

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

# Basic Properties of Schemes in Mathlib4

## A Companion to `Properties.lean`

## 1 Introduction

This document provides a natural language companion to the `Properties.lean` file in Mathlib4. The file establishes fundamental properties that schemes can satisfy, focusing on two crucial notions from classical algebraic geometry: reduced and integral schemes. These properties form the foundation for understanding the geometric behavior of schemes and their relationship to classical algebraic varieties.

## 2 Topological Properties of Schemes

### 2.1 $T_0$ Spaces

```
1 instance : TOSpace X :=
2   TOSpace.of_open_cover fun x => _ , X.affineCover.covers x,
3   (X.affineCover.map x).opensRange.2, IsEmbedding.tOSpace (Y := PrimeSpectrum _)
4   (isAffineOpen_opensRange (X.affineCover.map x)).isoSpec.schemeIsoToHomeo.
   isEmbedding
```

**Mathematical Significance:** Every scheme is a  $T_0$  topological space. This means that for any two distinct points, there exists an open set containing one but not the other. This property is fundamental to the geometric nature of schemes and ensures that the topological structure carries meaningful geometric information.

### 2.2 Quasi-Sober Spaces

```
1 instance : QuasiSober X := by
2   apply (config := { allowSynthFailures := true })
3   quasiSober_of_open_cover (Set.range fun x => Set.range <| (X.affineCover.map
   x).base)
```

**Mathematical Significance:** Every scheme is quasi-sober, meaning that every irreducible closed subset has a unique generic point. This property connects the topological structure of schemes to their algebraic nature, ensuring that irreducible components correspond to prime ideals in a natural way.

### 2.3 Prespectral Spaces

```
1 instance {X : Scheme.{u}} : PrespectralSpace X :=
2   have (Y : Scheme.{u}) (_ : IsAffine Y) : PrespectralSpace Y :=
3     .of_isClosedEmbedding (Y := PrimeSpectrum _) _
4     Y.isoSpec.hom.homeomorph.isClosedEmbedding
5   have (i : _) : PrespectralSpace (X.affineCover.map i).opensRange.1 :=
6     this (X.affineCover.map i).opensRange (isAffineOpen_opensRange (X.affineCover
   .map i))
7   .of_isOpenCover X.affineCover.isOpenCover_opensRange
```

**Mathematical Significance:** Every scheme is a prespectral space, which means it satisfies certain separation and compactness properties that are weaker than being spectral but still ensure good geometric behavior. This property is inherited from affine schemes being spectra of rings.

## 3 Reduced Schemes

### 3.1 Definition of Reduced Schemes

```
1 class IsReduced : Prop where
2   component_reduced :      U, _root_.IsReduced    (X, U) := by infer_instance
```

**Mathematical Significance:** A scheme  $X$  is reduced if all sections of its structure sheaf over any open set form a reduced ring (i.e., have no nonzero nilpotent elements). Geometrically, this means the scheme has no "infinitesimal thickening" - it corresponds to a classical algebraic variety without embedded points or multiple components.

In classical algebraic geometry, reduced schemes correspond to radical ideals. The condition ensures that the scheme captures only the "geometric" information without any scheme-theoretic multiplicities.

### 3.2 Characterization via Stalks

```
1 theorem isReduced_of_isReduced_stalk [      x : X, _root_.IsReduced (X.presheaf.
   stalk x)] :
2   IsReduced X := by
3   refine fun U => fun s hs => ? _
4   apply Presheaf.section_ext X.sheaf U s 0
5   intro x hx
6   change (X.sheaf.presheaf.germ U x hx) s = (X.sheaf.presheaf.germ U x hx) 0
7   rw [RingHom.map_zero]
8   change X.presheaf.germ U x hx s = 0
9   exact (hs.map _).eq_zero
```

**Mathematical Significance:** A scheme is reduced if and only if all its stalks are reduced. This provides a local characterization of the reduced property, showing that the global geometric property can be verified by checking local algebraic conditions at each point.

### 3.3 Reduced Property and Basic Opens

```
1 simp
2 theorem basicOpen_eq_bot_iff {X : Scheme} [IsReduced X] {U : X.Opens}
3   (s : (X, U)) : X.basicOpen s =      s = 0 := by
4   refine eq_zero_of_basicOpen_eq_bot      s, ? _
5   rintro rfl
6   simp
```

**Mathematical Significance:** In a reduced scheme, a section has empty basic open (meaning it vanishes everywhere) if and only if the section is zero. This captures the geometric intuition that in reduced schemes, a function that vanishes at all points must be the zero function - there are no "nilpotent directions" where a nonzero function can still have empty zero set.

## 4 Integral Schemes

### 4.1 Definition of Integral Schemes

```

1 class IsIntegral : Prop where
2   nonempty : Nonempty X := by infer_instance
3   component_integral : (U : X.Opens) [Nonempty U], IsDomain (X, U) := by
      infer_instance

```

**Mathematical Significance:** A scheme  $X$  is integral if it is nonempty and the ring of sections over any nonempty open set is an integral domain. This is the scheme-theoretic generalization of irreducible algebraic varieties - integral schemes correspond to geometric objects that cannot be decomposed as a union of proper closed subvarieties.

Integral schemes are fundamental in algebraic geometry as they correspond to:

- Irreducible algebraic varieties in classical geometry
- Geometric objects with a well-defined function field
- Schemes whose coordinate rings are integral domains

## 4.2 Relationship Between Integral and Reduced

```

1 instance (priority := 900) isReduced_of_isIntegral [IsIntegral X] : IsReduced X
   := by
2   constructor
3   intro U
4   rcases U.1.eq_empty_or_nonempty with h | h
5   have : U =      := SetLike.ext' h
6   haveI : Subsingleton (X, U) :=
7     CommRingCat.subsingleton_of_isTerminal (X.sheaf.isTerminalOfEqEmpty this)
8   infer_instance
9   haveI : Nonempty U := by simpa
10  infer_instance

```

**Mathematical Significance:** Every integral scheme is automatically reduced. This reflects the algebraic fact that integral domains have no nilpotent elements. Geometrically, irreducible varieties cannot have embedded components or multiplicities.

## 4.3 Integral Schemes and Irreducible Spaces

```

1 instance irreducibleSpace_of_isIntegral [IsIntegral X] : IrreducibleSpace X := by
2   by_contra H
3   replace H : IsPreirreducible (      : Set X) := fun h =>
4     H { toPreirreducible := h
5         toNonempty := inferInstance }
6   simp_rw [isPreirreducible_iff_isClosed_union_isClosed, not_forall, not_or] at H
7   rcases H with S , T, hS, hT, h , h , h
8   -- The proof continues by showing this leads to a contradiction

```

**Mathematical Significance:** Every integral scheme has an irreducible underlying topological space. This establishes the fundamental connection between the algebraic property (integral domains) and the topological property (irreducibility).

## 4.4 Equivalence Characterization

```

1 theorem isIntegral_iff_irreducibleSpace_and_isReduced :
2   IsIntegral X      IrreducibleSpace X      IsReduced X :=
3   fun _ => inferInstance , inferInstance , fun _ , _ =>
4     isIntegral_of_irreducibleSpace_of_isReduced X

```

**Mathematical Significance:** A scheme is integral if and only if it is both irreducible as a topological space and reduced. This provides a clean geometric characterization: integral schemes are exactly the reduced irreducible schemes.

This equivalence bridges algebraic and geometric viewpoints:

- Algebraic: Sections form integral domains
- Geometric: The space is irreducible and has no embedded components

## 5 Preservation Properties

### 5.1 Preservation Under Open Immersions

```

1 theorem isReduced_of_isOpenImmersion {X Y : Scheme} (f : X → Y) [H :
  IsOpenImmersion f]
2   [IsReduced Y] : IsReduced X := by
3   constructor
4   intro U
5   have : U = f ⁻¹U ∩ f ⁻¹U := by
6     ext1; exact (Set.preimage_image_eq _ H.base_open.injective).symm
7   rw [this]
8   exact isReduced_of_injective (inv <| f.app (f ⁻¹U ∩ U)).hom
9   (asIso <| f.app (f ⁻¹U ∩ U) : (Y, f ⁻¹U ∩ U) → (X, f ⁻¹U ∩ U)).symm.
  commRingCatIsoToRingEquiv.injective

```

**Mathematical Significance:** The reduced property is preserved under open immersions. This means that open subschemes of reduced schemes are reduced, reflecting the local nature of the reduced condition.

### 5.2 Preservation for Integral Schemes

```

1 theorem isIntegral_of_isOpenImmersion {X Y : Scheme} (f : X → Y) [H :
  IsOpenImmersion f]
2   [IsIntegral Y] [Nonempty X] : IsIntegral X := by
3   constructor; infer_instance
4   intro U hU
5   have : U = f ⁻¹U ∩ f ⁻¹U := by ext1; exact (Set.preimage_image_eq _ H.
  base_open.injective).symm
6   rw [this]
7   have : IsDomain (Y, f ⁻¹U ∩ U) := by
8     apply (config := { allowSynthFailures := true }) IsIntegral.
  component_integral
9   exact _ , _, hU.some.prop, r f l
10  exact (asIso <| f.app (f ⁻¹U ∩ U) :
11    (Y, f ⁻¹U ∩ U) → (Y, f ⁻¹U ∩ U)).symm.commRingCatIsoToRingEquiv.toMulEquiv.isDomain _

```

**Mathematical Significance:** Nonempty open subschemes of integral schemes are integral. This shows that integrality is preserved under "geometric inclusions" that don't introduce new irreducible components.

## 6 Affine Case

### 6.1 Characterization for Affine Schemes

```

1 theorem affine_isReduced_iff (R : CommRingCat) :
2   IsReduced (Spec R) ↔ _root_.IsReduced R := by
3   refine ?_, fun h => inferInstance

```

```

4  intro h
5  exact isReduced_of_injective (Scheme. SpecIso R).inv.hom
6    (Scheme. SpecIso R).symm.commRingCatIsoToRingEquiv.injective

```

**Mathematical Significance:** For affine schemes, the scheme is reduced if and only if the corresponding ring is reduced. This establishes the fundamental dictionary between algebraic properties of rings and geometric properties of their spectra.

```

1  theorem affine_isIntegral_iff (R : CommRingCat) :
2    IsIntegral (Spec R)      IsDomain R :=
3    fun _ => MulEquiv.isDomain (Spec R,      )
4    (Scheme. SpecIso R).symm.commRingCatIsoToRingEquiv.toMulEquiv, fun _ =>
      inferInstance

```

**Mathematical Significance:** For affine schemes, the scheme is integral if and only if the corresponding ring is an integral domain. This provides the direct translation between algebraic geometry and commutative algebra for the most basic geometric objects.

## 7 Geometric Significance

The properties defined in this file capture fundamental geometric concepts:

- **Reduced schemes** correspond to classical algebraic varieties without embedded points or nilpotent structure. They represent the "honest" geometric objects where functions that vanish everywhere are actually zero.
- **Integral schemes** are the irreducible geometric objects - they cannot be written as a union of proper closed subvarieties. These are the building blocks of algebraic geometry, analogous to irreducible varieties in classical algebraic geometry.
- The relationship  $\text{Integral} \Rightarrow \text{Reduced}$  reflects the fundamental fact that irreducible objects cannot have embedded components.
- The characterization  $\text{Integral} \Leftrightarrow \text{Irreducible} \cap \text{Reduced}$  provides a clean geometric understanding of integrality.

These properties form the foundation for more advanced concepts in scheme theory and are essential for understanding the geometric behavior of algebraic objects defined by polynomial equations.