Maintenance of Lean's mathlib and the Liquid Tensor Experiment

Riccardo Brasca

Université Paris Cité
Institut de Mathématiques de Jussieu-Paris Rive Gauche

24th September, 2022

Introduction

I will speak about

- The mathematical library mathlib and how we maintain it
- The *Liquid Tensor Experiment* and similar projects

Mathlib

Mathlib is Lean's mathematical library

It is a large *monolithic* library

It is meant to be used by "standard" mathematicians

It is about classical mathematics by design

Is uses the axiom of choice everywhere

Some statistics (as September 23th):

- 41257 definitions
- 99058 theorems
- 275 contributors

Mathlib "knows" a lot of undergraduate mathematics and some graduate level mathematics

Category theory: category, small category, functor, natural transformation, Yoneda embedding, adjunction, monad, comma category, limits, presheafed space, sheafed space, monoidal category, cartesian closed, abelian category

Numbers: natural number, integer, rational number, continued fraction, real number, extended real number, complex number, p-adic numbers, p-adic integers, hyper-real number

Group theory: group, group morphism, group action, class formula, Burnside lemma, subgroup, subgroup generated by a subset, quotient group, first isomorphism theorem, second isomorphism theorem, third isomorphism theorem, abelianization, free group, presented group, Schreier's lemma, cyclic group, nilpotent group, permutation group of a type, structure of fintely generated abelian groups

Rings: ring, ring morphism, the category of rings, subring, localization, local ring, noetherian ring, ordered ring

Ideals and quotients: ideal of a commutative ring, quotient ring, prime ideal, maximal ideal, Chinese remainder theorem, fractional ideal, first isomorphism theorem for commutative rings

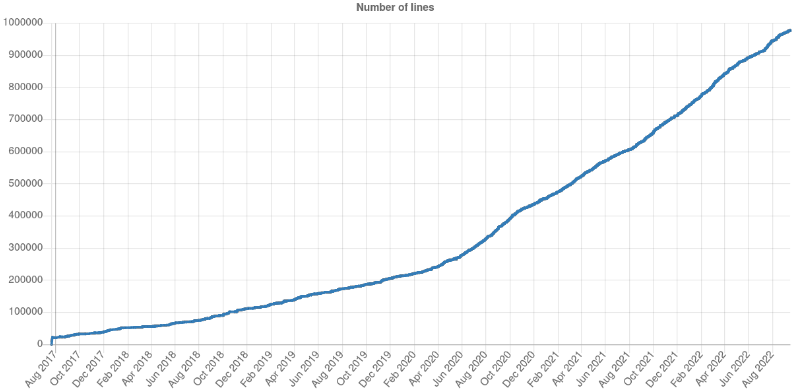
Divisibility in integral domains: irreducible element, coprime element, unique factorisation domain, greatest common divisor, least common multiple, principal ideal domain, Euclidean domain, Euclid's' algorithm, Euler's totient function, Lucas-Lehmer primality test

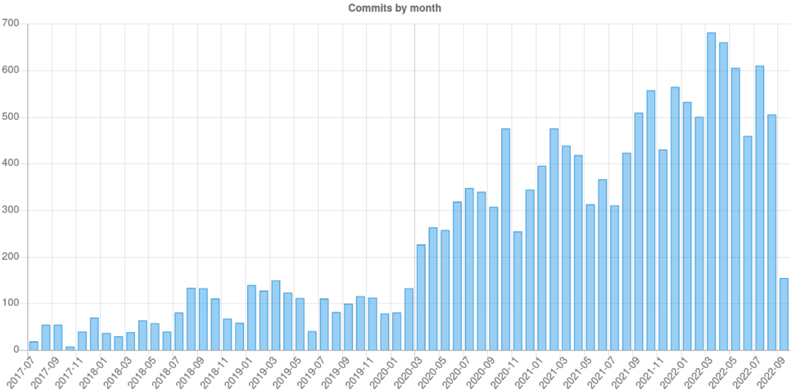
Polynomials and power series: polynomial in one indeterminate, roots of a polynomial, multiplicity, separable polynomial, $K[X]$ is Euclidean, Hilbert basis theorem, $A[X]$ has gcd and lcm if A does, $A[X]$ is a UFD when A is a UFD, irreducible polynomial, Eisenstein's criterion, polynomial in several indeterminates, power series

Algebras over a ring: associative algebra over a commutative ring, the category of algebras over a ring, free algebra of a commutative ring, tensor product of algebras, tensor algebra of a commutative ring, Lie algebra, exterior algebra, Clifford algebra

Field theory: field, characteristic of a ring, characteristic zero, characteristic p, Frobenius morphism, algebraically closed field, existence of algebraic closure of a field $\mathbb{C}$ is algebraically closed, field of fractions of an integral domain, algebraic extension, rupture field, splitting field, perfect closure, Galois correspondence, Abel-Ruffini theorem (one direction)
Homological algebra: chain complex, functorial homology
Number theory: sum of two squares, sum of four squares, quadratic reciprocity, solutions to Pell's equation, Matiyasevič's theorem, arithmetic functions, Bernoulli numbers, Chevalley-Warning theorem, Hensel's lemma (for $\mathbb{Z}_p$), ring of Witt vectors, perfection of a ring
Transcendental numbers: Liouville's theorem on existence of transcendental numbers

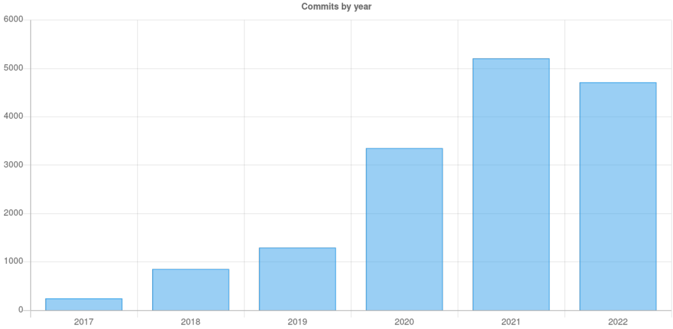
The library grows quickly
Around one million lines of code




Commits by month
700
600
500
400
300
200
100
0
2017-07
2017-09
2017-11
2018-01
2018-03
2018-05
2018-07
2018-09
2018-11
2019-01
2019-03
2019-05
2019-07
2019-09
2019-11
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09

This month in mathlib (in algebra in August 2022)

- proof that a ring morphism is finite if and only if it is integral and of finite type
- proof that localization maps of artinian rings are surjective
- definition of the module of Kähler differentials
- definition of the principal unit group of a valuation subring
- definition of divisible groups
- definition of the normal closure of a fields extension


Commits by year
6000
5000
4000
3000
2000
1000
0
2017
2018
2019
2020
2021
2022

Pull requests

Anyone can contribute, to any part of the library
There are more than 500 opened PRs, around 180 are marked *awaiting-review*

We have PRs about:

- All areas of mathematics
- Tactics
- The core library
- Various editors' extensions
- ...

I will focus on mathematical PRs

Adding a theorem

PR 13585:

```
/-- If ‘K‘ is a ‘p ^ k‘-th cyclotomic extension of
 ‘ℚ‘, then ‘(adjoin ℤ {ζ})‘ is the integral closure
   of ‘ℤ‘ in ‘K‘. -/
lemma
  is_integral_closure_adjoing_singleton_of_prime_pow
  {p : ℕ+} {k : ℕ} {K : Type*} [field K]
  [char_zero K] {ζ : K} [hp : fact (nat.prime ↑p)]
  [hcycl : is_cyclotomic_extension {p ^ k} ℚ K]
  (hζ : is_primitive_root ζ ↑(p ^ k)) :
   is_integral_closure (adjoin ℤ ({ζ} : set K)) ℤ K
```

- 275 lines added
- around 10 comments and suggestions
- one real addition
- merged in two weeks

A refactor. The $\mathbb{N}$-multiplication diamond

Any `add_monoid` has a natural $\mathbb{N}$-multiplication given by repeated addition

```
3 • m = m + m + m
```

For any `semiring` R, the polynomial ring $R[X]$ has a natural R-multiplication given by multiplication of the coefficients

```
r • (1 + X) = r + rX
```

Any `semiring` is an `add_monoid`, so $\mathbb{N}[X]$ has *two* “natural” $\mathbb{N}$-multiplications

These are propositionally equal, but not definitionally equal, in particular we can end up with goals like

```
3 • (1 + X) = 3 • (1 + X)
```

that are not closed by `refl`

Solution (by Sébastien Gouëzel): embed the $\mathbb{N}$-multiplication in the definition of `add_monoid` and a proof that it coincides with repeated addition

From

```
class add_monoid (M : Type u) extends add_semigroup M,
    add_zero_class M
```

To

```
class add_monoid (M : Type u) extends add_semigroup M,
    add_zero_class M :=
(nsmul : ℕ → M → M := nsmul_rec)
(nsmul_zero’ : ∀ x, nsmul 0 x = 0 . try_refl_tac)
(nsmul_succ’ : ∀ (n : ℕ) x, nsmul n.succ x = x +
   nsmul n x . try_refl_tac)
```

- 1010 lines added and 671 removed
- 117 files modified
- a lot of discussion *before* opening the PR
- certain proofs become slower and needed to be fixed
- merged quickly

Future perspectives

Can this way of handling things scale in the future?
Problems:

- We are only 25 maintainers
- The library is getting bigger and bigger
- Proofs are becoming more complex and slower

Partial solutions:

- We are expanding the admin team
- Lean 4 will help

Large refactors will be more and more difficult

Is it reasonable to split mathlib in *basic mathlib* and *advanced mathlib*?

Currently there three layers in a project using mathlib:

- The core library, Lean itself
- Mathlib
- Your own project

The Liquid Tensor Experiment

In December 2020, Peter Scholze asked for a formalization of the following theorem Dustin Clausen and himself

Theorem

Let $0 < p' < p \leq 1$ be real numbers, S a profinite set and V a p-Banach space. We have

$$\mathrm{Ext}^1_{\mathrm{Cond(Ab)}}(\mathcal{M}_{p'}(S), V) = 0$$

This is cutting edge mathematics. It is a difficult theorem about complicated objects, that uses all sort of prerequisites

Why do I want a formalization?

- *... I think the theorem is of utmost foundational importance, so being 99.9 % sure is not enough*
- *... As it will be used as a black box, a mistake in this proof could remain uncaught*
- *... 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*
- *From what I hear, it sounds like the goal is not completely out of reach. . . . If achieved, it would be a strong signal that a computer verification of current research in very abstract mathematics has become possible. I'll certainly be excited to watch any progress*

The project was complete on July 14th, 2022 by a team of around 15 people

Joahn Commelin

Riccardo Brasca

Mario Carneiro

Patrick Massot

Scott Morrison

Joël Riou

Andrew Yang

Adam Topaz

Kevin Buzzard

Heather Macbeth

Bhavik Mehta

Filippo A.E. Nuccio

Damiano Testa

many others

LTE statistics

More than 77000 lines of codes
6185 commits

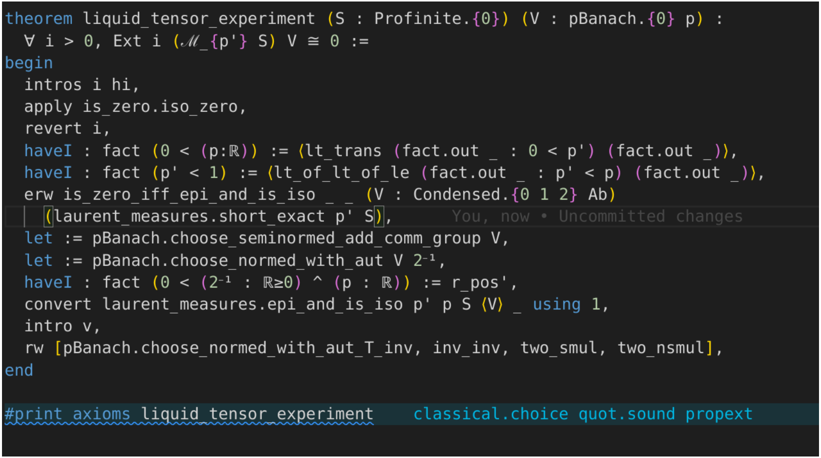
24780 Lean declarations to get to the final result :

- 5012 definitions, 3571 instancesand 15390 theorems
- 18173 in mathlib and 1082 in the core library

5525 declarations exist only in the project (these are slowly being integrated in mathlib)

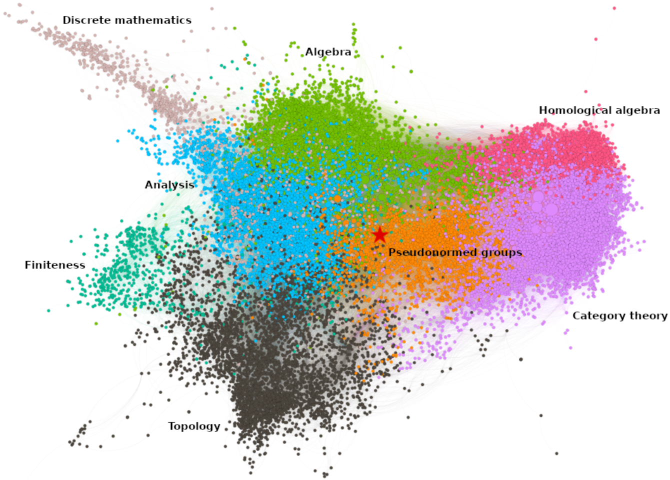
The final theorem

```
variables (p’ p : ℝ≥0) [fact (0 < p’)]
  [fact (p’ < p)] [fact (p ≤ 1)]

theorem liquid_tensor_experiment (S : Profinite.{0})
  (V : pBanach.{0} p) :
  ∀ i > 0, Ext i (ℳ_{p’} S) V ≅ 0 :=
```



```
theorem liquid_tensor_experiment (S : Profinite.{0}) (V : pBanach.{0} p) :
  ∀ i > 0, Ext i (ℳ_{p'} S) V ≅ 0 :=
begin
  intros i hi,
  apply is_zero.iso_zero,
  revert i,
  haveI : fact (0 < (p:ℝ)) := ⟨lt_trans (fact.out _ : 0 < p') (fact.out _)⟩,
  haveI : fact (p' < 1) := ⟨lt_of_lt_of_le (fact.out _ : p' < p) (fact.out _)⟩,
  erw is_zero_iff_epi_and_is_iso _ _ (V : Condensed.{0 1 2} Ab)
  │ (laurent_measures.short_exact p' S),      You, now • Uncommitted changes
  let := pBanach.choose_seminormed_add_comm_group V,
  let := pBanach.choose_normed_with_aut V 2⁻¹,
  haveI : fact (0 < (2⁻¹ : ℝ≥0) ^ (p : ℝ)) := r_pos',
  convert laurent_measures.epi_and_is_iso p' p S ⟨V⟩ _ using 1,
  intro v,
  rw [pBanach.choose_normed_with_aut_T_inv, inv_inv, two_smul, two_nsmul],
end

#print axioms liquid_tensor_experiment    classical.choice quot.sound propext
```

The dependency graph



The `for_mathlib` folder

We have a huge `for_mathlib` folder: more than 32000 lines of code

We tried at beginning to PR as much as possible into mathlib

Impossible to do while still working on the project

Maybe is it suitable for an *advanced mathlib*?

Mathlib bumps

Mathlib changes quickly, but the LTE uses a fixed version

If we don't upgrade mathlib often, the code will rot

Bumping mathlib after six months has been very difficult. We did it in several steps

What to do in the future?

The FLT-regular project

The goal is to formalize the proof of the following

Theorem

Let p be a regular prime. Then the equation

$$x^p + y^p = z^p$$

has no nontrivial solutions in $\mathbb{Z}$.

Much smaller than the LTE

Mathlib bumps are easy
We are able to include our results in mathlib almost in real time

Thank you for your attention!