A formalization of Dedekind domains and class groups of global fields

Anne Baanen
Sander R. Dahmen
Ashvni Narayanan
Filippo A. E. Nuccio

Introduction

Our project is the first formalization of several essential notions of algebraic number theory, in the Lean 3 prover as part of mathlib.

Goal: lay a useful foundation for theory-building.

mathlib is a community-driven project to build a tightly-integrated library of formalized mathematics.
Developing with mathlib means updating your code regularly in exchange for frequent new results and improvements.

Background: global fields

A number field is finite dimensional as a $\mathbb{Q}$-vector space, of the form $\mathbb{Q}(\alpha)$ for some algebraic α.

Background: global fields

A number field is finite dimensional as a $\mathbb{Q}$-vector space, of the form $\mathbb{Q}(\alpha)$ for some algebraic α.

Each number field K contains a ring of integers $\mathcal{O}_K$ mirroring the way $\mathbb{Q}$ contains $\mathbb{Z}$.
Example: the Gaussian integers $\mathbb{Z}[i]$ inside $\mathbb{Q}(i)$.

Background: global fields

A number field is finite dimensional as a $\mathbb{Q}$-vector space, of the form $\mathbb{Q}(\alpha)$ for some algebraic α.

Each number field K contains a ring of integers $\mathcal{O}_K$ mirroring the way $\mathbb{Q}$ contains $\mathbb{Z}$.
Example: the Gaussian integers $\mathbb{Z}[i]$ inside $\mathbb{Q}(i)$.

A global field is either a number field or a function field: finite extension of a field of rational functions $\mathbb{F}_q(t)$.

Function fields also have a ring of integers, mirroring $\mathbb{F}_q[t] \subset \mathbb{F}_q(t)$.

Background: global fields

A number field is finite dimensional as a $\mathbb{Q}$-vector space, of the form $\mathbb{Q}(\alpha)$ for some algebraic α.

Each number field K contains a ring of integers $\mathcal{O}_K$ mirroring the way $\mathbb{Q}$ contains $\mathbb{Z}$.
Example: the Gaussian integers $\mathbb{Z}[i]$ inside $\mathbb{Q}(i)$.

A global field is either a number field or a function field: finite extension of a field of rational functions $\mathbb{F}_q(t)$.

Function fields also have a ring of integers, mirroring $\mathbb{F}_q[t] \subset \mathbb{F}_q(t)$.

Theorem: rings of integers are Dedekind domains.

Background: class number

Fractional ideals extend (integral) ideals with division by scalars: a fractional ideal is of the form $\frac{1}{b}I$ (with $b \neq 0$).

Fractional ideals are not automatically invertible!

Fractional ideals extend (integral) ideals with division by scalars:
a fractional ideal is of the form $\frac{1}{b}I$ (with $b \neq 0$).
Fractional ideals are not automatically invertible!

Theorem: Dedekind domain $\iff$ fractional ideals $\neq 0$ are invertible.

Background: class number

Fractional ideals extend (integral) ideals with division by scalars: a fractional ideal is of the form $\frac{1}{b}I$ (with $b \neq 0$).
Fractional ideals are not automatically invertible!

Theorem: Dedekind domain $\iff$ fractional ideals $\neq 0$ are invertible.

Principal fractional ideals $\left\langle \frac{a}{b} \right\rangle = \frac{a}{b}\mathcal{O}_K$ for $\frac{a}{b} \in K$ form a subgroup of the fractional ideals; the quotient is the class group $\mathcal{C}l_{\mathcal{O}_K}$.

Theorem: if $\mathcal{O}_K$ is a ring of integers, $\mathcal{C}l_{\mathcal{O}_K}$ is a finite abelian group.
The class number of K is the cardinality of $\mathcal{C}l_{\mathcal{O}_K}$.

Background: class number

Fractional ideals extend (integral) ideals with division by scalars:
a fractional ideal is of the form $\frac{1}{b}I$ (with $b \neq 0$).
Fractional ideals are not automatically invertible!

Theorem: Dedekind domain $\iff$ fractional ideals $\neq 0$ are invertible.

Principal fractional ideals $\left\langle \frac{a}{b} \right\rangle = \frac{a}{b}\mathcal{O}_K$ for $\frac{a}{b} \in K$ form a subgroup of the fractional ideals; the quotient is the class group $\mathcal{C}l_{\mathcal{O}_K}$.

Theorem: if $\mathcal{O}_K$ is a ring of integers, $\mathcal{C}l_{\mathcal{O}_K}$ is a finite abelian group.
The class number of K is the cardinality of $\mathcal{C}l_{\mathcal{O}_K}$.

Theorem: A Dedekind domain is a UFD $\iff$ it is a PID
$\iff$ $\mathcal{C}l_{\mathcal{O}_K}$ is trivial $\iff$ class number of $K = 1$.

Number fields; field extensions

mathlib typically uses typeclasses for algebraic structures, e.g.

```
class is_number_field (K : Type*) [field K] : Prop :=
[cz : char_zero K] [fd : finite_dimensional ℚ K]
```

Typeclass inference automates the implications `char_zero K` → `algebra ℚ K` → `module ℚ K` required for `finite_dimensional ℚ K`.

Number fields; field extensions

mathlib typically uses typeclasses for algebraic structures, e.g.

```
class is_number_field (K : Type*) [field K] : Prop :=
[cz : char_zero K] [fd : finite_dimensional ℚ K]
```

Typeclass inference automates the implications `char_zero K` → `algebra ℚ K` → `module ℚ K` required for `finite_dimensional ℚ K`.

A field extension L/K is represented in mathlib by an instance `[algebra K L]` giving the canonical inclusion map `algebra_map K L`.

Number fields; field extensions

mathlib typically uses typeclasses for algebraic structures, e.g.

```
class is_number_field (K : Type*) [field K] : Prop :=
[cz : char_zero K] [fd : finite_dimensional ℚ K]
```

Typeclass inference automates the implications `char_zero K` → `algebra ℚ K` → `module ℚ K` required for `finite_dimensional ℚ K`.

A field extension L/K is represented in mathlib by an instance `[algebra K L]` giving the canonical inclusion map `algebra_map K L`.

A tower $L/K/F$ is given by inclusions `[algebra F K]` `[algebra K L]` `[algebra F L]` and an instance `[is_scalar_tower F K L]` stating the maps commute.
Coherence proof obligations are automated through typeclass search.

Representing $\mathbb{Q}(\alpha)$

Number fields have the form $\mathbb{Q}(\alpha)$, where α is algebraic:
the minimal polynomial $f_\alpha \in \mathbb{Q}[x]$ is irreducible and $f_\alpha(\alpha) = 0$.

Many constructions of $\mathbb{Q}(\alpha)$: subtype of $\mathbb{C}$, quotient type $\mathbb{Q}[x]/f_\alpha$, ...
These are isomorphic but not equal: how do we reason uniformly?

Representing $\mathbb{Q}(\alpha)$

Number fields have the form $\mathbb{Q}(\alpha)$, where α is algebraic:
the minimal polynomial $f_\alpha \in \mathbb{Q}[x]$ is irreducible and $f_\alpha(\alpha) = 0$.

Many constructions of $\mathbb{Q}(\alpha)$: subtype of $\mathbb{C}$, quotient type $\mathbb{Q}[x]/f_\alpha$, ...
These are isomorphic but not equal: how do we reason uniformly?

We used the power basis: $\mathbb{Q}(\alpha)$ has a $\mathbb{Q}$-basis $1, \alpha, \ldots, \alpha^{n-1}$.
Theorem: each construction of $\mathbb{Q}(\alpha)$ is a field with power basis generated by α.

Dedekind domains

We defined Dedekind domains as integral domains R with an `is_dedekind_domain R` instance:

```
class is_dedekind_domain (R : Type*) [integral_domain R] :
    Prop :=
(to_is_noetherian_ring : is_noetherian_ring R)
(dimension_le_one : ∀ (P : ideal R), P ≠ 0 →
  is_prime P → is_maximal P)
(is_integrally_closed :
  integral_closure R (fraction_ring R) = ⊥)
```

Fractional ideals

We formalized fractional ideals of R as a subtype:
R-submodules I of $\mathrm{Frac}(R)$ such that $\exists a : R, aI \subseteq R$.

Fractional ideals have a semiring structure (like submodules):

- $0 = \{0\}$
- $1 = \{x \mid x \in R\}$
- $I + J = \{x + y \mid x \in I, y \in J\}$
- $I * J$ is generated by $x * y$, $x \in I$, $y \in J$
- $x \in I/J \iff \forall y \in J, x * y \in I$

Fractional ideals

We formalized fractional ideals of R as a subtype:
R-submodules I of $\mathrm{Frac}(R)$ such that $\exists a : R, aI \subseteq R$.

Fractional ideals have a semiring structure (like submodules):

- $0 = \{0\}$
- $1 = \{x \mid x \in R\}$
- $I + J = \{x + y \mid x \in I, y \in J\}$
- $I * J$ is generated by $x * y$, $x \in I$, $y \in J$
- $x \in I/J \iff \forall y \in J, x * y \in I$

Theorem: $I * (1/I) = 1$ for all $I \neq 0$ iff R is a Dedekind domain.

Re-defining division

The `group_with_zero` typeclass used to define $x/y := x * y^{-1}$.
For fractional ideals we want to define $I^{-1} := 1/I$. How to deal with this circularity?

Re-defining division

The `group_with_zero` typeclass used to define $x/y := x * y^{-1}$.
For fractional ideals we want to define $I^{-1} := 1/I$. How to deal with this circularity?

Solution: turn defeq into propositional equality by adding a new operation $(/)$ to `group` (`_with_zero`) and an axiom $x/y = x * y^{-1}$.

This required about 500 changes in mathlib.

Dedekind domain theorems

Theorem: $I * (1/I) = 1$ for all $I \neq 0$ iff R is a Dedekind domain.

Difficulties:

- Showing $x \in I * J$ implies $x = \sum_k y_k z_k$ for $y_k \in I$, $z_k \in J$.
- Coercions: I can be an integral ideal or set $\subseteq R$ or a fractional ideal or submodule or set $\subseteq \mathsf{Frac}(R)$.

Dedekind domain theorems

Theorem: $I * (1/I) = 1$ for all $I \neq 0$ iff R is a Dedekind domain.

Difficulties:

- Showing $x \in I * J$ implies $x = \sum_k y_k z_k$ for $y_k \in I$, $z_k \in J$.
- Coercions: I can be an integral ideal or set $\subseteq R$ or a fractional ideal or submodule or set $\subseteq \mathrm{Frac}(R)$.

Theorem: Principal ideal domains are Dedekind domains.
Corollary: $\mathbb{Z}$ and $\mathbb{F}_q[t]$ are Dedekind domains.

Rings of integers are Dedekind domains

Theorem: The integral closure of a Dedekind domain R in a finite separable extension $K/\operatorname{Frac}(R)$ is a Dedekind domain.
Corollary: Rings of integers, closures of PIDs in finite separable extensions, are Dedekind domains.

Rings of integers are Dedekind domains

Theorem: The integral closure of a Dedekind domain R in a finite separable extension $K/\operatorname{Frac}(R)$ is a Dedekind domain.
Corollary: Rings of integers, closures of PIDs in finite separable extensions, are Dedekind domains.

"Accidental" collaboration with the Berkeley Galois theory group:

- We defined `intermediate_field`
- They used it to formalize the primitive element theorem
- We used that to show finite separable field extensions have a power basis
- They used that to show conjugate roots of α correspond to images $\sigma(\alpha)$ for $\sigma : F(\alpha) \to K$ fixing F
- We used that to show the trace form is nondegenerate

Finiteness of the class group

Theorem: If K is a global field, the class group of $\mathcal{O}_K$ is finite.

Typical proofs use Minkowski's lattice point theorem for number fields, extending this to function fields is complicated.

Finiteness of the class group

Theorem: If K is a global field, the class group of $\mathcal{O}_K$ is finite.

Typical proofs use Minkowski's lattice point theorem for number fields, extending this to function fields is complicated.

We introduced a new notion of admissible absolute value,
and proved if $\mathrm{abs} : R \to \mathbb{Z}$ is admissible,
this intermediate step in the classical proof holds:

```
theorem exists_mem_finset_approx'
  (a b : integral_closure R L) :=
  ∃ (q : integral_closure R L) (r ∈ finset_approx L f abs),
  abs_norm f abs (r • a - q * b) < abs_norm f abs b
```

Finiteness of the class group

After formalizing the remainder of the classical proof, it remained to find admissible absolute values.

For $\mathbb{Z}$, the usual absolute value is admissible.
For $\mathbb{F}_q[t]$, $|f|_{\mathrm{deg}} := q^{\deg f}$ is admissible.

Finiteness of the class group

After formalizing the remainder of the classical proof, it remained to find admissible absolute values.

For $\mathbb{Z}$, the usual absolute value is admissible.
For $\mathbb{F}_q[t]$, $|f|_{\deg} := q^{\deg f}$ is admissible.

```
def class_group (f : fraction_map R K) :=
quotient_group.quotient (to_principal_ideal f).range

noncomputable def number_field.class_number (K : Type*)
  [field K] [is_number_field K] : ℕ :=
card (class_group (ring_of_integers.fraction_map K))
```

Conclusions

Total contribution: $\pm$ 5000 lines of project-specific code, $\pm$ 2500 lines background work.
(Difficult to quantify exactly due to tight integration with mathlib.)

Conclusions

Total contribution: $\pm$ 5000 lines of project-specific code, $\pm$ 2500 lines background work.
(Difficult to quantify exactly due to tight integration with mathlib.)

Rules of thumb for our work:

- Parametrize (`is_scalar_tower`, `power_basis`, ...) instead of choosing a canonical construction.
- Refactoring allows deep integration between different viewpoints.
- Contribute quickly and often, so others do your work for you.