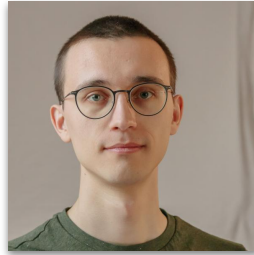
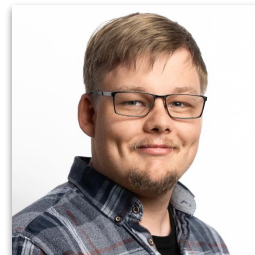
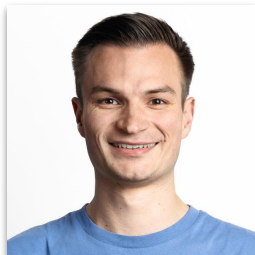
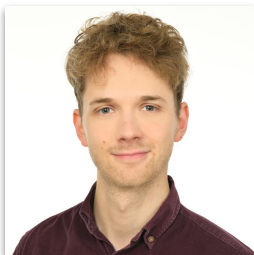
Sebastian Ullrich
Head of Engineering, Lean FRO

Lean Workshop, FLoC 2026

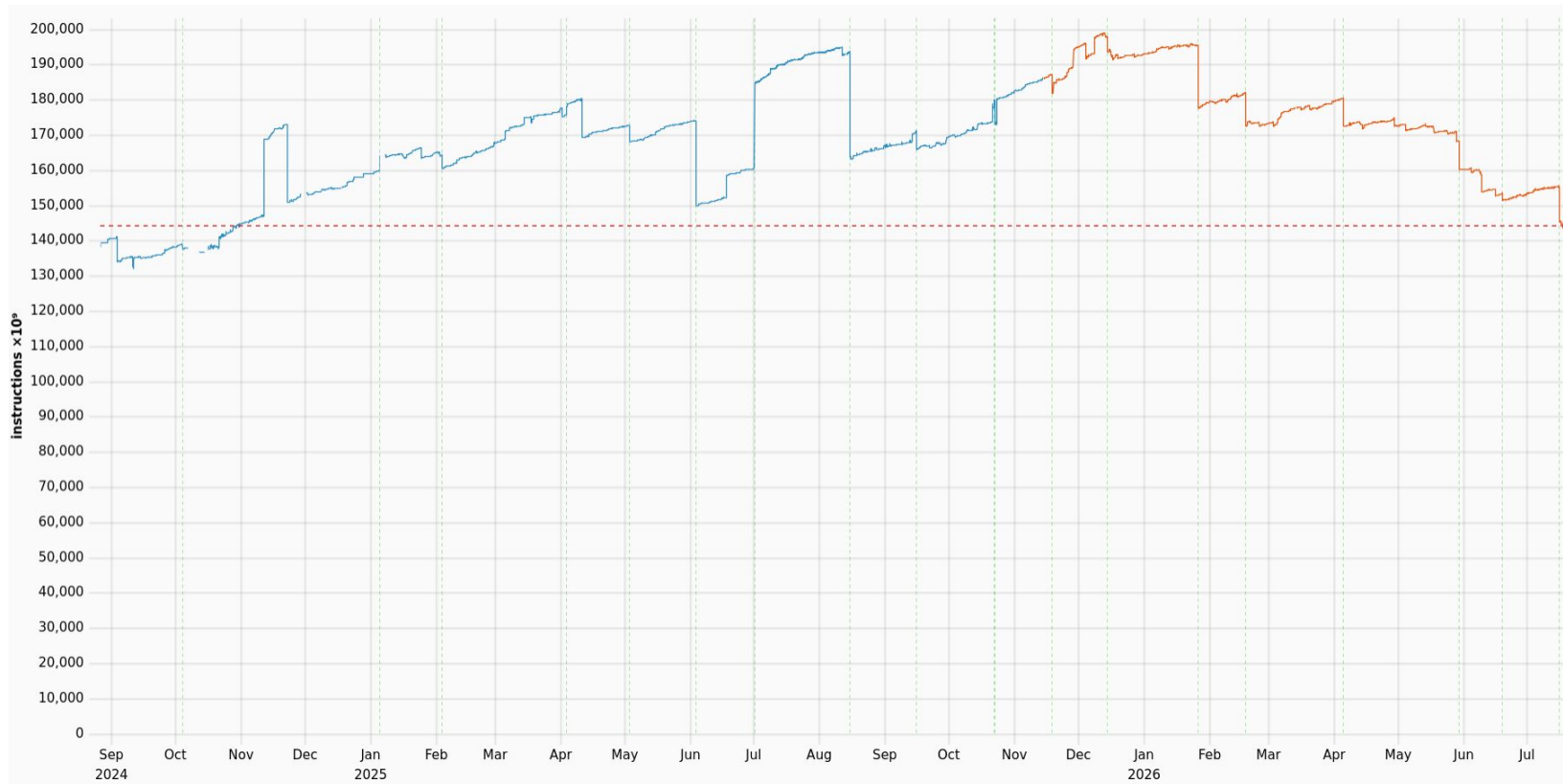
Looking Back

**Highlights of the past 12 months
at the Lean FRO**

The Lean FRO team



Compiler & Performance [Mathlib Radar]





Frontend

Error message index and improvements

Extended **go-to-def**, **call hierarchy**, **"Try this"**, ...

Further **IDE optimizations** (e.g. 3.5x faster **auto-completion**)

Error Explanations

This section provides explanations of errors and warnings that may be generated by Lean when processing a source file. All error names listed below have the Lean package prefix.

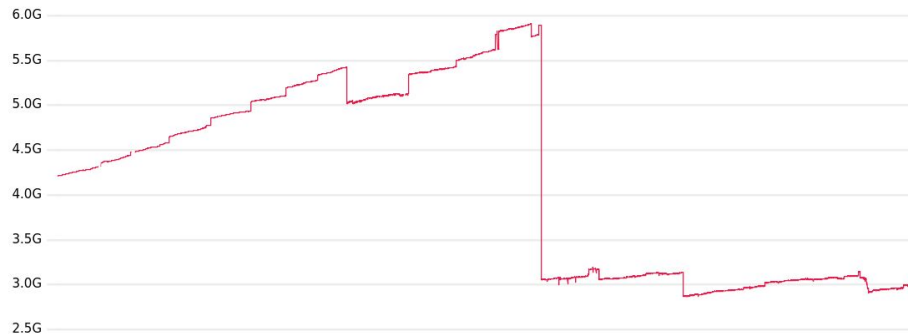
Name	Summary	Severity	Since
ctorResultingTypeMismatch	Resulting type of constructor was not the inductive type being declared.	Error	4.22.0
dependsOnNoncomputable	Declaration depends on noncomputable definitions but is not marked as noncomputable	Error	4.22.0

Language

New, **extensible do notation** with **intrinsic verification** syntax

Module system arrived in Mathlib, halving memory use to 3GB and enabling further refactorings

Defeq transparency overhaul to address Mathlib performance gaps





Standard library

Rebuilt **String** datatype, with slices and dependently-typed positions

Verifiable **Float** type, plus related runtime hardening

New **ranges** and **iterator** API

HTTP server & client

```
def sortNumbers (str : String) : String :=  
  str.split ' '  
    |>.filter (!·.isEmpty)  
    |>.map (numberToNumber[·.copy]!)  
    |>.toList  
    |>.mergeSort  
    |>.map (numbers[·]!)  
    |> String.intercalate " "
```

```
example : sortNumbers "five three" = "three five" := by cbv
```

```
example : sortNumbers "six one two" = "one two six" := by cbv
```

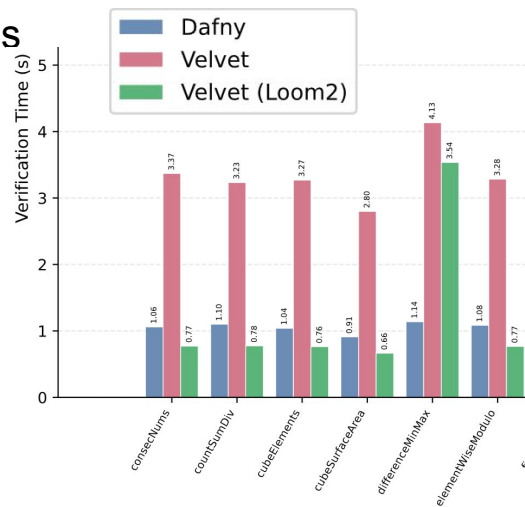
Automation

vcgen: verification condition generator for monadic programs and beyond

cbv: call-by-value evaluation without **native_decide**

grind enhancements, optimizations, and tool-guided curation of annotations

SymM: a new tactic framework for scalable symbolic reasoning
now powers **grind**, **vcgen**, **cbv**, **bv_decide** (2-6x perf), **Velvet**



Tooling

Lake local & global cache moving into production: lean4, downstream, Tau Ceti, ...

New **shake** import minimizer understands module system as well as metaprograms

Disk snapshots can speed up repeated importing and elaboration

Lean Beam prerelease for efficient agent interaction

Verso

New **reference manual** chapters: mvcgen, grind, cbv, do, ...

Verso support in **docstrings**

Literate programming support, used in e.g. Analysis companion

Verso Blueprint, Verso Slides

```
import Analysis.MeasureTheory.Section_1_3_3
import Analysis.MeasureTheory.Section_1_3_4
import Analysis.MeasureTheory.Section_1_3_5

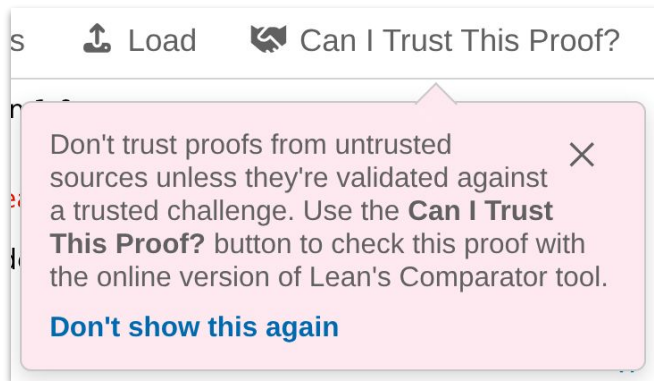
/-!
  The files in this directory contain a formaliza
```

Trust

"Validating a Lean Proof" reference manual chapter

comparator created as the gold standard for safe proof auditing, integrated into live.lean-lang.org

native_decide creates one axiom per invocation



Community Projects

Kernel Arena

Tau Ceti (under Mathlib Initiative's Formal Frontiers umbrella)

Signal Shot (with BAIF et al)



Formal Verification Certificate

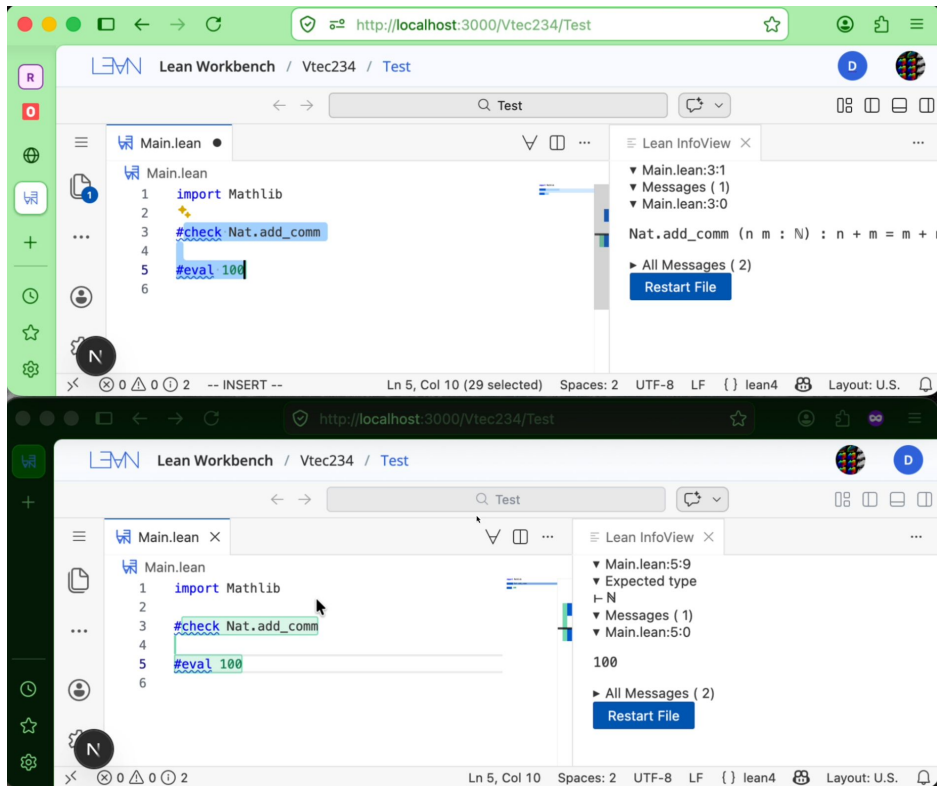
dalek-curve25519, elliptic curve cryptography library

Certified 

Looking Forward

Lean FRO Year 4: Key Projects

Mathematics: Collaborative Lean Workbench





Mathematics et al: scalability

Draft

perf: consolidate all experimental typeclass inference optimisations #14339

Kha wants to merge 22 commits into `master` from `push-nkopukwvwywz`

- ✓ build//instructions : -10.6T (-7.04%)

Large changes (98✓)

- ✓ build/module/Mathlib.Algebra.Algebra.NonUnitalSubalgebra//instructions : -12.1G (-17.65%)
- ✓ build/module/Mathlib.Algebra.Group.Irreducible.Indecomposable//instructions : -13.8G (-31.33%)
- ✓ build/module/Mathlib.Algebra.Module.Torsion.PrimaryComponent//instructions : -29.5G (-44.85%)
- ✓ build/module/Mathlib.Algebra.MonoidAlgebra.PointwiseSMul//instructions : -12.1G (-49.74%)
- ✓ build/module/Mathlib.Algebra.Order.Group.Pointwise.Interval//instructions : -9.0G (-16.99%)
- ✓ build/module/Mathlib.Algebra.Order.Monoid.Canonical.Basic//instructions : -10.5G (-33.88%)
- ✓ build/module/Mathlib.Algebra.Order.ToIntervalMod//instructions : -21.3G (-27.76%)
- ✓ build/module/Mathlib.Algebra.Polynomial.RuleOfSigns//instructions : -19.0G (-34.05%)
- ✓ build/module/Mathlib.Algebra.Ring.CentroidHom//instructions : -9.0G (-26.25%)
- ✓ build/module/Mathlib.Algebra.Star.NonUnitalSubalgebra//instructions : -27.6G (-25.76%)
- ✓ build/module/Mathlib.Algebra.TrivSqZeroExt.Basic//instructions : -12.8G (-17.77%)
- ✓ build/module/Mathlib.AlgebraicGeometry.EllipticCurve.Affine.AddSubMap//instructions : -17.1G (-28.18%)
- ✓ build/module/Mathlib.AlgebraicGeometry.EllipticCurve.Projective.Formula//instructions : -57.1G (-35.08%)

Lean FRO Year 4: Other Key Projects

Program verification: continue **SymM** scalability work, foundational tooling like **vcgen**

Programming: state-of-the-art ecosystem (**formatter**, robust **FFI**, ...)

Compatibility: stabilize Lean components to guarantee backwards compatibility

... look forward to the official roadmap for details!

Looking At You

Questions?