Rings, categories and schemes in Coq

a project to formalize algebraic geometry and related mathematics in Coq/SSReflect

Xuanrui Qi [1]

[1]Graduate School of Mathematics, Nagoya University

November 16, 2020
Theorem Proving and Provers 2020

Synopsis

Algebraic Geometry in Coq: an Experiment

Rings & Ideals Commutative algbra: maximal ideals, local rings, localization, Noetherian rings, etc.

Categories & Sheaves Categories, functors, sheaves, etc.

Scheme Defining a scheme

Goals

- Currently, focused on defining **schemes**
  - Just recently done in Lean, but not in any other theorem prover yet
- Long term
  - Formalize some important papers (e.g., Serre's FAC)
  - Write down some concrete examples of schemes
- Test how good Coq is at advanced, abstract mathematical reasoning.
- Experiment with packed classes and Mathematical Components library.

Goal: defining a scheme

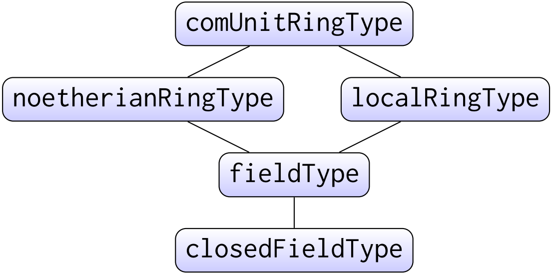
An **affine scheme** consists of a topological space $X = \mathrm{Spec}\ R$ which is homeomorphic to the *prime spectrum* of a (commutative) ring, and a *sheaf* $\mathcal{O}_X$ such that $\mathcal{O}_X(D(f)) = R_f$ (the *localization* of R at f).

Goal: defining a scheme

An **affine scheme** consists of a topological space $X = \mathrm{Spec}\ R$ which is homeomorphic to the *prime spectrum* of a (commutative) ring, and a *sheaf* $\mathcal{O}_X$ such that $\mathcal{O}_X(D(f)) = R_f$ (the *localization* of R at f).
This is an involved definition! Requires notion of:

- **sheaf** (and first, presheaf) (WIP)
- **localization** of a ring (WIP)
- **Zariski topology** (to be started)

Extending MathComp's algebraic hierarchy



Note that under classical reasoning `comUnitRingType` and `unitRingType` are the same.

Challenge 1: define localization of a ring

Definition: let R be a commutative ring and $S \subset R$ a multiplicatively closed set. The **localization of** R **at** S $S^{-1}R$ is defined as $(R \times S)/\sim$, where $(r_1, s_1) \sim (r_2, s_2)$ iff $\exists t \in S$ s.t. $t(r_1 s_2 - r_2 s_1 = 0)$.

Easy on paper, but difficult in a theorem prover!

Challenge 1: define localization of a ring

Three steps:

- define $\sim$ and show that it is indeed a equivalence relation;

Challenge 1: define localization of a ring

Three steps:

- define $\sim$ and show that it is indeed a equivalence relation;
- once the quotient is well-defined, show that $+$ and $\cdot$ lift through $\sim$ so that $S^{-1}R$ is a ring [troublesome in Coq!];

Challenge 1: define localization of a ring

Three steps:

- define $\sim$ and show that it is indeed a equivalence relation;
- once the quotient is well-defined, show that $+$ and $\cdot$ lift through $\sim$ so that $S^{-1}R$ is a ring [troublesome in Coq!];
- show that $S^{-1}R$ is indeed local.

Challenge 1: define localization of a ring

Three steps:

- define $\sim$ and show that it is indeed a equivalence relation;
- once the quotient is well-defined, show that $+$ and $\cdot$ lift through $\sim$ so that $S^{-1}R$ is a ring [troublesome in Coq!];
- show that $S^{-1}R$ is indeed local.

Phew!

First step: $\sim$ is w.f.

"Subset type" wrapper pattern:

```
Structure tS := MkMulType { elem : R ; _ : elem \in S }.
```

Often used, but can be tedious.

First step: $\sim$ is w.f.

"Subset type" wrapper pattern:

```
Structure tS := MkMulType { elem : R ; _ : elem \in S }.
```

Often used, but can be tedious.

Define $\sim$ and show that it is an equivalence:

```
Definition loc_equiv (p p' : R * tS) :=
match p, p' with
| (r1, s1), (r2, s2) =>
  `[< exists t, t * (r1 * (s2 : R) - r2 * (s1 : R)) = 0 >]
end.
...
Canonical loc_equiv_equiv :=
  EquivRelPack loc_equiv_is_equiv.
...
Definition localize := {eq_quot loc_equiv}.
```

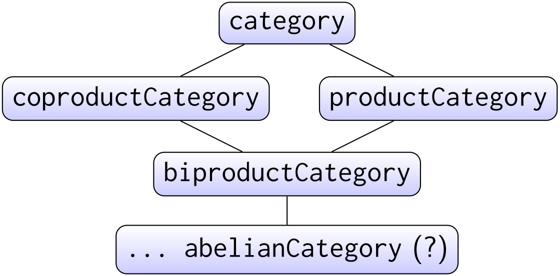
WIP: steps 2 and 3

Lifted addition and multiplication:

```
Definition add_localized := lift_op2 localize loc_add.
Definition mul_localized := lift_op2 localize loc_mul.
```

Now, need to prove their associativity and/or commutativity...
(WIP)

Category theory à la MathComp



Hierarchies for functors, natural transformations, etc.

Why category theory?

- By-product: need to describe algebraic geometry more accurately
- Allows us to formalize diagram chasing to simplify proofs
- A practical library for algebraic geometers

An example

```
Structure mixin_of (C : category) : Type := Mixin {
  prod : C -> C -> C ;
  proj1 : forall {X1 X2 : C}, prod X1 X2 ~> X1 ;
  proj2 : forall {X1 X2 : C}, prod X1 X2 ~> X2 ;
  _ : forall (X1 X2 Y : C) (f1 : Y ~> X1) (f2 : Y ~> X2),
      exists! (f : Y ~> prod X1 X2), proj1 \\o f = f1 /\
      proj2 \\o f = f2
  }.
```

A categorical structure defined as mixin over another categorical structure.

Defining a presheaf

Take 1: presheaf as contravariant functor $X^{\mathrm{op}} \to \mathcal{C}$
Problem: not easy to work with (e.g., applying the restriction map), trouble with defining sheaf axioms.

Defining a presheaf

Take 1: presheaf as contravariant functor $X^{\mathrm{op}} \to \mathcal{C}$
Problem: not easy to work with (e.g., applying the restriction map), trouble with defining sheaf axioms.

Take 2: direct definition (a topological space with a structure and a restriction map)
Benefit: easy to work with! But need a coercion to use as functor. We use this approach to define presheaves and sheaves (WIP).

See our code!

https://github.com/xuanruiqi/commalg
https://github.com/xuanruiqi/categories