

$-1-1_2\text{ }opop_2$
 $\mathcal{U}\mathcal{U}1\text{ } \mathbb{K}\mathbb{K}$
 $:=:=2$

Function Fields of Integral Schemes in Mathlib4

A Companion to `FunctionField.lean`

1 Introduction

This document provides a natural language companion to the `FunctionField.lean` file in Mathlib4. The file develops the theory of function fields for integral schemes, which is one of the most fundamental concepts in algebraic geometry. The function field captures the "generic" behavior of rational functions on an irreducible variety and serves as the algebraic counterpart to the geometric notion of the generic point.

In classical algebraic geometry, the function field of an irreducible variety is the field of rational functions on that variety. In scheme theory, this becomes the stalk of the structure sheaf at the generic point of an irreducible scheme. This generalization provides a powerful tool for understanding the global properties of schemes through local algebraic data.

2 The Function Field

2.1 Definition

```
1 noncomputable abbrev Scheme.functionField [IrreducibleSpace X] : CommRingCat :=
2   X.presheaf.stalk (genericPoint X)
```

Mathematical Significance: The function field of an irreducible scheme X is defined as the stalk of the structure sheaf at the generic point of X . Despite the name, this is only a field when the scheme is integral (both irreducible and reduced).

The generic point represents the "most general" point of the scheme - it specializes to every other point. Therefore, the stalk at the generic point captures the most general algebraic behavior of the scheme and contains the minimal amount of algebraic information needed to understand the entire scheme.

In classical terms, if $X = \text{Spec}(R)$ where R is an integral domain, then the function field is the field of fractions of R , which consists of all formal quotients $\frac{f}{g}$ where $f, g \in R$ and $g \neq 0$.

2.2 Canonical Germ Maps

```
1 noncomputable abbrev Scheme.germToFunctionField [IrreducibleSpace X] (U : X.Open
2   )
3   [h : Nonempty U] : (X, U) → X.functionField :=
4   X.presheaf.germ U
5   (genericPoint X)
6   (((genericPoint_spec X).mem_open_set_iff U.isOpen).mpr (by simp using h))
```

Mathematical Significance: For any nonempty open subset U of an irreducible scheme X , there is a canonical ring homomorphism from the sections over U to the function field. This map sends each section to its "germ" or "value" at the generic point.

This construction establishes the function field as the "limit" of all the section rings as we take smaller and smaller (but still nonempty) open sets. Since the generic point lies in every nonempty open set of an irreducible scheme, we always have such a canonical map.

3 Field Structure for Integral Schemes

3.1 Field Instance

```

1 noncomputable instance [IsIntegral X] : Field X.functionField := by
2   refine .ofIsUnitOrEqZero fun a      ?_
3   obtain U , m, s, rfl := TopCat.Presheaf.germ_exist _ _ a
4   rw [or_iff_not_imp_right, (X.presheaf.germ _ _ m).hom.map_zero]
5   intro ha
6   replace ha := ne_of_apply_ne _ ha
7   have hs : genericPoint X      RingedSpace.basicOpen _ s := by
8     rw [SetLike.mem_coe, (genericPoint_spec X).mem_open_set_iff,
9       Set.univ_inter, Set.nonempty_iff_ne_empty, Ne, Opens.coe_bot,
10      SetLike.ext'_iff]
11     erw [basicOpen_eq_bot_iff]
12     exact ha
13   have := (X.presheaf.germ _ _ hs).hom.isUnit_map (RingedSpace.
14     isUnit_res_basicOpen _ s)
15   rwa [Presheaf.germ_res_apply] at this

```

Mathematical Significance: When the scheme is integral (irreducible and reduced), the function field is indeed a field. The proof shows that every nonzero element has an inverse by using the fundamental property that in integral schemes, a section that doesn't vanish identically must be invertible on some basic open set, and since the generic point lies in every nonempty open, it lies in this basic open where the section is invertible.

This result establishes the crucial connection between the geometric notion of integrality and the algebraic notion of being a field. It shows that integral schemes have well-defined function fields that capture their rational function behavior.

4 Injectivity Properties

4.1 Germ Injectivity

```

1 theorem germ_injective_of_isIntegral [IsIntegral X] {U : X.Opens} (x : X) (hx : x
  U) :
2   Function.Injective (X.presheaf.germ U x hx) := by
3   rw [injective_iff_map_eq_zero]
4   intro y hy
5   rw [(X.presheaf.germ U x hx).hom.map_zero] at hy
6   obtain W , hW, iU, iV, e := X.presheaf.germ_eq _ hx hx _ _ hy
7   cases Subsingleton.elim iU iV
8   haveI : Nonempty W := _ , h W
9   exact map_injective_of_isIntegral X iU e

```

Mathematical Significance: In an integral scheme, the germ map from any open set to any stalk is injective. This fundamental result shows that in integral schemes, sections can be uniquely recovered from their germs, reflecting the fact that integral domains embed into their fields of fractions.

4.2 Function Field Map Injectivity

```

1 theorem Scheme.germToFunctionField_injective [IsIntegral X] (U : X.Opens) [
  Nonempty U] :
2   Function.Injective (X.germToFunctionField U) :=
3   germ_injective_of_isIntegral _ _ _

```

Mathematical Significance: The canonical map from sections over any nonempty open set to the function field is injective for integral schemes. This means that the function field contains a faithful copy of all the section rings, making it a universal repository for the rational functions on the scheme.

5 Behavior Under Morphisms

5.1 Generic Points and Open Immersions

```

1 theorem genericPoint_eq_of_isOpenImmersion {X Y : Scheme} (f : X → Y) [H :
  IsOpenImmersion f]
2   [hX : IrreducibleSpace X] [hY : IrreducibleSpace Y] :
3   f.base (genericPoint X) = genericPoint Y := by
4   apply ((genericPoint_spec Y).eq _).symm
5   convert (genericPoint_spec X).image (show Continuous f.base by fun_prop)
6   symm
7   rw [Set.univ_subset_iff]
8   convert subset_closure_inter_of_isPreirreducible_of_isOpen _ H.base_open.
  isOpen_range _

```

Mathematical Significance: When we have an open immersion between irreducible schemes, the generic point of the source maps to the generic point of the target. This fundamental result ensures that open subschemes of irreducible schemes have the "same" generic point, preserving the function field structure.

This property is crucial for understanding how function fields behave under geometric inclusions and ensures that rational functions on a scheme can be naturally extended to any larger scheme containing it as an open subscheme.

6 Affine Case

6.1 Generic Point in Spectra

```

1 simp
2 theorem genericPoint_eq_bot_of_affine (R : CommRingCat) [IsDomain R] :
3   genericPoint (Spec R) = (0 : PrimeSpectrum R) := by
4   apply (genericPoint_spec (Spec R)).eq
5   rw [isGenericPoint_def]
6   rw [PrimeSpectrum.zeroLocus_vanishingIdeal_eq_closure, PrimeSpectrum.
  vanishingIdeal_singleton]
7   rw [PrimeSpectrum.zeroLocus_singleton_zero]
8   rfl

```

Mathematical Significance: For the spectrum of an integral domain R , the generic point is the zero ideal (0) , which corresponds to the smallest prime ideal. This makes algebraic sense because the zero ideal represents the "most general" prime, and localizing at it gives the field of fractions.

6.2 Fraction Ring Structure

```

1 instance functionField_isFractionRing_of_affine (R : CommRingCat.{u}) [IsDomain R] :
2   IsFractionRing R (Spec R).functionField := by
3   convert StructureSheaf.IsLocalization.to_stalk R (genericPoint (Spec R))
4   delta IsFractionRing IsLocalization.AtPrime
5   -- Porting note: 'congr' does not work for 'Iff'
6   apply Eq.to_iff
7   congr 1
8   rw [genericPoint_eq_bot_of_affine]

```

```

9   ext
10  exact mem_nonZeroDivisors_iff_ne_zero

```

Mathematical Significance: For an affine integral scheme $\text{Spec}(R)$ where R is an integral domain, the function field is precisely the field of fractions of R . This establishes the fundamental dictionary between algebraic and geometric notions of function fields.

This result connects the abstract scheme-theoretic definition with the classical algebraic notion, showing that our geometric construction recovers the familiar algebraic object in the affine case.

7 Local Properties

7.1 Fraction Ring Property for Affine Opens

```

1  theorem functionField_isFractionRing_of_isAffineOpen [IsIntegral X] (U : X.Opens)
2    (hU : IsAffineOpen U) [Nonempty U] :
3    IsFractionRing (X, U) X.functionField := by
4  haveI : IsAffine _ := hU
5  haveI : IsIntegral U :=
6    @isIntegral_of_isAffine_of_isDomain _ _ _
7    (by rw [Scheme.Opens.toScheme_presheaf_obj, Opens.isOpenEmbedding_obj_top];
8     infer_instance)
9  delta IsFractionRing Scheme.functionField
10 convert hU.isLocalization_stalk genericPoint X,
    (((genericPoint_spec X).mem_open_set_iff U.isOpen).mpr (by simp using
      Nonempty U )) using 1

```

Mathematical Significance: For any nonempty affine open subset U of an integral scheme X , the function field of X is the field of fractions of $\Gamma(X, U)$. This shows that the function field can be constructed from the sections over any sufficiently large affine open set.

This result provides a practical way to compute function fields and shows that they have a local nature - they can be understood by examining sufficiently large affine pieces of the scheme.

7.2 Stalk-Function Field Relationship

```

1  instance [IsIntegral X] (x : X) :
2    IsFractionRing (X.presheaf.stalk x) X.functionField :=
3  let U : X.Opens := (X.affineCover.map x).opensRange
4  have hU : IsAffineOpen U := isAffineOpen_opensRange (X.affineCover.map x)
5  let x : U := x , X.affineCover.covers x
6  have : Nonempty U := x
7  -- Technical proof involving localizations

```

Mathematical Significance: For any point x in an integral scheme X , the function field is the field of fractions of the stalk at x . This establishes a fundamental relationship between local and global properties of integral schemes.

This result shows that every stalk naturally embeds into the function field, and the function field can be constructed as the field of fractions of any stalk. This provides multiple equivalent ways to think about and construct function fields.

8 Domain Properties

8.1 Stalks as Integral Domains

```

1 instance [IsIntegral X] {x : X} : IsDomain (X.presheaf.stalk x) :=
2   Function.Injective.isDomain _ (IsFractionRing.injective (X.presheaf.stalk x) (X
    .functionField))

```

Mathematical Significance: In an integral scheme, every stalk is an integral domain. This follows from the fact that each stalk embeds into the function field (which is a field), and subrings of fields are integral domains.

This result provides a local characterization of integrality and shows that the global geometric property of being integral translates directly to local algebraic properties at every point.

9 Geometric Significance

The function field theory developed in this file captures several fundamental aspects of algebraic geometry:

- **Rational Functions:** The function field provides the scheme-theoretic generalization of the field of rational functions on an algebraic variety. It captures the "generic" algebraic behavior of the scheme.
- **Generic Point Philosophy:** By defining the function field as the stalk at the generic point, the theory makes precise the intuitive idea that generic behavior determines specific behavior in algebraic geometry.
- **Local-Global Principle:** The various characterizations (via affine opens, stalks, etc.) show that global function fields can be understood through local algebraic data.
- **Birational Invariance:** The function field captures the birational geometry of schemes - schemes that are birationally equivalent have isomorphic function fields.
- **Integrality Detection:** The field property of function fields provides a way to detect and characterize integral schemes through their most generic algebraic data.

The function field serves as a bridge between the geometric intuition of "rational functions defined everywhere except on lower-dimensional sets" and the algebraic precision required for modern algebraic geometry. It provides the foundation for many advanced topics including birational geometry, function field arithmetic, and the study of rational points on schemes.