Annals of Formalized Mathematics
Volume **1** (2025), 1–42
DOI: https://doi.org/10.46298/afm.13609

Formalization of derived categories in Lean/Mathlib

Joël Riou

Abstract. This paper outlines the formalization of derived categories in the mathematical library of the proof assistant Lean 4. The derived category $D(\mathcal{C})$ of any abelian category $\mathcal{C}$ is formalized as the localization of the category of unbounded cochain complexes with respect to the class of quasi-isomorphisms, and it is endowed with a triangulated structure.

MSC 2020: 18G80, 18G15, 18G40, 18E35, 68V20
Keywords: derived category, homological algebra, spectral sequence

Contact information:
Joël Riou: Université Paris-Saclay, CNRS, Laboratoire de mathématiques d'Orsay, 91405 Orsay, France. joel.riou@universite-paris-saclay.fr

Received: May 17, 2024
Final form: January 20, 2025
Accepted: February 11, 2025

1 Introduction

1.1 Derived categories are not absolutely necessary in order to do homological algebra. Indeed, when I initially learnt about this subject, it was all about studying projective resolutions of modules, or homotopies between morphisms of chain complexes of free abelian groups in the context of the singular homology of topological spaces. I learnt very nice theorems, but some details surprised me, as in the universal coefficient theorem for singular homology:

Theorem 1.1.1 ([Spa95, Theorem 5.2.8]). Let X be a topological space. Let A be an abelian group. For any $n \in \mathbf{Z}$, there is a canonical short exact sequence:

$$0 \to H_n(X) \otimes_{\mathbf{Z}} A \to H_n(X, A) \to \mathrm{Tor}_1^{\mathbf{Z}}(H_{n-1}(X), A) \to 0$$

Moreover, this exact sequence splits (noncanonically).

I could understand the proof that the sequence splits, but the statement still looked mysterious to me. Which phenomenon was responsible for this? I understood this much better after learning about derived categories [Ver96]. In the derived category of an abelian category $\mathcal{C}$, instead of working up to homotopy, we formally invert quasi-isomorphisms (i.e. morphisms of complexes that induce isomorphisms in homology): for example, if $P_\bullet$ is a projective resolution of an object X in $\mathcal{C}$, then it can be understood as a quasi-isomorphism $P_\bullet \to X$,[1] so that we get an *isomorphism* $P_\bullet \cong X$ in the derived category $D(\mathcal{C})$. Because $\mathbf{Z}$ is a principal ring, any submodule of a free $\mathbf{Z}$-module is free, and then any $\mathbf{Z}$-module M admits a very short free resolution $0 \to P_1 \to P_0 \to M \to 0$, which implies the vanishing of Ext^q groups for $q \geq 2$. Using this, one may obtain that any object in the derived category of abelian groups, in particular the singular chain complex $C_\star X$ of a topological space X, decomposes (noncanonically) in the derived category as a direct sum $C_\star X \cong \oplus_{n \in \mathbf{N}} H_n(X)[n]$. This gives a more satisfactory explanation for the existence of the splitting in 1.1.1. Similarly, degeneracy of spectral sequences can be explained using splittings of objects in derived categories, as it was done in algebraic geometry for the degeneracy of the Leray spectral sequence in étale cohomology with $\mathbf{Q}_\ell$ coefficients for a projective and smooth morphism such that fibers satisfy the conclusion of the hard Leftschetz theorem [Del68].

1.2 Derived categories were initially introduced by Grothendieck and Verdier in order to study the cohomology of schemes, first for coherent sheaves [Har66], as an extension of Serre's duality, and secondly for étale sheaves, towards the proof of the Weil conjectures [Del74].

[1] Here, we identify X to the complex $\dots \to 0 \to 0 \to X \to 0 \to 0 \to \dots$ where X sits in degree 0.

Actually, in the étale context, the derived categories are not just a tool in order to prove theorems, but the important statements about the “six operations” in the étale formalism can be phrased *only* using derived categories.[2]

1.3 The main result in this paper is the formalization in Lean/mathlib of the derived category of any abelian category. An application to the construction of spectral sequences, in particular the Grothendieck spectral sequence for the composition of right derived functors is also obtained (see 5.4.5.1).

1.4 Derived categories already appeared in some form in the Liquid Tensor Experiment (LTE), a team effort led by Johan Commelin to formalize in Lean a highly nontrival result in condensed mathematics by Dustin Clausen and Peter Scholze [Sch21] [CT22]. However, only the bounded above derived category was considered and it was defined only for an abelian category $\mathcal{C}$ with enough projectives as the homotopy category of bounded above cochain complexes of projective objects in $\mathcal{C}$. It follows that the major novelty in my formalization is that it relies on the definition of the derived category in general as a localized category (see section 3) obtained by formally inverting quasi-isomorphisms between arbitrary cochain complexes. Spectral sequences also appeared in a prior work in Lean 2 [DRB17], where the Serre spectral sequence of a fibration [Ser51] was constructed. A spectacular formalization of the Brouwer fixed-point theorem in Lean was obtained by Brendan Murphy as a consequence of his formalization of singular homology [Mur22].

1.5 The Lean proof assistant is developed mainly by Leonardo de Moura [MU21]. The formalization work described in this paper builds on mathlib, which is the community developed mathematical library for Lean [The20]. Category theory was initially developed by Kim Morrison and general results on abelian categories were obtained by Markus Himmel [Him20]. This work is also very much a “post-LTE” development, because the design of homological complexes in mathlib owes a lot to the LTE project, and the lessons learned from it were very helpful in order to develop homology in mathlib (see 2.1).

1.6 Homological algebra has been formalized in other proof assistants and studied in an effective manner [RS13] in specialized software. The Kenzo program [DSS] is a tool that is able to compute homology groups and homotopy groups. Part of the theorems it relies on have been formally verified in the proof assistant Isabelle/HOL [ABR08]. The approach in this paper is decidedly non-constructive and non-effective! However, I wish that the software API

[2] See [Ill90] for more information about the development of derived categories.

I have contributed in Lean/mathlib could be used in order to formally certify computations in homology.

1.7 Automated methods for diagram chasing in homological algebra have been studied in [Mon22] and [GMP24]. This formalization of derived categories (and spectral sequences) includes a certain number of diagram chases,[3] but the strategy I have used (see 2.2) makes diagram chasing in general abelian categories almost as easy as it would be in the category of abelian groups. Then, even though attempts at automation of diagram chasing are very interesting developments, I have not felt it would have eased significantly this work if such tools had been available.

1.8 This work has made possible a redefinition of Ext-groups in mathlib using derived categories (see 5.1). Formal properties such as long exact sequences of Ext-groups have been obtained, which should allow the development of more cohomology theory in Lean/mathlib. In particular, it will be possible to develop sheaf cohomology as Ext-groups in categories of abelian sheaves (or using the right derived functor approach 5.3.3). This work should also enable more computations in group cohomology (which was introduced in mathlib by Amelia Livingston [Liv23]).

1.9 This formalization was carried on as the GitHub branch `jriou_localization` of mathlib. The about 150 pull requests (PR) to mathlib which were extracted from this branch are listed at `https://github.com/leanprover-community/mathlib4/pull/25848`. In order to support the content in this paper, it is accompanied with a Lean file in the project `https://github.com/joelriou/lean-derived-categories` which allows an easy cross-reference between mathematical statements and definitions formalized in Lean.

1.10 Throughout the paper, mathlib notations are used whenever it is possible. For example, the composition of two morphisms $f : X \to Y$ and $g : Y \to Z$ shall be denoted $f \gg g$ (and not gf). A functor from a category $\mathcal{C}$ to a category $\mathcal{D}$ shall be denoted $F : \mathcal{C} \Rightarrow \mathcal{D}$. Composition of functors is denoted $F \ggg G$ (and not GF).

1.11 Acknowledgements

I would like to acknowledge the Lean/mathlib community for creating this amazing framework for the formalization of mathematics. I would like to thank particularly Patrick

[3] Actually, for the formalization of derived categories, only a handful of diagram chases are necessary. Spectral sequences require much more!

Massot for mentioning the existence of Lean to me, Floris van Doorn and Kyle Miller for their deep understanding of Lean/mathlib which enabled them to answer my questions while they were both postdocs in Orsay in 2022/2023. I thank Kevin Buzzard for his enthusiasm about my formalization projects. I want to acknowledge the extreme dedication of Johan Commelin towards the mathlib community, and his massive reviewing work of my pull requests to mathlib. Finally, I would like to thank the referee for their suggestions.

2 Homology and diagram chasing in general abelian categories

2.1 The homology refactor

2.1.1 In an abelian category $\mathcal{C}$, given two composable morphisms $X_1 \xrightarrow{f} X_2 \xrightarrow{g} X_3$ such that the composition vanishes `w : f ≫ g = 0`, one may define the homology at X_2 as the cokernel of the canonical map $\operatorname{Im} f \to \ker g$. This was essentially the definition in MATHLIB until I completed the homology refactor (a sequence of about 70 pull requests which was finished by PR #8706). This homology refactor had several goals:

- change the definition of homology and exactness so that it becomes self-dual (i.e. we may easily relate these notions in the opposite category and the original category);
- develop a convenient software API in order to manipulate homology objects and exactness.

One of the main ideas in order to achieve this was to introduce the category of "short complexes". Instead of using Lean terms like `homology f g w`, the idea was to introduce a structure `ShortComplex C` which bundles all this data:

```
structure ShortComplex [HasZeroMorphisms C] where
  {X₁ X₂ X₃ : C}
  f : X₁ ⟶ X₂
  g : X₂ ⟶ X₃
  zero : f ≫ g = 0
```

With this, it becomes more convenient to introduce a single variable `S : ShortComplex C`, and refer to its homology as `S.homology`. That `ShortComplex C` is a category makes it easy to study morphisms `S.homology ⟶ S'.homology` induced by morphisms `S ⟶ S'` of short complexes.

2.1.2 In order to define the homology of S, I introduced the notion of left homology data for S. Such a `h : S.LeftHomologyData` involves the data of a morphism $h.K \to S.X_2$ which identifies to the kernel of $S.g : S.X_2 \to S.X_3$ and a morphism $h.K \to h.H$ which identifies to the cokernel of the induced morphism $S.X_1 \to h.K$. In dual terms, we define `h' : S.RightHomologyData` to be the data of a morphism $S.X_2 \to h'.Q$ which identifies to the cokernel of $S.f : S.X_1 \to S.X_2$ and a morphism $h'.H \to h'.Q$ which identifies to the kernel of the induced morphism $h'.Q \to S.X_3$. Then, a homology data of S consists of left and right homology data h and h', together with an isomorphism $h.H \cong h'.H$ which makes the pentagon commute:

$$\begin{tikzcd}
S.X_1 \arrow[rrr, "S.f"] \arrow[dr, "h.f'"'] & & & S.X_2 \arrow[rrr, "S.g"] \arrow[drr, "h'.p"'] & & & S.X_3 \\
& h.K \arrow[urr, hook, "h.i"'] \arrow[dr, "h.\pi"'] & & & & h'.Q \arrow[ur, "h'.g'"'] & \\
& & h.H \arrow[rr, "\sim"] & & h'.H \arrow[ur, hook, "h'.\iota"'] & &
\end{tikzcd}$$

When such a homology data exists, we say that S "has homology". Under this assumption, which is the type class `S.HasHomology`, the homology `S.homology` of S is defined as $h.H$ for an arbitrary choice of such a homology data.[4] We define the object `S.cycles`, the cycles of S, as $h.K$. We also introduce the dual notion `S.opcycles`, the "opcycles" of S, which is defined as $h'.Q$. Then, one may understand the homology of S both as a quotient of `S.cycles` and as a subobject of `S.opcycles`. We say that S is exact (property `S.Exact`) when there is such a homology data and that the homology is a zero object.

One of the key remarks in order to understand the reason for this change of definition is that in the diagram above, the object $h.K$ does not need to be defined as `kernel S.g`. What is important in this approach is that this object $h.K$ is equipped with a morphism $h.i : h.K \to S.X_2$ which is *a* kernel of the morphism $S.g : S.X_2 \to S.X_3$, which in MATHLIB terms is formulated as the fields `wi : i ≫ S.g = 0` and `hi : IsLimit (KernelFork.ofι i wi)` of the left homology data structure h. Similar remarks apply to the objects $h'.Q$, $h.H$ and $h'.H$.

Left and right homology data behave well with respect to the application of exact functors. Actually, I initially introduced the notion of left homology data as part of the LTE in order to show that "homology commutes with the application of exact functors". The idea of redefining homology by using a structure similar to "homology data" was first formulated by Adam Topaz.

[4] The homology could have been defined as $h'.H$ instead of $h.H$. Even though doing such a choice breaks the symmetry between the left and the right, it has no consequence because $h.H$ and $h'.H$ are canonically isomorphic.

As left and right homology data are switched by passing to the opposite category, it is clear that these notions of homology and exactness of a short complex are self-dual.

In an abelian category, it is possible to show that all short complexes "have homology", so that the notion of homology defined here is consistent with the standard mathematical definition.

2.1.3 A significant advantage of this definition of homology and exactness is that it makes sense in very general categories. For example, if S is a short complex in any preadditive category $\mathcal{C}$, we may introduce the notion of splitting of S:

```
structure Splitting (S : ShortComplex C) where
  /-- a retraction of `S.f` -/
  r : S.X₂ ⟶ S.X₁
  /-- a section of `S.g` -/
  s : S.X₃ ⟶ S.X₂
  /-- the condition that `r` is a retraction of `S.f` -/
  f_r : S.f ≫ r = 𝟙 S.X₁ := by aesop_cat
  /-- the condition that `s` is a section of `S.g` -/
  s_g : s ≫ S.g = 𝟙 S.X₃ := by aesop_cat
  /-- the compatibility between the given section and retraction -/
  id : r ≫ S.f + S.g ≫ s = 𝟙 S.X₂ := by aesop_cat
```

In order to construct a splitting of S, we have to provide the morphisms r and s, but usually some of the three equations `f_r`, `s_g` and `id` can be proven automatically, which is the reason why in this code, the default value for the three proofs is `by aesop_cat`: category theory in MATHLIB relies heavily on the `aesop` automation tactic [LHF23].

Even though not all morphisms in $\mathcal{C}$ may have kernels or cokernels, it is still possible to show that if S is split (and $\mathcal{C}$ has a zero object), then S is a (short) exact short complex.

2.1.4 In MATHLIB, we have a category `HomologicalComplex C c` of homological complexes for a category $\mathcal{C}$ (with zero morphisms) and `c : ComplexShape ι`. The type ι is the type of indices for the complexes (like $\mathbb{N}$ or $\mathbb{Z}$) and `c` determines what are the directions of differentials. For example, `CochainComplex C ℤ` is an abbreviation for the category `HomologicalComplex C (ComplexShape.up ℤ)` which means that for a cochain complex K, the relevant differentials are `K.d i j : K.X i ⟶ K.X j` when $i+1=j$, which can be represented informally as follows when $\iota = \mathbb{Z}$:

$$\dots \xrightarrow{d} K^{n-2} \xrightarrow{d} K^{n-1} \xrightarrow{d} K^{n} \xrightarrow{d} K^{n+1} \xrightarrow{d} K^{n+2} \xrightarrow{d} \dots$$

As part of the implementation design of homological complexes in MATHLIB, the differential `K.d i j` is defined even if $i+1 \neq j$, in which case it has to be zero.

In the general situation, if K is a homological complex, and if i, j and k are indices in ι, then we may consider the short complex `K.sc' i j k` corresponding to the diagram `K.X i` $\longrightarrow$ `K.X j` $\longrightarrow$ `K.X k`. If i and k are respectively the previous and the next element of j for the complex shape c, then the homology of this short complex is by definition the homology of K in degree j: all the software API for the homology of homological complexes is based on the corresponding API for short complexes.

2.1.5 Besides changing the definitions, most of the work in this homology refactor corresponds to the development of a basic software API in order to manipulate homology objects, cycles and "opcycles": this does not involve any significant lemma or theorem!

2.2 Diagram chasing

2.2.1 In the category of abelian groups, a morphism $f : X \to Y$ is a monomorphism (resp. an epimorphism) if and only if f is an injective (resp. surjective) map, and a short complex $X_1 \xrightarrow{f} X_2 \xrightarrow{g} X_3$ is exact if and only if for any $x_2 \in X_2$ such that $g(x_2) = 0$, there exists $x_1 \in X_1$ such that $x_2 = f(x_1)$. These criteria allow a type of reasoning known as "diagram chasing": categorical properties can be rephased in terms of properties of *elements* in the abelian groups which appear in a certain diagram. In the category of abelian groups, the five lemma or the snake lemma can be obtained in this way.

There is another well-known situation where diagram chasing is possible. Let S be a topological space. Let $f : X \to Y$ be a morphism of sheaves of abelian groups on S. Then, f is a monomorphism if and only if for any open subset U of S, the map $f_U : X(U) \to Y(U)$ is injective. However, epimorphisms of sheaves cannot be characterized in such an easy way: instead of saying that an element in $Y(U)$ can be lifted to an element of $X(U)$, we should only require that it can be lifted *locally*. More precisely, it is possible to show that f is an epimorphism in the category of sheaves of abelian groups if and only if f is locally surjective, i.e. for any open subset U of S and $y \in Y(U)$, there exists an open cover $(U_i)_{i\in I}$ of U and sections $x_i \in X(U_i)$ such that for all $i \in I$, $f(x_i)$ is the restriction of y to U_i. Using these criteria, it is possible to do diagram chasing in categories of sheaves.

2.2.2 In order to formalize homological algebra, it is important to be able to obtain lemmas like the five lemma in general abelian categories. An abstract approach could be given by the Freyd–Mitchell embedding theorem of (small) abelian categories in categories of modules over a ring [Mit64, Theorem 4.4].[5] Markus Himmel was able to obtain basic homological algebra lemmas in general abelian categories [Him20] by formalizing a certain type of pseudo-elements attached to any object in an abelian category [Bor94]. As this particular type of pseudo-elements has some issues[6], I have developed a different approach which does not require the introduction of auxiliary types like pseudo-elements: the argumentation shall only involve morphisms in the abelian category. As we shall see in 2.2.3, this approach also has a sheaf-theoretic interpretation.

The key observation is the following lemma which characterizes epimorphisms in any abelian category $\mathcal{C}$:

```
lemma epi_iff_surjective_up_to_refinements (f : X ⟶ Y) :
  Epi f ↔ ∀ ⦃A : C⦄ (y : A ⟶ Y),
    ∃ (A' : C) (π : A' ⟶ A) (_ : Epi π) (x : A' ⟶ X), π ≫ y = x ≫ f := ...
```

The content of this lemma is illustrated in the following diagram:

$$\begin{tikzcd} A' \arrow[r, dotted, "x"] \arrow[d, dotted, two heads, "\pi"'] & X \arrow[d, "f"] \\ A \arrow[r, "y"] & Y \end{tikzcd}$$

Indeed, let $f : X \to Y$ be an epimorphism, and $y : A \to Y$ be any morphism. It would be too optimistic to expect the existence of a morphism $A \to X$ which makes the triangle commute. However, there exists an epimorphism $\pi : A' \to A$ and a morphism $x : A' \to X$ such that the square above commutes: it suffices to take the fiber product A' of y and f. Conversely, when this property holds for any morphism $y : A \to Y$ (in particular for the identity of Y), then f is an epimorphism.

Then, the idea is to think of a morphism $y : A \to Y$ as an "element" of Y. If f is an epimorphism, it may not be possible to lift it to an element $A \to X$ of X. However, as the lemma above shows, it becomes possible if we allow the precomposition of y with a well chosen epimorphism $A' \to A$. This operation of precomposition shall be named "refinement". With this language, a morphism f is an epimorphism if and only if it is "surjective up to refinements".

[5]The Freyd-Mitchell theorem was formalized by Markus Himmel, Jakob von Raumer, Paul Reichert and myself. It entered MATHLIB in February 2025 (PR #22222), see also [HR25].

[6]See https://mathoverflow.net/questions/419888/pullback-and-pseudoelements/419951 for the problematic behavior of these pseudo-elements with respect to pullbacks, which was raised by Riccardo Brasca during the LTE.

As the exactness of a short complex $X_1 \xrightarrow{f} X_2 \xrightarrow{g} X_3$ in an abelian category can be rephased by saying that the induced map $X_1 \to \ker g$ is an epimorphism, it is possible to deduce that similarly, exactness is equivalent to "exactness up to refinements":

```
lemma ShortComplex.exact_iff_exact_up_to_refinements (S : ShortComplex C) :
  S.Exact ↔ ∀ ⦃A : C⦄ (x₂ : A ⟶ S.X₂) (_ : x₂ ≫ S.g = 0),
   ∃ (A' : C) (π : A' ⟶ A) (_ : Epi π) (x₁ : A' ⟶ S.X₁),
    π ≫ x₂ = x₁ ≫ S.f := ...
```

After I had formalized these lemmas, I found that this approach was described in the unpublished notes [Ber74]. I have used the word "refinement" because this is the terminology which appeared there.

This type of argumentation "up to refinements" was very efficient in the formalization of homological algebra: the snake lemma, the long exact homology sequence of a short exact sequence of homological complexes, etc.

2.2.3 This approach of diagram chasing "up to refinements" admits a sheaf-theoretic interpretation. Let $f : X \to Y$ be a morphism in an abelian category $\mathcal{C}$. We may consider the induced natural transformation $\mathrm{Hom}(-, f) : \mathrm{Hom}(-, X) \to \mathrm{Hom}(-, Y)$, which we should think of as a morphism in the category of presheaves (of sets or of abelian groups) on the category $\mathcal{C}$. Essentially by definition, f is a monomorphism if and only if for all $A \in \mathcal{C}$, $\mathrm{Hom}(A, f)$ is injective, i.e. $\mathrm{Hom}(-, f)$ is a monomorphism of presheaves. In order to characterize epimorphisms, one may introduce the following Grothendieck topology [Sga, II 1.1] on the abelian category $\mathcal{C}$: a sieve of an object X for this "refinements topology" is covering if and only if it contains an epimorphism.[7] One may easily show that the representable presheaves $\mathrm{Hom}(-, X)$ and $\mathrm{Hom}(-, Y)$ are sheaves for this Grothendieck topology. With these definitions, epimorphisms in $\mathcal{C}$ can be characterized as follows:

Lemma 2.2.3.1. Let $f : X \to Y$ be a morphism in an abelian category $\mathcal{C}$. Then, f is an epimorphism if and only if the morphism of sheaves $\mathrm{Hom}(-, f) : \mathrm{Hom}(-, X) \to \mathrm{Hom}(-, Y)$ is locally surjective for the refinements topology (i.e. it is an epimorphism of sheaves[8]).

[7] In an abelian category, all epimorphisms are effective, so that this "refinements" topology is a particular case of the "regular topology" that is defined in MATHLIB.

[8] In MATHLIB, the statement that epimorphisms of sheaves are exactly the locally surjective morphisms requires some constraints on the universe parameters of the category $\mathcal{C}$, but these do hold if $\mathcal{C}$ is a small category in a certain universe u.

Indeed, from 2.2.2, we know that f is an epimorphism if and only if it is "surjective up to refinements". Then, essentially by definition, f is "surjective up to refinements" if and only if the morphism $\mathrm{Hom}(-, f)$ is locally surjective for the refinements topology.

It follows from this lemma that arguing "up to refinements" in a general abelian category is essentially a particular case of the basic diagram chasing in categories of sheaves which was mentioned in 2.2.1, at least if we are ready to use Grothendieck topologies instead of topological spaces.

3 Localization of categories

As it was mentioned in the introduction, the main difference between this formalization of homological algebra and previous works is that it relies on the definition of the derived category of an abelian category $\mathcal{C}$ as a *localized category*, i.e. it is obtained by formally inverting the class of quasi-isomorphisms.

3.1 Let $\mathcal{C}$ be a category. Let $\mathcal{W}$ be a class of morphisms in $\mathcal{C}$. In MATHLIB, such a class is `W : MorphismProperty C`.[9] The localized category $\mathcal{C}[\mathcal{W}^{-1}]$ (named `W.Localization` in MATHLIB) should be thought as the category generated by $\mathcal{C}$ in which we formally invert the morphisms that are in $\mathcal{W}$ [GZ67, I 1.1]. More precisely, the objects in $\mathcal{C}[\mathcal{W}^{-1}]$ are the same as in $\mathcal{C}$, but morphisms from X to Y in $\mathcal{C}[\mathcal{W}^{-1}]$ are equivalence classes of zigzags modulo the equivalence relation which enforces that we have a functor $Q : \mathcal{C} \Rightarrow \mathcal{C}[\mathcal{W}^{-1}]$ and that the formal inverses that are introduced are actual left and right inverses, where a zigzag is a diagram like this

$$X \longrightarrow Z_1 \longleftarrow Z_2 \longrightarrow Z_3 \longleftarrow \ldots \longleftarrow Z_n \longrightarrow Y$$

which may involve morphisms in $\mathcal{C}$ going in both directions, but with the condition that morphisms going towards the left are in $\mathcal{W}$. More precisely, when I formalized this (initially in the MATHLIB 3 PR #14422), I defined a quiver with the same objects as $\mathcal{C}$ and such that the arrows are either a morphism in $\mathcal{C}$ or a morphism in $\mathcal{W}$ in the other direction. Then, the localized category was defined as a quotient[10] of the path category of this quiver.[11]

[9]Classes of morphisms in categories were first introduced in mathlib by Andrew Yang in order to formulate properties of morphisms of schemes in algebraic geometry.

[10]Quotients categories were formalized by David Wärn in 2020.

[11]The path category of a quiver was formalized by Kim Morrison in 2021.

3.2 The localized category $\mathcal{C}[\mathcal{W}^{-1}]$ and the functor $Q : \mathcal{C} \Rrightarrow \mathcal{C}[\mathcal{W}^{-1}]$ satisfy the universal property that for any functor $F : \mathcal{C} \Rrightarrow \mathcal{E}$ which sends morphisms in $\mathcal{W}$ to isomorphisms in $\mathcal{E}$, there exists a unique functor $\widetilde{F} : \mathcal{C}[\mathcal{W}^{-1}] \Rrightarrow \mathcal{E}$ such that $F = Q \ggg \widetilde{F}$. A similar result was obtained in Coq by Carlos Simpson [Sim06].

3.3 In commutative algebra, there is a parallel notion of the localization of a commutative ring R at a (multiplicative) set $S \subset R$. This localization is a R-algebra T which satisfies a certain universal property, which implies that it is well defined up to a unique isomorphism. There is also an explicit construction of this localization, that is denoted $R[S^{-1}]$. In the applications, we usually want to apply results not only to the constructed algebra $R[S^{-1}]$ but to any T which satisfies the universal property. In the development of the theory of schemes in Lean [Buz+22], it was important to introduce a nice predicate which expresses that a morphism of rings $R \to T$ identifies T to the localization $R[S^{-1}]$.

Similarly, given a functor $L : \mathcal{C} \Rrightarrow \mathcal{D}$ and a class of morphisms $\mathcal{W}$ in $\mathcal{C}$, we would like to express that $\mathcal{D}$ is "the" localized category of $\mathcal{C}$ with respect to $\mathcal{W}$. The conditions are:

- any morphism in $\mathcal{W}$ is mapped by L to an isomorphism;
- the induced functor $\mathcal{C}[\mathcal{W}^{-1}] \Rrightarrow \mathcal{D}$ from the constructed localized category is an equivalence of categories.

When these conditions hold, we shall say that L is a localization functor for $\mathcal{W}$, and this is the predicate `L.IsLocalization W`. The exact definition of this predicate is used only in the internals of the software API about the localization of categories. This is a practical choice which allows to circumvent the universe issues mentioned below (see 3.7), and it also relaxes the condition on the localized category, so that this notion behaves well with respect to equivalences of categories. Using these definitions, we obtain the following relaxed universal property:

Lemma 3.3.1. If $L : \mathcal{C} \Rrightarrow \mathcal{D}$ is a localization functor for a class of morphisms $\mathcal{W}$, then for any category $\mathcal{E}$, the composition with L induces an equivalence of categories from the category of functors $\mathcal{D} \Rrightarrow \mathcal{E}$ to the full subcategory of $\mathcal{C} \Rrightarrow \mathcal{E}$ consisting of functors which invert $\mathcal{W}$.

This lemma contains most of what is needed for the applications: it allows to lift functors $\mathcal{C} \Rrightarrow \mathcal{E}$ to $\mathcal{D} \Rrightarrow \mathcal{E}$, and similarly natural transformations and natural isomorphisms can be lifted. Obviously, if $L : \mathcal{C} \Rrightarrow \mathcal{D}$ and $L' : \mathcal{C} \Rrightarrow \mathcal{D}'$ are two localization functors for a class of morphisms $\mathcal{W}$, there is an equivalence of categories $F : \mathcal{D} \Rrightarrow \mathcal{D}'$ equipped with an isomorphism `L ⋙ F ≅ L'`.

3.4 We obtain various stability properties of localization functors:

Lemma 3.4.1. If $L : \mathcal{C} \Rightarrow \mathcal{D}$ is a localization functor for a class of morphisms $\mathcal{W}$, then the functor $L^{op} : \mathcal{C}^{op} \Rightarrow \mathcal{D}^{op}$ is a localization functor for the opposite class $\mathcal{W}^{op}$.

Lemma 3.4.2. If $L_1 : \mathcal{C}_1 \Rightarrow \mathcal{D}_1$ and $L_2 : \mathcal{C}_2 \Rightarrow \mathcal{D}_2$ are localization functors for classes of morphisms $\mathcal{W}_1$ and $\mathcal{W}_2$, then the product functor $L_1 \times L_2 : \mathcal{C}_1 \times \mathcal{C}_2 \Rightarrow \mathcal{D}_1 \times \mathcal{D}_2$ is a localization functor for the product class $\mathcal{W}_1 \times \mathcal{W}_2$ if both $\mathcal{W}_1$ and $\mathcal{W}_2$ contain identity morphisms in $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

Lemma 3.4.3. Let $L_1 : \mathcal{C}_1 \Rightarrow \mathcal{C}_2$ and $L_2 : \mathcal{C}_2 \Rightarrow \mathcal{C}_3$ be localization functors for classes of morphisms $\mathcal{W}_1$ and $\mathcal{W}_2$ on $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. Let $\mathcal{W}_3$ be a class of morphisms on $\mathcal{C}_1$ such that:

(1) $\mathcal{W}_3$ is inverted by $L_1 \ggg L_2$;
(2) $\mathcal{W}_1 \subset \mathcal{W}_3$;
(3) $\mathcal{W}_2$ is contained in the essential image of $\mathcal{W}_3$ by L_1.

Then, the functor $L_1 \ggg L_2$ is a localization functor for $\mathcal{W}_3$.[12]

In order to prove these lemmas, the general strategy is as follows:

- using equivalence of categories, show that we may assume that the given functors are the functors Q of the constructed localized categories 3.1;
- show that in this particular case, the expected functor is a localization functor because it satisfies the *strict* universal property 3.2;

The predicate `L.IsLocalization W` was made a *type class*, which informally means that it is a variable in Lean that we do not need to pass explicitly to lemmas and definitions. It is worth noting that the lemmas 3.4.1 and 3.4.2 are *instances*: this basically means that if we know `L.IsLocalization W`, and that for some reason, we need to know `L.op.IsLocalization W.op`, then the later assumption shall be found automatically by Lean's type class inference system.

3.5 Calculus of fractions

For the application to triangulated categories, and in particular for the construction of derived categories as triangulated categories, it is important to develop the notion of calculus of left or right fractions. If the class of morphisms $\mathcal{W}$ in the category is $\mathcal{C}$ is multiplicative (i.e. contains identities and is stable by composition), it admits a calculus of left fractions if the following conditions hold [GZ67, I 2.2]:

[12] This lemma 3.4.3 is parallel to the statement in commutative algebra that if $S_1 \subset S_2$ is an inclusion between two multiplicative subsets of a commutative ring R, then there is a canonical isomorphism $R[S_1^{-1}][S_2^{-1}] \cong R[S_2^{-1}]$.

(1) For any right fraction $X \xleftarrow{s} X' \xrightarrow{f} Y$ (i.e. $s \in \mathcal{W}$), there exists a left fraction $X \xrightarrow{f'} Y' \xleftarrow{s'} Y$ (i.e. $s' \in \mathcal{W}$) such that the following diagram is commutative:

$$\begin{tikzcd}
& Y' \\
X \arrow[ur, dotted, "f'"] & Y \arrow[u, dotted, "s'"'] \\
X' \arrow[u, "s"] \arrow[ur, "f"'] &
\end{tikzcd}$$

(2) If f and g are two morphisms $X \to Y$ and $s : X' \to X$ is a morphism in $\mathcal{W}$ such that $s \gg f = s \gg g$, then there exists $t : Y \to Y'$ in $\mathcal{W}$ such that $f \gg t = g \gg t$:

$$\begin{tikzcd}
X' \arrow[r, "s"] & X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[r, dotted, "t"] & Y'
\end{tikzcd}$$

If $L : \mathcal{C} \Rightarrow \mathcal{D}$ is a functor which inverts $\mathcal{W}$ (in particular, if L is a localization functor), then any left or right fraction as above induces a morphism $L(X) \to L(Y)$. We obtain the following lemma:

Lemma 3.5.1. Let $L : \mathcal{C} \Rightarrow \mathcal{D}$ be a localization functor for a class of morphisms $\mathcal{W}$ that admits a calculus of left fractions. If X and Y are objects in $\mathcal{C}$, any morphism $L(X) \to L(Y)$ can be represented by a left fraction.

Moreover, two left fractions $X \xrightarrow{f_1} Y_1 \xleftarrow{s_1} Y$ and $X \xrightarrow{f_2} Y_2 \xleftarrow{s_2} Y$ induce the same morphism $L(X) \to L(Y)$ if and only if there exists an object $Z \in \mathcal{C}$, and two morphisms $t_1 : Y_1 \to Z$ and $t_2 : Y_2 \to Z$ such that $f_1 \gg t_1 = f_2 \gg t_2$, and $s_1 \gg t_1 = s_2 \gg t_2 \in \mathcal{W}$:

$$\begin{tikzcd}
& Y_1 \arrow[dr, dotted, "t_1"] & \\
X \arrow[ur, "f_1"] \arrow[dr, "f_2"'] & Y \arrow[u, "s_1"'] \arrow[d, "s_2"] & Z \\
& Y_2 \arrow[ur, dotted, "t_2"'] &
\end{tikzcd}$$

This is essentially the mathematical content of [GZ67, I 2]. Many details about this construction can be found in the article [Sim06] by Carlos Simpson who also formalized this construction in Coq.

Similarly as in [Sim06], the proof of the lemma above consists in the verification that the equivalence classes of left fractions are the morphisms for a category $\mathcal{C}'$, and that the obvious functor $\mathcal{C} \Rrightarrow \mathcal{C}'$ satisfies the *strict* universal property 3.2.

3.6 Preadditive structure

Let $L : \mathcal{C} \Rrightarrow \mathcal{D}$ be a localization functor for a class of morphisms $\mathcal{W}$. In order to proceed with the localization of triangulated categories, we need to know that under certain circumstances the localized category $\mathcal{D}$ is additive (i.e. $\mathcal{D}$ is preadditive and has finite products).

It is a general fact that finite products indexed by a set I exists in $\mathcal{D}$ if and only if the diagonal functor $\mathcal{D} \Rrightarrow \mathcal{D}^I$ has a right adjoint. It follows that if we assume that $\mathcal{C}$ has products indexed by I, the functor $\mathcal{C} \Rrightarrow \mathcal{C}^I$ has such a right adjoint $F : \mathcal{C}^I \Rrightarrow \mathcal{C}$, and if we assume that the class $\mathcal{W}$ is compatible with products (i.e. if we have morphisms $f_i : X_i \to Y_i$ in $\mathcal{W}$, then the product map $\prod_i f_i$ is also in $\mathcal{W}$), then this functor F can be lifted to a functor $\widetilde{F} : \mathcal{D}^I \Rrightarrow \mathcal{D}$ if I is finite (this is related to 3.4.2). As I have formalized a theorem about the localization of adjunctions, one may obtain that $\widetilde{F}$ is the expected right adjoint, and then $\mathcal{D}$ also has finite products indexed by I.

If we assume that $\mathcal{C}$ is additive, we may use the previous construction in order to obtain that $\mathcal{D}$ has finite products. It remains to show that $\mathcal{D}$ is preadditive, i.e. that the sets of morphisms in $\mathcal{D}$ are naturally equipped with structures of abelian groups. In my first approach, I used the property that every object X in $\mathcal{C}$ is naturally equipped with an (internal) abelian group object structure, i.e. we have morphisms `0 : ⊤_ C ⟶ X` (where `⊤_ C` is the terminal object in $\mathcal{C}$), `neg : X ⟶ X` and `add : X ⨯ X ⟶ X` (where `X ⨯ X` is the categorical product of two copies of X) which satisfies the usual relations.[13] Then, as we know that the localization functor $L : \mathcal{C} \Rrightarrow \mathcal{D}$ preserves finite products, this functor from $\mathcal{C}$ to the category of commutative group objects in $\mathcal{C}$ localizes as a functor from $\mathcal{D}$ to the category of commutative abelian group objects in $\mathcal{D}$, and from this, one may obtain the expected preadditive structure on $\mathcal{D}$.

In a second approach, I have formalized the preadditive structure on the localized category using the calculus of left fractions: if $\mathcal{C}$ is preadditive and $\mathcal{W}$ admits a calculus of left fractions, then $\mathcal{D}$ is preadditive and L is an additive functor [GZ67, I 3.3]. (If we know that $\mathcal{C}$ is additive, one may deduce that $\mathcal{D}$ also has finite products by using that additive functors preserve finite products.)

[13] For example, commutative group schemes are defined as internal abelian group objects in the category of schemes (over a base).

3.7 Universe issues

3.7.1 In the context of set theory, universes were introduced in [Sga, I 0] as sets that are closed under certain set theoretic operations. The axiom of universes (an addition to ZFC) states that any set belongs to a universe. This axiom is equivalent to the condition that any cardinal is smaller than an inaccessible cardinal. In that context, if $\mathcal{U}$ is a universe, the sets that are built from sets which belong to $\mathcal{U}$ also belong to $\mathcal{U}$ (e.g. if $\varphi : I \to \mathcal{U}$ is a map, with $I \in \mathcal{U}$, then, $\cup_{i\in I} \varphi(i) \in \mathcal{U}$). However, due to the axiom of foundation, the set $\mathcal{U}$ of all sets which belong to $\mathcal{U}$ is not an element of $\mathcal{U}$, i.e. $\mathcal{U} \notin \mathcal{U}$.

In Lean, universes are part of the syntax. Types like $\mathbf{N}$ or $\mathbf{R}$ are types in the zeroth universe: they are terms in the type `Type 0` (or just `Type`). But, `Type 0` is also a type: if we ask Lean, using the syntax `#check Type 0`, we get that `Type 0` is a term in `Type 1`. Roughly speaking, we may think that `Type 0` plays the role of a universe $\mathcal{U}_0$ in set theory, and that `Type 1` is the smallest universe $\mathcal{U}_1$ which contains $\mathcal{U}_0$. More precisely, Mario Carneiro has proved in [Car19] that Lean's type theory is equiconsistent relative to ZFC and the existence of sequences of inaccessible cardinals of arbitrary finite length.

3.7.2 In MATHLIB's category theory, when we introduce the variables for a category $\mathcal{C}$, we may proceed like this:

```
universe v u
variable (C : Type u) [Category.{v} C]
```

It is important to note that two universes are involved. First, we say that the type of objects of $\mathcal{C}$ is in the universe u. The second variable `[Category.{v} C]` expresses that we have a category structure on $\mathcal{C}$ such that for all objects X and Y, the type of morphisms $X \to Y$ is in the universe v. In MATHLIB, a small category corresponds to the situation where `C : Type u` and `[Category.{u} C]` for some universe u. For a large category, we would have `C : Type (v + 1)` and `[Category.{v} C]` for some universe v.

3.7.3 Assume that $\mathcal{W}$ is a class of morphisms in a category $\mathcal{C}$ such we have `C : Type u` and `[Category.{v} C]` and examine the case of the constructed localized category $\mathcal{C}[\mathcal{W}^{-1}]$ from 3.1. By construction, the type of objects of $\mathcal{C}[\mathcal{W}^{-1}]$ is $\mathcal{C}$ (or more precisely, it is a type synonym for $\mathcal{C}$, which is a type which is obviously in bijection with $\mathcal{C}$). Then, $\mathcal{C}[\mathcal{W}^{-1}]$ is a type in the same universe u. The situation becomes more complicated for morphisms $X \to Y$ in the localized category. We recall that such morphisms are equivalence classes of zigzags:

$$X \longrightarrow Z_1 \longleftarrow Z_2 \longrightarrow Z_3 \longleftarrow \dots \longleftarrow Z_n \longrightarrow Y$$

In order to “parametrize” these zigzags, we have to specify a certain natural number which is the length of the zigzag, then we have the types of morphisms $Z_i \to Z_{i\pm1}$ which are in the universe v. A source of disappointment is that we also need to specify the intermediate objects $Z_1, \ldots, Z_n$, which belong to a type in the universe u. It follows that if we denote `W.Localization : Type u` the localized category, the types of morphisms are in the universe `max u v`, i.e. we have `[Category.{max u v} W.Localization]`. The same remark applies to the constructed localized category when there is a calculus of left or right fractions, because similarly as zigzags of arbitrary length contain the data of the auxiliary objects Z_i, the data of a fraction involves one auxiliary object.

3.7.4 In certain circumstances, it is possible to show that the sets of morphisms in the localized category are v-small (i.e. they are in bijection with a type in the universe v). This is the case of the homotopy category of a model category $\mathcal{C}$, which is the localized category with respect to the class of weak equivalences of the model structure. Indeed, the fundamental lemma of homotopical algebra states:

Lemma 3.7.4.1. [Qui67, Corollary 1, §I.1] Let $\mathcal{C}$ be a model category. Let X be a cofibrant object of $\mathcal{C}$. Let Y be a fibrant object of $\mathcal{C}$. The set of homotopy classes of morphisms $X \to Y$ in $\mathcal{C}$ identifies to the set of morphisms between the images of X and Y in the homotopy category of $\mathcal{C}$.[14]

It follows that sets of morphisms in the homotopy category of a model category $\mathcal{C}$ are v-small. Indeed, if X and Y are objects in $\mathcal{C}$, there exists a cofibrant replacement $X_c \to X$ of X and a fibrant replacement $Y \to Y_f$ of Y, i.e. X_c is cofibrant, and the map $X_c \to X$ is a trivial fibration (in particular it is a weak equivalence), and similarly Y_f is fibrant, and the map $Y \to Y_f$ is a trivial cofibration. Then, it follows from the lemma that any morphism in the homotopy category between X and Y can be represented as a zigzag of the form $X \leftarrow X_c \to Y_f \leftarrow Y$. If follows that the type of morphisms between X and Y in the homotopy category identifies to a quotient of $X_c \to Y_f$ which is in `Type v`.

3.7.5 The main result of [Hov01] is that if $\mathcal{C}$ is a Grothendieck abelian category, i.e. an abelian category that has a generator and exact filtered colimits, then there is a model category structure on the category of cochain complexes in $\mathcal{C}$ indexed by $\mathbb{Z}$ (i.e. unbounded complexes) such that the class of weak equivalences is the class of quasi-isomorphisms. Using 3.7.4, it follows that types of morphisms in the derived category of $\mathcal{C}$ must be v-small. This should

[14] I have formalized this lemma as part of a test of my localization of categories software API in Lean 3.

apply in particular to the categories of modules over a ring, and categories of sheaves on a ringed site.

3.7.6 I have introduced the following type class `HasLocalization.{w} W` in order to take into account this universe issue:

```
universe w v u
variable {C : Type u} [Category.{v} C]
class HasLocalization (W : MorphismProperty C) where
  /-- the objects of the localized category. -/
  {D : Type u}
  /-- the category structure. -/
  [hD : Category.{w} D]
  /-- the localization functor. -/
  L : C ⥤ D
  [hL : L.IsLocalization W]
```

In addition to the universes u and v that are involved in the category $\mathcal{C}$, there is a third universe w, and this type class `HasLocalization.{w} W` contains the data of a choice of a localization functor $L : \mathcal{C} \Rightarrow \mathcal{D}$ such that the types of morphisms in $\mathcal{D}$ are in `Type w`.[15] When this data is available, the chosen localization functor is denoted `W.Q' : C ⥤ W.Localization'`.

The design is that if some constructions (e.g. the derived category) require the choice of a localized category, then the user may introduce the variable `[HasLocalization.{w} W]`. If the user wants to formalize a theorem where the statement does not involve localized categories but the proof does, they may prove some auxiliary definitions and lemmas under the assumption `[HasLocalization.{w} W]`, but in the proof of the theorem, they may use the following code:

```
theorem ... : ... := by
  have : HasLocalization.{max u v} W := HasLocalization.standard W
  -- from now on, we access the localized category as `W.Localization'`
  ...
```

3.7.7 In the particular case of derived categories, there is an abbreviation:

```
abbrev HasDerivedCategory := MorphismProperty.HasLocalization.{w}
  (HomologicalComplex.quasiIso C (ComplexShape.up ℤ))
```

[15] I did not introduce a second auxiliary universe for the type of objects in $\mathcal{D}$: it is assumed to be in `Type u`, similary as $\mathcal{C}$. Indeed, the type of objects in the constructed localized category $\mathcal{C}[\mathcal{W}^{-1}]$ is in bijection with $\mathcal{C}$.

Then, after a few years, when we are able to obtain a v-smallness theorem for the type of morphisms in the derived category of a Grothendieck abelian category $\mathcal{C}$ (see 3.7.5), it shall be possible to construct a term in the type `HasDerivedCategory.{v} C`.

4 The derived category

4.1 Definitions

In this formalization of the derived category of an abelian category $\mathcal{C}$, we define the derived category $D(\mathcal{C})$ (`DerivedCategory C`) as the localization of the category of cochain complexes $C(\mathcal{C})$ (indexed by $\mathbb{Z}$) with respect to quasi-isomorphisms. By definition, we have a localization functor $Q : C(\mathcal{C}) \Rightarrow D(\mathcal{C})$.

Lemma 4.1.1. Let K be a cochain complex (indexed by $\mathbb{Z}$) in an additive category. There is a cochain complex $\mathrm{Cylinder}(K)$ such that for any cochain complex L, the data of a morphism $\mathrm{Cylinder}(K) \to L$ is naturally equivalent to the data of two morphisms f and g in $K \to L$ and a homotopy between f ang g. The identity of $\mathrm{Cylinder}(K)$ corresponds to two morphisms $\iota_0 : K \to \mathrm{Cylinder}(K)$, $\iota_1 : K \to \mathrm{Cylinder}(K)$ and a homotopy between ι_0 and ι_1. Moreover, there is a homotopy equivalence $\pi : \mathrm{Cylinder}(K) \to K$ such that $\iota_0 \gg \pi = \iota_1 \gg \pi = \mathrm{id}_K$.

As homotopy equivalences are quasi-isomorphisms, it follows from this lemma that $Q(\pi)$ is an isomorphism and that $Q(\iota_0) = Q(\iota_1)$. It follows more generally that if f and g are homotopic morphisms in $C(\mathcal{C})$, then we have an equality of morphisms $Q(f) = Q(g)$ in the derived category $D(\mathcal{C})$. In other words, the functor $Q : C(\mathcal{C}) \Rightarrow D(\mathcal{C})$ induces a functor $Qh : K(\mathcal{C}) \Rightarrow D(\mathcal{C})$ from the homotopy category. It also follows that $K(\mathcal{C})$, which is a quotient category of $C(\mathcal{C})$, also identifies to the localization of $C(\mathcal{C})$ with respect to the class of homotopy equivalences.

4.1.2

$$\begin{tikzcd} C(\mathcal{C}) \arrow[r] \arrow[dr, "Q"'] & K(\mathcal{C}) \arrow[d, "Qh"] \\ & D(\mathcal{C}) \end{tikzcd}$$

Using the lemma 3.4.3, one may deduce that *via* the functor Qh, the derived category $D(\mathcal{C})$ identifies to the localization of $K(\mathcal{C})$ with respect to the class of quasi-isomorphisms in $K(\mathcal{C})$. Then, our direct construction of the category $D(\mathcal{C})$ as a localized category of $C(\mathcal{C})$ is also consistent with the more standard definition of the derived category in two steps from the original sources [Ver77] and [Ver96]: first take the quotient by homotopies, and secondly localize with respect to quasi-isomorphisms. That Qh is a localization functor shall be very important in order to obtain more structure on the category $D(\mathcal{C})$, namely the triangulated structure.

4.2 Shifts

4.2.1 When it was first introduced by Kim Morrison in mathlib3 in 2020, the original definition of a shift on a category $\mathcal{C}$ consisted of the data of an auto-equivalence of the category $\mathcal{C}$. This means that we have a functor $F : \mathcal{C} \Rightarrow \mathcal{C}$, a choice of a quasi-inverse $G : \mathcal{C} \Rightarrow \mathcal{C}$, a unit isomorphism `𝟭 C ≅ F ⋙ G`, and a counit isomorphism `G ⋙ F ≅ 𝟭 C` which satisfy the triangle identity (similarly as adjoint functors do). This definition was essentially consistent with the mathematical literature on triangulated categories where it is assumed that F is an isomorphism of categories (i.e. we have equalities `F ⋙ G = 𝟭 C` and `G ⋙ F = 𝟭 C`): it is so in [Nee01] as well as in the original definition of triangulated categories [Ver96, II 1.1.1] where Verdier assumed that there is a structure of a "**Z**-*catégorie stricte*". In this context, we may define the iteration F^n of the functor F for any $n \in \mathbf{Z}$.

4.2.2 In 2021, the definition of shifts was changed by Johan Commelin and Andrew Yang in mathlib3 PR #10573. The definition became closer to what Verdier defined as a "**Z**-*catégorie*" in [Ver96, I 1.2.2]. It was defined as a monoidal functor `Discrete ℤ ⥤ (C ⥤ C)` where the category of endofunctors `C ⥤ C` is equipped with the monoidal structure given by the composition of functors. This means that as part of the data of the shift on the category, we have functors `F n : C ⥤ C` for all `n : ℤ`, an isomorphism `zero : F 0 ≅ 𝟭 C`, and a family of isomorphisms `add n m : F (n + m) ≅ F n ⋙ F m` for all `n m : ℤ`, which satisfy three compatibilities (associativity, left unitality and right unitality), which expresses a certain coherence relative to the identities $(n+m)+p = n+(m+p)$, $0+n = n$ and $n+0 = n$ in $\mathbb{Z}$. In mathlib, shifts on categories are defined in this way for any additive monoid A.

4.2.3 When I ported this from Lean 3 to Lean 4, I felt it was difficult to prove some identities because the definitions about shifts were always unfolded into terms revealing the internals of the API for monoidal functors. In the mathlib4 PR #3039, I decided to put a certain isolation between the API for shifts from that of monoidal functors: abbreviations were replaced by definitions, and more shift-specific lemmas were introduced. This improved automation significantly: for example, in mathlib4 PR #3047, almost all the proofs in the file about the rotation of triangles were now found automatically `by aesop_cat`.

4.2.4 Assume that the category $\mathcal{C}$ is equipped with a shift by an additive monoid A. For any `n : A`, the shift functor denoted `F n` above can be obtained as `shiftFunctor C n : C ⥤ C`, and we have the notation `X⟦n⟧` for `(F n).obj X`. Similarly, the isomorphisms `add` can be obtained as `shiftFunctorAdd C n m`. The associativity compatibility that was mentioned before expresses the commutativity of the following pentagon, where the maps are obtained by using the natural isomorphisms `add` and the functors `F`:

$$
\begin{tikzcd}
 & \mathtt{X⟦n + m⟧⟦p⟧} \arrow[dr, "\sim"] & \\
\mathtt{X⟦(n + m) + p⟧} \arrow[ur, "\sim"] \arrow[d, equal] & & \mathtt{X⟦n⟧⟦m⟧⟦p⟧} \\
\mathtt{X⟦n + (m + p)⟧} \arrow[dr, "\sim"'] & & \\
 & \mathtt{X⟦n⟧⟦m + p⟧} \arrow[uur, "\sim"'] &
\end{tikzcd}
$$

The morphism on the left corresponds to an equality between two objects which follows from the associativity relation $(n+m)+p = n+(m+p)$ in the additive monoid. Out of the context of a formalization, we may take this equality for granted, and we may not even make it appear on the diagram: indeed, it does not appear in the equation [Ver96, I (1.2.1.3)]. In the formalization in Lean, we have to take this into consideration.

4.2.5 So as to mitigate this issue, I have introduced a natural isomorphism `X⟦k⟧ ≅ X⟦i⟧⟦j⟧` whenever the equality `h : i + j = k` holds: this is the definition `shiftFunctorAdd' C i j k h`. In particular, in the pentagon diagram above, the composition `X⟦(n + m) + p⟧ ≅ X⟦n⟧⟦m + p⟧` can be obtained directly as `shiftFunctorAdd' C n (m + p) ((n + m) + p) _`. It follows that the associativity can be phrased more generally in terms of these isomorphisms `shiftFunctorAdd'` when we have elements a_1, a_2, a_3, a_{12}, a_{23} and a_{123} in the additive monoid which satisfy

$a_1 + a_2 = a_{12}$, $a_2 + a_3 = a_{23}$ and $a_1 + a_2 + a_3 = a_{123}$: it says that the two ways to identify `X⟦a₁₂₃⟧` and `X⟦a₁⟧⟦a₂⟧⟦a₃⟧` using `X⟦a₁₂⟧⟦a₃⟧` or `X⟦a₁⟧⟦a₂₃⟧` as an intermediate object are the same.

4.2.6 Let $\mathcal{C}$ be a preadditive category. The category of cochain complexes $C(\mathcal{C})$ in $\mathcal{C}$ is equipped with a shift by $\mathbb{Z}$. If `K : CochainComplex C ℤ` and `n : ℤ`, then `K⟦n⟧` is the cochain complex such that by definition we have `(K⟦n⟧).X i = K.X (i + n)` in degree `i` and the differentials are obtained by multiplying by $(-1)^n$ the differentials of `K`.

The very fact that we are able to describe all the shifts `K⟦n⟧` and not just `K⟦1⟧` and `K⟦−1⟧` shows how relevant the design change 4.2.2 by Johan Commelin and Andrew Yang was. Indeed, if only the shifts by ±1 were part of the structure of the shift on the category `CochainComplex C ℤ`, then the explicit description of the iterated shifts for all $n \in \mathbf{Z}$ would have to be phrased by saying that the nth iteration of the shift functor is isomorphic to the explicit functor above. As a result, future applications may require that we prove coherence properties of these isomorphisms! It is a much better design to bundle all of this data and properties in the definition of the shift.

4.2.7 The shift on the category $C(\mathcal{C})$ induces a shift on the category $K(\mathcal{C})$. The mathematical reason is that $K(\mathcal{C})$ is the quotient category of $C(\mathcal{C})$ by relations that are compatible with the shift on $C(\mathcal{C})$: if two morphisms of cochain complexes `f` and `g` are homotopic, then so are their shifts `f⟦n⟧'` and `g⟦n⟧'` for all `n : ℤ`. (In mathlib, shifts of objects are denoted `X⟦n⟧` while shifts of morphisms are denoted `f⟦n⟧'`.)

In a certain sense, up to isomorphisms, the shift on $K(\mathcal{C})$ is determined by the shift on the category $C(\mathcal{C})$. A simple way to express a compatibility between the shifts on $C(\mathcal{C})$ and $K(\mathcal{C})$ consists in the formulation of a compatibility of the quotient functor $C(\mathcal{C}) \Rightarrow K(\mathcal{C})$ with respect to the shifts, which is done in the next paragraph.

4.2.8 Let $F : \mathcal{C} \Rightarrow \mathcal{D}$ be a functor between two categories equipped with a shift by an additive monoid A. In order to express that F commutes with the shift, we should at least provide isomorphisms `shiftFunctor C a ⋙ F ≅ F ⋙ shiftFunctor D a` for all `a : A`, but these isomorphisms should also satisfy some compatibilities. Indeed, when $a = 0$, we always have an obvious isomorphism:

```
def CommShift.isoZero :
  shiftFunctor C (0 : A) ⋙ F ≅ F ⋙ shiftFunctor D (0 : A) := ...
```

From the data of such isomorphisms for two elements a and b, we may also construct a commutation isomorphism for the shift by $a+b$:

```
def CommShift.isoAdd {a b : A}
  (e₁ : shiftFunctor C a ⋙ F ≅ F ⋙ shiftFunctor D a)
  (e₂ : shiftFunctor C b ⋙ F ≅ F ⋙ shiftFunctor D b) :
  shiftFunctor C (a + b) ⋙ F ≅ F ⋙ shiftFunctor D (a + b) :=
```

Using these definitions, I have formalized a type class `F.CommShift A` which expresses that F commutes with the shifts by A as follows:

```
class CommShift where
 iso (a : A) : shiftFunctor C a ⋙ F ≅ F ⋙ shiftFunctor D a
 zero : iso 0 = CommShift.isoZero F A := by aesop_cat
 add (a b : A) : iso (a + b) = CommShift.isoAdd (iso a) (iso b) := by aesop_cat
```

Under the assumption `F.CommShift A`, the API for this type class `Functor.CommShift` allows to access the isomorphism `iso a` of the class above as `F.commShiftIso a`.

After I had formalized this, I found that the compatibility `add` was phrased in the commutative diagram [Ver96, I (1.2.3.2)]. (In Verdier's notations, the condition `zero` was automatically satisfied because in the language of fibered categories used in [Ver96, I 1], the shift functors on a category are the base-change functors given by a *normalized* cleavage.)

If `τ : F₁ ⟶ F₂` is a natural transformation between two functors $\mathcal{C} \Rightarrow \mathcal{D}$ which commute with the shifts on $\mathcal{C}$ and $\mathcal{D}$, I have also introduced a type class `NatTrans.CommShift τ A` which expresses a compatibility between τ and the isomorphisms `F₁.commShiftIso a` and `F₂.commShiftIso a` given by the commutation of F_1 and F_2 with the shifts.

4.2.9 Let $\mathcal{C}$ be an abelian category. As we have shown that the functor $Qh : K(\mathcal{C}) \Rightarrow D(\mathcal{C})$ is a localization functor, the shift on $K(\mathcal{C})$ induces a shift by $\mathbb{Z}$ on the derived category $D(\mathcal{C})$: this construction of a localized shift, and the previously mentioned construction of a quotient shift 4.2.7, are actually both a special case of a more general construction. Both localization functors and quotient functors share a common property: these are functors $F : \mathcal{C} \Rightarrow \mathcal{D}$ such that for any category $\mathcal{E}$, the functor $(\mathcal{D} \Rightarrow \mathcal{E}) \Rightarrow (\mathcal{C} \Rightarrow \mathcal{E})$ given by the precomposition with F is fully faithful. Under this assumption on F (with $\mathcal{E} := \mathcal{D}$), if $\mathcal{C}$ is equipped with a shift by an additive monoid A, and if the functors `shiftFunctor C a ⋙ F` can be lifted as functors `s a : D ⥤ D`, then the category $\mathcal{D}$ can be equipped with a shift by A with the functors `s a` as shift functors, and the functor F commutes with the shifts.

It follows that the three categories $C(\mathcal{C})$, $K(\mathcal{C})$ and $D(\mathcal{C})$ are equipped with shifts by $\mathbb{Z}$. As $Q : C(\mathcal{C}) \Rrightarrow D(\mathcal{C})$ identifies to the composition of the quotient functor $C(\mathcal{C}) \Rrightarrow K(\mathcal{C})$ and of the localization functor $K(\mathcal{C}) \Rrightarrow D(\mathcal{C})$, we may deduce that Q also commutes with the shifts.

4.3 The triangulated structure on the homotopy category

4.3.1 If a category $\mathcal{T}$ is equipped with a shift by $\mathbb{Z}$, a triangle is a diagram $\mathsf{X}_1 \longrightarrow \mathsf{X}_2 \longrightarrow \mathsf{X}_3 \longrightarrow \mathsf{X}_1[\![1]\!]$, which may be drawn as:

$$\begin{tikzcd} & X_3 \arrow[dl, "+1"'] & \\ X_1 \arrow[rr] & & X_2 \arrow[ul] \end{tikzcd}$$

A pretriangulated structure on a preadditive category $\mathcal{T}$ equipped with a shift by $\mathbb{Z}$ involves the data of a predicate on triangles: the triangles which satisfy this predicate are called *distinguished triangles* [Ver96, II 1.1.1]. The axioms of pretriangulated and triangulated categories are statements about these distinguished triangles. Pretriangulated categories were formalized in mathlib in 2021 by Luke Kershaw. I have added many basic lemmas about pretriangulated categories, and I have formalized the statement of the "octahedron axiom" (TR IV) of triangulated categories as this shall be used in 4.4.

Definition 4.3.2 ([Ver96, I 3])**.** If $\mathcal{C}$ is an additive category, we shall say that a triangle in the homotopy category of cochain complexes indexed by $\mathbb{Z}$ in $\mathcal{C}$ is a distinguished triangle if it is isomorphic to the image of a standard triangle attached to a morphism $f : K \to L$ in the category `CochainComplex C` $\mathbb{Z}$:

$$\begin{tikzcd} & \mathrm{Cone}(f) \arrow[dl, "+1"'] & \\ K \arrow[rr] & & L \arrow[ul] \end{tikzcd}$$

where $\mathrm{Cone}(f)$ (or `mappingCone f`) is the cochain complex defined by $\mathrm{Cone}(f)^n := K^{n+1} \boxplus L^n$ and the differentials are given by the matrix $\begin{pmatrix} -d & 0 \\ f & d \end{pmatrix}$. The map $L \to \mathrm{Cone}(f)$ is the obvious injection, while $\mathrm{Cone}(f) \to K[\![1]\!]$ is the *opposite* of the first projection.

The fact that the homotopy category of an additive category is a pretriangulated category was already obtained by Andrew Yang and Kim Morrison in the LTE. They used the definition of distinguished triangles from the Stacks project https://stacks.math.columbia.edu/tag/014P [Jon+]: a triangle in the homotopy category is distinguished if and only if it is isomorphic to the triangle $X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_1[\![1]\!]$ that is associated to a degreewise split short exact sequence of complexes $0 \longrightarrow X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow 0$ (the choice of a splitting in each degree allows the definition of a 1-cocycle from X_3 to X_1, which corresponds to a morphism $X_3 \longrightarrow X_1[\![1]\!]$). I have followed more closely the original definition 4.3.2 by Verdier [Ver96, I 3].[16]

4.3.3 Calculus of cochains

In order to verify the axioms of triangulated categories for the homotopy category of cochain complexes, it is convenient to introduce the cochain complex of morphisms $\mathrm{Hom}^\bullet(K, L)$ for two cochain complexes K and L [Con00, p. 10]. It is a cochain complex in the category of abelian groups which in degree n consists of families of morphisms $K^p \to L^q$ for all $(p,q) \in \mathbf{Z}^2$ such that $p + n = q$. The differentials on $\mathrm{Hom}^\bullet(K, L)$ are defined in such a way that an element in $\mathrm{Hom}^0(K, L)$ is a cocycle if and only if it corresponds to a morphism of cochain complexes $K \to L$. In mathlib, I have implemented this definition as the cochain complex `HomComplex K L`. However, the more convenient related definitions are the types of cochains `HomComplex.Cochain K L n` and cocycles `HomComplex.Cocycle K L n` in this complex. We obtain the expected correspondence between morphisms of cochain complexes and 0-cocycles as:

```
def equivHom : (K ⟶ L) ≃+ Cocycle K L 0 where
  ...
```

Similarly, two morphisms of cochain complexes $\varphi_i : K \to L$ for $i \in \{1, 2\}$ are homotopic if and only if the corresponding cochains are cohomologous:

```
def equivHomotopy (φ₁ φ₂ : K ⟶ L) :
    Homotopy φ₁ φ₂ ≃
      { z : Cochain K L (-1) //
        Cochain.ofHom φ₁ = δ (-1) 0 z + Cochain.ofHom φ₂ } where
  ...
```

Then, in the verification of the axioms of triangulated categories, as we need to construct morphisms from or to mapping cones of morphisms (and homotopies), it is very convenient to manipulate them as cochains. For example, we have the following definitions for the left and right inclusions in the mapping cone of a morphism $f : K \to L$ and the first and second projections from it:

[16] However, I have followed the better sign conventions of [Con00, p. 8]. Originally, there was no sign in the definition of the morphism $\mathrm{Cone}(f) \to K[\![1]\!]$ in [Ver96, I 3.2.2.5] and [Har66, I §2].

```
def inl : Cochain K (mappingCone f) (−1) := ...
def inr : L ⟶ mappingCone f := ....
def fst : Cocycle (mappingCone f) K 1 := ...
def snd : Cochain (mappingCone f) L 0 := ...
```

An important structure on cochains is that they can be composed: if `z₁ : Cochain K L a` and `z₂ : Cochain L M b`, we may construct their composition in `Cochain K M (a + b)`. Actually, similarly as for shifts 4.2.5, I defined the composition `z₁.comp z₂ h : Cochain K M c` for any `c : ℤ` such that `h : a + b = c` holds. Computations can be achieved using lemmas like:

```
lemma inl_fst :
  (inl f).comp (fst f).1 (neg_add_self 1) = Cochain.ofHom (𝟙 K) := ...
```

Here, `inl f` is of degree −1 and `fst f` is of degree 1. Then, their naive composition would be of degree $(-1)+1$. By using the design above, which forces the user to provide the equation `neg_add_self 1 : (−1) + 1 = 0`, we obtain a 0-cochain.

This design for the composition of cochains is different from the design for the product of homogeneous elements in graded rings in mathlib [WZ22]. If I had followed a similar design as for graded rings, I would have introduced the type of the direct sum of the abelian groups `Cochain K L n` for all $n \in \mathbb{Z}$,[17] and made computations in this type. On the one hand, doing so may have eased the automation of the proof of some identifies (especially those where the associativity of the composition is used), but in many situations, especially when K or L is obtained by shifting other cochain complexes, we need to specify explicitly well chosen integers in order to do computations.

4.3.4 The octahedron axiom

The main ingredient in order to obtain the octahedron axiom for the homotopy category is that if $f : X_1 \to X_2$ and $g : X_2 \to X_3$ are composable morphisms in the category of cochain complexes, then there is a distinguished triangle:

$$\begin{tikzcd} & \mathrm{Cone}(g) \arrow[dl, "+1"'] & \\ \mathrm{Cone}(f) \arrow[rr] & & \mathrm{Cone}(f \gg g) \arrow[ul] \end{tikzcd}$$

[17] This type could be defined as the subtype of the families of morphisms $\alpha_{p,q} : K^p \to L^q$ for all p and q in $\mathbb{Z}$ consisting of those families $(\alpha_{p,q})_{p,q}$ such that $\{q-p, \alpha_{_{p,q}} \neq 0\}$ is finite.

In order to do that, we need to construct an isomorphism in the homotopy category between $\mathrm{Cone}(g)$ and the mapping cone of the canonical map $\mathrm{Cone}(f) \to \mathrm{Cone}(f \gg g)$. We actually show that the former is a deformation retract of the latter. In order to verify this, we need to construct a homotopy between two endomorphisms of $\mathrm{Cone}(\mathrm{Cone}(f) \to \mathrm{Cone}(f \gg g))$. If we unfold the definitions, we see that in degree n, we have $\mathrm{Cone}(f)^n \simeq X_1^{n+1} \boxplus X_2^n$ and $\mathrm{Cone}(f \gg g)^n \simeq X_1^{n+1} \boxplus X_3^n$, and it follows that we have an isomorphism:

$$\mathrm{Cone}(\mathrm{Cone}(f) \to \mathrm{Cone}(f \gg g))^n \cong (X_1^{n+2} \boxplus X_2^{n+1}) \boxplus (X_1^{n+1} \boxplus X_3^n)$$

We see that the two endomorphisms we are trying to relate by a homotopy can be thought as 4×4-matrices consisting of cochains from X_i to X_j of various degrees $d \in \{-2, -1, 0, 1, 2\}$ for various tuples (i, j). Then, once the candidate homotopy is found, the equality of cochains that we need to show can be interpreted as an identity between two 4×4-matrices, which corresponds to 16 identities between cochains.

One of the difficulties when proving equalities involving cochains is related to the associativity of the composition of cochains. In category theory, when f, g and h are composable morphisms, a term $(f \gg g) \gg h$ is automatically replaced by the `simp` tactic as $f \gg (g \gg h)$ (and then, the parentheses are redundant). Automation in mathlib relies on this design choice that compositions are "associated towards the right". We may try to do the same for cochains, but the lemma expressing the associativity of the composition of cochains is phrased as follows:

```
lemma comp_assoc {n₁ n₂ n₃ n₁₂ n₂₃ n₁₂₃ : ℤ}
  (z₁ : Cochain F G n₁) (z₂ : Cochain G K n₂) (z₃ : Cochain K L n₃)
  (h₁₂ : n₁ + n₂ = n₁₂) (h₂₃ : n₂ + n₃ = n₂₃) (h₁₂₃ : n₁ + n₂ + n₃ = n₁₂₃) :
  (z₁.comp z₂ h₁₂).comp z₃ (show n₁₂ + n₃ = n₁₂₃ by rw [← h₁₂, h₁₂₃]) =
    z₁.comp (z₂.comp z₃ h₂₃) (by rw [← h₂₃, ← h₁₂₃, add_assoc]) := by ...
```

The issue is that if z_i for $i \in \{1, 2, 3\}$ are composable cochains of degrees n_i, we may consider the composition "$z_1 \gg z_2$" only if we provide an integer n_{12} such that $n_1 + n_2 = n_{12}$, and similarly for all the other compositions in the identity "$(z_1 \gg z_2) \gg z_3 = z_1 \gg (z_2 \gg z_3)$". We could be tempted to define the composition as a cochain of degree $n_1 + n_2$, but using $n_1 + n_2$ is not always the best choice: for example, if z_1 is of degree $n - 1$ for some integer n, and z_2 is of degree 1, we probably want to consider the composition "$z_1 \gg z_2$" as a cochain of degree n rather than $(n - 1) + 1$. To be more specific about the associativity, if we have all the data and properties in order to make sense of the LHS of the equality (i.e. we have the integers n_{12} and n_{123}), in general, there is no preferred choice for the integer $n_{23} = n_2 + n_3$ which appears in the RHS. This is the reason why we cannot make a nice general `simp` lemma out of `comp_assoc`. In a few carefully selected situations, there is a preferred choice for n_{23}, in which case we may state specialized `simp` lemmas, e.g. when one of the n_i is zero. Another case is when $n_2 = -n_3$: we may choose $n_{23} := 0$. For example, if α is a cochain of degree n, the identity "$(\alpha \gg \mathrm{inl}\, f) \gg \mathrm{fst}\, f = \alpha$" can be proved automatically `by simp`: as $\mathrm{inl}\, f$ and $\mathrm{fst}\, f$ are respectively of degrees -1 and 1, the associativity relation is applied, so that the `simp` tactic is able to get the successive equalities $(\alpha \gg \mathrm{inl}\, f) \gg \mathrm{fst}\, f = \alpha \gg (\mathrm{inl}\, f \gg \mathrm{fst}\, f) = \alpha \gg 1 = \alpha$.

These difficulties with the associativity relation are one of the reasons why, in the proof of the axioms of triangulated categories (including the octahedron axiom), we do not prove the expected identities between cochains by doing only computations in the types of cochains. In a few situations, in order to prove an equality between two cochains α and β in $\mathrm{Hom}^n(K, L)$, instead of proving directly $\alpha = \beta$, we show it componentwise, i.e. using suitable extensionality lemmas, we have to show that for all p and q such that $p + n = q$, the morphisms $K^p \to L^q$ that are part of α and β are equal. Here again, we have to make reasonable choices for the integers p and q, e.g. if $n = 0$, we may assume that q is p by definition: two 0-cochains α and β are equal if and only if the corresponding morphisms $K^p \to L^p$ are equal for all p. By doing so, the associativity issue with the composition of cochains disappears because we can use the associativity of the composition of morphisms in the category $\mathcal{C}$.

This formalization of the triangulated structure on the homotopy category of cochain complexes indexed by $\mathbb{Z}$ in any additive category entered mathlib in January 2024 (PR #9550).

4.4 The localization theorem for triangulated categories

I have formalized the following theorem, which is essentially [Ver96, II 2.2.6]:

Theorem 4.4.1. Let $\mathcal{T}$ be a pretriangulated category. Let $\mathcal{W}$ be a class of morphisms in $\mathcal{T}$ that has a calculus of left fractions (see 3.5) and is compatible with the triangulation. The localized category $\mathcal{T}[\mathcal{W}^{-1}]$ has a pretriangulated category structure such that the localization functor $\mathcal{T} \Rightarrow \mathcal{T}[\mathcal{W}^{-1}]$ is a triangulated functor. Moreover, if $\mathcal{T}$ is triangulated and that $\mathcal{W}$ also has a calculus of right fractions, then $\mathcal{T}[\mathcal{W}^{-1}]$ is a triangulated category.

The condition that $\mathcal{W}$ is compatible with the triangulation means that it is invariant by the shift functors and that if $T : X_1 \to X_2 \to X_3 \to X_1[1]$ and $T' : X'_1 \to X'_2 \to X'_3 \to X'_1[1]$ are distinguished triangles, then any commutative square where the maps s_1 and s_2 are in $\mathcal{W}$ can be extended to a morphism of triangles $T \to T'$ such that s_3 is also in $\mathcal{W}$:

$$\begin{tikzcd}
X_1 \arrow[r] \arrow[d, "s_1"] & X_2 \arrow[r] \arrow[d, "s_2"] & X_3 \arrow[r] \arrow[d, dotted, "s_3"] & X_1[1] \arrow[d, "s_1[1]"] \\
X_1' \arrow[r] & X_2' \arrow[r] & X_3' \arrow[r] & X_1'[1]
\end{tikzcd}$$

In my formalization, the category $\mathcal{T}[\mathcal{W}^{-1}]$ can be replaced by the target category of any localization functor $L : \mathcal{T} \Rightarrow \mathcal{D}$ with respect to $\mathcal{W}$. The verification of the axioms of (pre)triangulated categories for $\mathcal{D}$ is relatively easy. The only difficulty consists in the construction of the expected structures on the category $\mathcal{D}$: the preadditive structure is obtained using the calculus of fractions (see 3.6), and the shift functors are obtained by localization (see 4.2.9).

4.4.2 In order to construct classes of morphisms $\mathcal{W}$ satisfying the assumptions of 4.4.1, the main construction is that of a class of morphisms $\mathcal{W}_S$ attached to a triangulated subcategory S of a triangulated category $\mathcal{T}$. A triangulated subcategory S consists of the data of a predicate on objects on $\mathcal{T}$ which is satisfied by a zero object, is stable by shifts and such that if $X_1 \to X_2 \to X_3 \to X_1[1]$ is a distinguished triangle, and if X_1 and X_3 are in S, then X_2 is isomorphic to an object in S. Then, we define $\mathcal{W}_S$ as the class of morphisms $X_1 \to X_2$ which fit into a distinguished triangle $X_1 \to X_2 \to X_3 \to X_1[1]$ with $X_3 \in S$.

The verification of the expected properties for $\mathcal{W}_S$ uses the octahedron axiom, and this is the reason why this axiom was introduced [Ver96, II 2.2.12]. The full subcategory of $\mathcal{T}$ corresponding to S is automatically endowed with a triangulated structure, and the localized category with respect to $\mathcal{W}_S$ is also denoted using the quotient notation $\mathcal{C}/S$: this is known as the Verdier quotient of $\mathcal{C}$ by S.

4.4.3 The triangulated structure on the derived category $D(\mathcal{C})$ of an abelian category is obtained by taking the Verdier quotient of the triangulated category $K(\mathcal{C})$ by the triangulated subcategory $\mathcal{A}$ consisting of acyclic objects in the homotopy category $K(\mathcal{C})$.

In order to check that $\mathcal{A}$ is a triangulated subcategory and that the class $W_{\mathcal{A}}$ is precisely the class of quasi-isomorphisms, we show that the homology functor $H^0 : K(\mathcal{C}) \Rightarrow \mathcal{C}$ is a homological functor, i.e. that if $X_1 \to X_2 \to X_3 \to X_1[1]$ is a distinguished triangle in $K(\mathcal{C})$, then $H^0(X_1) \to H^0(X_2) \to H^0(X_3)$ is an exact sequence. This can be proven directly by diagram chasing using the definition 4.3.2 of distinguished triangles, or this can be deduced from the homology sequence associated to a short exact sequence of cochain complexes. Then, the triangulated subcategory $\mathcal{A}$ can be understood as the "kernel" of the homological functor

H^0: an object X belongs to $\mathcal{A}$ if and only if for any $n \in \mathbf{Z}$, $H^0(X[n])$ is zero, or equivalently, if for any $n \in \mathbf{Z}$, $H^n(X)$ is zero.

This construction of the derived category was submitted to mathlib in PR #11806 in March 2024, and it was merged in June 2024.

5 Ongoing works

In this section, I outline some ongoing works. Very significant parts of these are already formalized, but it may take a certain time before they enter mathlib.

5.1 Ext-groups

Before this work, Ext-groups (or Ext-modules) were defined in mathlib only in abelian categories that have enough projectives. This applies to the category of modules over a ring, which is sufficient for the application to group cohomology [Liv23] and to local cohomology (whose definition was formalized in 2023 by Emily Witt and Kim Morrison). However, we cannot use this definition in the context of categories of sheaves over a topological space, or a Grothendieck topology. Moreover, if $0 \to X_1 \to X_2 \to X_3 \to 0$ is a short exact sequence in an abelian category $\mathcal{C}$, and if $Y \in \mathcal{C}$, we should have two long exact sequences of Ext:

$$\cdots \to \mathrm{Ext}^n(Y,X_1) \to \mathrm{Ext}^n(Y,X_2) \to \mathrm{Ext}^n(Y,X_3) \xrightarrow{\delta} \mathrm{Ext}^{n+1}(Y,X_1) \to \dots$$

$$\cdots \to \mathrm{Ext}^n(X_3,Y) \to \mathrm{Ext}^n(X_2,Y) \to \mathrm{Ext}^n(X_1,Y) \xrightarrow{\delta} \mathrm{Ext}^{n+1}(X_3,Y) \to \dots$$

Before the present work, mathlib did not contain the statement of these exact sequences.[18] However, the second exact sequence was formalized in the LTE using the definition of $\mathrm{Ext}^n(-,Y)$ as the right derived functors of $\mathrm{Hom}(-,Y)$.

I have formalized the fact that the functor $\mathcal{C} \Rightarrow D(\mathcal{C})$ which sends an object $X \in \mathcal{C}$ to the cochain complex $\cdots \to 0 \to 0 \to X \to 0 \to 0 \to \dots$ where X sits in degree 0 is fully faithful (this is the inclusion of the heart of a t-structure, see 5.2). As a result, we may identify $\mathcal{C}$ to a full subcategory of the derived category $D(\mathcal{C})$. Given two objects X and Y in $\mathcal{C}$, we may define $\mathrm{Ext}^n(X,Y)$ as $\mathrm{Hom}_{D(\mathcal{C})}(X,Y[n])$ for any $n \in \mathbb{N}$. It is then easy to obtain the expected long exact sequences.

There are two difficulties in this process:

[18] These long exact sequences are now in mathlib (see PR #14515 and #15092.)

- make sign conventions for the definition of the connecting homomorphisms δ consistent with the existing mathematical literature [Con00, §1.3];
- find a partial solution to the universe issue 3.7: in general, the type of morphisms between two arbitrary objects in the derived category $\mathcal{C}$ may lie in a larger universe than the universe of morphisms in the category $\mathcal{C}$. However, if $\mathcal{C}$ has enough projectives or enough injectives, it is possible to show that the types of morphisms $\mathrm{Hom}_{D(\mathcal{C})}(X, Y[n])$ are small for X and Y in $\mathcal{C}$. It follows that Ext-groups can be defined by "shrinking" these types to the smaller universe.

5.2 t-structures

5.2.1 If $\mathcal{C}$ is an abelian category, the homology functors $H^q : D(\mathcal{C}) \Rightarrow \mathcal{C}$ can be used in order to define full subcategories $D(\mathcal{C})^{\geq n}$ and $D(\mathcal{C})^{\leq n}$ for all $n \in \mathbb{Z}$:

- an object $X \in D(\mathcal{C})$ is $\geq n$ if $H^q(X)$ is zero whenever $q < n$.
- an object $X \in D(\mathcal{C})$ is $\leq n$ if $H^q(X)$ is zero whenever $n < q$.

These full subcategories satisfy the following important properties:

- If $X \leq 0$ and $Y \geq 1$, then $\mathrm{Hom}_{D(\mathcal{C})}(X, Y) = 0$.
- For any $Z \in D(\mathcal{C})$, there exists a distinguished triangle $X \to Z \to Y \to X[1]$ with $X \leq 0$ and $Y \geq 1$.

More generally, a t-structure [Bei+18, §1.3] on a triangulated category $\mathcal{T}$ consists of the data of full subcategories $\mathcal{T}^{\geq n}$ and $\mathcal{T}^{\leq n}$ satisfying similar properties as those stated above. I would like to emphasize the clarity of the exposition in [Bei+18, §1.3]: it was easy to translate the written arguments into formal proofs in Lean/mathlib.

5.2.2 Given a t-structure on a triangulated category $\mathcal{T}$, I have formalized the verification that the heart $\mathcal{T}_{\geq 0} \cap \mathcal{T}_{\leq 0}$ of the t-structure is an abelian category [Bei+18, Théorème 1.3.6]. For example, the heart of the canonical t-structure defined above on the derived category of $D(\mathcal{C})$ of $\mathcal{C}$ is $\mathcal{C}$ itself.

5.2.3 An important feature of t-structures is that the distinguished triangle $X \to Z \to Y \to X[1]$ with $X \leq 0$ and $Y \geq 1$ is functorial in Z: one may define functors $\tau_{\leq 0}$, $\tau_{\geq 1}$ and a natural transformation $\delta : \tau_{\geq 1} Z \to (\tau_{\leq 0} Z)[1]$ such that the following triangle is distinguished for all Z:

$$\tau_{\leq 0} Z \to Z \to \tau_{\geq 1} Z \xrightarrow{\delta} (\tau_{\leq 0} Z)[1]$$

More generally, one may define functors $\tau_{\geq n}$, $\tau_{\leq n}$, $\tau_{>n} := \tau_{\geq n+1}$ and $\tau_{<n} := \tau_{\leq n-1}$. An important result is that there are natural isomorphisms $\tau_{\geq a}(\tau_{\leq b}Z) \cong \tau_{\leq b}(\tau_{\geq a}Z)$ for all a and b in $\mathbb{Z}$.

5.2.4 If $a \leq b \leq c$, then there is a natural distinguished triangle for all X:

$$\begin{tikzcd} & \tau_{<c}(\tau_{\geq b}X) \arrow[dl, "+1"'] & \\ \tau_{<b}(\tau_{\geq a}X) \arrow[rr] & & \tau_{<c}(\tau_{\geq a}X) \arrow[ul] \end{tikzcd}$$

This may be extended for a, b and c in $\mathbf{Z} \cup \{\pm\infty\}$ if we set $\tau_{<-\infty}X = \tau_{\geq+\infty}X = 0$ and $\tau_{\geq-\infty}X = \tau_{<+\infty}X = X$. Then, to any object X in a triangulated category $\mathcal{T}$ equipped with a t-structure is attached what Verdier calls *"un objet spectral de type $\mathbf{Z} \cup \{\pm\infty\}$ à valeurs dans $\mathcal{T}$"* [Ver96, II 4.1.2]. Surprisingly, I did not find any mention of this spectral object in [Bei+18]. The formalization in Lean is long and technical, but it shall be a very important tool in the construction of spectral sequences 5.4.4.

5.3 Derived functors

5.3.1 If $F : \mathcal{C} \Rightarrow \mathcal{D}$ is an additive functor between abelian categories, there is an induced triangulated functor $K(\mathcal{C}) \Rightarrow K(\mathcal{D})$ on the homotopy categories. In general, this functor does not preserve quasi-isomorphisms, unless F is exact. In other words, the composed functor $K(\mathcal{C}) \Rightarrow D(\mathcal{D})$ may not send quasi-isomorphisms in $K(\mathcal{C})$ to isomorphisms in $D(\mathcal{D})$, i.e. there is no "commutative diagram" of functors:

$$\begin{tikzcd} K(\mathcal{C}) \arrow[r, "F"] \arrow[d] & K(\mathcal{D}) \arrow[d] \\ D(\mathcal{C}) \arrow[r, dotted] & D(\mathcal{D}) \end{tikzcd}$$

However, it is often possible to construct a functor $RF : D(\mathcal{C}) \Rightarrow D(\mathcal{D})$, and instead of an isomorphism between the composed functors, we have a natural transformation α:

$$\begin{tikzcd} K(\mathcal{C}) \arrow[r, "F"] \arrow[d] \arrow[dr, phantom, "\swarrow\alpha"] & K(\mathcal{D}) \arrow[d] \\ D(\mathcal{C}) \arrow[r, "RF"] & D(\mathcal{D}) \end{tikzcd}$$

The tuple (RF, α) is said to be the right derived functor when it is universal (i.e. it is an initial object in the category of such diagrams). By definition, when it exists, such a right derived functor is a left Kan extension of $K(\mathcal{C}) \Rrightarrow D(\mathcal{D})$ along the localization functor $K(\mathcal{C}) \Rrightarrow D(\mathcal{C})$.

5.3.2 Kan extensions have been introduced in mathlib by Yuma Mizuno in PR #6552 in the context of bicategories. Mathematically speaking, the notion of right derived functor, which is a special case of a left Kan extension of functors, can be thought as a particular case of left Kan extension in bicategories. However, in terms of formalization in Lean, we cannot use the same software API for both because in the diagram above, the categories may not have the same universe parameters (see also 3.7).

I have formalized similar definitions of left Kan extensions in the context of categories and functors, and developed the particular case of right derived functors. I was able to formalize the following theorem:

Theorem 5.3.3. Let $F : \mathcal{C} \Rrightarrow \mathcal{D}$ be an additive functor between abelian categories. We assume that $\mathcal{C}$ has enough injectives. The induced triangulated functor $F : K^+(\mathcal{C}) \Rrightarrow K^+(\mathcal{D})$ on the homotopy categories of bounded below cochain complexes can be right derived as a triangulated functor $RF : D^+(\mathcal{C}) \Rrightarrow D^+(\mathcal{D})$.

The proof of this theorem involves two aspects. First, the main technical results are that if we denote Injectives($\mathcal{C}$) the full subcategory of $\mathcal{C}$ consisting of injective objects, for any $L \in K^+(\mathcal{C})$, there is a quasi-isomorphism $L \to L'$ with $L' \in K^+(\text{Injectives}(\mathcal{C}))$.[19] These statements generalize the well known fact that if $X \in \mathcal{C}$, then X admits an injective resolution $0 \to X \to I^0 \to I^1 \to I^2 \to \dots$. Secondly, I have formalized the notion of "derivability structure" introduced by Bruno Kahn and Georges Maltsiniotis [KM08]: this is a general abstract machinery in order to construct derived functors. (The details about this categorical notion are too technical to be described here.) Using the properties mentioned above, I have shown that the inclusion functor $K^+(\text{Injectives}(\mathcal{C})) \Rrightarrow K^+(\mathcal{C})$, thought as a morphism of localizers (here, it means that this functor sends isomorphisms to quasi-isomorphisms), is a right derivability structure; moreover, the induced functor $K^+(\text{Injectives}(\mathcal{C})) \Rrightarrow D^+(\mathcal{C})$ is an equivalence of categories.[20] Here, the consequence is that any functor $G : K^+(\mathcal{C}) \to \mathcal{E}$ from the bounded below homotopy category has a right derived functor $RG : D^+(\mathcal{C}) \to \mathcal{E}$, and

[19] The lemmas that I have formalized essentially correspond to the factorization axiom CM5 for the model category structure on $C^+(\mathcal{C})$ when $\mathcal{C}$ has enough injectives.

[20] The dual result of this is that if $\mathcal{C}$ has enough projectives, then the category $D^-(\mathcal{C})$ is equivalent to $K^-(\text{Projectives}(\mathcal{C}))$, which shows the compatibility of this approach and the LTE 1.4.

for any cochain complex $L \in K^+(\text{Injectives}(\mathcal{C}))$, the canonical map $\alpha_L : G(L) \to RG(L)$ is an isomorphism.

5.3.4 Even though this is not discussed in [KM08], the notion of derivability structure behaves well with respect to products of categories. It follows that this framework is suitable for the study of derived functors of functors of several variables. For example, if $\mathcal{A}$ is an abelian category that is equipped with a monoidal category structure, in such a way that any object is a quotient of a flat object in a functorial manner, there should be a "flat" left derivability structure on $K^-(\mathcal{A})$, and the product derivability structure of two copies of it should allow the construction of the derived functor of the tensor product functor $K^-(\mathcal{A}) \times K^-(\mathcal{A}) \Rightarrow K^-(\mathcal{A})$. Then, it should be possible to obtain a monoidal category structure on $D^-(\mathcal{A})$. Similarly, if we assume the existence of K-flat resolutions (in the sense of [Spa88, 5.1]), it should be possible to obtain a monoidal category structure on the full derived category $D(\mathcal{A})$. In particular, this could be used in order to define and study the properties of $\text{Tor}_n^A(M, N)$ when M and N are modules over a commutative ring A.

5.4 Spectral sequences

5.4.1 I have formalized the definition of a spectral sequence as follows:

```
variable (C : Type*) [Category C] [Abelian C]
  {ι : Type*} (c : ℤ → ComplexShape ι) (r₀ : ℤ)

structure SpectralSequence where
  -- the `r`th page of the spectral sequence
  page' (r : ℤ) (hr : r₀ ≤ r) : HomologicalComplex C (c r)
  -- the homology of a page identifies to the next page
  iso' (r r' : ℤ) (hrr' : r + 1 = r') (pq : ι) (hr : r₀ ≤ r) :
    (page' r hr).homology pq ≅ (page' r' (by omega)).X pq
```

With this definition, a spectral sequence E starting on page r_0 consists of a family of homological complexes, the pages of E, which are defined for all integers $r \geq r_0$. All the pages are complexes that are indexed by the same type ι (typically $\mathbf{Z}^2$ or $\mathbf{N}^2$), but the shapes of differentials are specified for each page individually. The data in E also contains an isomorphism saying that the homology of a page identifies to the next page.

For example, I have made an abbreviation `CohomologicalSpectralSequence` for spectral sequences indexed by $\mathbf{Z}^2$, with differentials of bidegree $(r, 1-r)$ on the rth page. Even though I have made general definitions, allowing general shapes of spectral sequences, in this exposition, I shall focus on this particular case, and use the standard mathematical notation $E_r^{p,q}$.

5.4.2 Stabilization

Given a spectral sequence E, let us fix (p,q). For any $r \geq r_0$, $E_{r+1}^{p,q}$ identifies to a subquotient of $E_r^{p,q}$. If the differentials of E_r are such that the differential to and from $E_r^{p,q}$ are both zero, then we have a canonical isomorphism $E_r^{p,q} \cong E_{r+1}^{p,q}$. If this holds for all big enough integers r, we may define the limit object $E_\infty^{p,q}$ in such a way that for a big enough r, we shall have a canonical isomorphism $E_r^{p,q} \cong E_\infty^{p,q}$. In my formalization, I have defined a type class `E.HasPageInfinityAt pq` for `pq : ι` in order to express that this stabilization phenomenon occurs. We may say that the spectral sequence E stabilizes if this property holds for all (p,q) (even though there may not be a uniform bound on r).

5.4.3 Convergence

Assuming that the spectral sequence E stabilizes, we may say that it strongly converges in degree n to a certain object H^n of the abelian category if we provide a filtration Fil_i on H^n such that the $E_\infty^{p,q}$ for all $p+q=n$ identify to the graded object of the successive quotients of this filtration, which should also satisfy $\mathrm{Fil}_i = 0$ for a small enough i and $\mathrm{Fil}_i = X$ for a big enough i.

The convergence can be used in order to facilitate computations. For example, I have formalized the 5-terms exact sequence in low degrees of a strongly convergent first quadrant E_2-cohomological spectral sequence:

$$0 \to E_2^{1,0} \to H^1 \to E_2^{0,1} \to E_2^{2,0} \to H^2$$

Indeed, the filtration on H^1 that is given by the convergence translates as a short exact sequence $0 \to E_\infty^{1,0} \to H^1 \to E_\infty^{0,1} \to 0$, and the automatic stabilization of first quadrant spectral sequences gives identifications $E_\infty^{1,0} \cong E_2^{1,0}$ and $E_\infty^{0,1} \cong E_3^{0,1}$. Now, as E_3 is the homology of the E_2-page, we see that $E_3^{0,1}$ identifies to the kernel of the differential $d_2 : E_2^{0,1} \to E_2^{2,0}$. Then, the cokernel of this differential d_2 identifies to $E_3^{2,0} \cong E_\infty^{2,0}$ which is a subobject of H^2 because of the convergence in degree 2.

5.4.4 Construction of spectral sequences

Let us assume that we have a spectral object $(E_{a,b})$ of type $\mathbf{Z} \cup \{\pm\infty\}$ in a triangulated category $\mathcal{T}$. The basic data of E include objects $E_{a,b} \in \mathcal{T}$ whenever we have an inequality $a \leq b$ in $\mathbf{Z} \cup \{\pm\infty\}$. More precisely, these $E_{a,b}$ should be part of a functor from the category of arrows in the ordered set $\mathbf{Z}\cup\{\pm\infty\}$: in particular, if $a\leq b$, $a'\leq b'$, $a\leq a'$ and $b\leq b'$, there is a map $E_{a,b}\to E_{a',b'}$. The additional data is that of a (functorial) connecting morphism $\delta: E_{b,c}\to E_{a,b}[1]$ whenever $a\leq b\leq c$, in such a way that the following triangle is distinguished :

$$E_{a,b}\to E_{a,c}\to E_{b,c}\to E_{a,b}[1]$$

Among the examples of spectral objects in triangulated categories, we have the spectral object $(\tau_{<b}(\tau_{\geq a}X))$ attached to any object in a triangulated category that is equipped with a t-structure 5.2.4. If a cochain complex K in an abelian category $\mathcal{C}$ is equipped with a filtration Fil_a indexed by $\mathbf{Z}\cup\{\pm\infty\}$, then there is an associated spectral object in the homotopy category $K(\mathcal{C})$ defined by $X_{a,b}:=\mathrm{Cone}(\mathrm{Fil}_a\to\mathrm{Fil}_b)$: the expected distinguished triangles are given by 4.3.4. After applying the triangulated functor $K(\mathcal{C})\Rightarrow D(\mathcal{C})$, we obtain the spectral object in $D(\mathcal{C})$ of a filtered object in $C(\mathcal{C})$.

If we have a homological functor $H^0:\mathcal{T}\to\mathcal{C}$ from the triangulated category $\mathcal{T}$ to an abelian category $\mathcal{C}$, these distinguished triangles lead to long exact sequences for all $n\in\mathbf{Z}$:

$$\cdots\to H^n(E_{a,b})\to H^n(E_{a,c})\to H^n(E_{b,c})\xrightarrow{\delta} H^{n+1}(E_{a,b})\to\dots$$

These objects $(H^n(E_{a,b}))$ are now part of a spectral object with values in the abelian category $\mathcal{C}$ as it was defined in [Ver96, II 4.1.4] and [CE99, XV §7]. Spectral sequences are attached to any spectral object in an abelian category [Ver96, II 4.3.3]. I have formalized this construction and studied the stabilization and convergence of the associated spectral sequences.

Spectral sequences can also be constructed using the notion of exact couple in an abelian category [Mas52]. Slightly less data are involved in exact couples as compared to spectral objects. For example, in the case of a filtered complex K in an abelian category, with the spectral object approach, we consider simultaneously all the long exact sequences in homology deduced from all the short exact sequences $0\to\mathrm{Fil}_b/\mathrm{Fil}_a\to\mathrm{Fil}_c/\mathrm{Fil}_a\to\mathrm{Fil}_c/\mathrm{Fil}_b\to 0$ whenever $a\leq b\leq c$ in $\mathbf{Z}\cup\{\pm\infty\}$. In the exact couple approach, the data would only involve the long exact sequences deduced from the exact sequences $0\to\mathrm{Fil}_{a-1}\to\mathrm{Fil}_a\to\mathrm{Fil}_a/\mathrm{Fil}_{a-1}\to 0$ for $a\in\mathbf{Z}$. I have opted for spectral objects because all the data involved in the spectral sequence can be described very directly in terms of the data of the spectral object, whereas in the exact couple approach, pages are constructed through an inductive process known as the "derived exact couple" [Mas52, I §4].

5.4.5 Examples of spectral sequences

The machinery for the construction of spectral sequences which was outlined above shows that in order to construct a spectral sequence in an abelian category $\mathcal{C}$, it suffices to provide two data:

- a homological functor $H^0 : \mathcal{T} \Rightarrow \mathcal{C}$,
- a spectral object E in the triangulated category $\mathcal{T}$.

The main example of a homological functor is the homology functor attached to a t-structure, which obviously includes the functor $H^0 : D(\mathcal{C}) \Rightarrow \mathcal{C}$ when $\mathcal{C}$ is an abelian category. It is also important to note that if $H^0 : \mathcal{T} \Rightarrow \mathcal{C}$ is a homological functor, then for any triangulated functor $F : \mathcal{T}' \Rightarrow \mathcal{T}$, the composition $F \ggg H^0 : \mathcal{T}' \Rightarrow \mathcal{C}$ is also a homological functor.

In order to construct a spectral object in the derived category, we may use any filtration on a cochain complex. In particular, we may use the canonical filtration which is related to the spectral objects attached to t-structures (see 5.2.4), but we may also use the stupid filtration. Similarly, the total complex of a bicomplex may also be equipped with a filtration by the rows or by the columns.

I have completely formalized the following theorem, which is the Grothendieck spectral sequence for the composition of right derived functors:

Theorem 5.4.5.1 ([Gro57, 2.4.1]). Let $F : \mathcal{A} \to \mathcal{B}$ and $G : \mathcal{B} \to \mathcal{C}$ be additive functors between abelian categories. We assume that $\mathcal{A}$ and $\mathcal{B}$ have enough injectives. Moreover, we assume that for any injective object I in $\mathcal{C}$, the object $F(I)$ is "acyclic" for G, i.e. the canonical map $G(F(I)) \to RG(F(I))$ is an isomorphism.[21]

Then, for any $X \in \mathcal{A}$, there is a first quadrant cohomological spectral sequence with first page $E_2^{p,q} \cong R^pG(R^qF(X))$ which converges to $R^{p+q}(F \ggg G)(X)$.

We apply the machinery of spectral sequences to the homological functor $RG \ggg H^0 : D^+(\mathcal{B}) \Rightarrow \mathcal{C}$ and the spectral object attached to $RF(X)$ using the canonical t-structure on $D^+(\mathcal{B})$. This spectral sequence converges to $H^{p+q}(RG(RF(X)))$, but the assumptions on F and G allow to show that the natural transformation $R(F \ggg G) \to RF \ggg RG$ is an isomorphism. It follows that there is a canonical isomorphism $R^{p+q}(F \ggg G)(X) \cong H^{p+q}(RG(RF(X)))$.

[21] If G is left exact, this means that $(R^pG)(F(I)) = 0$ for $p > 0$.

References

[ABR08] J. Aransay, C. Ballarin, and J. Rubio. "A mechanized proof of the basic perturbation lemma". In: *J. Automat. Reason.* 40.4 (2008), pages 271–292. ISSN: 0168-7433,1573-0670. DOI: 10.1007/s10817-007-9094-x.

[Bei+18] A. A. Beilinson, J. Bernstein, P. Deligne, and O. Gabber. *Analyse et topologie sur les espaces singuliers. I : Faisceaux pervers*. Volume 100. Astérisque. Société Mathématique de France, Paris, 2018, 172 pages.

[Ber74] G. Bergman. "A note on abelian categories – translating element-chasing proofs, and exact embedding in abelian groups". 1974. URL: http://math.berkeley.edu/~gbergman/papers/unpub/elem-chase.pdf.

[Bor94] F. Borceux. *Handbook of categorical algebra. 2*. Volume 51. Encyclopedia of Mathematics and its Applications. Categories and structures. Cambridge University Press, Cambridge, 1994, xviii+443 pages. ISBN: 0-521-44179-X.

[Buz+22] K. Buzzard, C. Hughes, K. Lau, A. Livingston, R. Fernández Mir, and S. Morrison. "Schemes in Lean". In: *Exp. Math.* 31.2 (2022), pages 355–363. ISSN: 1058-6458,1944-950X. DOI: 10.1080/10586458.2021.1983489.

[Car19] M. Carneiro. *The Type Theory of Lean*. Master thesis. 2019. URL: https://github.com/digama0/lean-type-theory/releases.

[CE99] H. Cartan and S. Eilenberg. *Homological algebra*. Princeton Landmarks in Mathematics. With an appendix by David A. Buchsbaum, Reprint of the 1956 original. Princeton University Press, Princeton, NJ, 1999, xvi+390 pages. ISBN: 0-691-04991-2.

[CT22] J. Commelin and A. Topaz. *Completion of the Liquid Tensor Experiment*. 2022. URL: https://leanprover-community.github.io/blog/posts/lte-final/.

[Con00] B. Conrad. *Grothendieck duality and base change*. Volume 1750. Lecture Notes in Mathematics. Springer-Verlag, Berlin, 2000, vi+296 pages. ISBN: 3-540-41134-8. DOI: 10.1007/b75857.

[Del68] P. Deligne. "Théorème de Lefschetz et critères de dégénérescence de suites spectrales". In: *Inst. Hautes Études Sci. Publ. Math.* 35 (1968), pages 259–278. ISSN: 0073-8301,1618-1913. URL: http://www.numdam.org/item/PMIHES_1968__35__107_0.

[Del74] P. Deligne. “La conjecture de Weil. I”. In: *Inst. Hautes Études Sci. Publ. Math.* 43 (1974), pages 273–307. ISSN: 0073-8301,1618-1913. URL: http://www.numdam.org/item?id=PMIHES_1974__43__273_0.

[DRB17] F. van Doorn, J. von Raumer, and U. Buchholtz. “Homotopy type theory in Lean”. In: *Interactive theorem proving*. Volume 10499. Lecture Notes in Comput. Sci. Springer, Cham, 2017, pages 479–495. ISBN: 978-3-319-66107-0; 978-3-319-66106-3. DOI: 10.1007/978-3-319-66107-0.

[DSS] X. Dousson, F. Sergeraert, and Y. Siret. *The Kenzo program*. URL: http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo/.

[GZ67] P. Gabriel and M. Zisman. *Calculus of fractions and homotopy theory*. Volume Band 35. Ergebnisse der Mathematik und ihrer Grenzgebiete [Results in Mathematics and Related Areas]. Springer-Verlag New York, Inc., New York, 1967, x+168 pages.

[Gro57] A. Grothendieck. “Sur quelques points d’algèbre homologique”. In: *Tohoku Math. J. (2)* 9 (1957), pages 119–221. ISSN: 0040-8735,2186-585X. DOI: 10.2748/tmj/1178244839.

[GMP24] B. Guillemet, A. Mahboubi, and M. Piquerez. *Machine-Checked Categorical Diagrammatic Reasoning*. 2024. arXiv: 2402.14485 [cs.LO]. URL: https://arxiv.org/abs/2402.14485.

[Har66] R. Hartshorne. *Residues and duality*. Volume No. 20. Lecture Notes in Mathematics. Lecture notes of a seminar on the work of A. Grothendieck, given at Harvard 1963/64, With an appendix by P. Deligne. Springer-Verlag, Berlin-New York, 1966, vii+423 pages.

[Him20] M. Himmel. “Diagram chasing in interactive theorem proving”. Bachelorarbeit. Karlsruher Institut für Technologie, 2020. URL: https://pp.ipd.kit.edu/uploads/publikationen/himmel20bachelorarbeit.pdf.

[HR25] M. Himmel and J. Riou. *Theorems about abelian categories*. 2025. URL: https://leanprover-community.github.io/blog/posts/abelian-categories/.

[Hov01] M. Hovey. “Model category structures on chain complexes of sheaves”. In: *Trans. Amer. Math. Soc.* 353.6 (2001), pages 2441–2457. ISSN: 0002-9947,1088-6850. DOI: 10.1090/S0002-9947-01-02721-0.

[Ill90] L. Illusie. “Catégories dérivées et dualité: travaux de J.-L. Verdier”. In: *Enseign. Math. (2)* 36.3-4 (1990), pages 369–391. ISSN: 0013-8584.

[Jon+] A. J. de Jong et al. *The Stacks project*. URL: https://stacks.math.columbia.edu/.

[KM08] B. Kahn and G. Maltsiniotis. "Structures de dérivabilité". In: *Adv. Math.* 218.4 (2008), pages 1286–1318. ISSN: 0001-8708,1090-2082. DOI: 10.1016/j.aim.2008.03.010.

[LHF23] J. Limperg and A. Halkjær From. "Aesop: White-Box Best-First Proof Search for Lean". In: *CPP 2023: Proceedings of the 12th ACM SIGPLAN Internation Conference on Certified Programs and Proofs*. 2023, pages 253–266. DOI: https://doi.org/10.1145/3573105.3575671.

[Liv23] A. Livingston. "Group Cohomology in the Lean Community Library". In: *14th International Conference on Interactive Theorem Proving (ITP 2023)*. Edited by A. Naumowicz and R. Thiemann. Volume 268. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 22:1–22:17. ISBN: 978-3-95977-284-6. DOI: 10.4230/LIPIcs.ITP.2023.22.

[Mas52] W. S. Massey. "Exact couples in algebraic topology. I, II". In: *Ann. of Math. (2)* 56 (1952), pages 363–396. ISSN: 0003-486X. DOI: 10.2307/1969805.

[Mit64] B. Mitchell. "The full imbedding theorem". In: *Amer. J. Math.* 86.3 (1964), pages 619–637. ISSN: 0002-9327,1080-6377. DOI: 10.2307/2373027.

[Mon22] Y. Monbru. "Towards automatic diagram chasing". M1 report. École Normale Supérieure Paris-Saclay. 2022. URL: https://github.com/ymonbru/Diagram-chasing/blob/main/MONBRU_Yannis_Rapport.pdf.

[MU21] L. de Moura and S. Ullrich. "The Lean 4 Theorem Prover and Programming Language". In: *Automated Deduction – CADE 28*. Edited by A. Platzer and G. Sutcliffe. Cham: Springer International Publishing, 2021, pages 625–635. ISBN: 978-3-030-79876-5. URL: https://link.springer.com/chapter/10.1007/978-3-030-79876-5_37.

[Mur22] B. Murphy. *Formalization of the Brouwer fixed-point theorem in Lean 3*. 2022. URL: https://github.com/Shamrock-Frost/BrouwerFixedPoint.

[Nee01] A. Neeman. *Triangulated categories*. Volume 148. Annals of Mathematics Studies. Princeton University Press, Princeton, NJ, 2001, pages viii+449. ISBN: 0-691-08685-0; 0-691-08686-9. DOI: 10.1515/9781400837212.

[Qui67] D. G. Quillen. *Homotopical algebra*. Volume No. 43. Lecture Notes in Mathematics. Springer-Verlag, Berlin-New York, 1967, iv+156 pp. (not consecutively paged).

[RS13] J. Rubio and F. Sergeraert. *Constructive Homological Algebra and Applications*. 2013. arXiv: 1208.3816 [math.KT]. URL: https://arxiv.org/abs/1208.3816.

[Sch21] P. Scholze. *Half a year of the Liquid tensor experiment: Amazing developments*. 2021. URL: https://xenaproject.wordpress.com/2021/06/05/half-a-year-of-the-liquid-tensor-experiment-amazing-developments/.

[Ser51] J.-P. Serre. “Homologie singulière des espaces fibrés. Applications”. In: *Ann. of Math. (2)* 54 (1951), pages 425–505. ISSN: 0003-486X. DOI: 10.2307/1969485.

[Sim06] C. Simpson. “Explaining Gabriel-Zisman localization to the computer”. In: *J. Automat. Reason.* 36.3 (2006), pages 259–285. ISSN: 0168-7433,1573-0670. DOI: 10.1007/s10817-006-9038-x.

[Spa88] N. Spaltenstein. “Resolutions of unbounded complexes”. In: *Compositio Math.* 65.2 (1988), pages 121–154. ISSN: 0010-437X,1570-5846. URL: http://www.numdam.org/item?id=CM_1988__65_2_121_0.

[Spa95] E. H. Spanier. *Algebraic topology*. Springer-Verlag, New York, 1995, xvi+528 pages. ISBN: 0-387-94426-5.

[The20] The mathlib community. “The Lean mathematical library”. In: *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*. 2020, pages 367–381. DOI: 10.1145/3372885.3373824.

[Sga] *Théorie des topos et cohomologie étale des schémas. Tome 1: Théorie des topos*. Volume Vol. 269. Lecture Notes in Mathematics. Séminaire de Géométrie Algébrique du Bois-Marie 1963–1964 (SGA 4), Dirigé par M. Artin, A. Grothendieck, et J. L. Verdier. Avec la collaboration de N. Bourbaki, P. Deligne et B. Saint-Donat. Springer-Verlag, Berlin-New York, 1972, xix+525 pages.

[Ver77] J.-L. Verdier. “Catégories dérivées: quelques résultats (état 0)”. In: *Cohomologie étale*. Volume 569. Lecture Notes in Math. Springer, Berlin, 1977, pages 262–311. ISBN: 3-540-08066-X; 0-387-08066-X.

[Ver96] J.-L. Verdier. “Des catégories dérivées des catégories abéliennes”. In: *Astérisque* 239 (1996). With a preface by Luc Illusie, Edited and with a note by Georges Maltsiniotis, xii+253 pages. ISSN: 0303-1179,2492-5926.

[WZ22] E. Wieser and J. Zhang. "Graded Rings in Lean's Dependent Type Theory". In: *Intelligent Computer Mathematics - 15th International Conference, CICM 2022, Tbilisi, Georgia, September 19-23, 2022, Proceedings*. Edited by K. Buzzard and T. Kutsia. Volume 13467. Lecture Notes in Computer Science. Springer, 2022, pages 122–137. DOI: 10.1007/978-3-031-16681-5_8.