mathlib:
Lean's mathematical library

Johannes Hölzl

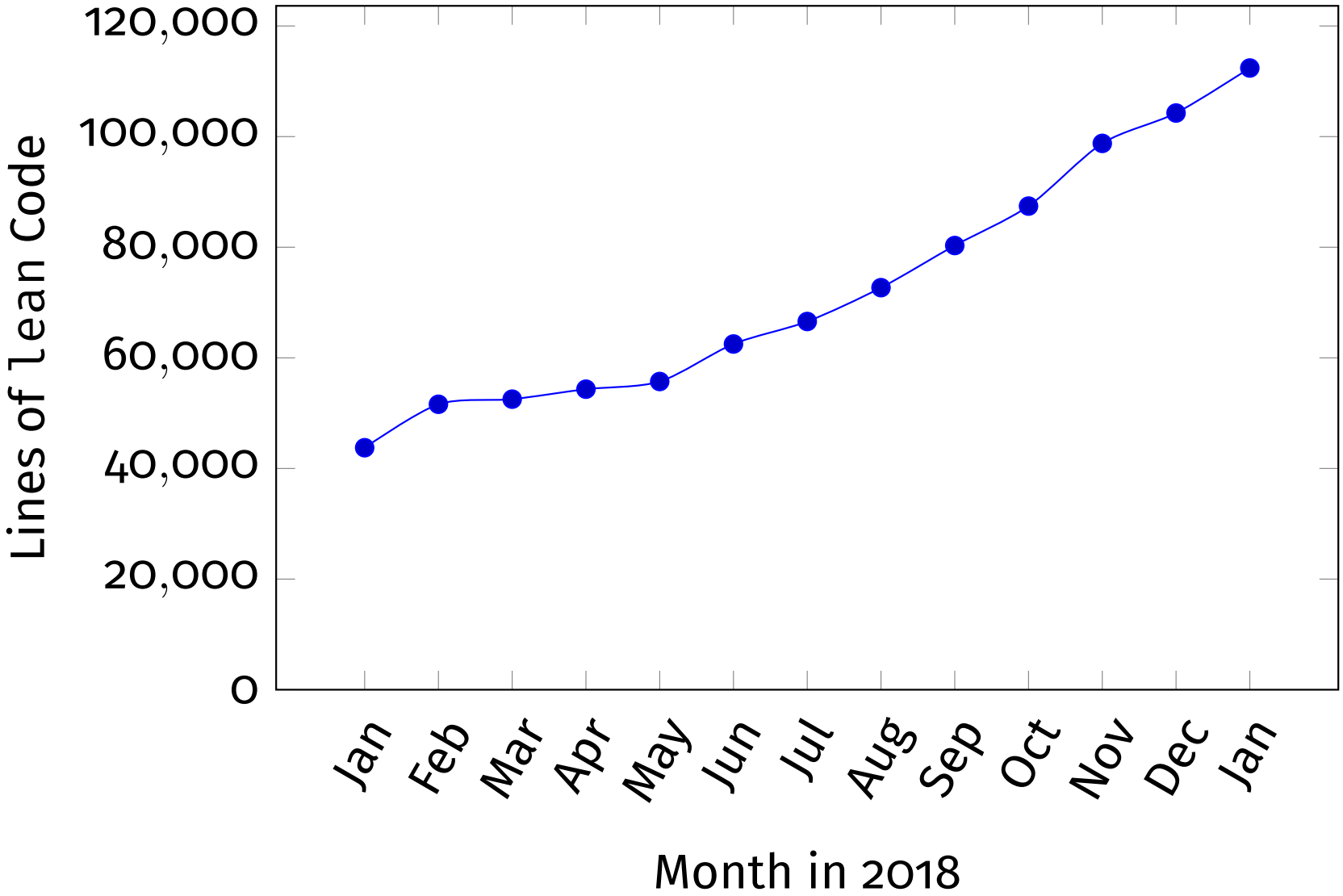
VU Amsterdam

Lean Together 2019
Amsterdam

Introduction — what is `mathlib`

- (classical) mathematical library for Lean
  classical – linear algebra, number theory, analysis, ...
  classical – using choice and LEM
- Formerly distributed with Lean itself
  Leo wanted more flexibility
- Some (current) topics:
  (Linear) Algebra, Analysis, Set Theory, Number Theory, ...
- Goal: be comparable to
  Coq's mathematical components,
  Isabelle's `HOL-Analysis`

Development



Repository starts July 2017 with 22.000 LoC

- Organization
- Constructions
- Tactics
- Theories

Organization

Basic parts

`tactic`	`mathlib`'s tactics
`logic`	Lemmas about logical connectices; classical function inverse; Schröder-Bernstein; ...
`order`	Lemmas about orders; further order and lattice type classes
`algebra`	Lemmas about algebraic structures; more algebraic and order type classes, upto modules
`category`	`Type` category (Haskell style): Functor, Applicative, Monads, Traversable
`data`	Generic (mostly computable) types

Theories

set_theory	Ordinals, Cardinals, a model of ZFC
category_theory	Category theory: functors, natural trans., limits, ...
linear_algebra	Modules, Vector spaces, Tensor products, Dimensions, ...
analysis	Analysis, Topology, Measure theory, ...
group_theory	Free (abelian) groups, Order of an element, ...
ring_theory	Principal ideal domains, ...
field_theory	Finite fields, perfect closure, ...
computability	Turing machines etc.
number_theory	Diophantine equations, Pell's equation (?)

data – Generic (mostly computable) types

nat $\mathbb{N}$, gcd, mod, prime, $\binom{n}{p}$, ...

int $\mathbb{Z}$ (as datatype, not quotient)

rat $\mathbb{Q}$ (as datatype, not quotient)

real $\mathbb{R}$ (as quotient over Cauchy sequences)

equiv Isomorphism between Type

fin fin $n = \{0, \ldots, n-1\}$

set set $\alpha := \alpha \to$ Prop

list list α

multiset multiset $\alpha :=$ list $\alpha/_{\text{perm}}$

finset finset $\alpha := \{m :$ multiset $\alpha \mid$ nodup $m\}$

...

Constructions

Constructions – Ordered groups and rings

`nonneg_comm_group`, `nonneg_ring` and `linear_nonneg_ring`

- Helper type classes to construct `ordered_group` and `ordered_ring`
- Specify `nonneg` : $\alpha \to$ `Prop` and (optionally) `pos` : $\alpha \to$ `Prop`
- Constructs the order on the group / ring
- Proves the relation btw order and algebraic operations

Constructions – Adjoin $\top$, $\bot$ or 0

`with_top`, `with_bot` and `with_zero` := `option`

- Ext. nonnneg. reals: $\mathbb{R}_{\geq 0} \uplus \infty$, extended nats $\overline{\mathbb{N}} = \mathbb{N} \uplus \infty$
- Multisets with infinity (e.g. sets of factors, ∞ to represent 0)
- Example instances:
  - `partial_order` $\alpha \to$ `partial_order` (`with_top` α)
  - `add_monoid` $\alpha \to$ `add_monoid` (`with_top` α)
  - `canonically_ordered_comm_semiring` $\alpha \to$ `canonically_ordered_comm_semiring` (`with_top` α)

Structures as Functors and Complete Lattices

There are many `set` like structures:

`filter`, `topology`, `uniformity`, `submodule`, `measurable`

- Form complete lattices, e.g. `topological_space` α
- (Co)functors, e.g. `map` : $(\alpha \to \beta) \to (\texttt{filter}\ \alpha \to \texttt{filter}\ \beta)$
  Exception: `uniformity` only cofunctor
- Order + `comap` is usually the morphism definition
- Used as light-weight category theory
  (simpler to setup, base constructions require less setup)

Example: Topologies

```
structure topo (α : Type) :=
(is_open : set α → Prop) ... -- univ, ∩, ⋃

cont : (α → β) → topo α → topo β → Prop
cont f T₁ T₂ :↔
  ∀s, is_open T₂ s → is_open T₁ (f⁻¹'s)

(≤) : topo α → topo α → Prop
T₁ ≤ T₂ :↔ ∀s, is_open T₁ s → is_open T₂ s

map : (α → β) → (topo α → topo β)
is_open (map T f) s :↔ is_open T (f ⁻¹' s)

cont f T₁ T₂ :↔ T₂ ≤ T₁.map f
```

Example: Generating Topologies

```
generate : set (set α) → topo α
∀s∈g, is_open (generate g) s
∀T, (∀s∈g, is_open T s) → generate g ≤ T

map : (α → β) → (topo α → topo β)
is_open (map T f) s :↔ is_open T (f ⁻¹' s)

comap : (α → β) → (topo β → topo α)
is_open (comap T f) s :↔
  ∃t, is_open T t ∧ s = f⁻¹'t
```

Example: Product of Topologies

What do we want:

```
_×_ : topo α → topo β → topo (α × β)
cont π₁ (T₁ × T₂) T₁
cont π₂ (T₁ × T₂) T₂
cont (π₁ ∘ f) T T₁ → cont (π₂ ∘ f) T T₂ →
  cont f T (T₁ × T₂)
```

Traditional definition:

```
T₁ × T₂ =
  generate {s × t | is_open T₁ s ∧ is_open T₂ t}
```

Generic definition (set, filter, topology, σ-algebra):

```
T₁ × T₂ = comap π₁ T₁ ⊔ comap π₂ T₂
```

Example: Lattice Structure on Topologies

Supremum as generating by the union:

```
T₁ ⊔ T₂ =
  generate {s | is_open T₁ s ∨ is_open T₂ t}
```

Problem: Proof that it is a supremum.

Define directly complete lattice over topologies:

$$\bigsqcap \mathscr{T} = \bigcap_{T \in \mathscr{T}} \left\{ s \middle| \mathit{is\_open}\ T\ s \right\}$$

Problem: we might want different def eq for $\top$, $\sqcap$, and $\bigsqcap$.

Solution: Use Galois insertion (extending a Galois connection) to get the complete lattice structure

Constructions

$\mathbb{N}$ $T_{\mathbb{N}} := \top$

Indiscrete $\bot$

Sum $T_\alpha \oplus T_\beta := map\ \iota_1\ T_\alpha \sqcap map\ \iota_2\ T_\beta$

$$\stackrel{\text{def}}{=} \{s | open\ (\iota_1^{-1}[s]) \wedge open\ (\iota_2^{-1}[s])\}$$

Product $T_\alpha \times T_\beta := comap\ \pi_1\ T_\alpha \sqcup comap\ \pi_2\ T_\beta$

Pi $\prod_i T_{\alpha_i} := \bigsqcup_i comap\ (\lambda \omega, \omega\ i)\ T_{\alpha_i}$

Sigma $\Sigma_i T_{\alpha_i} := \bigsqcap_i map\ (\lambda a, (i, a))\ T_{\alpha_i}$

List (not done yet)

$\texttt{list}\ T_\alpha := lfp\ (\lambda T,\ T_() \oplus T_\alpha \times T)$

$\texttt{list}\ T_\alpha := gfp\ (\lambda T,\ T_() \oplus T_\alpha \times T)$

Tactics and Commands

Theories

mathlib

- Basic (computable) data
- Type class hierarchies:

  Orders orders, lattices
  Algebraic (commutative) groups, rings, fields
  Spaces measurable, topological, uniform, metric
- Cardinals & ordinals
- Analysis: topology, measure theory, …
- Linear algebra
- Group / ring / field theory
- Number theory

Cardinals and Ordinals

Definition (Equivalence, Cardinals and Ordinals)

$$\texttt{structure}\ \alpha \simeq \beta :=$$
$$(f : \alpha \to \beta)\ (g : \beta \to \alpha)\ (f\_g : f \circ g = id)\ (g\_f : g \circ f = id)$$
$$\texttt{cardinal}_u \ : \ \texttt{Type}_{u+1} \ := \ \texttt{Type}_u/_{\simeq}$$
$$\texttt{ordinal}_u \ : \ \texttt{Type}_{u+1} \ := \ \texttt{Well\_order}_u/_{\simeq_{ord}}$$

- Well-ordered, semiring, etc...
- Semiring structure of `cardinal` from $\simeq$ constructions
- $\kappa + \kappa = \kappa = \kappa * \kappa$ (for $\kappa \geq \omega$)
- Existence of inaccessible cardinals (i.e. in the next universe)
- **Application:** Dimension of subspaces

p-adic numbers $\mathbb{Q}_p$ by Rob Y. Lewis

- Cauchy construction of $\mathbb{Q}$ "in the other direction"
- Interesting result: Hensel's lemma (see Rob's talk)

Quadratic Reciprocity by Chris Hughes ($\sim$ 1300 lines patch)

```
theorem quadratic_reciprocity
  (hp : prime p) (hq : prime q) (hpq : p ≠ q)
  (hp1 : p % 2 = 1) (hq1 : q % 2 = 1) :
  legendre p q hq * legendre q p hp =
    (-1) ^ (p/2 * q/2)
```

Analysis

Filter generalizes limits (derived from Isabelle/HOL)

Topology nhds filter, open & closed & compact sets, interior, closure

Uniformity complete, totally bounded, completion
`compact` $\leftrightarrow$ `complete` & `totally bounded`

Metric instance of uniformities

Norm only rudimentary (no deriviatives yet...)

Measure Lebesgue measure etc...

Analysis: Constructing Reals

Construct $\mathbb{R}$ using completion of $\mathbb{Q}$

- For foundational reasons metric completion is not possible (alt: $\alpha \to \alpha \to \mathbb{Q} \to$ `Prop`, c.f. Krebbers & Spitters)
- Formalize uniform spaces (using the filter library!)
- Use completion on the uniform space $\mathbb{Q}$
- Is it worth it?
  Mario went back to Cauchy sequences...
- Anyway: $\mathbb{R}$ as order & topologically complete field

(Outer) Measures

```
structure outer_measure (α : Type*) :=
(μ : set α → ennreal) ...

structure measure (α) [measurable_space α]
  extends outer_measure α := ...
```

- Outer measures provide natural totalization of measures
- Carathéodory's extension theorem
- Lebesgue Measure
- Completion measurable space

Probability Theory

```
def pmf (α) := { f : α → nnreal // is_sum f 1 }

def pure (a : α) : pmf α :=
⟨λa', if a' = a then 1 else 0, is_sum_ite _ _⟩

def bind (p : pmf α) (f : α → pmf β) : pmf β :=
⟨λb, (∑a, p a * f a b), ...⟩
```

- prove rules of the Giry monad
- generality of $\sum$ helps!

Analysis: Infinite Sum

Definition (Infinite sum)

$$\sum_{i:\iota} f\ i \quad = \quad \lim_{s \to \texttt{at\_top}} \sum_{i \in s} f\ i$$

Assuming: $f : \iota \to \alpha$, [topological_add_monoid α]

- at_top : filter (finset ι) finite sets approaching univ : set ι
- $\mathbb{R}$, normed vector spaces, ennreal, $\mathbb{Q}$, $\mathbb{N}$, $\mathbb{Z}$, ...
- $\sum_b \sum_c f\ (b, c) = \sum_{(b,c)} f\ (b, c)$ — α regular
- $\sum_{n:\mathbb{N}} f\ n = \lim_{i \to \infty} \sum_{n=0}^{i} f\ n$

Definition (Module)

```
class module
  (α : out_param (Type u)) (β : Type v)
  [out_param (ring α)] [add_comm_group β]
  extends semimodule α β
```

Constr.: Subspace, Linear maps, Quotient, Product, Tensor Product

Dim.: `dim (β) [vector_space α β] : cardinal`

Laws:

$$\frac{dom(f)}{ker(f)} \simeq_\ell im(f) \qquad \frac{s}{s \cap t} \simeq_\ell \frac{s \oplus t}{t}$$

$$R \oplus M \simeq_\ell M \quad M \oplus N \simeq_\ell N \oplus M \quad M \oplus (N \oplus L) \simeq_\ell (M \oplus N) \oplus L$$