1. Introduction

This document provides a natural language companion to the `Spec.lean` file in Mathlib4. The file defines the spectrum functor Spec from commutative rings to locally ringed spaces, which is fundamental to algebraic geometry. The construction proceeds in three stages, building up structure incrementally: first as topological spaces, then as sheafed spaces, and finally as locally ringed spaces.

The spectrum construction is the bridge between algebra and geometry, allowing us to translate problems about rings into geometric problems about spaces and vice versa.

2. Implementation Strategy

The Spec functor is defined in three consecutive steps, each with more structure:

- Spec.toTop: from rings to topological spaces
- Spec.toSheafedSpace: from rings to sheafed spaces
- Spec.toLocallyRingedSpace: from rings to locally ringed spaces

This layered approach makes the construction more manageable and allows for reuse of the intermediate stages.

3. Topological Space Level

3.1. Objects

```
def Spec.topObj (R : CommRingCat.{u}) : TopCat :=
  TopCat.of (PrimeSpectrum R)
```

Natural Language: The spectrum of a commutative ring R as a topological space is simply the prime spectrum $\mathrm{Spec}(R)$, consisting of all prime ideals of R equipped with the Zariski topology.

3.2. Morphisms

```
def Spec.topMap {R S : CommRingCat.{u}} (f : R → S) : Spec.topObj S → Spec.topObj
R :=
  TopCat.ofHom (PrimeSpectrum.comap f.hom)
```

Natural Language: A ring homomorphism $f : R \to S$ induces a continuous map $\mathrm{Spec}(S) \to \mathrm{Spec}(R)$ by taking the preimage of prime ideals. Note the contravariant nature: morphisms of rings induce morphisms in the opposite direction on spectra.

3.3. Functoriality

```
theorem Spec.topMap_id (R : CommRingCat.{u}) : Spec.topMap (𝟙 R) = 𝟙 (Spec.topObj R)

theorem Spec.topMap_comp {R S T : CommRingCat.{u}} (f : R → S) (g : S → T) :
    Spec.topMap (f ≫ g) = Spec.topMap g ≫ Spec.topMap f
```

Natural Language: The construction respects identities and composition, making it a genuine contravariant functor. The composition formula shows the order reversal characteristic of contravariant functors.

3.4. The Topological Functor

```
def Spec.toTop : CommRingCat.{u}ᵒᵖ ⥤ TopCat where
  obj R := Spec.topObj (unop R)
  map {_ _} f := Spec.topMap f.unop
```

Natural Language: By working in the opposite category of rings, we can present Spec as a covariant functor from $\mathbf{CRing}^{\mathrm{op}}$ to **Top**.

4. Sheafed Space Level

4.1. Objects with Structure Sheaves

```
def Spec.sheafedSpaceObj (R : CommRingCat.{u}) : SheafedSpace CommRingCat where
  carrier := Spec.topObj R
  presheaf := (structureSheaf R).1
  IsSheaf := (structureSheaf R).2
```

Natural Language: The spectrum of R as a sheafed space consists of the prime spectrum as the underlying topological space, equipped with the structure sheaf. The structure sheaf assigns to each open set the appropriate localization of R.

4.2. Morphisms of Sheafed Spaces

```
def Spec.sheafedSpaceMap {R S : CommRingCat.{u}} (f : R → S) :
    Spec.sheafedSpaceObj S → Spec.sheafedSpaceObj R where
  base := Spec.topMap f
  c :=
    { app := fun U => CommRingCat.ofHom <|
        comap f.hom (unop U) ((TopologicalSpace.Opens.map (Spec.topMap f)).obj (unop
U)) fun _ => id
      naturality := fun {_ _} _ => by ext; rfl }
```

Natural Language: A ring homomorphism $f : R \to S$ induces a morphism of sheafed spaces. The underlying map is the topological map defined earlier, and the sheaf morphism is given by the comap construction, which provides the appropriate maps between structure sheaves.

4.3. Functoriality Properties

```
theorem Spec.sheafedSpaceMap_id {R : CommRingCat.{u}} :
    Spec.sheafedSpaceMap (𝟙 R) = 𝟙 (Spec.sheafedSpaceObj R)

theorem Spec.sheafedSpaceMap_comp {R S T : CommRingCat.{u}} (f : R → S) (g : S → T) :
    Spec.sheafedSpaceMap (f ≫ g) = Spec.sheafedSpaceMap g ≫ Spec.sheafedSpaceMap f
```

Natural Language: The sheafed space construction also preserves identities and composition, confirming that we have a well-defined contravariant functor at the sheafed space level.

4.4. The Sheafed Space Functor

```
def Spec.toSheafedSpace : CommRingCat.{u}ᵒᵖ ⥤ SheafedSpace CommRingCat where
  obj R := Spec.sheafedSpaceObj (unop R)
  map f := Spec.sheafedSpaceMap f.unop
  map_comp f g := by simp [Spec.sheafedSpaceMap_comp]
```

Natural Language: This gives us a contravariant functor from commutative rings to sheafed spaces over commutative rings.

5. Presheafed Space Perspective

5.1. Forgetful Functor

```
def Spec.toPresheafedSpace : CommRingCat.{u}ᵒᵖ ⥤ PresheafedSpace CommRingCat :=
  Spec.toSheafedSpace ⋙ SheafedSpace.forgetToPresheafedSpace
```

Natural Language: By composing with the forgetful functor, we can also view Spec as producing presheafed spaces. This perspective is sometimes useful for certain constructions and proofs.

6. Locally Ringed Space Level

6.1. Local Rings at Stalks

```
def Spec.locallyRingedSpaceObj (R : CommRingCat.{u}) : LocallyRingedSpace :=
  { toSheafedSpace := Spec.sheafedSpaceObj R
    localRing := fun x => by
      exact (structureSheaf R).isUnit_res_basicOpen (R := R) ⟨x, rfl⟩ }
```

Natural Language: The spectrum becomes a locally ringed space by verifying that the stalk at each point is indeed a local ring. This follows from the fundamental property of the structure sheaf that stalks are localizations at prime ideals.

6.2. Stalk Maps and Locality

```
theorem stalkMap_toStalk {R S : CommRingCat.{u}} (f : R ⟶ S) (p : PrimeSpectrum S) :
  StructureSheaf.stalkIso S p ≪≫
  (Spec.sheafedSpaceMap f).c.app (op (PrimeSpectrum.basicOpen ⊤)) ≪≫
  (StructureSheaf.stalkIso R (PrimeSpectrum.comap f.hom p)).inv =
  CommRingCat.ofHom (Localization.localRingHom _ _ f.hom rfl)
```

Natural Language: The induced maps on stalks are exactly the localization maps. This shows that morphisms of locally ringed spaces induced by ring homomorphisms have the correct local behavior.

6.3. Morphisms of Locally Ringed Spaces

```
def Spec.locallyRingedSpaceMap {R S : CommRingCat.{u}} (f : R ⟶ S) :
    Spec.locallyRingedSpaceObj S ⟶ Spec.locallyRingedSpaceObj R :=
  { toSheafedSpaceHom := Spec.sheafedSpaceMap f
    isLocalAtTarget := fun x => by
      -- Proof that stalk maps are local ring homomorphisms
      -- ... }
```

Natural Language: A ring homomorphism induces a morphism of locally ringed spaces, where the key additional property is that the induced stalk maps are local ring homomorphisms.

6.4. The Main Functor

```
def Spec.toLocallyRingedSpace : CommRingCat.{u}ᵒᵖ ⥤ LocallyRingedSpace where
  obj R := Spec.locallyRingedSpaceObj (unop R)
  map f := Spec.locallyRingedSpaceMap f.unop
```

Natural Language: This is the main result: a contravariant functor from commutative rings to locally ringed spaces. This functor is the foundation for defining schemes.

7. The Spec-Γ Relationship

7.1. Natural Transformation to Global Sections

```
def toSpecΓ (R : CommRingCat.{u}) : R → Γ.obj (op (Spec.toLocallyRingedSpace.obj (op
R))) :=
  StructureSheaf.toOpen R ⊤

instance isIso_toSpecΓ (R : CommRingCat.{u}) : IsIso (toSpecΓ R)
```

Natural Language: There is a natural isomorphism between any commutative ring R and the global sections of Spec(R). This is a fundamental relationship that will be part of the adjunction between Γ and Spec.

7.2. Naturality

```
theorem Spec_Γ_naturality {R S : CommRingCat.{u}} (f : R → S) :
  toSpecΓ R ≫ Γ.map (Spec.toLocallyRingedSpace.map f.op).op = f
```

Natural Language: The isomorphisms between rings and their spectra's global sections are natural with respect to ring homomorphisms. This naturality is crucial for establishing the adjunction.

7.3. Identity Natural Transformation

```
def LocallyRingedSpace.SpecΓIdentity : Spec.toLocallyRingedSpace.rightOp ≫ Γ ≅ 1 _ :=
  asIso
    { app := toSpecΓ
      naturality := Spec_Γ_naturality }
```

Natural Language: The collection of all these natural isomorphisms forms a natural isomorphism between the composite functor Spec ∘ Γ and the identity functor. This is one of the triangle identities needed for the adjunction.

8. Localization Properties

8.1. Localization Maps

```
theorem Spec_map_localization_isIso (R : CommRingCat.{u}) (M : Submonoid R)
    [M.IsUnit] : IsIso (Spec.locallyRingedSpaceMap (CommRingCat.ofHom (algebraMap R
(Localization M))))
```

Natural Language: When M consists entirely of units in R, the localization map $R \to M^{\{-1\}}R$ is an isomorphism, and consequently the induced map on spectra is also an isomorphism. This captures the idea that "localizing by units does nothing."

9. Pushforward and Pullback Properties

9.1. Pushforward to Stalks

```
def toPushforwardStalk : S → (Spec.topMap f _* (structureSheaf S).1).stalk p :=
  toStalk S (PrimeSpectrum.comap f.hom p) ≫
  (Spec.topMap f _* (structureSheaf S).1).germ ⟨p, rfl⟩
```

Natural Language: There are canonical maps from a ring to the stalks of pushforward sheaves. This construction is fundamental for understanding how ring homomorphisms affect the local structure of spectra.

9.2. Algebraic Structure

```
instance : Algebra R ((Spec.topMap f _* (structureSheaf S).1).stalk p) :=
  RingHom.toAlgebra (toPushforwardStalk f p)
```

Natural Language: The stalks of pushforward sheaves naturally carry algebra structures over the original ring. This provides an algebraic framework for understanding the geometric relationship between spectra.

10. Module-Theoretic Aspects

10.1. Localized Modules

```
def toPushforwardStalkAlgHom :
  S →ₐ[R] (Spec.topMap f _* (structureSheaf S).1).stalk p :=
  { toFun := toPushforwardStalk f p
    -- Algebra homomorphism properties ... }

instance isLocalizedModule_toPushforwardStalkAlgHom :
  IsLocalizedModule (Ideal.primeCompl p.asIdeal) (toPushforwardStalkAlgHom f p)
```

Natural Language: The stalks of pushforward sheaves are localized modules in a precise sense. This connects the geometric construction with the algebraic theory of localization, providing tools for computing stalks and understanding their properties.

11. Conclusion

The Spec functor construction in this file provides the essential bridge between commutative algebra and algebraic geometry. By building up the structure in stages (topological → sheafed → locally ringed spaces), the implementation is both modular and mathematically transparent.

The key insights captured are:

- Prime ideals correspond to points in geometric spaces
- Ring homomorphisms induce geometric maps in the reverse direction
- Local algebraic properties (like being a local ring) have geometric meanings
- The spectrum construction preserves and reflects important algebraic information

This foundation enables the definition of schemes and the development of the full theory of algebraic geometry in Mathlib4.