

# Constructive Category Theory and Applications to Equivariant Sheaves

DISSERTATION

zur Erlangung des Grades eines  
Doktors der Naturwissenschaften

vorgelegt von  
Sebastian Posur

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät  
der Universität Siegen  
Siegen 2017

GUTACHTER

Prof. Dr. Mohamed Barakat, Universität Siegen  
Prof. Dr. Frank-Olaf Schreyer, Universität des Saarlandes

TAG DER MÜNDLICHEN PRÜFUNG: 21.06.2017

## Preface

I would like to thank all the people who helped and supported me in the last years.

I express my deep gratitude to Mohamed Barakat for introducing me to the wonderful field of constructive mathematics, for his invaluable advices and continuous encouragement.

I am also very grateful for the inspiring comments I received from Frank-Olaf Schreyer and his honest interest in my thesis.

Special thanks to my great colleague Sebastian Gutsche for the exciting time we had creating and implementing our software project CAP.

I also express my thanks to all my friends and colleagues in Aachen and Siegen for the pleasant working atmosphere we have in our departments and the mutual support.

Last, I would like to thank Tassja and our families for their unconditional support and love.



## Summary

In this thesis we create a purely categorical framework for cohomology computations of  $G$ -equivariant coherent sheaves on projective space for a finite group  $G$ . For this, we develop three different sub-frameworks: First, we construct a skeletal tensor category  $\mathrm{SRep}_k(G)$  equivalent to the representation category  $\mathrm{Rep}_k(G)$  of  $G$ . Second, we design, in the context of an arbitrary abelian category, an algorithm for computing spectral sequences which is suitable for a direct computer implementation, i.e., it only uses categorical constructions provided by the axioms of an abelian category. Last, we describe how to internalize the exterior algebra  $E$  and its modules in a tensor category.

Combining our three sub-frameworks yields an algorithm for computing spectral sequences within the category of  $E$ -modules internal to  $\mathrm{SRep}_k(G)$ . Thanks to an equivariant version of the famous BGG-correspondence, we can use such an algorithm for computing cohomology groups of  $G$ -equivariant sheaves on projective space. Furthermore, this algorithm allows us to compute a new invariant called spectral cohomology table which in this thesis is proven to be stronger than the classical cohomology table.

Since our framework can be described in purely categorical language, a software project in **GAP** facilitating the implementation of abstract categories and categorical algorithms was born during the writing of this thesis: **CAP** (Categories, Algorithms, Programming). The categorical framework along with all algorithms presented in this thesis is implemented in **CAP**.



## Zusammenfassung

In dieser Arbeit geben wir der Kohomologieberechnung  $G$ -äquivarianter Garben auf dem projektiven Raum für endliche Gruppen  $G$  einen konstruktiven kategoriellen Rahmen. Dazu gehen wir in drei Schritten vor: Wir konstruieren zuerst eine skeletale Tensor-kategorie  $\text{SRep}_k(G)$ , welche äquivalent zur Darstellungskategorie  $\text{Rep}_k(G)$  von  $G$  ist. Danach entwerfen wir einen ausschließlich auf den Axiomen einer abelschen Kategorie beruhenden Algorithmus zur Berechnung von Spektralsequenzen. Im Anschluss behandeln wir die äußere Algebra  $E$  und ihre Moduln intern in einer Tensor-kategorie.

Die Kombination dieser drei Schritte ergibt einen Algorithmus zur Berechnung von Spektralsequenzen innerhalb der Kategorie von  $E$ -Moduln intern in  $\text{SRep}_k(G)$ . Dank der berühmten BGG-Korrespondenz kann dieser Algorithmus zur Bestimmung von Kohomologiegruppen  $G$ -äquivarianter Garben auf dem projektiven Raum genutzt werden. Darüber hinaus ermöglicht er die Berechnung von Spektral-Kohomologietabellen – eine neue Invariante, welche stärker ist als klassische Kohomologietabellen, wie in dieser Arbeit gezeigt wird.

Durch die konstruktive Anwendung rein kategorieller Konzepte entstand während des Verfassens dieser Arbeit ein Software-Projekt in GAP zur Vereinfachung der Implementierung abstrakter Kategorien und kategorieller Algorithmen: CAP (Categories, Algorithms, Programming). Alle Ergebnisse und Algorithmen dieser Arbeit wurden in CAP realisiert und implementiert.





## Contents

Introduction	9
The CAP Project	11
1. Syntax	11
2. Semantics	13
Chapter 1. Constructive Category Theory	17
1. Preliminaries	18
1.1. Categories, Functors, Natural Transformations	19
1.2. Naturality	21
1.3. Images	26
1.4. Limits	28
2. Additive, Abelian, and Coproduct Categories	34
2.1. Additive Categories	34
2.2. Abelian Categories	37
2.3. Coproduct Categories	41
3. Constructing Tensor Categories	44
3.1. Bilinear Bifunctors	44
3.2. Monoidal Categories	47
3.3. Skeletal Tensor Categories	55
3.3.1. Representation Category of Finite Groups	55
3.3.2. Defining a Bifunctor	56
3.3.3. Defining an Associator	58
3.3.4. Defining a Braiding	61
3.3.5. Defining Unitors	63
3.3.6. Defining Duals	65
3.3.7. Skeletal Representation Category of Finite Groups	69
3.3.8. Graded Group Representations	69
3.3.9. Example: $S_3$	70
3.3.10. Example: $D_8$ and $Q_8$	78
3.3.11. Example: Subgroup of Order 1000 of the Automorphism Group of the Horrocks-Mumford Bundle	80
Chapter 2. Constructive Homological Algebra	83
1. Generalized Morphisms	85
1.1. Additive Relations	85

1.2. Categorification of Additive Relations	87
1.3. Computation Rules for Generalized Morphisms	92
1.4. Data Structures for Generalized Morphisms	98
1.5. Epi-Mono Factorizations of Generalized Morphisms	105
1.6. Attributes and Properties of Generalized Morphisms	107
1.6.1. Canonical Objects in the Underlying Abelian Category	107
1.6.2. Honest Morphisms	110
1.7. Reasoning with the Canonical Objects	111
2. Diagram Chases and Spectral Sequences	115
2.1. Constructive Diagram Chases	115
2.2. Generalized Cochain Complexes	116
2.3. Spectral Sequence of a Filtered Complex	119
2.4. Computing Spectral Sequences	124
Chapter 3. Applications to Equivariant Sheaves	127
1. (Co)homological Invariants	129
1.1. Natural Filtrations	129
1.2. Spectral Betti Tables	131
2. Equivariant Modules over the Exterior Algebra	132
2.1. Actions and Coactions	132
2.2. Equivariant Modules	136
2.3. Internal Algebra	137
2.3.1. Exterior Algebra	137
2.3.2. Dual of Exterior Algebra	139
2.3.3. Internal Free Resolutions	140
2.3.4. Internal Cofree Resolutions	142
3. Computations with Equivariant Sheaves	143
3.1. BGG Correspondence	143
3.2. Equivariant Cohomology Tables	146
3.2.1. Equivariant BGG Correspondence	146
3.2.2. Equivariant Cohomology Table of the Horrocks-Mumford Bundle	147
3.3. Spectral Cohomology Tables	154
3.3.1. Definition	154
3.3.2. Spectral Cohomology Table of $\Omega_{\mathbb{P}_k^2}$	155
3.3.3. Hilbert Series of Unbounded Cochain Complexes	156
3.3.4. Spectral Cohomology Tables of Supernatural Sheaves	157
3.3.5. Spectral Cohomology Tables vs. Cohomology Tables	159
3.3.6. Spectral Cohomology Table of the Horrocks-Mumford Bundle	160
List of Figures	167
Bibliography	169
Index	171

## Introduction

Constructive methods in mathematics are a powerful tool to enhance and deepen our understanding of mathematical structures. They provide computational means for otherwise inaccessible invariants and let us rethink and replace non-constructive with direct approaches.

In this thesis we illustrate the power of constructive methods in mathematics by pursuing the following computational goal: Create a framework for cohomology computations of  $G$ -equivariant coherent sheaves on projective space over a field  $k$  for a finite group  $G$  and implement this framework in the computer algebra system GAP [GAP14].

Our approach relies on an equivariant version of the famous BGG-correspondence [BGG78], which cleverly links cohomology computations with resolutions of  $\mathbb{Z}$ -graded modules over an exterior algebra  $E$  (see [EFS03]). A classical method in computer algebra to implement a framework for such computations would begin with an implementation of the ring in question, which in our case is the crossed product ring  $E \rtimes G$ , a ring that equals the presumably very high dimensional vector space  $E \otimes_k k[G]$  on the level of elements. Such an implementation has to rely on performing effective arithmetics with  $E$  and  $G$ . The next step would be the implementation of a solver for linear equations with coefficients in  $E \rtimes G$ , since we ultimately aim at projective resolutions of modules. Such a solver would have to deal with very high dimensional equations and for making it work satisfactorily, it would have to exploit the special arithmetics of  $E \rtimes G$  in technical demanding ways.

This thesis suggests following a new path in computer algebra laid out by category theory. Invariants of mathematical objects, including cohomology groups of equivariant sheaves, often admit purely categorical descriptions. This simple fact has a significant computational implication: These invariants do not depend on a particular choice of data structure for the mathematical object in question, but solely rely on its abstract categorical context. Building on that, we want to approach our computational goal using the powerful machinery of constructive category theory. We will focus on purely categorical constructions like limits, tensor products, and duals, i.e., constructions working with objects and morphisms, and ignore those set-theoretic constructions relying on elements and functions. This paradigm shift, though, comes at a price. We give up easily axiomatizable algebraic structures like rings and get heavily loaded categorical structures like tensor categories. The upshot is a bigger variety in our choice of data structures, the design of algorithms flexible in use, and the possibility to concretely employ far-reaching categorical concepts like the internalization of modules, which ultimately lets us compute with  $E \rtimes G$ -modules

without touching elements of  $E$  or  $G$  at all (see Subsection III.2.2 for categorical models of  $E \rtimes G$ -modules).

We achieve our computational goal by describing, implementing, and in the end combining three different categorical sub-frameworks: In the first chapter we construct a skeletal version  $\text{SRep}_k(G)$  of the representation category  $\text{Rep}_k(G)$  of  $G$  regarded as a tensor category over a splitting field  $k$  for  $G$ . The objects in this category are simply given by finite lists of non-negative integers, so a set-theoretic concept such as “evaluate a representation at a group element” cannot be applied. However, thanks to category theory we know that for our computational objectives, only  $\text{SRep}_k(G)$ ’s tensor structure matters. Finding this tensor structure is a computationally hard but feasible task that only has to be solved *once and for all*. Once properly constructed, working within  $\text{SRep}_k(G)$  becomes very easy.

In the second chapter we design, in the context of an arbitrary abelian category  $\mathbf{A}$ , an algorithm for computing spectral sequences, which is suitable for a direct computer implementation, i.e., it only uses categorical constructions directly provided by the axioms of  $\mathbf{A}$ . For this, we develop a calculus for dealing and reasoning with additive relations in  $\mathbf{A}$ , which we will refer to as *generalized morphisms*.

In the third chapter we describe how to *internalize* the exterior algebra  $E$  and its modules in a tensor category. The concept of an internal module allows us to abstract from the idea that a module must have an underlying set of elements, and uses the structure of tensor categories instead. Thus, we can smoothly apply the theory of internal modules to our model  $\text{SRep}_k(G)$ .

The combination of our three categorical sub-frameworks yields an algorithm for computing spectral sequences within the category of  $E$ -modules internal to  $\text{SRep}_k(G)$ . As mentioned above, an equivariant version of the BGG-correspondence can use such an algorithm for computing cohomology groups of  $G$ -equivariant sheaves on projective space. Furthermore, this algorithm allows us to compute a new invariant called *spectral cohomology table* which in this thesis is proven to be stronger than the classical cohomology table (see Theorem III.3.20).

Since our framework can be described in purely categorical language, a software project in GAP facilitating the implementation of abstract categories and categorical algorithms was born during the writing of this thesis: CAP, which is an acronym for *Categories, Algorithms, Programming*. This thesis contains a brief introductory chapter to the CAP project and the categorical framework along with all algorithms presented in this thesis is implemented in CAP.

Our highly general approach allows to reach further. The framework can be smoothly extended to equivariant computations with modules over path algebras and may in the future lead to the computation of equivariant cohomology of sheaves over varieties other than projective space, exploiting derived equivalences induced by full exceptional collections.

# The Cap Project

The CAP project is a collection of software packages for category theory implemented in the computer algebra system GAP. Its development started in December 2013 by Sebastian Gutsche and the author of this thesis, followed by major contributions to the core system from Øystein Skartsæterhagen in 2015. So far, four CAP related software packages<sup>1</sup> are distributed via the current GAP release<sup>2</sup>, more packages still under development are available at the GitHub pages<sup>3</sup> of the CAP-authors.

The name CAP is an acronym for *Categories, Algorithms, Programming*. The core system provides templates for categories possessing or equipped with extra structure, e.g., direct products, addition for morphisms, kernels and cokernels, tensor products, et cetera. On the one hand, these templates can be used to create instances of categories, for example the category of finite dimensional vector spaces over a computable field  $k$  or the category of finitely presented modules over a computable ring  $R$  (see [BLH11] for a definition of computable rings). On the other hand, these templates provide the syntax for implementing generic categorical algorithms, such as the computation of specific differentials on a page of a spectral sequence in the context of an arbitrary abelian category. In the following, we will briefly explain the syntax and semantics of these templates. For a deeper discussion of CAP and its functionalities we refer the reader to [Gut17].

## 1. Syntax

CAP supports lots of important notions of category theory, which we also call *categorical constructions*. An example of such a categorical construction is the *cokernel*.

**Definition.** Let  $\mathbf{A}$  be an additive category. Given objects  $A, B \in \mathbf{A}$  and a morphism  $\phi \in \text{Hom}_{\mathbf{A}}(A, B)$ , a **cokernel of  $\phi$**  consists of the following data:

---

<sup>1</sup> These packages are:

- CAP (the core system)
- LinearAlgebraForCAP (an implementation of the category of finite dimensional vector spaces)
- ModulePresentationsForCAP (an implementation of the category of finitely presented modules)
- GeneralizedMorphismsForCAP (an implementation of additive relations in abelian categories, see Chapter II)

<sup>2</sup> Version 4.8.6, as of November 2016

<sup>3</sup> GitHub pages of the CAP-authors:

- Sebastian Gutsche: <https://github.com/sebasguts>
- Sebastian Posur: <https://github.com/sebastianpos>

- (1) An object  $C \in \mathbf{A}$ .
- (2) A morphism  $\pi : B \rightarrow C$  such that  $\pi \circ \phi = 0$ .
- (3) A dependent function<sup>4</sup>  $u$  mapping any pair  $(T, \tau)$  consisting of an object  $T \in \mathbf{A}$  and a morphism  $\tau : B \rightarrow T$  such that  $\tau \circ \phi = 0$  to a morphism  $u(T, \tau) : C \rightarrow T$  which has to be uniquely determined by the property  $\tau = u(T, \tau) \circ \pi$ .

The category  $\mathbf{A}$  **has cokernels** if it comes equipped with a dependent function mapping any morphism  $\phi \in \text{Hom}_{\mathbf{A}}(A, B)$  for  $A, B \in \mathbf{A}$  to a cokernel  $(C, \pi, u)$  of  $\phi$ .

CAP provides the following three primitives accessing the three components of the triple  $(C, \pi, u)$  for an additive category  $\mathbf{A}$  having cokernels:

- (1) CokernelObject :

$$\text{Hom}_{\mathbf{A}}(A, B) \rightarrow \text{Obj}_{\mathbf{A}} : \phi \mapsto C.$$

- (2) CokernelProjection :

$$\prod_{\phi \in \text{Hom}_{\mathbf{A}}(A, B)} \text{Hom}_{\mathbf{A}}(B, \text{CokernelObject}(\phi)) : \phi \mapsto \pi.$$

- (3) CokernelColift :

$$\prod_{\substack{\phi \in \text{Hom}_{\mathbf{A}}(A, B) \\ \tau \in \{\sigma \in \text{Hom}_{\mathbf{A}}(B, T) \mid \sigma \circ \phi = 0\}}} \text{Hom}_{\mathbf{A}}(\text{CokernelObject}(\phi), T) : (\phi, \tau) \mapsto u(T, \tau).$$

We also wrote down the dependent types of these primitives for given objects  $A, B, T$ , in order to highlight their interdependencies. For example, the dependent type of the primitive CokernelProjection tells us that given a morphism  $\phi : A \rightarrow B$ , the output CokernelProjection( $\phi$ ) will be a morphism  $B \rightarrow \text{CokernelObject}(\phi)$ , i.e., a morphism with range depending on the primitive CokernelObject.

These three primitives suffice for building up other functionalities of the cokernel, e.g., its functoriality.

**Example.** Given a commutative diagram in  $\mathbf{A}$  of the form

$$D := \begin{array}{ccc} A & \xrightarrow{\alpha} & B \\ \nu \downarrow & & \downarrow \mu \\ A' & \xrightarrow{\alpha'} & B' \end{array}$$

the functoriality of the cokernel is given by the term

$$(\dagger) \quad \text{CokernelFunctorial}(D) := \text{CokernelColift}(\alpha, \text{CokernelProjection}(\alpha') \circ \mu).$$

---

<sup>4</sup>See Definition I.1.1

$$\begin{array}{ccc}
A & \xrightarrow{\alpha} & B & \xrightarrow{\text{CokernelProjection}(\alpha)} & \text{CokernelObject}(\alpha) \\
\downarrow \nu & & \downarrow \mu & & \downarrow \text{CokernelFunctorial}(D) \\
A' & \xrightarrow{\alpha'} & B' & \xrightarrow{\text{CokernelProjection}(\alpha')} & \text{CokernelObject}(\alpha')
\end{array}$$

The primitives for categorical constructions provided by CAP are powerful enough for a functorial implementation of a spectral sequence algorithm working in the context of an arbitrary abelian category. Such an algorithm takes as arguments a morphism of descending filtered cochain complexes  $F^\bullet A^\bullet \rightarrow F^\bullet B^\bullet$  and a triple of integers  $(r, p, q)$  where  $r \geq 0$ . The output is the  $(p, q)$ -th differential on the  $r$ -th page of the associated spectral sequence connected in a commutative diagram of the form

$$\begin{array}{ccc}
F^\bullet A^\bullet E_r^{pq} & \xrightarrow{\partial} & F^\bullet A^\bullet E_r^{p+r, q-(r-1)} \\
\downarrow & & \downarrow \\
F^\bullet B^\bullet E_r^{pq} & \xrightarrow{\partial} & F^\bullet B^\bullet E_r^{p+r, q-(r-1)}
\end{array}$$

induced by the functoriality of spectral sequences. In this thesis we will demonstrate, among other things, how an implementation of such a high-level categorical construction can be realized with CAP's primitives (see Chapter II).

## 2. Semantics

The purpose of CAP is to model categories. Classically, a set of objects  $\text{Obj}_{\mathbf{A}}$  and a set of morphisms  $\text{Hom}_{\mathbf{A}}(A, B)$  for all pairs  $A, B \in \text{Obj}_{\mathbf{A}}$  are part of the data defining a (small) category  $\mathbf{A}$ .

CAP models a slightly more general and computer-friendlier notion of a category: Homomorphisms  $\text{Hom}_{\mathbf{A}}(A, B)$  are not only sets but *setoids*, i.e., a set equipped with an equivalence relation on it as an *extra datum*. The formal definition of this kind of category looks as follows:

**Definition.** A **Cap category**  $\mathbf{A}$  consists of the following data:

- (1) A set  $\text{Obj}_{\mathbf{A}}$  of **objects**.
- (2) For every pair  $A, B \in \text{Obj}_{\mathbf{A}}$ , a set  $\text{Hom}_{\mathbf{A}}(A, B)$  of **morphisms**. If two morphisms  $\alpha, \beta \in \text{Hom}_{\mathbf{A}}(A, B)$  are equal as elements of this set, we say they are **equal**.
- (3) For every pair  $A, B \in \text{Obj}_{\mathbf{A}}$ , an equivalence relation  $\sim_{A, B}$  on  $\text{Hom}_{\mathbf{A}}(A, B)$ . If  $\alpha \sim_{A, B} \beta$  for two morphisms  $\alpha, \beta \in \text{Hom}_{\mathbf{A}}(A, B)$ , we say they are **congruent**.
- (4) For every  $A \in \text{Obj}_{\mathbf{A}}$ , an **identity morphism**  $\text{id}_A \in \text{Hom}_{\mathbf{A}}(A, A)$ .
- (5) For every triple  $A, B, C \in \text{Obj}_{\mathbf{A}}$ , a **composition function**

$$\circ : \text{Hom}_{\mathbf{A}}(B, C) \times \text{Hom}_{\mathbf{A}}(A, B) \rightarrow \text{Hom}_{\mathbf{A}}(A, C)$$

compatible with the equivalence relation, i.e., if  $\alpha, \alpha' \in \text{Hom}_{\mathbf{A}}(A, B)$ ,  $\beta, \beta' \in \text{Hom}_{\mathbf{A}}(B, C)$ ,  $\alpha \sim_{A,B} \alpha'$  and  $\beta \sim_{B,C} \beta'$ , then  $\beta \circ \alpha \sim_{A,C} \beta' \circ \alpha'$ .

(6) For all  $A, B \in \text{Obj}_{\mathbf{A}}$ ,  $\alpha \in \text{Hom}_{\mathbf{A}}(A, B)$ , we have

$$(\text{id}_B \circ \alpha) \sim_{A,B} \alpha$$

and

$$\alpha \sim_{A,B} (\alpha \circ \text{id}_A).$$

(7) For all  $A, B, C, D \in \text{Obj}_{\mathbf{A}}$ ,  $\alpha \in \text{Hom}_{\mathbf{A}}(A, B)$ ,  $\beta \in \text{Hom}_{\mathbf{A}}(B, C)$ ,  $\gamma \in \text{Hom}_{\mathbf{A}}(C, D)$ , we have

$$((\gamma \circ \beta) \circ \alpha) \sim_{A,D} (\gamma \circ (\beta \circ \alpha))$$

**REMARK.** In terms of higher category theory, a CAP category is a 2-category such that the 2-morphism sets are either empty or a singleton, and such that its underlying object class is a set. Using this point of view, we can derive the notion of a functor between CAP categories: A CAP functor consists of an object and a morphism function such that the usual axioms of a functor hold up to congruence.

Given a CAP category  $\mathbf{A}$ , passing to the quotient sets  $\text{Hom}_{\mathbf{A}}(A, B)/\sim_{A,B}$  gives rise to a classical category  $\overline{\mathbf{A}}$ , because all constructions and axioms respect the congruence for morphisms. It is usually the case that we actually want to compute with  $\overline{\mathbf{A}}$ , but that it is easier to implement a CAP category  $\mathbf{A}$  giving rise to  $\overline{\mathbf{A}}$ . We demonstrate this principle by means of an example.

**Example.** Let  $R\text{-fpmo}$  be the category of finitely presented left  $R$ -modules for a computable ring  $R$ . We are going to model  $R\text{-fpmo}$  by a CAP category  $R\text{-fpres}$ . We define  $\text{Obj}_{R\text{-fpres}}$  as the set of all matrices with entries in  $R$ . Note that each such matrix  $A \in R^{m \times n}$  can be interpreted as a homomorphism between free modules  $R^{1 \times m} \xrightarrow{A} R^{1 \times n} \in R\text{-fpmo}$  presenting its cokernel. For  $A \in R^{m \times n}$ ,  $B \in R^{o \times p}$ , we define  $\text{Hom}_{R\text{-fpmo}}(A, B)$  as the set of matrices  $M \in R^{n \times p}$  such that the following diagram can be completed to a commutative diagram by inserting a matrix  $\nu$  on the left:

$$\begin{array}{ccc} R^{1 \times m} & \xrightarrow{A} & R^{1 \times n} \\ \exists \nu \downarrow \text{dashed} & & \downarrow M \\ R^{1 \times o} & \xrightarrow{B} & R^{1 \times p} \end{array}$$

Note that by the functoriality of the cokernel, such a diagram induces a morphism between the modules presented by  $A$  and  $B$  independent of the choice of  $\nu$  (since  $\nu$  does not appear in the CAP term ( $\dagger$ ) defining  $\text{CokernelFunctorial}$ ). Conversely, every morphism in  $R\text{-fpmo}$  between the cokernels can be lifted to such a diagram since row modules are projective.

In our definition of the homomorphism sets, two morphisms  $M, N \in \text{Hom}_{R\text{-fpres}}(A, B)$  are *equal* if they are equal as matrices. We say  $M$  and  $N$  are *congruent* if and only if they induce equal morphisms between the modules presented by  $A$  and  $B$ , which is the case if and only if there exists a matrix rendering the diagram



$$\begin{array}{ccc}
& & R^{1 \times n} \\
& \exists & \downarrow M - N \\
R^{1 \times o} & \xrightarrow{B} & R^{1 \times p}
\end{array}$$

commutative (this is a direct consequence of the comparison theorem [Wei94]).

Thus, we equipped the homomorphism sets  $\text{Hom}_{R\text{-fpres}}(A, B)$  with an equivalence relation such that passing to the quotient yields a category  $\overline{R\text{-fpres}}$  equivalent to  $R\text{-fpm}\text{od}$ . We can see the advantage of the model  $R\text{-fpres}$  over  $\overline{R\text{-fpres}}$  when we start defining the *function*

$$\text{CokernelObject} : \text{Hom}_{R\text{-fpres}}(A, B) \rightarrow \text{Obj}_{R\text{-fpres}}$$

for given  $A, B \in R\text{-fpres}$ . In the case of  $R\text{-fpres}$ , for  $M \in R^{n \times p}$ , we can simply set

$$\text{CokernelObject}(M) := \begin{pmatrix} M \\ B \end{pmatrix}$$

which yields a function since *equal* input yields *equal* output. The same mapping rule in the context of  $\overline{R\text{-fpres}}$  does not yield a function: For example,  $M_1 = (0)$  and  $M_2 = (2)$  both represent the same module homomorphism in

$$\begin{array}{ccc}
0 & \longrightarrow & \mathbb{Z}^{1 \times 1} \\
\downarrow & & \downarrow M_i \\
\mathbb{Z}^{1 \times 1} & \xrightarrow{(2)} & \mathbb{Z}^{1 \times 1}
\end{array}$$

for  $i = 1, 2$ , but

$$\text{CokernelObject}(M_1) = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \neq \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \text{CokernelObject}(M_2)$$

on the level of matrices and thus on the level of objects in  $\overline{R\text{-fpres}}$ . This issue can be fixed by making (possibly unnatural) choices of representatives, but this can be very expensive in an actual implementation.

We further define

$$\text{CokernelProjection}(M) := I_p$$

where  $I_p$  denotes the  $p \times p$  identity matrix, and

$$\text{CokernelColift}(M, T) := T$$

which are dependent functions of the correct types for our model  $R\text{-fpres}$ .

The following interpretation underlines the naturality of our model  $R\text{-fpres}$ : Not only is  $\text{CokernelObject}$  a function in the context of  $R\text{-fpres}$ , but actually a functor between CAP categories. This can be made precise as follows:  $\text{Hom}_{R\text{-fpres}}(A, B)$  equipped with its equivalence relation can be seen as a category, where there is a morphism from  $M$  to  $N$  if and only if  $M \sim N$ . Furthermore, every category trivially can be turned into a CAP

category, so  $\text{Hom}_{R\text{-fpres}}(A, B)$  is also a CAP category. The primitive  $\text{CokernelObject}$  can now be regarded as a CAP functor

$$\text{CokernelObject} : \text{Hom}_{R\text{-fpres}}(A, B) \rightarrow R\text{-fpres}$$

whose action on morphisms  $\text{CokernelObject}(M \sim M')$  is given by

$$\text{CokernelObject}(M) \xrightarrow{\text{CokernelColift}(M, \text{CokernelProjection}(M'))} \text{CokernelObject}(M').$$

It is well-defined since it respects composition and identities up to congruence and thus defines a CAP functor.

Problems similar to the issues with the cokernel arise when we want to deal with other categorical constructions, like kernels, pullbacks, or pushout, and the CAP category  $R\text{-fpres}$  provides a natural solution for all them.

## CHAPTER 1

# Constructive Category Theory

This chapter covers basic notions of category theory with an emphasis on its constructive aspects. We highlight data structures that are sometimes hidden in propositions, but essential for a concrete implementation of categories in a computer algebra system. As an example, let us take a look at the statement “*Any additive functor  $F$  commutes with direct sums*”. It actually is a shortcut for a more detailed proposition (see Lemma I.2.10) which claims the existence of a unique natural isomorphism  $\sigma : F(A \oplus B) \xrightarrow{\sim} F(A) \oplus F(B)$ . Such an isomorphism can safely be ignored in most cases when we do abstract theory, due to coherence theorems. But when we work with concrete instances of categories, e.g., a skeletal version of the category of  $k$ -vector spaces for a field  $k$ , the distributivity of tensor products  $\otimes$  and direct sums  $\oplus$  can have non-trivial representations (see Computation I.3.16), and treating  $\sigma$  as the identity simply yields wrong results. So, making categorical data structures as explicit as possible is crucial for a correct implementation of category theory. For this reason, whenever we define a categorical notion in this thesis, we will even declare equalities of morphisms which are involved in this definition as part of the defining data.

Another special feature of our presentation is the introduction of *natural dependent functions* and the *dependent sum category* in Subsection I.1.2. Natural dependent functions are also known under the concept of *extraordinary Set-naturality*, and can for example be found in [Kel05] (in the context of enriched category theory). They are more general than natural transformations (see Example I.1.17) since they can give precise meaning to the assertion that a term such as  $\epsilon_A : A \otimes A^\vee \rightarrow 1$  naturally depends on  $A$  both in a covariant and contravariant way (see Example I.1.18), where  $A$  is an object in a rigid monoidal category  $\mathbf{A}$  with tensor unit  $1$ .

We encountered natural dependent functions during the design of our templates for categorical constructions in our software project CAP. We needed expressive *types* for our categorical constructions that were able to capture interdependencies such as “the result of the operation CokernelObject is equal to the range of the result of the operation CokernelProjection” (see CAP project chapter). *Dependent types* are such expressive types. They are the key technology in proof assistants like Coq or in new approaches to homotopy theory [Uni13]. In our context every dependent function has a dependent type, e.g.,  $\epsilon_A$  is of dependent type  $\prod_{A \in \mathbf{A}} \text{Hom}(A \otimes A^\vee, 1)$  (see Definition I.1.1). From a computational point of view, these dependent types provide solutions for two problems. First, they dictate the correct specifications of each categorical construction, e.g., if you want to implement  $A \mapsto \epsilon_A$  correctly, you have to ensure that the output is a morphism whose range equals

the unit object  $1$  and whose source is given by the output of the operation  $\otimes$  applied to  $A$  and  $A^\vee$ . Second, in category theory we are often tempted to replace a given object with an isomorphic one and work with these objects as if they were the same. Of course, performing such a substitution in CAP can only be valid if it is respected by all constructions provided by CAP. Since we attached a dependent type to all constructions in CAP, we can coherently add another specification to all these constructions by simply requiring that every dependent function of a given dependent type in CAP shall be *natural*. Knowing whether a given dependent function  $f$  is *natural* or not tells us if we may coherently change the representation of a given object  $A$  without messing up the result of  $f(A)$ .

Dependent sum categories are a handy tool for the creation of new categories. Whenever we want to form a category whose objects are given by lists containing objects and morphisms, dependent sum categories automatically provide the correct notion of morphism between such lists. Furthermore, they subsume the notions of coslice, slice, and comma categories (Example I.1.25), and natural dependent functions can be interpreted as sections of their natural projection functor (Remark I.1.28). Natural dependent functions and dependent sum categories are used throughout this thesis as a concise and precise way to express naturality and for a convenient construction of categories.

In Subsections I.1.3 - I.2.2 we discuss images and limits. In Section I.2 we give a concrete example of how to implement the category of finite dimensional vector spaces  $k\text{-vec}$  in a computer algebra system as an abelian category well-suited for computations. Although this example is extremely simple,  $k\text{-vec}$  serves as a building block for more complicated categories throughout this thesis.

In Section I.3 we attack the following structure problem of tensor categories: Given a finite group  $G$  such that  $k$  is a splitting field for  $G$ , denote its set of irreducible  $k$ -representations by  $\text{Irr}(G)$ . Then the problem is to compute a tensor product  $\otimes$  and an associator

$$\alpha : A \otimes (B \otimes C) \xrightarrow{\sim} (A \otimes B) \otimes C$$

on the additive coproduct category  $\bigoplus_{i \in \text{Irr}(G)} k\text{-vec}$  that turn it into a monoidal category equivalent to the monoidal category of representations of  $G$ . This is an instance of the «general question of *categorification* of based rings, which is one of the main problems in the structure theory of tensor categories» ([EGNO15]).

These tensor categories will in turn be the building blocks for our categories of  $G$ -equivariant modules in Chapter III. CAP examples of such tensor categories are given in Subsections I.3.3.9, I.3.3.10, and I.3.3.11.

## 1. Preliminaries

Before we start, we again emphasize that our exposition highlights the constructive aspects of category theory and presents them in a way such that they become implementable in a computer algebra system. In particular, the categorical constructions which we present are given by dependent functions having a dependent type, a concept which has the following interpretation on a set-theoretic level.

**Definition 1.1.** Let  $A$  be a set and let  $(B_a)_{a \in A}$  be an  $A$ -indexed family of sets. Then we denote the set of all sections of the natural projection  $\sqcup_{a \in A} B_a \rightarrow A$  by

$$\prod_{a \in A} B_a := \{\sigma : A \rightarrow \sqcup_{a \in A} B_a \mid \sigma(a) \in B_a\}.$$

An element  $\sigma \in \prod_{a \in A} B_a$  is called a **dependent function of dependent type** (or simply of **type**)  $\prod_{a \in A} B_a$ .

**Example 1.2.** As we have seen in the [syntax section](#) of the [CAP project chapter](#), CAP’s primitives have dependent types. Here is a concrete interpretation of the dependent type of the primitive `CokernelProjection`: Let  $\mathbf{A}$  be an additive category with cokernels. For all  $A, B \in \mathbf{A}$ , we have a dependent function `CokernelProjection` of type

$$\prod_{\phi \in \text{Hom}_{\mathbf{A}}(A, B)} \text{Hom}_{\mathbf{A}}(B, \text{CokernelObject}(\phi))$$

that maps any morphism  $\phi : A \rightarrow B$  to another morphism whose range *depends* on the given input  $\phi$ , namely to the natural projection of  $B$  into the cokernel object of  $\phi$ .

In Subsection [1.1.2](#), we will generalize the concept of a dependent function to the case where the indexing set  $A$  is a category  $\mathbf{A}$ , and then speak about *natural* dependent functions, which are roughly speaking given by dependent functions “compatible with the morphisms in  $\mathbf{A}$ ”.

A general reference for category theory is [\[ML71\]](#).

### 1.1. Categories, Functors, Natural Transformations.

**Definition 1.3.** A **category  $\mathbf{C}$**  consists of the following data:

- (1) A class  $\text{Obj}_{\mathbf{C}}$  of **objects**.
- (2) For every pair  $A, B \in \text{Obj}_{\mathbf{C}}$ , a set  $\text{Hom}_{\mathbf{C}}(A, B)$  of **morphisms**.
- (3) For every  $A \in \text{Obj}_{\mathbf{C}}$ , an **identity morphism**  $\text{id}_A \in \text{Hom}_{\mathbf{C}}(A, A)$ .
- (4) For every triple  $A, B, C \in \text{Obj}_{\mathbf{C}}$ , a **composition function**

$$\circ : \text{Hom}_{\mathbf{C}}(B, C) \times \text{Hom}_{\mathbf{C}}(A, B) \rightarrow \text{Hom}_{\mathbf{C}}(A, C).$$

- (5) For all  $A, B \in \text{Obj}_{\mathbf{C}}$ ,  $\alpha \in \text{Hom}_{\mathbf{C}}(A, B)$ , we have

$$\text{id}_B \circ \alpha = \alpha$$

and

$$\alpha = \alpha \circ \text{id}_A.$$

- (6) For all  $A, B, C, D \in \text{Obj}_{\mathbf{C}}$ ,  $\alpha \in \text{Hom}_{\mathbf{C}}(A, B)$ ,  $\beta \in \text{Hom}_{\mathbf{C}}(B, C)$ ,  $\gamma \in \text{Hom}_{\mathbf{C}}(C, D)$ , we have

$$(\gamma \circ \beta) \circ \alpha = \gamma \circ (\beta \circ \alpha).$$

**Notation 1.4.** We write  $\alpha : A \rightarrow B$  for emphasizing the source and range of a morphism  $\alpha \in \text{Hom}_{\mathbf{C}}(A, B)$ . If  $\beta : B \rightarrow C$  is another morphism, we either write  $\beta \circ \alpha$  or  $\alpha \cdot \beta$  for their composite. Omitting any symbol as in the term  $\beta\alpha$  will always mean  $\beta \circ \alpha$ .

**Definition 1.5.** Let  $A, B \in \text{Obj}_{\mathbf{C}}$ . We call a pair  $(\phi, \psi) \in \text{Hom}_{\mathbf{C}}(A, B) \times \text{Hom}_{\mathbf{C}}(B, A)$  an **isomorphism** if it satisfies  $\phi \circ \psi = \text{id}_B$  and  $\psi \circ \phi = \text{id}_A$ , i.e., if  $\phi$  and  $\psi$  are mutual inverses.

Given two isomorphisms  $(\phi, \psi)$  and  $(\phi, \chi)$ , we trivially have  $\psi = \chi$ , which is why we will simply say  $\phi$  is an *isomorphism*, and denote its uniquely determined inverse by  $\phi^{-1}$ . We write  $\phi : A \xrightarrow{\sim} B$  to indicate that  $\phi$  is an isomorphism.

**Definition 1.6.** A **functor**  $F$  between two categories  $\mathbf{C}$  and  $\mathbf{D}$  consists of the following data:

- (1) An object function  $\text{Obj}_{\mathbf{C}} \rightarrow \text{Obj}_{\mathbf{D}}$ , which we also denote by  $F$ .
- (2) For  $A, B \in \text{Obj}_{\mathbf{C}}$ , a function  $F_{A,B} : \text{Hom}_{\mathbf{C}}(A, B) \rightarrow \text{Hom}_{\mathbf{D}}(F(A), F(B))$ .
- (3) For  $A \in \text{Obj}_{\mathbf{C}}$ ,  $F_{A,A}(\text{id}_A) = \text{id}_{F(A)}$ .
- (4) For  $A, B, C \in \text{Obj}_{\mathbf{C}}$ ,  $\alpha : A \rightarrow B$ ,  $\beta : B \rightarrow C$ , we have

$$F_{A,C}(\beta \circ \alpha) = F_{B,C}(\beta) \circ F_{A,B}(\alpha).$$

We usually suppress subscripts and write  $F(\alpha)$  instead of  $F_{A,B}(\alpha)$ .

There is an obvious notion of composition of functors.

**Definition 1.7.** A **natural transformation**  $\nu$  between two functors  $F, G : \mathbf{C} \rightarrow \mathbf{D}$  consists of the following data:

- (1) A dependent function of type  $\prod_{C \in \mathbf{C}} \text{Hom}_{\mathbf{D}}(FC, GC)$ , i.e., a dependent function mapping an object  $C \in \mathbf{C}$  to a morphism  $\nu_C \in \text{Hom}_{\mathbf{D}}(FC, GC)$ .
- (2) For  $C, D \in \mathbf{C}$ ,  $\alpha : C \rightarrow D$ , we have  $G(\alpha) \circ \nu_C = \nu_D \circ F(\alpha)$ , i.e., the following diagram commutes:

$$\begin{array}{ccc} FC & \xrightarrow{\nu_C} & GC \\ F(\alpha) \downarrow & & \downarrow G(\alpha) \\ FD & \xrightarrow{\nu_D} & GD \end{array}$$

The morphisms  $\nu_C$  are called the **components** of  $\nu$ .

A natural transformation can be nicely depicted as a 2-cell:

$$\begin{array}{ccc} & F & \\ \text{C} & \begin{array}{c} \curvearrowright \\ \Downarrow \nu \\ \curvearrowleft \end{array} & \text{D} \\ & G & \end{array}$$

**REMARK 1.8.** Natural transformations can be composed in two ways.

- (1) Given three functors  $F, G, H : \mathbf{C} \rightarrow \mathbf{D}$  and two natural transformations  $\nu : F \rightarrow G$ ,  $\mu : G \rightarrow H$  we can define  $\mu \circ \nu : F \rightarrow H$  componentwise. This is called **vertical composition**.

$$\begin{array}{ccc}
& F & \\
\curvearrowright & \Downarrow \nu & \curvearrowleft \\
\mathbf{C} & \xrightarrow{G} & \mathbf{D} \\
\curvearrowleft & \Downarrow \mu & \curvearrowright \\
& H &
\end{array}$$

- (2) Given categories  $\mathbf{C}, \mathbf{D}, \mathbf{E}$ , functors  $F, G : \mathbf{C} \rightarrow \mathbf{D}$ ,  $H, I : \mathbf{D} \rightarrow \mathbf{E}$ , and natural transformations  $\mu : F \rightarrow G$ ,  $\nu : H \rightarrow I$ , there are two ways to define the **horizontal composition**  $\nu \otimes \mu : HF \rightarrow IG$ . The components are given by one of the following two formulas, which both yield equal results:
- $C \mapsto \nu_{G(C)} \circ H(\mu_C)$ ,
  - $C \mapsto I(\mu_C) \circ \nu_{F(C)}$ .

$$\begin{array}{ccc}
& F & \\
\curvearrowright & \Downarrow \mu & \curvearrowleft \\
\mathbf{C} & & \mathbf{D} \\
\curvearrowleft & & \curvearrowright \\
& G &
\end{array}
\quad
\begin{array}{ccc}
& H & \\
\curvearrowright & \Downarrow \nu & \curvearrowleft \\
& & \mathbf{D} \\
\curvearrowleft & & \curvearrowright \\
& I &
\end{array}$$

We set  $\nu F := \nu \otimes \text{id}_F$  and  $H\mu := \text{id}_H \otimes \mu$ .

The notation  $\nu \otimes \mu$  is motivated by the fact that the functor category  $\text{Hom}(F, G)$  can be seen as a monoidal category with  $\otimes$  as a tensor product.

The following definition captures the idea of an isomorphism between functors.

**Definition 1.9.** A **natural isomorphism** from a functor  $F : \mathbf{C} \rightarrow \mathbf{D}$  to a functor  $G : \mathbf{C} \rightarrow \mathbf{D}$  consists of the following data:

- (1) A natural transformation  $\nu : F \rightarrow G$ .
- (2) For all  $C \in \mathbf{C}$ , the component  $\nu_C$  is an isomorphism.

We also write  $\nu : F \xrightarrow{\sim} G$  to indicate that  $\nu$  is a natural isomorphism.

**Definition 1.10.** An **equivalence**  $\nu$  between two categories  $\mathbf{C}$  and  $\mathbf{D}$  consists of the following data:

- (1) A functor  $F : \mathbf{C} \rightarrow \mathbf{D}$ .
- (2) A functor  $G : \mathbf{D} \rightarrow \mathbf{C}$ .
- (3) A natural isomorphism  $\eta : \text{id}_{\mathbf{C}} \xrightarrow{\sim} GF$ .
- (4) A natural isomorphism  $\epsilon : FG \xrightarrow{\sim} \text{id}_{\mathbf{D}}$ .

**1.2. Naturality.** In this section we clarify the notion of naturality for a categorical term depending on an object variable. For a complete treatment, we have to take covariance and contravariance concurrently into account.

**Definition 1.11.** Let  $\mathbf{C}$  be a category. We call a functor  $F : \mathbf{C}^{\text{op}} \times \mathbf{C} \rightarrow \mathbf{Set}$  a **discrete natural family with base category  $\mathbf{C}$** .

**Example 1.12.** Let  $\mathbf{C}$  be a category. The functor

$$\mathrm{Hom} : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \rightarrow \mathbf{Set}$$

is a discrete natural family. More examples will follow as the types of natural dependent functions.

The following notion can be found in [Kel05], in the more general context of enriched category theory under the term *extraordinary naturality for a family of maps*.

**Definition 1.13.** Let  $F : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \rightarrow \mathbf{Set}$  be a discrete natural family. A **natural dependent function of dependent type** (or simply of **type**)  $\prod_{C \in \mathbf{C}} F(C, C)$  consists of the following data:

- (1) A dependent function  $\delta$  mapping an object  $C \in \mathbf{C}$  to an element  $\delta_C \in F(C, C)$ .
- (2) For all  $\alpha : C \rightarrow C' \in \mathbf{C}$ , the equality  $F(\alpha, C')(\delta_{C'}) = F(C, \alpha)(\delta_C)$  holds.

The set of all natural dependent functions of a given type is also denoted by  $\prod_{C \in \mathbf{C}} F(C, C)$ .

$$\begin{array}{ccc}
 \delta_C \in F(C, C) & & F(C', C') \ni \delta_{C'} \\
 \swarrow & & \swarrow \\
 & F(C, \alpha) & F(\alpha, C') \\
 & \searrow & \swarrow \\
 & F(C, C') & \\
 & \Psi & \\
 & F(C, \alpha)(\delta_C) = F(\alpha, C')(\delta_{C'}) & 
 \end{array}$$

**REMARK 1.14.** Definition 1.13 generalizes the concept of a dependent function (see Definition 1.1.1) from a set-theoretic to a category-theoretic level. That means the family of sets is replaced by a discrete natural family, and only those dependent functions “compatible with morphisms” are considered as *natural*.

**REMARK 1.15.** Every functor  $F : \mathbf{C} \rightarrow \mathbf{Set}$  (or  $G : \mathbf{C}^{\mathrm{op}} \rightarrow \mathbf{Set}$ ) gives rise to a discrete natural family by composition with the natural projection  $\mathbf{C}^{\mathrm{op}} \times \mathbf{C} \rightarrow \mathbf{C}$  (or  $\mathbf{C}^{\mathrm{op}} \times \mathbf{C} \rightarrow \mathbf{C}^{\mathrm{op}}$ ). We then say this family is **dummy** in the first (or second) component. We simply write  $\prod_{C \in \mathbf{C}} F(C)$  (or  $\prod_{C \in \mathbf{C}} G(C)$ ) for the sets of dependent functions associated to  $F$  (or  $G$ ) considered as a discrete natural family.

We give a couple of examples to illustrate the usefulness of natural dependent functions in various contexts.

**Example 1.16.** The dependent function  $C \mapsto \mathrm{id}_C$  is natural of type

$$\prod_{C \in \mathbf{C}} \mathrm{Hom}_{\mathbf{C}}(C, C),$$

where the discrete natural family is given by the Hom functor (see Example 1.1.12).



**Example 1.17.** Let  $F, G : \mathbf{C} \rightarrow \mathbf{D}$  be functors. A natural dependent function  $\delta$  of type

$$\prod_{C \in \mathbf{C}} \text{Hom}_{\mathbf{D}}(F(C), G(C))$$

is a natural transformation from  $F$  to  $G$  (see Definition 1.1.7), because it satisfies

$$\begin{aligned} \delta_{C'} \circ F\alpha &= \text{Hom}_{\mathbf{D}}(F\alpha, G(C'))(\delta_{C'}) \\ &= \text{Hom}_{\mathbf{D}}(F(C), G\alpha)(\delta_C) = G\alpha \circ \delta_C \end{aligned}$$

for every morphism  $\alpha : C \rightarrow C'$  in  $\mathbf{C}$ .

**Example 1.18.** The dependent function mapping an object  $A$  to the evaluation morphism  $\epsilon_A$  in a rigid symmetric monoidal category  $\mathbf{C}$  (see Definition 1.3.28) is a natural dependent function of type  $\prod_{A \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(A \otimes A^\vee, 1)$ . The naturality constraint states that for every morphism  $\alpha : A \rightarrow B$  in  $\mathbf{C}$ , the diagram

$$\begin{array}{ccc} B \otimes B^\vee & \xrightarrow{\epsilon_B} & 1 \\ \alpha \otimes B^\vee \uparrow & & \uparrow \epsilon_A \\ A \otimes B^\vee & \xrightarrow{A \otimes \alpha^\vee} & A \otimes A^\vee \end{array}$$

commutes. Here, the concept of a natural dependent function captures the way in which the term  $\epsilon_A$  depends on  $A$  both in a covariant and contravariant way.

**Example 1.19.** The concept of a **dinatural transformation** [ML71] between two functors  $F, G : \mathbf{C}^{\text{op}} \times \mathbf{C} \rightarrow \mathbf{D}$  can be expressed using natural dependent functions: We start with the discrete natural family  $(C, C') \mapsto \text{Hom}_{\mathbf{D}}(F(C', C), G(C, C'))$  with base category  $\mathbf{C}$ . Then, a natural dependent function  $\delta$  of type

$$\prod_{C \in \mathbf{C}} \text{Hom}_{\mathbf{D}}(F(C, C), G(C, C))$$

is a dinatural transformation, because it satisfies

$$\begin{aligned} G(\alpha, C') \circ \delta_{C'} \circ F(C', \alpha) &= \text{Hom}_{\mathbf{D}}(F(C', \alpha), G(\alpha, C'))(\delta_{C'}) \\ &= \text{Hom}_{\mathbf{D}}(F(\alpha, C), G(C, \alpha))(\delta_C) = G(C, \alpha) \circ \delta_C \circ F(\alpha, C) \end{aligned}$$

for every morphism  $\alpha : C \rightarrow C'$  in  $\mathbf{C}$ .

**Example 1.20.** An object  $\perp$  in a category  $\mathbf{C}$  is called **initial** if for every  $C \in \mathbf{C}$ ,  $\text{Hom}_{\mathbf{C}}(\perp, C)$  is a singleton. We claim that an object  $\perp$  is initial if and only if there exists a natural dependent function  $f$  of type  $\prod_{C \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(\perp, C)$  such that  $f_\perp = \text{id}_\perp$ . For if  $\perp$  is initial, the unique maps  $\perp \rightarrow C$  clearly give a natural dependent function. Now, let  $f$  be a natural dependent function of type  $\prod_{C \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(\perp, C)$  such that  $f_\perp = \text{id}_\perp$ . Then every set  $\text{Hom}_{\mathbf{C}}(\perp, C)$  is inhabited by  $f_C$ . And given two morphisms  $\alpha, \beta : \perp \rightarrow C$ , naturality of  $f$  implies

$$\alpha = \alpha \circ \text{id}_\perp = \alpha \circ f_\perp = f_C = \beta \circ f_\perp = \beta \circ \text{id}_\perp = \beta.$$

We now turn to the case of several object variables and prove that naturality of a dependent function can be checked componentwise.

**Lemma 1.21.** *Let  $\mathbf{C}, \mathbf{D}$  be categories,  $F : (\mathbf{C} \times \mathbf{D})^{\text{op}} \times (\mathbf{C} \times \mathbf{D}) \rightarrow \mathbf{Set}$  be a discrete natural family. Let furthermore  $\delta$  be a dependent function mapping a pair  $(C, D) \in \mathbf{C} \times \mathbf{D}$  to a morphism  $\delta_{(C,D)} \in F((C, D), (C, D))$ . Then  $\delta$  is natural if and only if*

- (1) *for all  $C \in \mathbf{C}$ , the dependent function  $D \mapsto \delta_{(C,D)}$  is natural,*
- (2) *for all  $D \in \mathbf{D}$ , the dependent function  $C \mapsto \delta_{(C,D)}$  is natural.*

PROOF. If  $\delta$  is natural then so are its restrictions to subcategories of the base category. Conversely, let its restrictions be natural. Let  $\alpha : C \rightarrow C' \in \mathbf{C}, \beta : D \rightarrow D' \in \mathbf{D}$  be morphisms. We compute:

$$\begin{aligned}
F((\alpha, \beta), (C', D'))(\delta_{(C', D')}) &= F((\alpha, D), (C', D')) \circ F((C', \beta), (C', D'))(\delta_{(C', D')}) \\
&= F((\alpha, D), (C', D')) \circ F((C', D), (C', \beta))(\delta_{(C', D)}) \\
&= F((\alpha, D), (C', \beta))(\delta_{(C', D)}) \\
&= F((C, D), (C', \beta)) \circ F((\alpha, D), (C', D))(\delta_{(C', D)}) \\
&= F((C, D), (C', \beta)) \circ F((C, D), (\alpha, D))(\delta_{(C, D)}) \\
&= F((C, D), (\alpha, \beta))(\delta_{(C, D)}). \quad \square
\end{aligned}$$

**Corollary 1.22.** *Naturality of a natural transformation  $\nu : S \rightarrow T$  between functors  $S, T : \mathbf{A} \times \mathbf{B} \rightarrow \mathbf{C}$  can be checked componentwise.*

PROOF. Use Lemma 1.1.21 and Example 1.1.17. □

If we have a natural dependent function  $\delta$  of type

$$\coprod_{(C_1, \dots, C_r) \in \mathbf{C}_1 \times \dots \times \mathbf{C}_r} F(C_1, \dots, C_r, C_1, \dots, C_r),$$

we treat it like a dependent function in several variables and say it is **natural in each component**. We also denote its type by

$$\coprod_{C_1 \in \mathbf{C}_1, \dots, C_r \in \mathbf{C}_r} F(C_1, \dots, C_r, C_1, \dots, C_r).$$

This notation and terminology is justified by Lemma 1.1.21.

The dual construction to the set of all dependent functions associated to a discrete natural family is given by the dependent sum category.

**Definition 1.23.** Let  $F : \mathbf{C}^{\text{op}} \times \mathbf{C} \rightarrow \mathbf{Set}$  be a discrete natural family. Its associated **dependent sum category** or **category of elements** is given by the following data:

- (1) Objects are pairs  $(C, X)$  where  $C \in \mathbf{C}$  and  $X \in F(C, C)$ .
- (2) A morphism from  $(C, X)$  to  $(D, Y)$  where  $C, D \in \mathbf{C}, X \in F(C, C), Y \in F(D, D)$  is given by a morphism  $f : C \rightarrow D$  such that  $F(C, f)(X) = F(f, D)(Y)$ .

Composition and identities are given by composition and identities in  $\mathbf{C}$ . We denote this category by  $\sum_{C \in \mathbf{C}} F(C, C)$ .

As natural dependent functions can be seen as a generalization of dependent functions from a set-theoretic to a category-theoretic level (see Remark I.1.14), the dependent sum category  $\sum_{C \in \mathbf{C}} F(C, C)$  can be seen as a categorical version of the disjoint union  $\uplus_{C \in \mathbf{C}} F(C, C)$

To prove that composition in  $\sum_{C \in \mathbf{C}} F(C, C)$  is well-defined, let  $f : (C, X) \rightarrow (D, Y)$  and  $g : (D, Y) \rightarrow (E, Z)$  be morphisms. We compute

$$\begin{aligned} F(C, g \circ f)(X) &= F(C, g) \circ F(C, f)(X) \\ &= F(C, g) \circ F(f, D)(Y) \\ &= F(f, g)(Y) \\ &= F(f, E) \circ F(D, g)(Y) \\ &= F(f, E) \circ F(g, E)(Z) = F(g \circ f, E)(Z). \end{aligned}$$

Let  $F : \mathbf{C} \rightarrow \mathbf{Set}$  (or  $G : \mathbf{C}^{\text{op}} \rightarrow \mathbf{Set}$ ) be an ordinary functor. We simply write  $\sum_{C \in \mathbf{C}} F(C)$  (or  $\sum_{C \in \mathbf{C}} G(C)$ ) for the dependent sum category associated to  $F$  (or  $G$ ) considered as a discrete natural family (see Remark I.1.15).

**Example 1.24.** The category  $\sum_{C \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(C, C)$  has as objects pairs  $(C, \phi)$  where  $\phi$  is an endomorphism of an object  $C \in \mathbf{C}$ . A morphism from  $(C, \phi)$  to  $(D, \psi)$  is a morphism  $f : C \rightarrow D$  such that

$$f \circ \phi = \text{Hom}_{\mathbf{C}}(C, f)(\phi) = \text{Hom}_{\mathbf{C}}(f, D)(\psi) = \psi \circ f,$$

i.e.,  $f$  is compatible with the endomorphisms.

**Example 1.25.** Let  $F : \mathbf{A} \rightarrow \mathbf{B}$  be a functor and  $B \in \mathbf{B}$ . We can think of the category  $\sum_{A \in \mathbf{A}} \text{Hom}_{\mathbf{B}}(B, F(A))$  as the “left-fiber” of  $B$ , since it consists of all pairs  $(A, \phi)$  such that  $B \xrightarrow{\phi} F(A)$ . Similarly, one can call  $\sum_{A \in \mathbf{A}} \text{Hom}_{\mathbf{B}}(F(A), B)$  a “right-fiber” of  $B$ . These categories are usually called **coslice category** and **slice category**, respectively. More generally, let  $G : \mathbf{C} \rightarrow \mathbf{B}$  be another functor. The category  $\sum_{A \in \mathbf{A}, C \in \mathbf{C}} \text{Hom}_{\mathbf{B}}(FA, GC)$  is the **comma category** of  $F$  and  $G$ .

REMARK 1.26. Using the terminology of Example I.1.25, the standard theorems on the existence of adjoint functors read as follows:

- (1) The functor  $F$  admits a left adjoint if and only if all left-fibers have an initial object.
- (2) The functor  $F$  admits a right adjoint if and only if all right-fibers have a terminal object.

REMARK 1.27. The dependent sum category  $\sum_{C \in \mathbf{C}} F(C, C)$  admits a functor

$$\pi : \sum_{C \in \mathbf{C}} F(C, C) \rightarrow \mathbf{C}$$

which projects an object  $(C, X)$  to its first component. If  $F$  is dummy in one of its variables, then  $\pi$  is the projection of a so-called discrete fibration.

**REMARK 1.28.** A natural dependent function  $\delta$  of type  $\prod_{C \in \mathbf{C}} F(C, C)$  gives rise to a section of the projection functor  $\pi$  defined in Remark 1.1.27 by sending  $f : A \rightarrow B$  to  $(A, \delta(A)) \xrightarrow{f} (B, \delta(B))$ . Conversely, every section of  $\pi$  defines a natural dependent function by projecting to the second component. Both directions are mutually inverse.

We will use dependent sum categories for our constructions throughout this thesis. Furthermore, as already stated in the introduction of this chapter, knowing whether a given dependent function  $f$  is natural or not tells us if we may coherently change the representation of a given object  $A$  without messing up the result of  $f(A)$ . It is thus important for our implementation of categorical constructions to explicitly study their compatibility with morphisms (as we do it in Subsection 1.1.4 in the case of limits).

**1.3. Images.** Images and their dual notion of coimages provide an important construction tool in homological algebra, which is why we will briefly sketch their theory in the general categorical setup.

**Definition 1.29.** Let  $\mathbf{C}$  be a category and  $A, B \in \mathbf{C}$  be objects. A morphism  $\alpha : A \rightarrow B$  is called **monomorphism** if for all objects  $T \in \mathbf{C}$  and all pairs  $\beta, \gamma : T \rightarrow A$  of morphisms,  $\alpha \circ \beta = \alpha \circ \gamma$  implies  $\beta = \gamma$ . Dually,  $\alpha$  is called **epimorphism** if for all objects  $T \in \mathbf{C}$  and all pairs  $\beta, \gamma : B \rightarrow T$  of morphisms,  $\beta \circ \alpha = \gamma \circ \alpha$  implies  $\beta = \gamma$ .

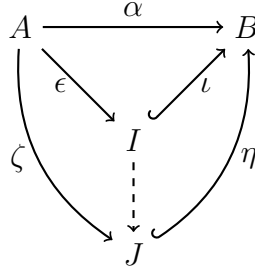
**Definition 1.30.** Let  $\alpha : A \rightarrow B$  be a morphism in a category  $\mathbf{C}$ . A **mono factorization of  $\alpha$**  consists of an object  $(I, \epsilon, \iota)$  in  $\sum_{I \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(A, I) \times \text{Hom}_{\mathbf{C}}(I, B)$  such that  $\iota$  is a monomorphism and  $\iota \circ \epsilon = \alpha$ .

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & B \\ & \searrow \epsilon & \nearrow \iota \\ & & I \end{array}$$

Note that the morphism  $\epsilon$  in a mono factorization is uniquely determined by  $\iota$ . Now, we turn to the categorical notion of an image, which is given by a universal mono factorization.

**Definition 1.31.** Let  $\alpha : A \rightarrow B$  be a morphism in a category  $\mathbf{C}$ . An **image** of  $\alpha$  consists of the following data:

- (1) A mono factorization  $(I, \epsilon, \iota)$  of  $\alpha$ .
- (2) A dependent function  $u$  mapping each mono factorization  $(J, \zeta, \eta)$  of  $\alpha$  to a morphism  $(I, \epsilon, \iota) \rightarrow (J, \zeta, \eta)$  in  $\sum_{I \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(A, I) \times \text{Hom}_{\mathbf{C}}(I, B)$ .



REMARK 1.32. For all dependent functions  $u, v$  satisfying (2) in Definition 1.1.31, it is easy to see that we have  $u = v$ .

The dual notion of a mono factorization is given by an **epi factorization**. The dual notion of an image is given by a **coimage**. We omit the obvious definitions and turn to epi-mono factorizations.

**Definition 1.33.** An **epi-mono factorization** of  $\alpha : A \rightarrow B$  consists of a mono factorization  $(I, \epsilon, \iota)$  which is also an epi factorization.

**Definition 1.34.** A **universal epi-mono factorization** of  $\alpha : A \rightarrow B$  is given by the following data:

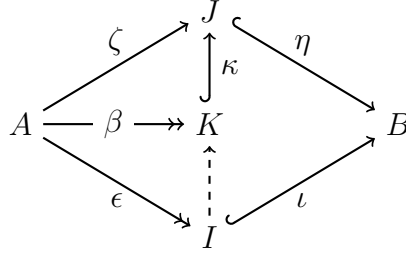
- (1) An epi-mono factorization  $(I, \epsilon, \iota)$  of  $\alpha$ .
- (2) A dependent function  $u$  mapping each epi-mono factorization  $(J, \zeta, \eta)$  to an isomorphism  $(I, \epsilon, \iota) \xrightarrow{\sim} (J, \zeta, \eta)$  in  $\sum_{I \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(A, I) \times \text{Hom}_{\mathbf{C}}(I, B)$ .

**Definition 1.35.** We say a category  $\mathbf{C}$  has

- **images** if it is equipped with a dependent function mapping each morphism  $\alpha$  to an image of  $\alpha$ , denoted by  $\text{im}(\alpha)$ ,
- **coimages** if it is equipped with a dependent function mapping each morphism  $\alpha$  to a coimage of  $\alpha$ , denoted by  $\text{coim}(\alpha)$ ,
- **universal epi-mono factorizations** if it is equipped with a dependent function mapping each morphism  $\alpha$  to a universal epi-mono factorization.

**Lemma 1.36.** *A category  $\mathbf{C}$  which has universal epi-mono factorizations also has images and coimages.*

PROOF. By duality, it suffices to show that  $\mathbf{C}$  has images. Let  $(I, \epsilon, \iota)$  be a universal epi-mono factorization of a morphism  $\alpha : A \rightarrow B$ . Let  $(J, \zeta, \eta)$  be a mono factorization of  $\alpha$ . Apply the epi-mono factorization to  $\zeta$ , which yields  $(K, \beta, \kappa)$ . Now  $\kappa \circ u((K, \beta, \eta \circ \kappa))$  is the desired morphism from  $(I, \epsilon, \iota)$  to  $(J, \zeta, \eta)$ .



□

**Definition 1.37.** Let  $\mathbf{C}$  be a category and  $A \in \mathbf{C}$ . An object  $(B, \iota : B \rightarrow A)$  in  $\sum_{B \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(B, A)$  such that  $\iota$  is a monomorphism is called a **subobject of  $A$** . We also write  $B \subseteq A$ . Two subobjects are said to be **equal as subobjects** if they are isomorphic (as objects in the dependent sum category). Dually, an object  $(B, \epsilon : A \rightarrow B)$  in  $\sum_{B \in \mathbf{C}} \text{Hom}_{\mathbf{C}}(A, B)$  such that  $\epsilon$  is an epimorphism is called a **quotient object of  $A$** . Two quotient objects are said to be **equal as quotient objects** if they are isomorphic (as objects in the dependent sum category).

As we would expect, every image  $(I, \epsilon, \iota)$  of  $\alpha : A \rightarrow B$  gives rise to a subobject  $(I, \iota)$  of  $B$ , and dually, every coimage  $(C, \epsilon, \iota)$  gives rise to a quotient object  $(C, \epsilon)$  of  $A$ .

**1.4. Limits.** Limits subsume important mathematical constructions in various different contexts, such as products, kernels or pullbacks of abelian groups, modules, or sheaves (see Example I.1.47). The theory of limits in category theory highlights the fact that such mathematical constructions are not merely given by single objects, but by objects equipped with *additional data*. We illustrate this point by means of an example construction: The pullback. Let  $\mathbf{C}$  be a category and let

$$\begin{array}{ccc}
 & C & \\
 & \downarrow \gamma & \\
 A & \xrightarrow{\alpha} & B
 \end{array}$$

be a *diagram* in  $\mathbf{C}$ . A pullback of that diagram consists of

- (1) an object  $A \times_B C$

to which we refer as the *pullback object*,

- (2) two morphisms  $\alpha^* : A \times_B C \rightarrow C$  and  $\gamma^* : A \times_B C \rightarrow A$ , rendering the diagram

$$\begin{array}{ccc}
 A \times_B C & \xrightarrow{\alpha^*} & C \\
 \gamma^* \downarrow & & \downarrow \gamma \\
 A & \xrightarrow{\alpha} & B
 \end{array}$$

commutative,

which we also call the *pullback projections*. Furthermore, the *universal property* of the pullback can also be seen as an additional datum:

- (3) A dependent function  $u$  mapping every triple  $(T, \tau_1 : T \rightarrow C, \tau_2 : T \rightarrow A)$  which also renders the above diagram commutative to a uniquely determined morphism

$$u(T, \tau_1, \tau_2) : T \rightarrow A \times_B C$$

such that the diagram

$$\begin{array}{ccccc}
 & & & & \tau_1 \\
 & & & & \curvearrowright \\
 T & & & & C \\
 \searrow & & & & \downarrow \gamma \\
 & u(T, \tau_1, \tau_2) & \longrightarrow & A \times_B C & \xrightarrow{\alpha^*} \\
 & & & \downarrow \gamma^* & \\
 & & & A & \xrightarrow{\alpha} B \\
 \searrow & & & & \\
 & \tau_2 & \longrightarrow & & 
 \end{array}$$

commutes.

Thus, a pullback can be fully described by the tuple  $(A \times_B C, (\alpha^*, \gamma^*), u)$ . Similarly, the constructions of kernels or direct products can also be seen as objects carrying additional data.

Now, we give a constructive exposition of limits, with which we can handle all these special instances of interest at one stroke. The role of a diagram is played by any functor  $D : \mathbf{I} \rightarrow \mathbf{C}$  from some index category  $\mathbf{I}$ . In the case of the pullback, such an index category is depicted by  $\bullet \rightarrow \bullet \leftarrow \bullet$ . The role of the pullback objects and the pullback projections is played by *sources*.

**Definition 1.38.** Let  $D : \mathbf{I} \rightarrow \mathbf{C}$  be a functor. A **source** of  $D$  consists of the following data:

- (1) An object  $S \in \mathbf{C}$ .
- (2) A dependent function  $s$  mapping an object  $i \in \mathbf{I}$  to a morphism  $s(i) : S \rightarrow D(i)$  such that for all  $i, j \in \mathbf{I}$ ,  $\iota : i \rightarrow j$ , we have  $D(\iota) \circ s(i) = s(j)$

Sources can be combined with functors and natural transformations.

**Notation 1.39.** For a natural transformation  $\delta : D \rightarrow E$  between functors  $D, E : \mathbf{I} \rightarrow \mathbf{C}$ , and a source  $s$  of  $D$ , we denote by  $\delta \circ s$  the source  $\left( S, (S \xrightarrow{s(i)} D(i) \xrightarrow{\delta(i)} E(i))_{i \in \mathbf{I}} \right)$ . For a functor  $F : \mathbf{C} \rightarrow \mathbf{D}$ , we denote by  $Fs$  the source  $\left( FS, (FS \xrightarrow{F(s(i))} FD(i))_{i \in \mathbf{I}} \right)$  of  $FD$ .

As in the case of the pullback, a limit is given by a source equipped with a *universal property*:

**Definition 1.40.** Let  $D : \mathbf{I} \rightarrow \mathbf{C}$  be a functor. A **limit** of  $D$  consists of the following data:

- (1) A source of  $D$  given by the data  $(\lim D, (\lambda(i) : \lim D \rightarrow D(i))_{i \in \mathbf{I}})$ .
- (2) A dependent function  $u$  mapping every source  $\tau = (T, (\tau(i) : T \rightarrow D(i))_{i \in \mathbf{I}})$  to a morphism  $u(\tau) : T \rightarrow \lim D$  such that  $\lambda(i) \circ u(\tau) = \tau(i)$  for all  $i \in \mathbf{I}$ .
- (3) For any other dependent function  $v$  satisfying (2), we have  $u = v$ .

In abelian categories, the pullback of every pair of arrows having the same range exists. So, we can say that abelian categories *have pullbacks*. Here is the general definition in the case of limits.

**Definition 1.41.** Let  $\mathbf{I}$  be a category. We say a category  $\mathbf{C}$  **has limits of type  $\mathbf{I}$**  if it is equipped with a dependent function  $\lambda$  mapping a functor  $D : \mathbf{I} \rightarrow \mathbf{C}$  to a limit  $(\lim D, \lambda_D, u_D)$  of  $D$ .

If two functors from  $\mathbf{I}$  to  $\mathbf{C}$  are related by natural transformations, we can ask in which way such natural transformations are compatible with the dependent function  $\lambda$ .

**Lemma 1.42.** *Let  $(\mathbf{C}, \lambda)$  be a category having limits of type  $\mathbf{I}$ . Then it can uniquely be equipped with the following naturality data: For every natural transformation  $\delta : D \rightarrow E$  between functors  $D, E : \mathbf{I} \rightarrow \mathbf{C}$ , we have*

- (1) a morphism  $\lambda(\delta) : \lim D \rightarrow \lim E$ ,
- (2) the equation  $\delta(i) \circ \lambda_D(i) = \lambda_E(i) \circ \lambda(\delta)$  holds for all  $i \in \mathbf{I}$ ,
- (3) for every source  $\tau = (T, (\tau(i) : T \rightarrow D(i))_{i \in \mathbf{I}})$ , the equation  $u_E(\delta \circ \tau) = \lambda(\delta) \circ u_D(\tau)$  holds.

Thus, we get a commutative diagram:

$$\begin{array}{ccccc}
 & & \tau(i) & & \\
 & \curvearrowright & & \curvearrowleft & \\
 T & \overset{u_D(\tau)}{\dashrightarrow} & \lim D & \xrightarrow{\lambda_D(i)} & D(i) \\
 & \searrow^{u_E(\delta \circ \tau)} & \downarrow \lambda(\delta) & & \downarrow \delta(i) \\
 & & \lim E & \xrightarrow{\lambda_E(i)} & E(i)
 \end{array}$$

**PROOF.** We set  $\lambda(\delta) := u_E(\delta \circ \lambda_D)$ . Using the universal properties, it is easy to check that the equations hold. The uniqueness of  $\lambda(\delta)$  follows from property (3) applied to the source  $(\lim D, \lambda_D)$ .  $\square$

The next natural question to ask is how a functor  $F : \mathbf{C} \rightarrow \mathbf{C}'$  relates the limits of its domain to the limits of its codomain. Note that since a limit is not merely an object, we have to understand how  $F$  acts on *all the defining data* of a limit.

**Lemma 1.43.** *Let  $(\mathbf{C}, \lambda)$  and  $(\mathbf{C}', \mu)$  be categories having limits of type  $\mathbf{I}$ , where  $\lambda$  denotes a dependent function  $D \mapsto (\lim D, \lambda_D, u_D)$  and  $\mu$  denotes a dependent function  $\Delta \mapsto (\lim \Delta, \mu_\Delta, v_\Delta)$ . Given a functor  $F : \mathbf{C} \rightarrow \mathbf{C}'$  between their underlying categories, it relates all the data involved in their limits of diagrams  $D : \mathbf{I} \rightarrow \mathbf{C}$ :*

$$"F((\lim D, \lambda_D, u_D)) \rightarrow (\lim FD, \mu_{FD}, v_{FD})".$$



More precisely,  $F$  can be equipped with the following data in a unique way: For every diagram  $D : \mathbf{I} \rightarrow \mathbf{C}$ , we have

- (1) a morphism  $f_D : F(\lim D) \rightarrow \lim FD$  which is natural in  $D$ , i.e.,  $\mu(F\delta) \circ f_D = f_E \circ F(\lambda(\delta))$  for any diagram  $E : \mathbf{I} \rightarrow \mathbf{C}$  and natural transformation  $\delta : D \rightarrow E$ ,
- (2) for all  $i \in \mathbf{I}$ , the equation  $\mu_{FD}(i) \circ f_D = F(\lambda_D(i))$  holds,
- (3) for every source  $\tau = (T, (\tau(i) : T \rightarrow D(i))_{i \in \mathbf{I}})$ , the equation  $f_D \circ F(u_D(\tau)) = v_{FD}(F\tau)$  holds.

$$\begin{array}{ccccc}
 & & \nu_{FD}(F\tau) & & \\
 & \text{---} & \text{---} & \text{---} & \\
 & & \text{---} & & \\
 FT & \xrightarrow{F(u_D(\tau))} & F(\lim D) & \xrightarrow{f_D} & \lim FD \\
 & \searrow & \downarrow & \swarrow & \\
 & & F(\lambda_D(i)) & & \\
 & \searrow & \downarrow & \swarrow & \\
 & & FD(i) & & \\
 & \swarrow & \downarrow & \searrow & \\
 & & F(\tau(i)) & & \\
 & \swarrow & \downarrow & \searrow & \\
 & & FD(i) & & 
 \end{array}$$

PROOF. We set  $f_D := v_{FD}(F\lambda_D)$ . With this definition, property (2) is nothing but the commutativity of the limit cone diagram of  $\lim FD$  together with  $(F(\lim D), F(\lambda))$  as a test object. Property (3) follows from the defining uniqueness property of  $\nu$ . The uniqueness of  $f_D$  follows from property (3) applied to the source  $(\lim D, \lambda_D)$ .

For the naturality of  $f_D$  in  $D$ , we compute for  $i \in \mathbf{I}$

$$\begin{aligned}
 \mu_{FE}(i) \circ \mu(F\delta) \circ f_D &= F(\delta(i)) \circ \mu_{FD}(i) \circ f_D && \text{Lemma I.1.42 (2)} \\
 &= F(\delta(i)) \circ F(\lambda_D(i)) && \text{property (2)} \\
 &= F(\delta(i) \circ \lambda_D(i)) \\
 &= F(\lambda_E(i) \circ \lambda(\delta)) && \text{Lemma I.1.42 (2)} \\
 &= F(\lambda_E(i)) \circ F(\lambda(\delta)) \\
 &= \mu_{FE}(i) \circ f_E \circ F(\lambda(\delta)) && \text{property (2)}.
 \end{aligned}$$

Now the desired equation follows from the universal property of  $\lim FE$ .  $\square$

**Definition 1.44.** If  $f_D$  is an isomorphism for every  $D$ , we say that  $F$  **preserves limits of type I**.

We saw in Lemma I.1.43 that a functor  $F : \mathbf{C} \rightarrow \mathbf{C}'$  relates the limits of its domain to the limits of its codomain by means of a morphism  $f_D : F(\lim D) \rightarrow \lim FD$ . Since functors are themselves related by natural transformations, the last natural question we want to answer is in which way  $f_D$  is compatible with such natural transformations.

**Lemma 1.45.** We use the notation of Lemma I.1.43. Given categories  $(\mathbf{C}, \lambda)$  and  $(\mathbf{C}', \mu)$  having limits of type I, two functors  $F, G : \mathbf{C} \rightarrow \mathbf{C}'$  equipped with the data

$(f_D)_D, (g_D)_D$  from Lemma I.1.43, respectively, and a natural transformation  $\epsilon : F \rightarrow G$ , then the equation

$$\mu(\epsilon D) \circ f_D = g_D \circ \epsilon_{\lim D}$$

holds. This means that the following diagram commutes:

$$\begin{array}{ccc} F(\lim D) & \xrightarrow{f_D} & \lim FD \\ \epsilon_{\lim D} \downarrow & & \downarrow \mu(\epsilon D) \\ G(\lim D) & \xrightarrow{g_D} & \lim GD \end{array}$$

PROOF. The above diagram is the left inner rectangular part of the following larger diagram for  $i \in \mathbf{I}$ :

$$\begin{array}{ccccc} F(\lim D) & \xrightarrow{F\lambda_D(i)} & & & FD(i) \\ & \searrow f_D & & & \downarrow \epsilon_{FD(i)} \\ & & \lim FD & \xrightarrow{\mu_{FD}(i)} & \\ \epsilon_{\lim D} \downarrow & & \downarrow \mu(\epsilon D) & & \\ & & \lim GD & \xrightarrow{\mu_{GD}(i)} & \\ G(\lim D) & \xrightarrow{G\lambda_D(i)} & & & GD(i) \end{array}$$

Its inner triangular parts commute by definition of  $f_D$  and  $g_D$ . Its right inner rectangular part commutes by definition of  $\mu(\epsilon D)$ . Its outer rectangle commutes by naturality of  $\epsilon$ . From these commutativities, we conclude

$$\mu_{GD}(i) \circ (\mu(\epsilon D) \circ f_D) = \mu_{GD}(i) \circ (g_D \circ \epsilon_{\lim D}).$$

Now the desired equation follows from the universal property of  $\lim GD$ .  $\square$

**Corollary 1.46.** *Under the hypotheses of Lemma I.1.45, if  $G$  preserves the limit of  $D$ , then  $\epsilon_{\lim D}$  is uniquely determined by the components  $\epsilon_{D(i)}$  for  $i \in \mathbf{I}$ .*

Studying limits in  $\mathbf{C}^{\text{op}}$  yields the theory of colimits in  $\mathbf{C}$ .

**Example 1.47.** Depending on  $\mathbf{I}$  some limits and colimits have special names:

$\mathbf{I}$	limit	colimit
$\emptyset$	terminal object	initial object
a set	direct product	coproduct
$\bullet \rightarrow \bullet \leftarrow \bullet$	pullback	-
$\bullet \leftarrow \bullet \rightarrow \bullet$	-	pushout
$\bullet \rightrightarrows \bullet$	equalizer	coequalizer

We have adopted this level of generality in this subsection since for our implementation of CAP, we wanted to know in what ways a given term (e.g., a limit) is compatible with the morphisms of its context, answering the question if it is possible to coherently change the representation of an object.

## 2. Additive, Abelian, and Coproduct Categories

The ultimate goal of this chapter is to construct a category  $\text{SRep}_k(G)$  well-suited for computations and equivalent to the representation category of a finite group  $G$ , where  $k$  is a splitting field for  $G$ . As a fundamental building block for  $\text{SRep}_k(G)$  will serve the category  $k\text{-vec}$ , which is also a category well-suited for computations and equivalent to the category of finite dimensional vector spaces over  $k$ .

**Construction 2.1.** We construct the category  $k\text{-vec}$  as follows:

- (1)  $\text{Obj}_{k\text{-vec}} := \mathbb{N}_0$ .
- (2) For  $m, n \in \mathbb{N}_0$ ,  $\text{Hom}_{k\text{-vec}}(m, n) := k^{m \times n}$ .
- (3) For  $n \in \mathbb{N}_0$ ,  $\text{id}_n$  is the  $n \times n$  identity matrix  $I_n$ .
- (4) For  $m, n, o \in \mathbb{N}_0$ ,  $A \in k^{m \times n}$ ,  $B \in k^{n \times o}$ ,  $B \circ A := A \cdot B$ .

So, objects in  $k\text{-vec}$  are simply modeled by natural numbers and morphisms by matrices with entries in  $k$ . Taking  $k\text{-vec}$  as a model for the category of  $k$ -vector spaces forces us to think in categorical terms: It does not make sense to address “vectors” of objects in  $k\text{-vec}$  since the objects are just natural numbers. Nevertheless, constructions like subspaces and quotient spaces can be described with the purely categorical terms *subobject* and *quotient object* (see Definition I.1.37). In fact, we aim at describing *all constructions that we need for the computation of  $G$ -equivariant sheaf cohomology on projective space* in terms of category theory. There is a long way to go until we will finally reach this computational goal in Section III.3, and it all starts with equipping  $k\text{-vec}$  with more and more categorical structure.

**2.1. Additive Categories.** Addition of matrices defines an abelian group structure on the homomorphism sets of  $k\text{-vec}$ . This turns  $k\text{-vec}$  into an Ab-category.

**Definition 2.2.** To define an **Ab-category**, take Definition I.1.3 of a category, replace the homomorphism sets  $\text{Hom}_{\mathbf{C}}(A, B)$  by abelian groups and the composition function  $\circ : \text{Hom}_{\mathbf{C}}(B, C) \times \text{Hom}_{\mathbf{C}}(A, B) \rightarrow \text{Hom}_{\mathbf{C}}(A, C)$  by a bilinear map. To define an **Ab-functor** between two Ab-categories, take Definition I.1.6 of a functor and replace the functions  $F_{A,B}$  between the homomorphism sets by homomorphisms of abelian groups. An **Ab-natural transformation** between two Ab-functors is simply an ordinary natural transformation (see Definition I.1.7).

In an Ab-category, we usually denote the group operation of  $\text{Hom}_{\mathbf{C}}(A, B)$  by  $+$ , and its neutral element by  $0_{A,B}$  (or simply by  $0$  if the context is clear).

**REMARK 2.3.** More generally, we can also study  $k$ -linear categories (or  $k$ -categories), where  $k$  is any commutative ring. The homomorphism sets are replaced by  $k$ -modules,  $\circ$  is a  $k$ -bilinear mapping. The corresponding functors are called  $k$ -linear functors (or  $k$ -functors), where the functions  $F_{A,B}$  between the homomorphism sets are given by  $k$ -module homomorphisms. The corresponding natural transformations are called  $k$ -linear natural transformations (or  $k$ -natural transformations). Even more generally, there is the concept of  $\mathcal{V}$ -enriched categories, where the homomorphism sets are replaced by objects in

some symmetric monoidal category  $(\mathcal{V}, \otimes)$  (see Definition I.3.25), and composition is given by a morphism  $\text{Hom}_{\mathbf{C}}(B, C) \otimes \text{Hom}_{\mathbf{C}}(A, B) \rightarrow \text{Hom}_{\mathbf{C}}(A, C)$  in  $\mathcal{V}$ . See [Kel05] for details.

Of course,  $k\text{-vec}$  is a  $k$ -linear category. We turn to the categorical characterizations of the zero dimensional vector space and direct sums.

**Definition 2.4.** An object  $Z$  in an Ab-category  $\mathbf{C}$  is called a **zero object** if  $\text{id}_Z = 0$ .

**Definition 2.5.** Let  $\mathbf{C}$  be an Ab-category,  $I$  a finite set,  $C_i \in \mathbf{C}$  for  $i \in I$ . A **direct sum** of  $(C_i)_{i \in I}$  consists of the following data:

- (1) An object  $C \in \mathbf{C}$ .
- (2) For each  $i \in I$ , a morphism  $\pi_i : C \rightarrow C_i$ , called **projection**.
- (3) For each  $i \in I$ , a morphism  $\iota_i : C_i \rightarrow C$ , called **injection**.
- (4) The equation  $\sum_{i \in I} \iota_i \pi_i = \text{id}_C$  holds.
- (5) For all  $i, j \in I$ , we have  $\pi_j \iota_i = \delta_{ij} \text{id}_{C_i}$ , where  $\delta_{ij}$  denotes the Kronecker delta.

REMARK 2.6. If  $I$  is empty, then we have the equation  $0 = \sum_{i \in \emptyset} \iota_i \pi_i = \text{id}_C$ . This shows that the direct sum of the empty family is a zero object.

**Construction 2.7.** We can easily equip  $k\text{-vec}$  with direct sums: For  $(m_1, \dots, m_r) \in k\text{-vec}^r = \mathbb{N}_0^r$ , we set

- (1)  $\oplus_{i=1}^r m_i := \sum_{i=1}^r m_i$ ,
- (2)  $\iota_i := \left( \dots \ 0^{m_i \times m_{i-1}} \ I_{m_i} \ 0^{m_i \times m_{i+1}} \ \dots \right)$ , and  $\pi_i := \iota_i^{\text{tr}}$  for  $i \in \{1, \dots, r\}$ .

This turns  $k\text{-vec}$  into an additive category.

**Definition 2.8.** An **additive category** consists of the following data:

- (1) An Ab-category  $\mathbf{A}$ .
- (2) A dependent function  $\oplus^{\mathbf{A}}$  (or simply  $\oplus$ ) mapping a *finite* set  $I$  and a collection  $(A_i)_{i \in I}$  of objects in  $\mathbf{A}$  to a corresponding direct sum  $(\oplus_{i \in I}^{\mathbf{A}} A_i, (\pi_i)_{i \in I}, (\iota_i)_{i \in I})$ .

By Remark I.2.6,  $\oplus^{\mathbf{A}}(\emptyset)$  is a zero object, which we denote by  $0^{\mathbf{A}}$ .

**Notation 2.9.** If  $I = \{1, \dots, n\}$  and  $(C_i)_{i \in I}$  are given, we set  $C_1 \oplus \dots \oplus C_n := \oplus_{i \in I} C_i$ .

The following lemma is often stated as “*Any additive functor  $F$  commutes with direct sums*”. Behind this shortcut lies a specific isomorphism  $\sigma : F(A \oplus B) \xrightarrow{\sim} F(A) \oplus F(B)$  from which we will greatly benefit in concrete computations (see Computation I.3.16 and Construction I.3.38) as well as in theoretical results (see for example the classification of  $k$ -linear functors in Lemma I.2.22, which in turn is used for understanding possible tensor product functors on the coproduct category  $\oplus_{i \in I} k\text{-vec}$  in Lemma I.3.10).

**Lemma 2.10.** *Let  $(\mathbf{A}, \oplus^{\mathbf{A}})$  and  $(\mathbf{B}, \oplus^{\mathbf{B}})$  be additive categories whose projections and injections we denote by  $\pi_i, \iota_i$  and  $p_i, q_i$ , respectively. An Ab-functor  $F : \mathbf{A} \rightarrow \mathbf{B}$  between their underlying Ab-categories preserves the additive structure. More precisely,  $F$  can be equipped with the following data in a unique way:*

- (1) A dependent function mapping a finite set  $I$  and a collection  $(A_i)_{i \in I}$  of objects in  $\mathbf{A}$  to an isomorphism

$$\sigma_F((A_i)_{i \in I}) : F(\oplus_{i \in I}^{\mathbf{A}} A_i) \xrightarrow{\sim} \oplus_{i \in I}^{\mathbf{B}} F(A_i),$$

which we simply denote by  $\sigma_F$  or  $\sigma$  if no confusion may occur.

(2) The equalities  $p_i \circ \sigma = F(\pi_i)$  and  $\sigma \circ F(\iota_i) = q_i$  hold.

In particular, we get an isomorphism  $z : F(0^{\mathbf{A}}) \xrightarrow{\sim} 0^{\mathbf{B}}$ .

PROOF. Let  $\sigma, \sigma'$  be morphisms satisfying  $p_i \circ \sigma = F(\pi_i) = p_i \circ \sigma'$ . It follows that  $\sum_i q_i p_i \sigma = \sum_i q_i p_i \sigma'$ . Applying equality (4) in Definition 1.2.5 gives us  $\sigma = \sigma'$  and thus proves uniqueness. Now, we set  $\sigma := \sum_{i \in I} q_i \circ F(\pi_i)$  and  $\tau := \sum_{i \in I} F(\iota_i) \circ p_i$ . Then a short computation using the equalities in Definition 1.2.5 proves that  $\sigma$  is an isomorphism with inverse  $\tau$ . Furthermore,  $\sigma$  satisfies the desired equations.  $\square$

Additive categories allow a matrix calculus that we will use several times (e.g., in Construction 1.2.20, or Lemma 1.3.54).

**Definition 2.11** (Matrix calculus). Let  $I, J$  be finite sets,  $(A_i)_{i \in I}, (B_j)_{j \in J}$  families of objects in  $\mathbf{A}$ , and  $(\alpha_{ij} : A_i \rightarrow B_j)_{i \in I, j \in J}$  a family of morphisms in  $\mathbf{A}$ . These data give rise to a morphism  $\bigoplus_{i \in I} A_i \rightarrow \bigoplus_{j \in J} B_j$  defined by  $\sum_{i,j} \iota_j \circ \alpha_{ij} \circ \pi_i$ , which we denote by the matrix  $(\alpha_{ij})_{ij}$ . Note that this is the **row convention**, since we label the rows of the matrix  $(\alpha_{ij})_{ij}$  according to the objects of the domain. If we are only given a finite set  $I$  and matrices  $(\alpha_i : A_i \rightarrow B_i)_{i \in I}$ , then we define  $\bigoplus_{i \in I} \alpha_i := \sum_i \iota_i \circ \alpha_i \circ \pi_i$ , which gives us a block diagonal matrix.

If  $\mu$  is a natural transformation between Ab-functors, we are able to deduce from the following lemma that the component  $\mu_{\bigoplus_{i \in I} A_i}$  is already determined by the components  $\mu_{A_i}$ . We will need this fact whenever we want to compute  $\mu_{\bigoplus_{i \in I} A_i}$  from the  $\mu_{A_i}$  (as in Lemma 1.2.22 or in Lemma 1.3.48). This is very useful especially in categories where every object can be written as a direct sum of simple objects (as it is the case in the coproduct category introduced in Definition 1.2.21).

**Lemma 2.12.** *Let  $F, G : \mathbf{A} \rightarrow \mathbf{B}$  be Ab-functors equipped with the data  $\sigma_F, \sigma_G$  of Lemma 1.2.10, respectively. Let  $\mu : F \rightarrow G$  be a natural transformation. Then the equation*

$$\sigma_G \circ \mu_{\bigoplus_{i \in I} A_i} = (\bigoplus_{i \in I} \mu_{A_i}) \circ \sigma_F$$

holds, i.e., the following diagram commutes:

$$\begin{array}{ccc} F(\bigoplus_{i \in I} A_i) & \xrightarrow{\sigma_F} & \bigoplus_{i \in I} F(A_i) \\ \mu_{\bigoplus_{i \in I} A_i} \downarrow & & \downarrow \bigoplus_{i \in I} \mu_{A_i} \\ G(\bigoplus_{i \in I} A_i) & \xrightarrow{\sigma_G} & \bigoplus_{i \in I} G(A_i) \end{array}$$

In particular,  $\mu_{0^{\mathbf{A}}} = 0$ .

PROOF. This can be seen as a special instance of Lemma 1.1.45, since direct sums are in particular direct products, or be checked directly using the definition of  $\sigma$  in the proof

of Lemma I.2.10 and the naturality of  $\mu$ :

$$\begin{aligned}
(\oplus_{i \in I} \mu_{A_i}) \circ \sigma_F &= \left( \sum_j q_j \circ \mu_{A_j} \circ p_j \right) \circ \left( \sum_i q_i \circ F(\pi_i) \right) \\
&= \sum_j q_j \circ \mu_{A_j} \circ F(\pi_j) \\
&= \sum_j q_j \circ G(\pi_j) \circ \mu_{\oplus_{i \in I} A_i} = \sigma_G \circ \mu_{\oplus_{i \in I} A_i}. \quad \square
\end{aligned}$$

**2.2. Abelian Categories.** Important constructions for  $k$ -vector spaces are kernels and cokernels. Categorically, a kernel of a morphism  $\alpha$  is the equalizer of the diagram

$$A \begin{array}{c} \xrightarrow{\alpha} \\ \rightrightarrows \\ \xrightarrow{0} \end{array} B$$

and thus a special instance of a limit. Nevertheless, we write down its definition explicitly in order to give special names to its defining data.

**Definition 2.13.** Let  $\alpha : A \rightarrow B$  be a morphism in an Ab-category  $\mathbf{A}$ . A **kernel of  $\alpha$**  consists of the following data:

- (1) An object  $\ker(\alpha) \in \mathbf{A}$ .
- (2) A morphism  $\text{KernelEmbedding}(\alpha) : \ker(\alpha) \rightarrow A$  such that

$$\alpha \circ \text{KernelEmbedding}(\alpha) = 0.$$

- (3) A dependent function  $\text{KernelLift}(\alpha, -)$  mapping a morphism  $\tau : T \rightarrow A$  with  $\alpha \circ \tau = 0$  to a morphism  $T \rightarrow \ker(\alpha)$  such that

$$\tau = \text{KernelEmbedding}(\alpha) \circ \text{KernelLift}(\alpha, \tau).$$

- (4) For any other dependent function  $v$  satisfying (3),  $v = \text{KernelLift}(\alpha, -)$ .

$$\begin{array}{ccccc}
& & \text{KernelEmbedding}(\alpha) & & \\
& & \xrightarrow{\quad} & & \\
\ker(\alpha) & \xleftarrow{\quad} & A & \xrightarrow{\alpha} & B \\
\uparrow \text{KernelLift}(\alpha, \tau) & & \nearrow & & \\
T & & \xrightarrow{\tau} & & 
\end{array}$$

Dually, a **cokernel of  $\alpha$**  consists of the following data:

- (1) An object  $\text{coker}(\alpha) \in \mathbf{A}$ .
- (2) A morphism  $\text{CokernelProjection}(\alpha) : B \rightarrow \text{coker}(\alpha)$  such that

$$\text{CokernelProjection}(\alpha) \circ \alpha = 0.$$

- (3) A dependent function  $\text{CokernelColift}(\alpha, -)$  mapping a morphism  $\tau : B \rightarrow T$  with  $\tau \circ \alpha = 0$  to a morphism  $\text{coker}(\alpha) \rightarrow T$  such that

$$\tau = \text{CokernelColift}(\alpha, \tau) \circ \text{CokernelProjection}(\alpha).$$

- (4) For any other dependent function  $v$  satisfying (3),  $v = \text{CokernelColift}(\alpha, -)$ .

$$\begin{array}{ccccc}
& & \text{CokernelProjection}(\alpha) & & \\
& & \longleftarrow & & \longleftarrow \alpha \\
& & \text{coker}(\alpha) & \longleftarrow & B & \longleftarrow & A \\
& \text{CokernelColift}(\alpha, \tau) & \downarrow & & & & \\
& & T & \longleftarrow & & & \\
& & & \tau & & & 
\end{array}$$

**Construction 2.14.** The category  $k\text{-vec}$  can be equipped with kernels and cokernels. We introduce 4 algorithms that can all be deduced from Gaussian elimination.

- (1) **SyzygiesOfRows:** The argument is a matrix  $A \in k^{m \times n}$ . The output is a matrix  $K^{o \times m}$  whose rows form a basis of the row kernel of  $A$ .
- (2) **SyzygiesOfColumns:** The argument is a matrix  $A \in k^{m \times n}$ . The output is a matrix  $K^{n \times o}$  whose columns form a basis of the column kernel of  $A$ .
- (3) **LeftDivide:** The arguments are matrices  $A \in k^{m \times n}, B \in k^{m \times o}$ . The output is a matrix  $X \in k^{n \times o}$  such that  $AX = B$ , if it exists.
- (4) **RightDivide:** The arguments are matrices  $B \in k^{m \times n}, A \in k^{o \times n}$ . The output is a matrix  $X \in k^{m \times o}$  such that  $XA = B$ , if it exists.

We can build kernels and cokernels from these algorithms:

- (1) Let  $A \in k^{m \times n}$ , then we define:
  - (a)  $\ker(A) :=$  number of rows of  $\text{SyzygiesOfRows}(A)$ ,
  - (b)  $\text{KernelEmbedding}(A) := \text{SyzygiesOfRows}(A) \in \text{Hom}_{k\text{-vec}}(\ker(A), m)$ ,
  - (c) for any  $T \in k^{o \times m}$  such that  $TA = 0$ , set the universal morphism

$$\text{KernelLift}(A, T) := \text{RightDivide}(T, \text{SyzygiesOfRows}(A)).$$

- (2) Let  $A \in k^{m \times n}$ , then we define:
  - (a)  $\text{coker}(A) :=$  number of columns of  $\text{SyzygiesOfColumns}(A)$ ,
  - (b)  $\text{CokernelProjection}(A) := \text{SyzygiesOfColumns}(A) \in \text{Hom}_{k\text{-vec}}(n, \text{coker}(A))$ ,
  - (c) for any  $T \in k^{n \times o}$  such that  $AT = 0$ , set the universal morphism

$$\text{CokernelColift}(A, T) := \text{LeftDivide}(\text{SyzygiesOfColumns}(A), T).$$

Once these methods are implemented for  $k\text{-vec}$ , we do not have to care about their internals anymore, since we want to focus on working with  $k\text{-vec}$  from a purely categorical point of view. Up till now,  $k\text{-vec}$  has become a pre-abelian category.

**Definition 2.15.** A **pre-abelian category** consists of the following data:

- (1) An additive category  $\mathbf{A}$ .
- (2) A dependent function mapping every morphism  $\alpha : A \rightarrow B$  for  $A, B \in \mathbf{A}$  to a kernel of  $\alpha$ .
- (3) A dependent function mapping every morphism  $\alpha : A \rightarrow B$  for  $A, B \in \mathbf{A}$  to a cokernel of  $\alpha$ .

Functors between pre-abelian categories preserving kernels and cokernels (see Definition I.1.44) are called **exact**. Due to the Lemma I.1.43 such functors can be uniquely equipped



with additional data relating the pre-abelian structures. Moreover, natural transformations between exact functors are compatible with kernels and cokernels by Lemma I.1.45.

**Definition 2.16.** Let  $\mathbf{C}$  be a category.

- (1) Let  $A, K, T \in \mathbf{C}$  be objects and  $\iota : K \rightarrow A$ ,  $\tau : T \rightarrow A$  morphisms. A **lift of  $\tau$  along  $\iota$**  is given by a morphism  $u : T \rightarrow K$  such that  $\iota \circ u = \tau$ .

$$\begin{array}{ccc} & K & \\ \text{---} u \text{---} & \nearrow \iota & \\ T & \xrightarrow{\tau} & A \end{array}$$

- (2) Let  $A, C, T \in \mathbf{C}$  be objects and  $\epsilon : A \rightarrow C$ ,  $\tau : A \rightarrow T$  morphisms. A **colift of  $\tau$  along  $\epsilon$**  is given by morphism  $u : C \rightarrow T$  such that  $u \circ \epsilon = \tau$ .

$$\begin{array}{ccc} A & \xrightarrow{\tau} & T \\ \text{---} \epsilon \text{---} & & \text{---} u \text{---} \\ & C & \end{array}$$

**Definition 2.17.** An **abelian category** consists of the following data:

- (1) A pre-abelian category  $\mathbf{A}$ .
- (2) A dependent function  $(-/-)$  mapping a pair  $\iota : K \rightarrow A$ ,  $\tau : T \rightarrow A$  to a lift  $\tau/\iota$  of  $\tau$  along  $\iota$ , where  $A, K, T \in \mathbf{A}$ ,  $\iota$  is a monomorphism and  $\text{CokernelProjection}(\iota) \circ \tau = 0$ .
- (3) A dependent function  $(-\backslash-)$  mapping a pair  $\epsilon : A \rightarrow C$ ,  $\tau : A \rightarrow T$  to a colift  $\epsilon \backslash \tau$  of  $\tau$  along  $\epsilon$ , where  $A, C, T \in \mathbf{A}$ ,  $\epsilon$  is an epimorphism and  $\tau \circ \text{KernelEmbedding}(\epsilon) = 0$ .

Usually, an abelian category is defined as a pre-abelian category such that *every monomorphism is the kernel of its cokernel*, and dually, *every epimorphism is the cokernel of its kernel*. Unwrapping this definition yields the data enlisted in Definition I.2.17.

So far, whenever we equipped a category with new data we have answered the question of how a functor interacts with these additional data (see Lemma I.1.43 for categories with limits or Lemma I.2.10 for additive categories). For the sake of completeness, we do the same for abelian categories.

**Lemma 2.18.** *Let  $\mathbf{A}$ ,  $\mathbf{B}$  be abelian categories. Let  $F : \mathbf{A} \rightarrow \mathbf{B}$  be an exact functor. Then  $F$  is compatible with the abelian structures. More precisely, the following equations hold:*

- (1) *For every pair  $\iota, \tau$  as in Definition I.2.17, we have  $F(\tau/\iota) = F(\tau)/F(\iota)$ .*
- (2) *For every pair  $\epsilon, \tau$  as in Definition I.2.17, we have  $F(\epsilon \backslash \tau) = F(\epsilon) \backslash F(\tau)$ .*

**PROOF.** Exact functors respect monomorphisms and epimorphisms, thus the equations are well-defined. Furthermore, lifts along monomorphisms and colifts along epimorphisms are unique, and every functor maps a lift to a lift and a colift to a colift.  $\square$

Because the compatibility data in [I.2.18](#) only involve equations of morphisms, there are no interesting new data for natural transformations between exact functors (in contrast to the case of natural transformations between additive functors, see [Lemma I.2.12](#)).

**Construction 2.19.** We can turn  $k\text{-vec}$  into an abelian category as follows:

- (1)  $B/A := \text{RightDivide}(B, A)$ ,  
where  $B \in k^{m \times n}$ ,  $A \in k^{o \times n}$ ,  $B \cdot \text{CokernelProjection}(A) = 0$ .
- (2)  $A \setminus B := \text{LeftDivide}(A, B)$ ,  
where  $A \in k^{m \times n}$ ,  $B \in k^{m \times o}$ ,  $\text{KernelEmbedding}(A) \cdot B = 0$ .

We end this subsection with an explicit construction of pullbacks in abelian categories. Note that since we know how to compute in  $k\text{-vec}$  as an abelian category, we can apply this explicit construction for the computation of pullbacks in  $k\text{-vec}$  (which gives us, as a special case, an algorithm for the intersection of subspaces or for the preimage of a subspace under a homomorphism).

**Construction 2.20.** Given a diagram in an abelian category  $\mathbf{A}$  of the form

$$\begin{array}{ccc} & & C \\ & & \downarrow \gamma \\ A & \xrightarrow{\alpha} & B \end{array}$$

we define the **diagonal difference**<sup>1</sup>

$$\delta := \begin{pmatrix} \alpha \\ -\gamma \end{pmatrix} : A \oplus C \rightarrow B.$$

Then, we can construct a pullback of  $\alpha, \gamma$  as follows:

- (1)  $A \times_B C := \ker(\delta)$ .
- (2) Let  $\pi_C : A \oplus C \rightarrow C$  be the natural projection. We set

$$\alpha^* := \pi_C \circ \text{KernelEmbedding}(\delta) : A \times_B C \rightarrow C.$$

- (3) Let  $\pi_A : A \oplus C \rightarrow A$  be the natural projection. We set

$$\gamma^* := \pi_A \circ \text{KernelEmbedding}(\delta) : A \times_B C \rightarrow A.$$

- (4) For any pair  $\tau_A : T \rightarrow A$ ,  $\tau_C : T \rightarrow C$  such that  $\alpha \circ \tau_A = \gamma \circ \tau_C$ , we set the universal morphism into the pullback as

$$u(T, \tau_A, \tau_C) := \text{KernelLift} \left( \delta, \begin{pmatrix} \tau_A & \tau_C \end{pmatrix} \right) : T \rightarrow A \times_B C.$$

Correctness of this construction follows from the fact that the equation  $\alpha \circ \tau_A = \gamma \circ \tau_C$  is equivalent to  $\begin{pmatrix} \tau_A & \tau_C \end{pmatrix} \cdot \delta = 0$ .

---

<sup>1</sup> See [Definition I.2.11](#) for an explanation of the matrix notation.

**2.3. Coproduct Categories.** We conclude this section with the introduction of the coproduct category  $\bigoplus_{i \in I} k\text{-vec}$ , which will serve as the underlying abelian category of  $\text{SRep}_k(G)$ , our computational model for the category of representations of a finite group  $G$ .

**Definition 2.21.** Let  $I$  be a set. For  $i \in I$ , we define the objects

$$\chi^i := (\delta_{ij})_{j \in I}$$

of the product category  $\prod_{i \in I} k\text{-vec}$ , where  $\delta_{ij}$  denotes the Kronecker delta. The **coproduct category**

$$\bigoplus_{i \in I} k\text{-vec}$$

is defined as the full subcategory of  $\prod_{i \in I} k\text{-vec}$  generated by all objects of the form  $\bigoplus_{i \in J} a_i \chi^i$ , where  $J \subseteq I$  is a finite subset of  $I$  and  $a_i \in \mathbb{N}_0$  for  $i \in J$ . We also write  $\bigoplus_{i \in I} a_i \chi^i$  for such objects, and assume that  $a_i \neq 0$  for only finitely many  $i$ 's.

Note that the coproduct category  $\bigoplus_{i \in I} k\text{-vec}$  is an abelian  $k$ -linear category (see Remark 1.2.3), in which all constructions are performed componentwise in  $k\text{-vec}$ .

Now, we classify all  $k$ -linear functors starting from  $\bigoplus_{i \in I} k\text{-vec}$ . This classification will in turn be used to classify possible tensor product functors on  $\bigoplus_{i \in I} k\text{-vec}$  in Lemma 1.3.10. Furthermore, it tells us when two functors  $F, G : \bigoplus_{i \in I} k\text{-vec} \rightarrow \bigoplus_{j \in J} k\text{-vec}$  are naturally isomorphic (see Corollary 1.2.23).

**Lemma 2.22.** *Let  $I$  be a set,  $\mathbf{A}$  be an additive  $k$ -linear category. Then evaluation at the objects  $\chi^i$*

$$F \mapsto (F(\chi^i))_{i \in I}$$

*yields an equivalence of categories:*

$$\text{Hom}_k \left( \bigoplus_{i \in I} k\text{-vec}, \mathbf{A} \right) \xrightarrow{\sim} \prod_{i \in I} \mathbf{A}$$

*where the left hand side denotes the category of  $k$ -linear functors.*

PROOF. We will describe mutually inverse functors

$$\text{Hom}_k \left( \bigoplus_{i \in I} k\text{-vec}, \mathbf{A} \right) \begin{array}{c} \xrightarrow{G} \\ \xleftarrow{H} \end{array} \prod_{i \in I} \mathbf{A}$$

The action of  $G$  is given by evaluation on all  $\chi^i$ 's:

$$\begin{array}{ccc} \begin{array}{ccc} & F & \\ \oplus_{i \in I} k\text{-vec} & \begin{array}{c} \downarrow \mu \\ \downarrow \mu \end{array} & \mathbf{A} \\ & F' & \end{array} & \mapsto & \left( \begin{array}{c} F(\chi^i) \\ \downarrow \mu_{\chi^i} \\ F'(\chi^i) \end{array} \right)_{i \in I} \end{array}$$

Conversely, given a family of objects  $(A_i)_{i \in I}$  in  $\mathbf{A}$ , we construct a functor with the help of the matrix notation for the morphism in  $\mathbf{A}$  introduced in Definition 1.2.11.

$$\begin{array}{ccc} \oplus_{i \in I} k\text{-vec} & \xrightarrow{H((A_i)_{i \in I})} & \mathbf{A} : \\ \begin{array}{ccc} \oplus_{i \in I} a_i \chi^i & & \oplus_{i \in I} \oplus_{j=1}^{a_i} A_i \\ (\alpha^i)_{i \in I} \downarrow & \mapsto & \downarrow \oplus_{i \in I} (\alpha_{jl}^i \cdot \text{id}_{A_i})_{jl} \\ \oplus_{i \in I} b_i \chi^i & & \oplus_{i \in I} \oplus_{j=1}^{b_i} A_i \end{array} \end{array}$$

The action of  $H$  on morphisms between families can also be described by matrices:

$$\left( \begin{array}{c} A^i \\ \downarrow \gamma^i \\ B^i \end{array} \right)_{i \in I} \mapsto \left( \begin{array}{c} \oplus_{i \in I} \oplus_{j=1}^{a_i} A_i \\ \downarrow \oplus_{i \in I} \oplus_{j=1}^{a_i} \gamma^i \\ \oplus_{i \in I} \oplus_{j=1}^{a_i} B_i \end{array} \right)_{(\oplus_{i \in I} a_i \chi^i) \in \oplus_{i \in I} k\text{-vec}}$$

where the right hand side are the components of a natural transformation

$$H((A_i)_{i \in I}) \rightarrow H((B_i)_{i \in I}).$$

The natural isomorphism  $\oplus_{j=1}^1 A_i \cong A_i$  induces a natural isomorphism from  $G \circ H((A_i)_{i \in I})$  to  $(A_i)_{i \in I}$ . The isomorphism  $\sigma$  from Lemma 1.2.10 induces a natural isomorphism from  $F$  to  $(H \circ G)(F)$  which is also natural in  $F$  due to Lemma 1.2.12:

$$\begin{array}{ccc} F(\oplus_{i \in I} a_i \chi^i) & \xrightarrow{\sigma_F} & \oplus_{i \in I} \oplus_{j=1}^{a_i} F(\chi^i) \\ \mu_{\oplus_{i \in I} \chi^i} \downarrow & & \downarrow \oplus_{i \in I} \mu_{\chi^i} \\ F'(\oplus_{i \in I} a_i \chi^i) & \xrightarrow{\sigma_{F'}} & \oplus_{i \in I} \oplus_{j=1}^{a_i} F'(\chi^i) \end{array}$$

for any natural transformation  $\mu : F \rightarrow F'$  and object  $(\oplus_{i \in I} a_i \chi^i) \in \oplus_{i \in I} k\text{-vec}$ .  $\square$

**Corollary 2.23.** *Any two  $k$ -functors  $F, G : \oplus_{i \in I} k\text{-vec} \rightarrow \oplus_{j \in J} k\text{-vec}$  are naturally isomorphic if and only if they yield equal functions on objects.*

PROOF. From Lemma [I.2.22](#) we get:

$$\begin{aligned} F \simeq G &\iff (F(\chi^i))_{i \in I} \simeq (G(\chi^i))_{i \in I} \\ &\iff (F(\chi^i))_{i \in I} = (G(\chi^i))_{i \in I} \end{aligned}$$

But  $F$  and  $G$  coincide on all  $\chi^i$ 's if and only if they coincide on all objects of  $\bigoplus_{i \in I} k\text{-vec}$ , which is due to the commutativity of  $F$  and  $G$  with direct sums (see Lemma [I.2.10](#)) and the fact that any object in  $\bigoplus_{i \in I} k\text{-vec}$  is given as a direct sum of the  $\chi^i$ 's.  $\square$

### 3. Constructing Tensor Categories

In this section we reach our goal of constructing  $\text{SRep}_k(G)$ , a tensor category well-suited for computations and equivalent *as tensor categories* to the representation category of a finite group  $G$ , where  $k$  is a splitting field for  $G$ . The underlying abelian category of  $\text{SRep}_k(G)$  will be given by the coproduct category  $\bigoplus_{i \in I} k\text{-vec}$ , where  $I$  is in bijection to the set of irreducible  $k$ -characters of  $G$ .

Until now, the structure with which we equipped  $\bigoplus_{i \in I} k\text{-vec}$  was completely canonical, since being an abelian category is a categorical property: Any two abelian structures on  $\bigoplus_{i \in I} k\text{-vec}$  are equivalent. However, there are many inequivalent ways to turn  $\bigoplus_{i \in I} k\text{-vec}$  into a tensor category (see Definition I.3.32).

**3.1. Bilinear Bifunctors.** We start this section with a classification of bilinear bifunctors which can be defined on  $\bigoplus_{i \in I} k\text{-vec}$  (and may serve as possible tensor products). Here,  $k$  denotes an arbitrary field.

**Definition 3.1.** Let  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  be categories. Functors of the form  $T : \mathbf{A} \times \mathbf{B} \rightarrow \mathbf{C}$  are called **bifunctors**. Their compatibility with composition is called the **interchange law**:

$$T(\beta \circ \alpha, \delta \circ \gamma) = T(\beta, \delta) \circ T(\alpha, \gamma)$$

for all morphisms  $\alpha : A \rightarrow A', \beta : A' \rightarrow A'' \in \mathbf{A}, \gamma : B \rightarrow B', \delta : B' \rightarrow B'' \in \mathbf{B}$ .

Every bifunctor  $T$  trivially gives rise to functors  $T(A, -) : \mathbf{B} \rightarrow \mathbf{C}$  for  $A \in \mathbf{A}$  and  $T(-, B) : \mathbf{A} \rightarrow \mathbf{C}$  for  $B \in \mathbf{B}$ . In the case of Ab-categories we want  $T(A, -)$  and  $T(-, B)$  to become additive, which leads to the notion of a bilinear bifunctor.

**Definition 3.2.** Let  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  be Ab-categories. A bifunctor  $T : \mathbf{A} \times \mathbf{B} \rightarrow \mathbf{C}$  is called **bilinear** if its maps on homomorphisms are bilinear. If  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  are  $k$ -linear categories,  $T$  is called  **$k$ -bilinear** (or simply bilinear) if its maps on homomorphisms are  $k$ -bilinear.

**Construction 3.3.** On  $k\text{-vec}$ , a bilinear bifunctor

$$\otimes : k\text{-vec} \times k\text{-vec} \rightarrow k\text{-vec}$$

can be defined via Kronecker products:

- (1) For  $m, n \in \mathbb{N}_0$ ,  $m \otimes n := mn$ .
- (2) For  $m, m', n, n' \in \mathbb{N}_0$ ,  $M \in k^{m \times m'}, N \in k^{n \times n'}$ ,  $M \otimes N \in \text{Hom}_{k\text{-vec}}(m \otimes m', n \otimes n')$  is given by the Kronecker product of  $M$  and  $N$ .

The bilinear bifunctor  $\otimes$  turns  $k\text{-vec}$  into a category with bifunctor.

**Definition 3.4.** A **category with bifunctor** consists of the following data:

- (1) A category  $\mathbf{C}$ .
- (2) A bifunctor  $\otimes : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ .

It is natural to ask if  $(k\text{-vec}, \otimes)$  is the unique way to equip  $k\text{-vec}$  with a bilinear bifunctor. Of course, uniqueness can only be expected up to a categorical notion of equivalence.

**Definition 3.5.** Let  $(\mathbf{C}, \otimes_{\mathbf{C}})$  and  $(\mathbf{D}, \otimes_{\mathbf{D}})$  be categories with bifunctors. A **functor between categories with bifunctor** from  $\mathbf{C}$  to  $\mathbf{D}$  consists of the following data:

- (1) A functor  $F : \mathbf{C} \rightarrow \mathbf{D}$ .
- (2) For  $A, B \in \mathbf{C}$ , a natural isomorphism  $F_2(A, B) : F(A) \otimes_{\mathbf{D}} F(B) \xrightarrow{\sim} F(A \otimes_{\mathbf{C}} B)$ .

It is called an **equivalence of categories with bifunctor** if the underlying functor  $F$  is an equivalence of categories.

**REMARK 3.6.** If  $(F, F_2)$  and  $(G, G_2)$  are functors between categories with bifunctors such that  $F$  and  $G$  are composable, then we equip  $G \circ F$  with the following data to become a functor between categories with bifunctors:

- $(G \circ F)_2(A, B) := G(F_2(A, B)) \circ G_2(F(A), F(B))$ .

For understanding the ways in which  $k$ -vec and more generally  $\bigoplus_{i \in I} k$ -vec can be turned into categories with bifunctor, we first need tools to transfer the classification result of  $k$ -linear functors on  $\bigoplus_{i \in I} k$ -vec (see Lemma I.2.22) to bilinear bifunctors.

**Definition 3.7.** Let  $\mathbf{A}, \mathbf{B}$  be Ab-categories. We define the **Ab-product category**  $\mathbf{A} \otimes \mathbf{B}$  as follows:

- (1)  $\text{Obj}_{\mathbf{A} \otimes \mathbf{B}} := \text{Obj}_{\mathbf{A}} \times \text{Obj}_{\mathbf{B}}$ .
- (2) For  $A, A' \in \mathbf{A}$ ,  $B, B' \in \mathbf{B}$ , we set

$$\text{Hom}_{\mathbf{A} \otimes \mathbf{B}}((A, B), (A', B')) := \text{Hom}_{\mathbf{A}}(A, A') \otimes_{\mathbb{Z}} \text{Hom}_{\mathbf{B}}(B, B').$$

- (3) Composition is given on elementary tensors by the interchange law of  $\otimes_{\mathbb{Z}}$ .
- (4) Identity of  $(A, B)$  is given by  $\text{id}_A \otimes_{\mathbb{Z}} \text{id}_B$ .

Clearly,  $\mathbf{A} \otimes \mathbf{B}$  is an Ab-category.

**REMARK 3.8.** Bilinear bifunctors  $T : \mathbf{A} \times \mathbf{B} \rightarrow \mathbf{C}$  correspond bijectively to Ab-functors  $B : \mathbf{A} \otimes \mathbf{B} \rightarrow \mathbf{C}$ . Under this correspondence, we have equality on the level of objects, and the bilinear maps on homomorphisms of  $T$  are in 1 : 1 correspondence to linear maps of the tensor products over  $\mathbb{Z}$ . In particular, we obtain a notion of equivalence of bilinear bifunctors.

**Lemma 3.9.** *Let  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  be Ab-categories. Then currying gives rise to an equivalence of categories:*

$$\text{Hom}_{\text{Ab}}(\mathbf{A} \otimes \mathbf{B}, \mathbf{C}) \simeq \text{Hom}_{\text{Ab}}(\mathbf{A}, \text{Hom}_{\text{Ab}}(\mathbf{B}, \mathbf{C})),$$

where  $\text{Hom}_{\text{Ab}}$  denotes the category of Ab-functors.

**PROOF.** We will describe mutually inverse functors

$$\begin{array}{ccc} & G & \\ & \curvearrowright & \\ \text{Hom}_{\text{Ab}}(\mathbf{A} \otimes \mathbf{B}, \mathbf{C}) & & \text{Hom}_{\text{Ab}}(\mathbf{A}, \text{Hom}_{\text{Ab}}(\mathbf{B}, \mathbf{C})) \\ & \curvearrowleft & \\ & H & \end{array}$$

We will only describe them on objects, for the action on morphisms can be given canonically.

By Remark I.3.8 we may describe  $G$  by starting with a bilinear bifunctor  $T : \mathbf{A} \times \mathbf{B} \rightarrow \mathbf{C}$ . We set  $G(T)(\alpha : A \rightarrow A') := B(\alpha, -) : B(A, -) \rightarrow B(A', -)$ . Conversely, given  $F : \mathbf{A} \rightarrow \text{Hom}_{\text{Ab}}(\mathbf{B}, \mathbf{C})$ , we define a bilinear functor  $H(F)(\alpha, \beta) := F(\alpha)(\beta)$ . We have  $GH(F) = F$  and  $HG(T) = T$ .  $\square$

Lemma I.3.9 has an obvious generalization to  $k$ -linear categories.

The next lemma and corollary give a classification of bilinear bifunctors  $F$  on  $\bigoplus_{i \in I} k\text{-vec}$ : They are uniquely determined by their values on all pairs of the form  $(\chi^i, \chi^j)$ . Thus, we can use this lemma to understand in what different ways  $\bigoplus_{i \in I} k\text{-vec}$  can be equipped with a bilinear bifunctor (see Construction I.3.12).

**Lemma 3.10.** *Let  $I, J$  be sets. Let  $\mathbf{A}$  be a  $k$ -linear category. Then evaluation at the objects  $(\chi^i, \chi^j)$*

$$F \mapsto \left( F((\chi^i, \chi^j)) \right)_{i \in I, j \in J}$$

*yields an equivalence of categories:*

$$\text{Hom}_k \left( \left( \bigoplus_{i \in I} k\text{-vec} \right) \otimes \left( \bigoplus_{j \in J} k\text{-vec} \right), \mathbf{A} \right) \simeq \prod_{i \in I} \prod_{j \in J} \mathbf{A}.$$

PROOF. Combining the results of Lemma I.3.9 and Lemma I.2.22, we get a chain of equivalences:

$$\begin{aligned} \text{Hom}_k \left( \left( \bigoplus_{i \in I} k\text{-vec} \right) \otimes \left( \bigoplus_{j \in J} k\text{-vec} \right), \mathbf{A} \right) &\simeq \text{Hom}_k \left( \bigoplus_{i \in I} k\text{-vec}, \text{Hom}_k \left( \bigoplus_{j \in J} k\text{-vec}, \mathbf{A} \right) \right) \\ &\simeq \prod_{i \in I} \text{Hom}_k \left( \bigoplus_{j \in J} k\text{-vec}, \mathbf{A} \right) \simeq \prod_{i \in I} \prod_{j \in J} \mathbf{A}. \end{aligned} \quad \square$$

**Corollary 3.11.** *Let  $I, J, L$  be sets. Any two  $k$ -functors*

$$F, G : \left( \bigoplus_{i \in I} k\text{-vec} \right) \otimes \left( \bigoplus_{j \in J} k\text{-vec} \right) \rightarrow \bigoplus_{l \in L} k\text{-vec}$$

*are naturally isomorphic if and only if they yield equal functions on objects.*

PROOF. This is a direct consequence of Lemma I.3.10.  $\square$

It follows that  $\otimes$  on  $k\text{-vec}$  is uniquely determined as a  $k$ -bilinear bifunctor up to natural isomorphism by the equation  $1 \otimes 1 = 1$ .

In the next construction we give formulas for  $k$ -bilinear bifunctors on  $\bigoplus_{i \in I} k\text{-vec}$  which are based on Kronecker products. Such a formula is needed for an implementation of a specific instance of the category  $\bigoplus_{i \in I} k\text{-vec}$  equipped with a bifunctor.

**Construction 3.12.** Let  $I$  be a set and let

$$n := \left( \bigoplus_{l \in I} n(i, j)_l \cdot \chi^l \right)_{i, j \in I} \in \prod_{i, j \in I} k\text{-vec}$$



be an  $I \times I$ -indexed family of objects in  $k\text{-vec}$ . By Lemma I.3.10, there exists a bilinear bifunctor (unique up to natural isomorphism)

$$\otimes_n : \left( \bigoplus_{i \in I} k\text{-vec} \times \bigoplus_{i \in I} k\text{-vec} \right) \rightarrow \bigoplus_{i \in I} k\text{-vec}$$

with the property

$$\chi^i \otimes_n \chi^j = \bigoplus_{l \in I} n(i, j)_l \cdot \chi^l.$$

Now, we construct one particular instance of such a functor for given  $n$ : Given morphisms  $\alpha = (\alpha^i)_{i \in I}$  and  $\gamma = (\gamma^j)_{j \in I}$ , we set the  $l$ -th component of  $\alpha \otimes_n \gamma$  as

$$(\alpha \otimes_n \gamma)_l := \bigoplus_{i, j} (\alpha^i \otimes \gamma^j \otimes I_{n(i, j)_l}),$$

where  $I$  denotes the identity matrix and  $\otimes$  the Kronecker product of matrices. Functoriality of  $\otimes_n$  follows from the interchange law of the Kronecker product.

The next theorem states that all ways to equip  $\bigoplus_{i \in I} k\text{-vec}$  with a bilinear bifunctor are given by  $(\bigoplus_{i \in I} k\text{-vec}, \otimes_n)$ . Since it is our goal to turn  $\bigoplus_{i \in I} k\text{-vec}$  into a tensor category equivalent to  $\text{Rep}_k(G)$ , this theorem already tells us how to define the tensor product functor on  $\bigoplus_{i \in I} k\text{-vec}$  (see Theorem I.3.36).

**Theorem 3.13.** *Let  $I$  be a set and let*

$$\otimes : \bigoplus_{i \in I} k\text{-vec} \times \bigoplus_{i \in I} k\text{-vec} \rightarrow \bigoplus_{i \in I} k\text{-vec}$$

*be a  $k$ -bilinear bifunctor. Furthermore, let*

$$n := (\chi^i \otimes \chi^j)_{i, j \in I} \in \prod_{i, j \in I} k\text{-vec}$$

*and let  $\otimes_n$  be the  $k$ -bilinear bifunctor from Construction I.3.12. Then there is an equivalence of categories with bifunctors*

$$\left( \bigoplus_{i \in I} k\text{-vec}, \otimes \right) \simeq \left( \bigoplus_{i \in I} k\text{-vec}, \otimes_n \right)$$

*whose underlying functor is given by the identity.*

**PROOF.** By Lemma I.3.10, the  $k$ -bilinear bifunctor  $\otimes$  is naturally isomorphic to  $\otimes_n$ , which is the claim.  $\square$

**3.2. Monoidal Categories.** In this subsection we start with the definition of a *monoidal category* and end with the definition of a *tensor category*. Tensor categories provide the categorical structure needed for the computation of  $G$ -equivariant cohomology groups in Section III.3.

**Definition 3.14** ([ML71]). A **monoidal category** consists of the following data:

- (1) A category with bifunctor  $(\mathbf{C}, \otimes : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C})$ .
- (2) An object  $1 \in \mathbf{C}$ , called **tensor unit**.

- (3) For  $A, B, C \in \mathbf{C}$ , a natural isomorphism  $\alpha_{A,B,C} : A \otimes (B \otimes C) \xrightarrow{\sim} (A \otimes B) \otimes C$ , called **associator**.
- (4) For  $A \in \mathbf{C}$ , a natural isomorphism  $\lambda_A : 1 \otimes A \xrightarrow{\sim} A$ , called **left unitor**.
- (5) For  $A \in \mathbf{C}$ , a natural isomorphism  $\rho_A : A \otimes 1 \xrightarrow{\sim} A$ , called **right unitor**.
- (6) For  $A, B, C, D \in \mathbf{C}$ , the following pentagonal diagram commutes:

$$\begin{array}{ccc}
 & A \otimes (B \otimes (C \otimes D)) & \\
 & \swarrow^{A \otimes \alpha_{B,C,D}} \quad \searrow^{\alpha_{A,B,C \otimes D}} & \\
 A \otimes ((B \otimes C) \otimes D) & & (A \otimes B) \otimes (C \otimes D) \\
 & \swarrow^{\alpha_{A,B \otimes C,D}} \quad \searrow^{\alpha_{A \otimes B,C,D}} & \\
 (A \otimes (B \otimes C)) \otimes D & \xrightarrow{\alpha_{A,B,C \otimes D}} & ((A \otimes B) \otimes C) \otimes D
 \end{array}$$

- (7) For  $A, C \in \mathbf{C}$ , the following triangular diagram commutes:

$$\begin{array}{ccc}
 A \otimes (1 \otimes C) & \xrightarrow{\alpha_{A,1,C}} & (A \otimes 1) \otimes C \\
 & \searrow^{A \otimes \lambda_C} & \downarrow \rho_{A \otimes C} \\
 & & A \otimes C
 \end{array}$$

**Construction 3.15.** In the case of  $k$ -vec and Kronecker products  $\otimes$ , it is simple to define a monoidal structure, since Kronecker products are strictly associative:

- (1) The tensor unit is given by  $1 \in \mathbb{N}_0$ .
- (2) The associator, left unitor, and right unitor are given by identity matrices.

**Computation 3.16.** Note that even though the associator and the unitors in  $k$ -vec are given by identity matrices, this is not the case for all structure morphisms. To see an example, we compute the natural left distributivity morphism

$$2 \otimes (1 \oplus 1) \xrightarrow{\sim} (2 \otimes 1) \oplus (2 \otimes 1)$$

with the CAP package `LinearAlgebraForCAP`, which yields the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

as a result.

```

gap> Q := HomalgFieldOfRationals();;
gap> U := TensorUnit( MatrixCategory( Q ) );
<A vector space object over Q of dimension 1>
gap> V := DirectSum( U, U );

```

```

<A vector space object over Q of dimension 2>
gap> Display( LeftDistributivityExpanding( V, [ U, U ] ) );
[ [ 1, 0, 0, 0 ],
  [ 0, 0, 1, 0 ],
  [ 0, 1, 0, 0 ],
  [ 0, 0, 0, 1 ] ]

A morphism in Category of matrices over Q

```

Here, CAP uses the constructions of Lemma 1.2.10 for the computation.

An arbitrary functor between the underlying categories of two monoidal categories does not have to respect the monoidal structures. Thus, such a preservation has to be ensured by an extra datum. Note that this situation differs from the case of additive categories, where Ab-functors automatically preserve direct sums and the extra datum comes for free (see Lemma 1.2.10).

**Definition 3.17.** Let  $(\mathbf{C}, \otimes_{\mathbf{C}}, 1^{\mathbf{C}}, \alpha^{\mathbf{C}}, \lambda^{\mathbf{C}}, \rho^{\mathbf{C}})$  and  $(\mathbf{D}, \otimes_{\mathbf{D}}, 1^{\mathbf{D}}, \alpha^{\mathbf{D}}, \lambda^{\mathbf{D}}, \rho^{\mathbf{D}})$  be monoidal categories. A **monoidal functor** from  $\mathbf{C}$  to  $\mathbf{D}$  consists of the following data:

- (1) A functor between categories with bifunctor<sup>2</sup>  $(F, F_2) : (\mathbf{C}, \otimes_{\mathbf{C}}) \rightarrow (\mathbf{D}, \otimes_{\mathbf{D}})$ .
- (2) An isomorphism  $F_0 : 1^{\mathbf{D}} \xrightarrow{\sim} F(1^{\mathbf{C}})$ .
- (3) For  $A, B, C \in \mathbf{C}$ , the following diagram commutes:

$$\begin{array}{ccc}
 F(A) \otimes_{\mathbf{D}} (F(B) \otimes_{\mathbf{D}} F(C)) & \xrightarrow{\alpha_{F(A), F(B), F(C)}^{\mathbf{D}}} & (F(A) \otimes_{\mathbf{D}} F(B)) \otimes_{\mathbf{D}} F(C) \\
 \downarrow F(A) \otimes_{\mathbf{D}} F_2(B, C) & & \downarrow F_2(A, B) \otimes_{\mathbf{D}} F(C) \\
 F(A) \otimes_{\mathbf{D}} F(B \otimes_{\mathbf{C}} C) & & F(A \otimes_{\mathbf{C}} B) \otimes_{\mathbf{D}} F(C) \\
 \downarrow F_2(A, B \otimes_{\mathbf{C}} C) & & \downarrow F_2(A \otimes_{\mathbf{C}} B, C) \\
 F(A \otimes_{\mathbf{C}} (B \otimes_{\mathbf{C}} C)) & \xrightarrow{F(\alpha_{A, B, C}^{\mathbf{C}})} & F((A \otimes_{\mathbf{C}} B) \otimes_{\mathbf{C}} C)
 \end{array}$$

- (4) For  $A \in \mathbf{C}$ , the following diagram commutes:

$$\begin{array}{ccc}
 F(A) \otimes_{\mathbf{D}} 1^{\mathbf{D}} & \xrightarrow{\rho_{F(A)}} & F(A) \\
 \downarrow F(A) \otimes_{\mathbf{D}} F_0 & & \uparrow F(\rho_A) \\
 F(A) \otimes_{\mathbf{D}} F(1^{\mathbf{C}}) & \xrightarrow{F_2(A, 1^{\mathbf{C}})} & F(A \otimes_{\mathbf{C}} 1^{\mathbf{C}})
 \end{array}$$

<sup>2</sup>Cf. Definition 1.3.5

(5) For  $A \in \mathbf{C}$ , the following diagram commutes:

$$\begin{array}{ccc}
 1^{\mathbf{D}} \otimes_{\mathbf{D}} F(A) & \xrightarrow{\lambda_{F(A)}} & F(A) \\
 F_0 \otimes_{\mathbf{D}} F(A) \downarrow & & \uparrow F(\lambda_A) \\
 F(1^{\mathbf{C}}) \otimes_{\mathbf{D}} F(A) & \xrightarrow{F_2(1^{\mathbf{C}}, A)} & F(1^{\mathbf{C}} \otimes_{\mathbf{C}} A)
 \end{array}$$

In the definition of a monoidal functor  $F$  given in [ML71],  $F_0$  and  $F_2$  only have to be morphisms. If they are isomorphisms,  $F$  is called strong. Since we will only deal with the case where  $F_0$  and  $F_2$  are isomorphisms, we omit this adjective.

REMARK 3.18. For two composable monoidal functors  $F, G$ , we equip  $G \circ F$  with the following data to become a monoidal functor:

- (1)  $(G \circ F)_0 := G(F_0) \circ G_0$ ,
- (2)  $(G \circ F)_2(A, B) := G(F_2(A, B)) \circ G_2(F(A), F(B))$ .

**Definition 3.19** ([ML71]). Given two monoidal functors  $(F, F_0, F_2), (G, G_0, G_2)$  from  $(\mathbf{C}, \otimes_{\mathbf{C}}, 1^{\mathbf{C}})$  to  $(\mathbf{D}, \otimes_{\mathbf{D}}, 1^{\mathbf{D}})$ , a **monoidal natural transformation** consists of the following data:

- (1) A natural transformation  $\nu : F \rightarrow G$ .
- (2) For  $A, B \in \mathbf{C}$ , the following diagram commutes:

$$\begin{array}{ccc}
 F(A) \otimes_{\mathbf{D}} F(B) & \xrightarrow{F_2(A, B)} & F(A \otimes_{\mathbf{C}} B) \\
 \nu_A \otimes_{\mathbf{D}} \nu_B \downarrow & & \downarrow \nu_{A \otimes_{\mathbf{C}} B} \\
 G(A) \otimes_{\mathbf{D}} G(B) & \xrightarrow{G_2(A, B)} & G(A \otimes_{\mathbf{C}} B)
 \end{array}$$

(3) The following diagram commutes:

$$\begin{array}{ccc}
 1^{\mathbf{D}} & \xrightarrow{F_0} & F(1^{\mathbf{C}}) \\
 & \searrow G_0 & \downarrow \nu_{1^{\mathbf{C}}} \\
 & & G(1^{\mathbf{C}})
 \end{array}$$

**Definition 3.20.** We call a monoidal functor

$$(F, F_0, F_2) : (\mathbf{C}, \otimes_{\mathbf{C}}, 1^{\mathbf{C}}, \alpha^{\mathbf{C}}, \lambda^{\mathbf{C}}, \rho^{\mathbf{C}}) \rightarrow (\mathbf{D}, \otimes_{\mathbf{D}}, 1^{\mathbf{D}}, \alpha^{\mathbf{D}}, \lambda^{\mathbf{D}}, \rho^{\mathbf{D}})$$

a **monoidal equivalence** if  $F$  is an equivalence of categories.

This notion of a monoidal equivalence is justified in [SR72] in the sense that there exists an appropriate inverse. In Remark 1.3.39 we will see two inequivalent monoidal

categories with equal underlying category with bifunctors. This illustrates the importance of treating the associator as an extra datum and not merely as a property (however, the unitors may be seen as a categorical property, see [SR72] for details).

**Definition 3.21** ([ML71]). A **braiding** for a monoidal category  $(\mathbf{C}, \otimes, 1, \alpha, \lambda, \rho)$  consists of the following data:

- (1) For  $A, B \in \mathbf{C}$ , a natural isomorphism  $\gamma_{A,B} : A \otimes B \xrightarrow{\sim} B \otimes A$ .
- (2) Compatibility with unitors: For  $A \in \mathbf{C}$ , the following diagram commutes:

$$\begin{array}{ccc} A \otimes 1 & \xrightarrow{\gamma_{A,1}} & 1 \otimes A \\ & \searrow \rho_A & \swarrow \lambda_A \\ & & A \end{array}$$

- (3) For  $A, B, C \in \mathbf{C}$ , the following diagram swapping  $B$  and  $C$  commutes:

$$\begin{array}{ccc} (A \otimes B) \otimes C & \xrightarrow{\gamma_{A \otimes B, C}} & C \otimes (A \otimes B) \\ \alpha_{A,B,C}^{-1} \downarrow & & \downarrow \alpha_{C,A,B} \\ A \otimes (B \otimes C) & & (C \otimes A) \otimes B \\ A \otimes \gamma_{B,C} \downarrow & & \downarrow \gamma_{C,A} \otimes B \\ A \otimes (C \otimes B) & \xrightarrow{\alpha_{A,C,B}} & (A \otimes C) \otimes B \end{array}$$

- (4) For  $A, B, C \in \mathbf{C}$ , the following diagram swapping  $A$  and  $B$  commutes:

$$\begin{array}{ccc} A \otimes (B \otimes C) & \xrightarrow{\gamma_{A, B \otimes C}} & (B \otimes C) \otimes A \\ \alpha_{A,B,C} \downarrow & & \downarrow \alpha_{B,C,A}^{-1} \\ (A \otimes B) \otimes C & & B \otimes (C \otimes A) \\ \gamma_{A,B} \otimes C \downarrow & & \downarrow B \otimes \gamma_{C,A} \\ (B \otimes A) \otimes C & \xrightarrow{\alpha_{B,A,C}^{-1}} & B \otimes (A \otimes C) \end{array}$$

Kronecker products are not commutative. Of course, this does not prevent  $k\text{-vec}$  from being a braided monoidal category, it simply means that we cannot expect the braiding to be given by identity matrices.

**Construction 3.22.** Here is how we define the braiding on  $k\text{-vec}$ :

- Let  $m, n \in \mathbb{N}_0$ ,  $B := (\{1, \dots, m\} \times \{1, \dots, n\}, <)$  be the totally ordered set with the lexicographical ordering, i.e.,  $(i, j) < (i', j') \Leftrightarrow (i < i') \text{ or } ((i = i') \wedge (j < j'))$ . Let further  $\nu : (\{1, \dots, mn\}, <) \rightarrow B$  denote the unique order preserving map. The braiding  $\gamma_{m,n}$  is given by the row permutation matrix defined by the permutation

$$\{1, \dots, mn\} \xrightarrow{\nu} B \xrightarrow{(i, j) \mapsto (j, i)} B \xrightarrow{\nu^{-1}} \{1, \dots, mn\}.$$

For example, if  $m = n = 2$ , we have

$$\begin{aligned} \nu : \{1, 2, 3, 4\} &\rightarrow \{1, 2\} \times \{1, 2\} \\ 1 &\mapsto (1, 1) \\ 2 &\mapsto (1, 2) \\ 3 &\mapsto (2, 1) \\ 4 &\mapsto (2, 2) \end{aligned}$$

which induces the permutation  $(2, 3)$  (as a cycle in the permutation group on  $\{1, 2, 3, 4\}$ ). Thus, the braiding  $\gamma_{2,2}$  is given by the  $4 \times 4$  permutation matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

which is not the identity matrix.

**Definition 3.23** ([ML71]). Let  $(\mathbf{C}, \otimes_{\mathbf{C}}, \alpha^{\mathbf{C}}, \gamma^{\mathbf{C}})$  and let  $(\mathbf{D}, \otimes_{\mathbf{D}}, \alpha^{\mathbf{D}}, \gamma^{\mathbf{D}})$  be braided monoidal categories. A **braided monoidal functor** from  $\mathbf{C}$  to  $\mathbf{D}$  consists of the following data:

- (1) A monoidal functor  $(F, F_0, F_2)$ .
- (2) For  $A, B \in \mathbf{C}$ , the following diagram commutes (compatibility with braiding):

$$\begin{array}{ccc} FA \otimes_{\mathbf{D}} FB & \xrightarrow{\gamma_{FA, FB}^{\mathbf{D}}} & FB \otimes_{\mathbf{D}} FA \\ \downarrow F_2(A, B) & & \downarrow F_2(B, A) \\ F(A \otimes_{\mathbf{C}} B) & \xrightarrow{F(\gamma_{A, B}^{\mathbf{C}})} & F(B \otimes_{\mathbf{C}} A). \end{array}$$

Since a braided monoidal category equips a monoidal category only with a natural family of isomorphisms, a braided monoidal functor does not have to be equipped with any object or morphism datum. In particular, the notion of a braided monoidal natural transformation coincides with the one of a monoidal natural transformation.

**REMARK 3.24.** Mac Lane’s coherence theorem [ML71] states that all formal expressions of natural isomorphisms which are built up from  $\alpha, \lambda, \rho$  coincide if they have the same formal source and the same formal range. It is due to this coherence theorem that we may omit parentheses whenever we formally work with monoidal categories. Such a coherence theorem cannot be expected for braided monoidal categories, since  $\gamma_{B,A} \circ \gamma_{A,B} \neq \text{id}_{A \otimes B}$  in general. However, there exists a graphical language, called **string diagrams**, which ‘compiles’ any formal expression of a braided monoidal category to a graph consisting of strings and nodes. Equality of two formal expressions can be decided by comparing their corresponding graphs. In this sense string diagrams are a generalization of Mac Lane’s coherence theorem. A very good survey on the string diagram calculus for numerous kinds of monoidal categories can be found in [Sel11].

**Definition 3.25.** A braided monoidal category  $(\mathbf{C}, \otimes, \gamma)$  is called **symmetric** if

$$\gamma_{B,A} \circ \gamma_{A,B} = \text{id}_{A \otimes B}$$

for all objects  $A, B \in \mathbf{C}$ .

The braiding which we constructed for  $k\text{-vec}$  is clearly symmetric.

**Definition 3.26** ([ML71]). A **closed category** consists of the following data:

- (1) A symmetric monoidal category  $(\mathbf{C}, \otimes)$ .
- (2) For each  $A \in \mathbf{C}$ , a right adjoint  $\underline{\text{Hom}}(A, -) : \mathbf{C} \rightarrow \mathbf{C}$  to  $(-) \otimes A$ , called the **internal Hom**.

**Definition 3.27.** For an object  $A$  in a closed category  $\mathbf{C}$ , we define its **dual object** as  $A^\vee := \underline{\text{Hom}}(A, 1)$ .

There is a more restrictive context in which dual objects can be introduced.

**Definition 3.28.** A **compact closed category** or **rigid symmetric monoidal category** consists of the following data:

- (1) A symmetric monoidal category  $(\mathbf{C}, \otimes, 1)$ .
- (2) A dependent function mapping an object  $A \in \mathbf{C}$  to an **exact pairing**, i.e., an object  $A^\vee \in \mathbf{C}$  (called **dual object of  $A$** ) and morphisms  $\eta_A : 1 \rightarrow A^\vee \otimes A$  and  $\epsilon_A : A \otimes A^\vee \rightarrow 1$  satisfying the zig-zag identities:

$$\begin{array}{ccc}
 A & \xrightarrow{A \otimes \eta_A} & A \otimes A^\vee \otimes A \\
 & \searrow \text{id}_A & \downarrow \epsilon_A \otimes A \\
 & & A
 \end{array}
 \qquad
 \begin{array}{ccc}
 A^\vee & \xrightarrow{\eta_A \otimes A^\vee} & A^\vee \otimes A \otimes A^\vee \\
 & \searrow \text{id}_{A^\vee} & \downarrow A^\vee \otimes \epsilon_A \\
 & & A^\vee
 \end{array}$$

REMARK 3.29. In a compact closed category, assigning to  $A$  its dual object  $A^\vee$  is a contravariant functorial operation: Dualizing a morphism  $\alpha : A \rightarrow B$  is given by the composition

$$B^\vee \xrightarrow{\eta_A \otimes B^\vee} A^\vee \otimes A \otimes B^\vee \xrightarrow{A^\vee \otimes \alpha \otimes B^\vee} A^\vee \otimes B \otimes B^\vee \xrightarrow{A^\vee \otimes \epsilon_B} A^\vee.$$

REMARK 3.30. A compact closed category gives rise to a closed category by setting  $\underline{\text{Hom}}(A, B) := A^\vee \otimes B$ . In particular,  $\underline{\text{Hom}}(A, 1) \cong A^\vee$ , which justifies the notation of dual objects in Definition 1.3.27 and Definition 1.3.28. In the case of a compact closed category the natural morphism  $A \rightarrow (A^\vee)^\vee$  is always an isomorphism.

**Construction 3.31.** We introduce dual objects in  $k\text{-vec}$ :

- For  $m \in \mathbb{N}_0$ ,  $m^\vee := m$ .
- For  $m \in \mathbb{N}_0$ ,  $\nu_m : 1 \rightarrow m^\vee \otimes m$  is given by the concatenation of the rows of the  $m \times m$  identity matrix, which yields one single row.
- For  $m \in \mathbb{N}_0$ ,  $\epsilon_m : m \otimes m^\vee \rightarrow 1$  is given by the transposed matrix of  $\nu_m$ .

From the definition of  $\nu$  and  $\epsilon$ , it follows that the action of  $(-)^\vee$  on morphisms is given by transposing matrices (see Remark 1.3.29).

This turns  $k\text{-vec}$  into a tensor category.

**Definition 3.32** ([Del90]). Let  $k$  be a field. A **tensor category over  $k$**  is an abelian rigid symmetric monoidal  $k$ -linear category such that the tensor product is  $k$ -bilinear and  $\text{End}(1) = k \cdot \text{id}_1$ .

**Example 3.33.** The abelian  $k$ -linear category  $\bigoplus_{d \in \mathbb{Z}} k\text{-vec}$  can be regarded as the category of  $\mathbb{Z}$ -graded vector spaces. We give two inequivalent choices of tensor category structures on  $\bigoplus_{d \in \mathbb{Z}} k\text{-vec}$ . A first one can be defined using the forgetful functor  $\bigoplus_{d \in \mathbb{Z}} k\text{-vec} \rightarrow k\text{-vec}$  and structure transport. We denote the braiding of this first tensor structure by  $\gamma$ . A second tensor structure can be defined using the embedding of  $\bigoplus_{d \in \mathbb{Z}} k\text{-vec}$  in the category of chain complexes over  $k$  and structure transport. We denote the braiding of this second tensor structure by  $\gamma'$ . If  $V$  denotes a 1-dimensional space sitting in degree 1, then there is no way of rendering the diagram

$$\begin{array}{ccc} V \otimes V & \xrightarrow{\gamma_{V,V} = (1)} & V \otimes V \\ \downarrow F_2(V, V) & & \downarrow F_2(V, V) \\ V \otimes V & \xrightarrow{\gamma'_{V,V} = (-1)} & V \otimes V \end{array}$$

commutative, which shows the inequivalence of these tensor structures.



**Convention.** Without further specification, we regard the category of graded  $k$ -vector spaces as a tensor category equipped with the first structure presented in Example I.3.33, i.e., no signs are involved in the braiding.

**3.3. Skeletal Tensor Categories.** In this subsection we turn to the construction of a tensor category  $\text{SRep}_k(G)$  equivalent to the category of group representations  $\text{Rep}_k(G)$  for a finite group  $G$  over a splitting field  $k$  for  $G$ . The underlying abelian category of  $\text{SRep}_k(G)$  is given by  $\bigoplus_{i \in I} k\text{-vec}$  for a set  $I$  which is in bijection to irreducible characters of  $G$ . In particular, the objects in  $\text{SRep}_k(G)$  are simply given by finite lists of non-negative integers, and just like in the case of  $k\text{-vec}$ , we are forced by this model to think in purely categorical terms: It does not make sense to ‘evaluate an object in  $\text{SRep}_k(G)$  at a group element’, a process which we can do for objects in  $\text{Rep}_k(G)$ . However, all we need for the computation of  $G$ -equivariant cohomology groups in Section III.3 is the tensor category structure of  $\text{Rep}_k(G)$ , and since the data structures for objects and morphisms in  $\text{SRep}_k(G)$  are very easy, we will prefer  $\text{SRep}_k(G)$  as our computational model.

Here is the plan for defining a tensor category structure on  $\text{SRep}_k(G)$ :

- (1) Start with an equivalence

$$F : \text{SRep}_k(G) \xrightarrow{\sim} \text{Rep}_k(G)$$

between abelian categories.

- (2) Use structure transport to transfer the well-known tensor product of  $\text{Rep}_k(G)$  to  $\text{SRep}_k(G)$  (see Theorem I.3.36).
- (3) Use structure transport to induce
  - an associator (see Subsection I.3.3.3),
  - a braiding (see Subsection I.3.3.4),
 on  $\text{SRep}_k(G)$  compatible with  $F$ .
- (4) Define unitors (see Subsection I.3.3.5) compatible with the associator and braiding.
- (5) Define duals (see Subsection I.3.3.6).

All these construction steps are quite technical, but suitable for a computer implementation. And once these methods are implemented for  $\text{SRep}_k$ , we do not have to care about their internals anymore, just like in the case of computing kernels in  $k\text{-vec}$  via Gaussian elimination.

**Notation 3.34.** For our notational conventions concerning  $\bigoplus_{i \in I} k\text{-vec}$  see Definition I.2.21.

**3.3.1. Representation Category of Finite Groups.** Let  $G$  be a finite group and let  $k$  be a splitting field for  $G$ , i.e., we have  $\text{char}(k) \nmid |G|$  and any irreducible representation is absolutely irreducible. By  $\text{Irr}(G)$  we denote the set of irreducible  $k$ -characters equipped with an arbitrary total order (so that we are able to address the  $i$ -th element in  $\text{Irr}(G)$ ).

By  $\mathbf{B}G$ , we denote the delooping of  $G$ , i.e., the category with only one object  $\bullet$ , homomorphisms given by  $\text{Hom}_{\mathbf{B}G}(\bullet, \bullet) = G$ , composition  $\circ$  given by group multiplication, and identity given by  $e$ , the neutral element of  $G$ . The **representation category of  $G$**  is

defined as the functor category

$$\mathrm{Rep}_k(G) := \mathrm{Hom}(\mathbf{B}G, k\text{-vec}).$$

It inherits its tensor category structure over  $k$  from  $k\text{-vec}$  in a natural way: Given two representations  $V, V' \in \mathrm{Hom}(\mathbf{B}G, k\text{-vec})$ , their tensor product is given by

$$(V \otimes V')(g) = V(g) \otimes V'(g)$$

for  $g \in G$ . The associator, braiding, unitors, and duals are directly induced by the corresponding concepts in  $k\text{-vec}$ . See [Day74] for a general treatment of monoidal structures on categories of functors. Furthermore,  $\mathrm{Rep}_k(G)$  is a strict monoidal category, i.e., we have:

- For all  $V \in \mathrm{Rep}_k(G)$ :  $1 \otimes V = V \otimes 1 = V$ ,
- For all  $V \in \mathrm{Rep}_k(G)$ :  $\lambda_V = \mathrm{id}_V = \rho_V$ ,
- For all  $A, B, C \in \mathrm{Rep}_k(G)$ :  $(A \otimes B) \otimes C = A \otimes (B \otimes C)$ ,
- For all  $A, B, C \in \mathrm{Rep}_k(G)$ :  $\alpha_{A,B,C} = \mathrm{id}_{(A \otimes B) \otimes C}$ .

This is true since we constructed  $k\text{-vec}$  as a strict monoidal category.

Let  $I := \{1, \dots, |\mathrm{Irr}(G)|\}$ . For  $i \in I$ , we can use algorithms from representation theory to construct representations  $V^i$  affording the  $i$ -th irreducible character in  $\mathrm{Irr}(G)$ . Such algorithms are for example provided by the GAP package `reprsn` [Dab11]. We define a functor

$$F : \bigoplus_{i \in I} k\text{-vec} \rightarrow \mathrm{Rep}_k(G)$$

by setting  $F(\chi^i) = V^i$  and then employing the construction in the proof of Lemma 1.2.22 to extend  $F$  to a functor strictly commuting with direct sums. Due to our assumptions on  $k$ , all objects in  $\mathrm{Rep}_k(G)$  are semisimple by Maschke's theorem and  $F$  gives rise to an equivalence of categories.

For every pair  $a, b \in I$ , there exist an isomorphism

$$\epsilon_{a,b} : \bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} V^i \xrightarrow{\sim} V^a \otimes V^b$$

for natural numbers  $n(a,b)_i \in \mathbb{N}_0$ . Such isomorphisms can be constructed by computing a  $k$ -basis of the  $G$  fixed points of  $\mathrm{Hom}_k(V^i, V^a \otimes V^b)$  for all  $i \in I$ , which can be done by solving linear equations.

3.3.2. *Defining a Bifunctor.* We will use equivalences of categories to transport a bifunctor from one category to another.

**Construction 3.35** (Structure transport of bifunctors). Let  $\mathbf{A}$  be a  $k$ -linear category and  $(\mathbf{B}, \otimes_{\mathbf{B}})$  be a  $k$ -linear category with a  $k$ -bilinear bifunctor. Let furthermore

$$\begin{array}{ccc} & F & \\ \mathbf{A} & \xrightarrow{\quad} & (\mathbf{B}, \otimes_{\mathbf{B}}) \\ & R & \end{array}$$

be an equivalence with natural isomorphism  $\mu_B : FR(B) \xrightarrow{\sim} B$  for  $B \in \mathbf{B}$ . Then we construct the  $k$ -bilinear bifunctor

$$\otimes_{\mathbf{A}} := \mathbf{A} \times \mathbf{A} \xrightarrow{F \times F} \mathbf{B} \times \mathbf{B} \xrightarrow{\otimes_{\mathbf{B}}} \mathbf{B} \xrightarrow{R} \mathbf{A}$$

on  $\mathbf{A}$ . Furthermore, we set

$$F_2(A_1, A_2) := \mu_{FA_1 \otimes_{\mathbf{B}} FA_2}^{-1} : FA_1 \otimes_{\mathbf{B}} FA_2 \xrightarrow{\sim} F(R(FA_1 \otimes_{\mathbf{B}} FA_2)) = F(A_1 \otimes_{\mathbf{A}} A_2)$$

which gives us an equivalence

$$(F, F_2) : (\mathbf{A}, \otimes_{\mathbf{A}}) \xrightarrow{\sim} (\mathbf{B}, \otimes_{\mathbf{B}})$$

of categories with bifunctor.

Now, we transport the tensor product of  $\text{Rep}_k(G)$  to  $\bigoplus_{i \in I} k\text{-vec}$ .

**Theorem 3.36.** *Let  $\text{Rep}_k(G)$  be the representation category of  $G$  with the given equivalence*

$$F : \bigoplus_{i \in I} k\text{-vec} \xrightarrow{\sim} \text{Rep}_k(G)$$

and isomorphisms

$$\epsilon_{a,b} : \bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} V^i \xrightarrow{\sim} V^a \otimes V^b$$

for  $a, b \in I$  (see Subsection [I.3.3.1](#)). We further define the family

$$n := \left( \bigoplus_{i \in I} n(a,b)_i \cdot \chi^i \right)_{a,b \in I} \in \prod_{a,b \in I} k\text{-vec}.$$

Then there exists an equivalence of categories with bifunctors

$$(F, F_2) : \left( \bigoplus_{i \in I} k\text{-vec}, \otimes_n \right) \xrightarrow{\sim} \text{Rep}_k(G)$$

where  $\otimes_n$  denotes the  $k$ -bilinear bifunctor associated to  $n$  defined in Construction [I.3.12](#). This equivalence of categories with bifunctors has  $F$  as its underlying functor and for  $a, b \in I$ , we further have

$$F_2(\chi^a, \chi^b) = \epsilon_{a,b}^{-1}.$$

**PROOF.** We apply structure transport (see Construction [I.3.35](#)) to the equivalence  $F$  which gives us a  $k$ -bilinear bifunctor  $\otimes$  on  $\bigoplus_{i \in I} k\text{-vec}$  and an equivalence of categories with bifunctors

$$(F, F_2') : \left( \bigoplus_{i \in I} k\text{-vec}, \otimes \right) \xrightarrow{\sim} \text{Rep}_k(G).$$

From this equivalence, it follows that  $\chi^a \otimes \chi^b = \bigoplus_{i \in I} n(a,b)_i \cdot \chi^i$ . Thus, we also have an equivalence of categories with bifunctors

$$(\text{id}, F_2'') : \left( \bigoplus_{i \in I} k\text{-vec}, \otimes_n \right) \xrightarrow{\sim} \left( \bigoplus_{i \in I} k\text{-vec}, \otimes \right)$$

due to Theorem I.3.13, which yields by composition an equivalence of categories with bifunctors

$$(F, F_2''') : \left( \bigoplus_{i \in I} k\text{-vec}, \otimes_n \right) \xrightarrow{\sim} \text{Rep}_k(G).$$

As a last step, we will modify  $F_2'''$  as follows: Any natural isomorphism  $\tau : \otimes_n \xrightarrow{\sim} \otimes_n$  between  $k$ -bilinear bifunctors defines an automorphism  $(\text{id}, \tau)$  of  $(\bigoplus_{i \in I} k\text{-vec}, \otimes_n)$ . Any family of isomorphisms  $\tau_{ab} : \left( \chi^a \otimes_n \chi^b \xrightarrow{\sim} \chi^a \otimes_n \chi^b \right)_{ab}$  uniquely determines such a  $\tau$  by Lemma I.3.10. We compose  $(\text{id}, \tau)$  with  $(F, F_2''')$  with the formula in Remark I.3.6:

$$F(\chi^a) \otimes F(\chi^b) \xrightarrow{F_2'''(\chi^a, \chi^b)} F(\chi^a \otimes_n \chi^b) \xrightarrow{F(\tau_{ab})} F(\chi^a \otimes_n \chi^b)$$

Since  $F$  is an equivalence, it is full and faithful. Thus,  $F(\tau_{ab})$  can be any isomorphism we want, and we choose it such that the above composition yields  $\epsilon_{a,b}^{-1}$ . This is possible since source and range are correct:

$$F(\chi^a) \otimes F(\chi^b) = V^a \otimes V^b$$

and

$$F(\chi^a \otimes_n \chi^b) = F\left(\bigoplus_{i \in I} n(a, b)_i \cdot \chi^i\right) = \bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} F(\chi^i) = \bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} V^i$$

by the strict commutativity of  $F$  with  $\oplus$ . □

**Notation 3.37.** We will abbreviate the tensor product  $\otimes_n$  of Theorem I.3.36 on  $\bigoplus_{i \in I} k\text{-vec}$  simply by  $\otimes$ , if no confusion might occur.

3.3.3. *Defining an Associator.* In [SR72] it is shown that we can use the equivalence

$$(F, F_2) : \left( \bigoplus_{i \in I} k\text{-vec}, \otimes_n \right) \xrightarrow{\sim} \text{Rep}_k(G)$$

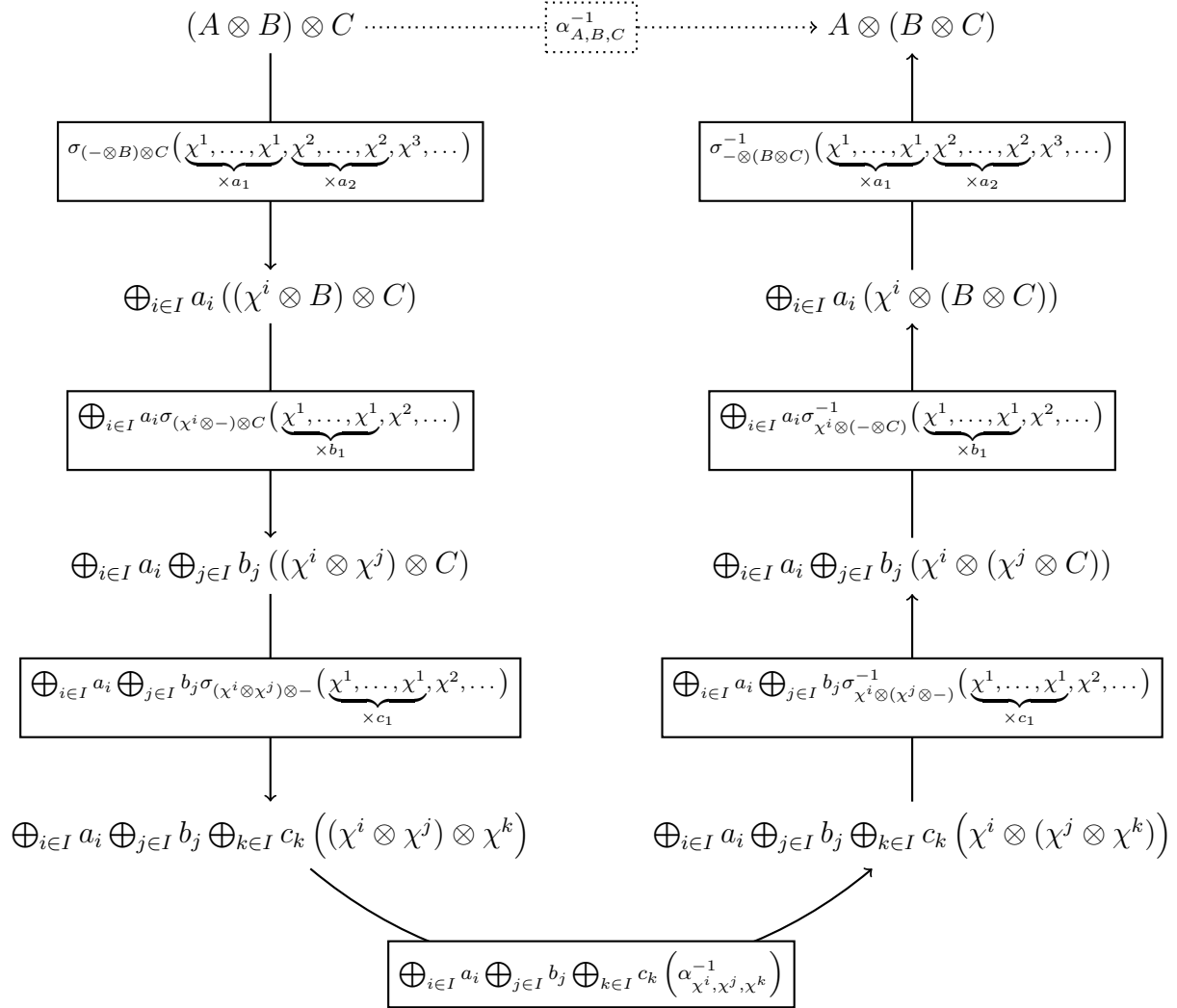
of Theorem I.3.36 to define a uniquely determined associator on  $(\bigoplus_{i \in I} k\text{-vec}, \otimes_n)$  compatible with  $(F, F_2)$ .

**Construction 3.38.** We are going to use  $(F, F_2) : \bigoplus_{i \in I} k\text{-vec} \xrightarrow{\sim} \text{Rep}_k(G)$  for computing an associator on  $\bigoplus_{i \in I} k\text{-vec}$ . Let  $a, b, c \in I$ . We define  $\alpha_{\chi^a, \chi^b, \chi^c}$  as the unique morphism such that  $F(\alpha_{\chi^a, \chi^b, \chi^c})$  satisfies the associator compatibility in Definition I.3.17. Unwrapping the definitions, we end up with the following commutative diagram in  $\text{Rep}_k(G)$ :

$$\begin{array}{ccc}
(V^a \otimes V^b) \otimes V^c & \xrightarrow{=} & V^a \otimes (V^b \otimes V^c) \\
\uparrow & & \downarrow \\
\boxed{F_2(\chi^a, \chi^b) \otimes V^c} & & \boxed{V^a \otimes F_2(\chi^b, \chi^c)^{-1}} \\
\uparrow & & \downarrow \\
(\bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} V^i) \otimes V^c & & V^a \otimes (\bigoplus_{i \in I} \bigoplus_{j=1}^{n(b,c)_i} V^i) \\
\uparrow & & \downarrow \\
\boxed{=} & & \boxed{\sigma_{V^a \otimes -} \left( \left( \bigoplus_{j=1}^{n(b,c)_i} V^i \right)_{i \in I} \right)} \\
\uparrow & & \downarrow \\
\bigoplus_{i \in I} \left( \bigoplus_{j=1}^{n(a,b)_i} V^i \otimes V^c \right) & & \bigoplus_{i \in I} \left( V^a \otimes \left( \bigoplus_{j=1}^{n(b,c)_i} V^i \right) \right) \\
\uparrow & & \downarrow \\
\boxed{=} & & \boxed{\bigoplus_{i \in I} \sigma_{V^a \otimes -} \left( (V^i)_{j=1}^{n(b,c)_i} \right)} \\
\uparrow & & \downarrow \\
\bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} (V^i \otimes V^c) & & \bigoplus_{i \in I} \bigoplus_{j=1}^{n(b,c)_i} (V^a \otimes V^i) \\
\uparrow & & \downarrow \\
\boxed{\bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} F_2(\chi^i, \chi^c)} & & \boxed{\bigoplus_{i \in I} \bigoplus_{j=1}^{n(b,c)_i} F_2(\chi^a, \chi^i)^{-1}} \\
\uparrow & & \downarrow \\
\bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,b)_i} \left( \bigoplus_{k \in I} \bigoplus_{l=1}^{n(i,c)_k} V^k \right) & \xrightarrow{F(\alpha_{\chi^a, \chi^b, \chi^c}^{-1})} & \bigoplus_{i \in I} \bigoplus_{j=1}^{n(b,c)_i} \left( \bigoplus_{k \in I} \bigoplus_{l=1}^{n(a,i)_k} V^k \right)
\end{array}$$

Recall that we denote the natural distributivity isomorphisms by  $\sigma$  (see Lemma I.2.10). The equalities in the diagram above are due to the fact that the associator and right distributivity in  $k\text{-vec}$  and thus in  $\text{Rep}_k(G)$  are given by identity matrices. But since the left distributivity may be non-trivial (see Computation I.3.16), we have to take it into account. In theory, the above approach is not restricted to triples of simple objects. But since the involved matrices may become very big, it is better to compute the associators for an arbitrary triple of objects  $A = \bigoplus_{i \in I} a_i \chi^i$ ,  $B = \bigoplus_{j \in I} b_j \chi^j$ ,  $C = \bigoplus_{k \in I} c_k \chi^k$  using the distributivity laws of  $\otimes$  in  $\bigoplus_{i \in I} k\text{-vec}$ . Since we have already computed the action of  $\otimes$  on morphisms in Theorem I.3.36, we can also compute the natural distributivity isomorphisms.

The corresponding diagram for computing arbitrary associators in  $\bigoplus_{i \in I} k\text{-vec}$  using those on simple objects looks as follows:



Note that it is possible that each of the morphisms involved in this computation is non-trivial (i.e., not equal to the identity).

**REMARK 3.39.** The author knows no good choice of the isomorphisms  $\epsilon_{a,b}$  such that the associator on  $\bigoplus_{i \in I} k\text{-vec}$  gets a particular simple shape. In general, we cannot expect to find a choice such that the associator  $\alpha$  becomes the identity. For example, consider the finite groups  $D_8$  and  $Q_8$ . They have equal character tables, which implies that the categories with bifunctor  $\bigoplus_{i \in \text{Irr}(D_8)} k\text{-vec}$  and  $\bigoplus_{i \in \text{Irr}(Q_8)} k\text{-vec}$  are equal, where the bifunctor is of the form  $\otimes_n$  (see Theorem I.3.36). In Subsection I.3.3.10, we show that the representation categories  $\text{Rep}_k(D_8)$  and  $\text{Rep}_k(Q_8)$  are not equivalent as *braided monoidal categories*. In [EG01], it is furthermore shown that these groups are not isocategorical, which means that their representation categories are not equivalent even as *monoidal categories*, i.e., when we

forget the braiding. In particular, the associator  $\alpha$  cannot be the identity in both categories. Alternatively, a direct computation shows that setting  $\alpha$  to the identity in both cases violates the pentagonal equality in Definition 1.3.14, when we set  $A = B = C = D = \chi$ , where  $\chi$  corresponds to the unique irreducible character of degree 2.

**Example 3.40.** One-dimensional representations in  $\text{Rep}_k(G)$  form a group with  $\otimes$  as multiplication. For  $a, b \in I$  such that  $V^a, V^b$  are one-dimensional, denote by  $ab$  the index such that  $V^{ab} = V^a \otimes V^b$ . Now, we set the isomorphisms

$$\epsilon_{a,b} : V^{ab} \rightarrow V^a \otimes V^b$$

all to the identity matrix. With this particular choice, the Construction 1.3.38 of an associator on  $\bigoplus_{i \in I} k\text{-vec}$  yields the identity for all triples  $\chi^a, \chi^b, \chi^c$ , where  $a, b, c \in I$  correspond to one-dimensional characters. In particular, if  $G$  is abelian, the associator on simple objects can be given by the identity. Note that in the abelian case,  $G$  is determined up to isomorphism by  $\text{Rep}_k(G)$  regarded as a category with bifunctor.

3.3.4. *Defining a Braiding.* In [SR72] it is shown that we can use the equivalence

$$(F, F_2) : \left( \bigoplus_{i \in I} k\text{-vec}, \otimes_n \right) \xrightarrow{\sim} \text{Rep}_k(G)$$

of Theorem 1.3.36 to define a uniquely determined braiding on  $(\bigoplus_{i \in I} k\text{-vec}, \otimes_n)$  compatible with  $(F, F_2)$  and compatible with the associator defined in Subsection 1.3.3.3.

**Construction 3.41.** In this construction we adjust the isomorphisms  $\epsilon_{a,b}$  for simplifying our future computation of the braiding on  $\bigoplus_{i \in I} k\text{-vec}$ . For  $a, b \in I$  such that  $a < b$ , we reset

$$\epsilon_{b,a} := \gamma_{V^a, V^b} \circ \epsilon_{a,b},$$

where  $\gamma_{V^a, V^b}$  is the braiding in  $\text{Rep}_k(G)$ . Furthermore, consider the automorphism

$$\bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,a)_i} V^i \xrightarrow{\epsilon_{a,a}} V^a \otimes V^a \xrightarrow{\gamma_{V^a, V^a}} V^a \otimes V^a \xrightarrow{\epsilon_{a,a}^{-1}} \bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,a)_i} V^i.$$

Since  $\text{Rep}_k(G)$  is a symmetric category, this automorphism is of order 2. If we assume that  $\text{char}(k) \neq 2$ , then it can be conjugated by another automorphism  $\kappa_a : \bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,a)_i} V^i \rightarrow \bigoplus_{i \in I} \bigoplus_{j=1}^{n(a,a)_i} V^i$  such that the result is given by a diagonal matrix having only 1 and  $-1$  on the diagonal (see Definition 1.2.11 for the matrix calculus in additive categories). We reset  $\epsilon_{a,a}$  by  $\epsilon_{a,a} \circ \kappa_a$ . Since all constructions so far worked with arbitrary choices for  $\epsilon_{a,b}$ , this resetting does not violate any of the previous results.

**From now on, we assume that  $\text{char}(k) \neq 2$ .**

**Construction 3.42.** We are going to use  $(F, F_2) : \bigoplus_{i \in I} k\text{-vec} \xrightarrow{\sim} \text{Rep}_k(G)$  for computing a braiding on  $\bigoplus_{i \in I} k\text{-vec}$ . Let  $a, b \in I$ . We define  $\gamma_{\chi^a, \chi^b}$  as the unique morphism such that  $F(\gamma_{\chi^a, \chi^b})$  satisfies the braiding compatibility in Definition 1.3.23. Due to our resetting of  $\epsilon$  in Construction 1.3.41, we end up with

$$\gamma_{\chi^a, \chi^b} = \text{id}$$

for  $a \neq b$ . Again, we use the distributivity laws of  $\otimes$  in  $\bigoplus_{i \in I} k\text{-vec}$  to compute the braiding for an arbitrary pair of objects  $A = \bigoplus_{i \in I} a_i \chi^i$ ,  $B = \bigoplus_{j \in I} b_j \chi^j$ :

$$\begin{array}{ccc}
A \otimes B & \xrightarrow{\gamma_{A,B}} & B \otimes A \\
\downarrow \sigma_{-\otimes B}(\underbrace{\chi^1, \dots, \chi^1}_{\times a_1}, \underbrace{\chi^2, \dots, \chi^2}_{\times a_2}, \chi^3, \dots) & & \downarrow \sigma_{B \otimes -}^{-1}(\underbrace{\chi^1, \dots, \chi^1}_{\times a_1}, \underbrace{\chi^2, \dots, \chi^2}_{\times a_2}, \chi^3, \dots) \\
\bigoplus_{i \in I} a_i (\chi^i \otimes B) & & \bigoplus_{i \in I} a_i (B \otimes \chi^i) \\
\downarrow \bigoplus_{i \in I} a_i \sigma_{\chi^i \otimes -}(\underbrace{\chi^1, \dots, \chi^1}_{\times b_1}, \chi^2, \dots) & & \downarrow \bigoplus_{i \in I} a_i \sigma_{-\otimes \chi^i}^{-1}(\underbrace{\chi^1, \dots, \chi^1}_{\times b_1}, \chi^2, \dots) \\
\bigoplus_{i \in I} a_i \bigoplus_{j \in I} b_j (\chi^i \otimes \chi^j) & \xrightarrow{\bigoplus_{i \in I} a_i \bigoplus_{j \in I} b_j (\gamma_{\chi^i, \chi^j})} & \bigoplus_{i \in I} a_i \bigoplus_{j \in I} b_j (\chi^j \otimes \chi^i)
\end{array}$$

It is possible that each of the morphisms involved in this computation is not equal to the identity.

How does  $\gamma_{\chi^a, \chi^a}$  look like?

**Definition 3.43.** Given an object  $A$  in a tensor category over  $k$ , we define  $\wedge^2 A$  as the cokernel object of the morphism  $\text{id}_{A \otimes A} + \gamma_{A,A}$ .

**Theorem 3.44.** Let  $a, l \in I$ . The  $l$ -th component of  $\gamma_{\chi^a, \chi^a}$  (given by Construction I.3.42) is a diagonal matrix having only 1 and  $-1$  on the diagonal. Furthermore, the number of  $-1$  entries is equal to  $\dim_k(\text{Hom}(\chi^l, \wedge^2 \chi^a))$ .

**PROOF.** Due to our resetting of  $\epsilon_{a,a}$  in Construction I.3.41, the  $l$ -th component of  $\gamma_{\chi^a, \chi^a}$  clearly is a diagonal matrix having only 1 and  $-1$  on the diagonal. Its number of  $-1$  entries equals the number of 0 entries in the  $l$ -th component of  $\text{id}_{\chi^a \otimes \chi^a} + \gamma_{\chi^a, \chi^a}$ , since we assume  $\text{char}(k) \neq 2$ . But the number of 0-entries of the latter can be read off from the cokernel object's dimension.  $\square$

The  $-1$  and 1 entries can be arranged arbitrarily on the diagonal of  $\gamma_{\chi^a, \chi^a}$ . We choose the following convention:

$$\text{Diag}(1, \dots, 1, \underbrace{-1, \dots, -1}_{\times \dim_k(\chi^l, \wedge^2 \chi^a)}).$$

**REMARK 3.45.** The number  $\dim_k(\text{Hom}(\chi^l, \wedge^2 \chi^a))$  can simply be computed using the character table and the 2-power map of  $G$ . Concretely, if  $V$  is a representation of  $G$  with



$k$ -character  $\chi_V$ , then the  $k$ -character of the representation  $\wedge^2 V$  is given by

$$\chi_{\wedge^2 V} = \frac{\chi_V(g)^2 - \chi_V(g^2)}{2}.$$

Thus, for the computation of  $\chi_{\wedge^2 V}$ , all we need to know is the following:

- (1) The values of  $\chi_V$  on conjugacy classes, which is the information provided by the character table.
- (2) The map sending a conjugacy class of an element  $g$  to the conjugacy class of  $g^2$ .

**REMARK 3.46.** The natural distributivity morphisms between  $\otimes$  and  $\oplus$  in  $\bigoplus_{i \in I} k\text{-vec}$  are all given by permutation matrices. This can be seen as follows: From the construction of  $\sigma$  in Lemma 1.2.10 and the definition of  $\otimes$  in  $\bigoplus_{i \in I} k\text{-vec}$  (see Construction 1.3.12), it follows that each component of  $\sigma$  contains exactly one non-zero entry per column, which is given by 1. But since  $\sigma$  is an isomorphism, it has to be a permutation matrix. A direct computation of the corresponding permutation yields a more efficient implementation of  $\sigma$  than using the construction in Lemma 1.2.10.

**3.3.5. Defining Unitors.** There is a uniquely determined index  $u \in I$  such that  $V^u$  is the unit object in  $\text{Rep}_k(G)$ . In this subsection we will see that for  $\chi^u$ , we can define unitors in  $\bigoplus_{i \in I} k\text{-vec}$  compatible with the associator and braiding of Subsections 1.3.3.3 and 1.3.3.4.

**Construction 3.47.** In this construction we adjust once again the isomorphisms  $\epsilon_{a,b}$  for simplifying our computation of the unitors. For  $a \in I$ , we have

$$V^a \otimes V^u = V^a = V^u \otimes V^a$$

since  $\text{Rep}_k(G)$  is a strict monoidal category. Thus, we may reset

$$\begin{aligned} \epsilon_{a,u} &:= \text{id}_{V^a}, \\ \epsilon_{u,a} &:= \text{id}_{V^a}. \end{aligned}$$

Because  $\gamma_{V^a, V^u} = \text{id}_{V^a} = \gamma_{V^u, V^a}$  in  $\text{Rep}_k(G)$ , this resetting is compatible with the resetting in Construction 1.3.41.

In [SR72] it is shown that for a category with associator and braiding, having a unit and unitors which are compatible with the associator and the braiding is a categorical property. Thus, it suffices to find a left unitor  $\lambda$  and a right unitor  $\rho$  for the data  $(\bigoplus_{i \in I} k\text{-vec}, \otimes, \alpha, \gamma, V^u)$  that we have constructed so far.

Since  $\bigoplus_{i \in I} k\text{-vec}$  is skeletal (see Definition 1.3.57), we have

$$A \otimes \chi^u = A = \chi^u \otimes A$$

for all  $A \in \bigoplus_{i \in I} k\text{-vec}$ . Thus, we may simply define

$$\begin{aligned} \rho_A &:= \text{id}_A, \\ \lambda_A &:= \text{id}_A. \end{aligned}$$

It remains to prove that this choice is compatible with the associator and the braiding. We prove the following auxiliary lemma.

**Lemma 3.48.** *Let*

$$(F, F_2) : \bigoplus_{i \in I} k\text{-vec} \xrightarrow{\sim} \text{Rep}_k(G)$$

be the functor of Theorem I.3.36. Then

$$F_2(A, \chi^u) = \text{id}_{FA} = F_2(\chi^u, A)$$

for all objects  $A \in \bigoplus_{i \in I} k\text{-vec}$ .

PROOF. By our resetting in Construction I.3.47, the claim holds for  $A = \chi^i$ , where  $i \in I$ . For arbitrary  $A = \bigoplus_{i \in I} a_i \chi^i$ , the value of  $F_2(\bigoplus_{i \in I} a_i \chi^i, \chi^u)$  is uniquely determined by the values of  $F_2(\chi^i, \chi^u)$  thanks to Lemma I.2.12:

$$\begin{array}{ccc} F(\bigoplus_{i \in I} a_i \chi^i) \otimes F(\chi^u) & \xrightarrow{=} & \bigoplus_{i \in I} \bigoplus_{j=1}^{a_i} (F(\chi^i) \otimes F(\chi^u)) \\ F_2(\bigoplus_{i \in I} a_i \chi^i, \chi^u) \downarrow & & \downarrow (\bigoplus_{i \in I} \bigoplus_{j=1}^{a_i} F_2(\chi^i, \chi^u)) = \text{id} \\ F((\bigoplus_{i \in I} a_i \chi^i) \otimes \chi^u) & \xrightarrow{=} & \bigoplus_{i \in I} \bigoplus_{j=1}^{a_i} F(\chi^i \otimes \chi^u) \end{array}$$

The horizontal morphisms are given by identities since  $F$ ,  $(- \otimes F(\chi^u))$ , and  $(- \otimes \chi^u)$  strictly commute with direct sums. Analogously,  $F_2(\chi^u, A) = \text{id}_{FA}$ .  $\square$

**Lemma 3.49.** *The unitors  $\rho_A$  and  $\lambda_A$  for  $A \in \bigoplus_{i \in I} k\text{-vec}$  are compatible with the associator.*

PROOF. Compatibility with the associator  $\alpha$  means that the diagram in Definition I.3.14 involving  $\lambda, \rho$  and  $\alpha$  commutes, which in our case is equivalent to the assertion

$$\alpha_{A, \chi^u, C} = \text{id}_{A \otimes C}$$

for  $A, C \in \bigoplus_{i \in I} k\text{-vec}$ . Now,  $\alpha_{A, \chi^u, C}$  was defined via structure transport, i.e., as the uniquely determined morphism such that

$$\begin{array}{ccc} F(A) \otimes (F(\chi^u) \otimes F(C)) & \xrightarrow{\text{id}} & (F(A) \otimes F(\chi^u)) \otimes F(C) \\ \text{id} = (F(A) \otimes F_2(\chi^u, C)) \downarrow & & \text{id} = (F_2(A, \chi^u) \otimes F(C)) \downarrow \\ F(A) \otimes F(\chi^u \otimes C) & & F(A \otimes \chi^u) \otimes F(C) \\ F_2(A, C) = F_2(A, \chi^u \otimes C) \downarrow & & F_2(A, C) = F_2(A \otimes \chi^u, C) \downarrow \\ F(A \otimes (\chi^u \otimes C)) & \xrightarrow{F(\alpha_{A, \chi^u, C})} & F((A \otimes \chi^u) \otimes C) \end{array}$$

commutes. The two vertical identities are due to Lemma I.3.48. From this diagram, we can deduce  $\alpha_{A, \chi^u, C} = \text{id}_{A \otimes C}$ .  $\square$

**Lemma 3.50.** *The unitors  $\rho_A$  and  $\lambda_A$  for  $A \in \bigoplus_{i \in I} k\text{-vec}$  are compatible with the braiding.*

PROOF. Compatibility with the braiding  $\gamma$  means that the diagram in Definition I.3.21 involving  $\lambda, \rho$  and  $\gamma$  commutes, which in our case is equivalent to the assertion

$$\gamma_{A, \chi^u} = \text{id}_A$$

for  $A \in \bigoplus_{i \in I} k\text{-vec}$ . Now,  $\gamma_{A, \chi^u}$  was defined via structure transport, i.e., as the uniquely determined morphism such that

$$\begin{array}{ccc} FA \otimes F\chi^u & \xrightarrow{\text{id}} & F\chi^u \otimes FA \\ \text{id} = F_2(A, \chi^u) \downarrow & & \downarrow \text{id} = F_2(\chi^u, A) \\ F(A \otimes \chi^u) & \xrightarrow{F(\gamma_{A, \chi^u})} & F(\chi^u \otimes A). \end{array}$$

commutes. The two vertical identities are due to Lemma I.3.48. From this diagram, we can deduce  $\gamma_{A, \chi^u} = \text{id}_A$ .  $\square$

3.3.6. *Defining Duals.* Having dual objects is a property of a symmetric monoidal category. This means that all ways in which a category can be equipped with exact pairings are equivalent [Sel11]. Since we have established a monoidal equivalence between  $\bigoplus_{i \in I} k\text{-vec}$  and  $\text{Rep}_k(G)$ , the former inherits this property from the latter.

Because  $\bigoplus_{i \in I} k\text{-vec}$  is skeletal, the dual object  $A^\vee$  associated to  $A \in \bigoplus_{i \in I} k\text{-vec}$  is uniquely determined. Furthermore, Lemma I.2.22 gives us a chain of equivalences of functor categories

$$\text{Hom}_k \left( \left( \bigoplus_{i \in I} k\text{-vec} \right)^{\text{op}}, \bigoplus_{j \in I} k\text{-vec} \right) \simeq \text{Hom}_k \left( \left( \bigoplus_{i \in I} k\text{-vec} \right), \bigoplus_{j \in I} k\text{-vec} \right) \simeq \prod_{i \in I} \bigoplus_{j \in I} k\text{-vec},$$

which proves that the dualization functor is uniquely determined up to natural isomorphism by its action on objects. A good instance of such a dualization functor is given by the contravariant functor which acts on objects as dictated by the assignment  $A \mapsto A^\vee$ , and on morphisms by transposing matrices. We denote this functor by

$$(-)^* : \left( \bigoplus_{i \in I} k\text{-vec} \right)^{\text{op}} \rightarrow \bigoplus_{i \in I} k\text{-vec}.$$

It is easy to see that  $(-)^*$  strictly commutes with direct sums, which is a particularly convenient computational feature.

Our goal is to describe exact pairings (see Definition I.3.28) of the objects in  $\bigoplus_{i \in I} k\text{-vec}$  such that the induced dualization functor  $(-)^\vee$  (see Remark I.3.29) equals  $(-)^*$ . We will use the following lemma to achieve this goal.

**Lemma 3.51.** *Let  $\mathbf{A}$  be a rigid symmetric monoidal category. Denote its dualization functor by  $(-)^{\vee}$  (see Remark I.3.29). Let  $(-)^* : \mathbf{A}^{\text{op}} \rightarrow \mathbf{A}$  be another functor coinciding with  $(-)^{\vee}$  on objects, such that for all morphisms  $\alpha : A \rightarrow B$ , the diagram*

$$\begin{array}{ccc} A \otimes B^* & \xrightarrow{\alpha \otimes B^*} & B \otimes B^* \\ \downarrow A \otimes \alpha^* & & \downarrow \epsilon_B \\ A \otimes A^* & \xrightarrow{\epsilon_A} & 1 \end{array}$$

*commutes. Then  $(-)^*$  coincides with  $(-)^{\vee}$  on morphisms.*

PROOF. We calculate

$$\begin{aligned} \alpha^{\vee} &= (A^{\vee} \otimes \epsilon_B) \circ (A^{\vee} \otimes \alpha \otimes B^{\vee}) \circ (\eta_A \otimes B^{\vee}) \\ &= (A^{\vee} \otimes (\epsilon_B \circ (\alpha \otimes B^{\vee}))) \circ (\eta_A \otimes B^{\vee}) \\ &= (A^{\vee} \otimes (\epsilon_A \circ (A \otimes \alpha^*))) \circ (\eta_A \otimes B^{\vee}) \\ &= (A^{\vee} \otimes \epsilon_A) \circ (A^{\vee} \otimes A \otimes \alpha^*) \circ (\eta_A \otimes B^{\vee}) \\ &= (A^{\vee} \otimes \epsilon_A) \circ (\eta_A \otimes \alpha^*) = \alpha^*, \end{aligned}$$

where the last equation holds in every rigid symmetric monoidal category, as can simply be seen from the corresponding equation of string diagrams (see Remark I.3.24):

□

REMARK 3.52. The diagram in Lemma I.3.51 is motivated by the diagram coming from the natural dependent function  $A \mapsto \epsilon_A$  of type  $\prod_{A \in \mathbf{A}} \text{Hom}_{\mathbf{A}}(A \otimes A^{\vee}, 1)$  (see Example I.1.18).

**Construction 3.53.** We construct exact pairings for all simple objects. So let  $i \in I$ . Since

$$\text{Hom}(1, (\chi^i)^{\vee} \otimes \chi^i) \simeq \text{Hom}(\chi^i, \chi^i) = k \cdot \text{id}_{\chi^i}$$

and

$$\text{Hom}(\chi^i \otimes (\chi^i)^{\vee}, 1) \simeq \text{Hom}(\chi^i, \chi^i) = k \cdot \text{id}_{\chi^i},$$

$\eta_{\chi^i}$  and  $\epsilon_{\chi^i}$  are both given by a scalar in  $k$ . If  $(\eta, \epsilon)$  is an exact pairing, then so is  $(a\eta, \frac{1}{a}\epsilon)$  for every non-zero scalar  $a \in k$ . Thus we simply set  $\eta_{\chi^i}$  to be given by the scalar 1. The scalar of  $\epsilon_{\chi^i}$  now is uniquely determined by the commutativity of the diagrams in Definition I.3.28.

Using the following lemma, we get exact pairings for arbitrary objects in  $\bigoplus_{i \in I} k\text{-vec}$ .

**Lemma 3.54.** *Let  $\mathbf{A}$  be a symmetric monoidal category. Let  $(-)^{\vee} : \mathbf{A}^{\text{op}} \rightarrow \mathbf{A}$  be an equivalence. If  $(A^{\vee}, \eta_A, \epsilon_A)$  and  $(B^{\vee}, \eta_B, \epsilon_B)$  are exact pairings for  $A, B$ , respectively, then so is  $((A \oplus B)^{\vee}, \eta_{A \oplus B}, \epsilon_{A \oplus B})$  for  $A \oplus B$ , where  $\eta_{A \oplus B}$  is induced by the row*

$$1 \xrightarrow{\begin{pmatrix} \eta_A & \cdot & \cdot & \eta_B \end{pmatrix}} (A^{\vee} \otimes A) \oplus (A^{\vee} \otimes B) \oplus (B^{\vee} \otimes A) \oplus (B^{\vee} \otimes B)$$

and  $\epsilon_{A \oplus B}$  is induced by the column

$$(A \otimes A^{\vee}) \oplus (A \otimes B^{\vee}) \oplus (B \otimes A^{\vee}) \oplus (B \otimes B^{\vee}) \xrightarrow{\begin{pmatrix} \epsilon_A \\ \cdot \\ \cdot \\ \epsilon_B \end{pmatrix}} 1.$$

**PROOF.** The verification of the zig-zag identities boils down to a simple calculation of matrices with entries given by homomorphisms in  $\mathbf{A}$ :

$$\left( \begin{pmatrix} \epsilon_A \\ \cdot \\ \cdot \\ \epsilon_B \end{pmatrix} \otimes \begin{pmatrix} \text{id}_A & \cdot \\ \cdot & \text{id}_B \end{pmatrix} \right) \circ \left( \begin{pmatrix} \text{id}_A & \cdot \\ \cdot & \text{id}_B \end{pmatrix} \otimes \begin{pmatrix} \eta_A & \cdot & \cdot & \eta_B \end{pmatrix} \right) = \text{id}_{A \oplus B}$$

and

$$\left( \begin{pmatrix} \text{id}_{A^{\vee}} & \cdot \\ \cdot & \text{id}_{B^{\vee}} \end{pmatrix} \otimes \begin{pmatrix} \epsilon_A \\ \cdot \\ \cdot \\ \epsilon_B \end{pmatrix} \right) \circ \left( \begin{pmatrix} \eta_A & \cdot & \cdot & \eta_B \end{pmatrix} \otimes \begin{pmatrix} \text{id}_{A^{\vee}} & \cdot \\ \cdot & \text{id}_{B^{\vee}} \end{pmatrix} \right) = \text{id}_{A^{\vee} \oplus B^{\vee}}. \quad \square$$

**Construction 3.55.** Lemma 1.3.54 justifies the following construction of  $\eta_A$  and  $\epsilon_A$  for an arbitrary object  $A = \bigoplus_{i \in I} a_i \chi^i \in \bigoplus_{i \in I} k\text{-vec}$ . Here, Row and Column will denote functions taking a matrix as an input and constructing one single row/column by concatenating the rows/columns of the given input:

$$\begin{array}{c}
1 \longrightarrow \boxed{\text{Row} \left( \text{Diag}(\underbrace{\eta_{\chi^1}, \dots, \eta_{\chi^1}}_{\times a_1}, \underbrace{\eta_{\chi^2}, \dots, \eta_{\chi^2}}_{\times a_2}, \eta_{\chi^3}, \dots) \right)} \longrightarrow \bigoplus_{i \in I} a_i \bigoplus_{j \in I} a_j ((\chi^i)^\vee \otimes \chi^j) \\
\vdots \downarrow \text{---} \boxed{\eta_A} \text{---} \downarrow \vdots \\
A^\vee \otimes A \longleftarrow \boxed{=} \longleftarrow \begin{array}{c} \bigoplus_{i \in I} a_i \sigma_{(\chi^i)^\vee \otimes -}^{-1}(\underbrace{\chi^1, \dots, \chi^1}_{\times a_1}, \chi^2, \dots) \\ \downarrow \\ \bigoplus_{i \in I} a_i ((\chi^i)^\vee \otimes A) \\ \downarrow \\ \sigma_{-\otimes A}^{-1}(\underbrace{(\chi^1)^\vee, \dots, (\chi^1)^\vee}_{\times a_1}, (\chi^2)^\vee, \dots) \\ \downarrow \\ (\bigoplus_{i \in I} a_i (\chi^i)^\vee) \otimes A \end{array}
\end{array}$$

is the diagram for computing  $\eta_A$  and

$$\begin{array}{c}
1 \longleftarrow \boxed{\text{Column} \left( \text{Diag}(\underbrace{\epsilon_{\chi^1}, \dots, \epsilon_{\chi^1}}_{\times a_1}, \underbrace{\epsilon_{\chi^2}, \dots, \epsilon_{\chi^2}}_{\times a_2}, \epsilon_{\chi^3}, \dots) \right)} \longleftarrow \bigoplus_{i \in I} a_i \bigoplus_{j \in I} a_j (\chi^j \otimes (\chi^i)^\vee) \\
\vdots \uparrow \text{---} \boxed{\epsilon_A} \text{---} \uparrow \vdots \\
A \otimes A^\vee \longrightarrow \boxed{=} \longrightarrow \begin{array}{c} \bigoplus_{i \in I} a_i \sigma_{-\otimes (\chi^i)^\vee}(\underbrace{\chi^1, \dots, \chi^1}_{\times a_1}, \chi^2, \dots) \\ \uparrow \\ \bigoplus_{i \in I} a_i (A \otimes (\chi^i)^\vee) \\ \uparrow \\ \sigma_{A \otimes -}(\underbrace{(\chi^1)^\vee, \dots, (\chi^1)^\vee}_{\times a_1}, (\chi^2)^\vee, \dots) \\ \uparrow \\ A \otimes (\bigoplus_{i \in I} a_i (\chi^i)^\vee) \end{array}
\end{array}$$

is the diagram for computing  $\epsilon_A$ . The equalities are due to the fact that  $(-)^*$  strictly commutes with direct sums.

**Theorem 3.56.** *The exact pairings of constructions 1.3.53 and 1.3.55 give rise to a functor  $(-)^{\vee}$  as described in Remark 1.3.29 that acts on morphisms by transposing matrices.*

**PROOF.** We are going to check the commutativity condition of Lemma 1.3.51. By definition,  $(-)^{\vee}$  and  $(-)^*$  coincide on objects. If  $A, B$  are simple objects, then  $\alpha$  has to be 0 if  $A \neq B$ , and is given by a scalar multiplication if  $A = B$ . Since transposing a scalar does not change the scalar, the commutativity condition holds. Now, take direct sums

$A = A_1 \oplus A_2$ ,  $B = B_1 \oplus B_2$ , and a morphism

$$\alpha = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} : A \rightarrow B.$$

Since we chose  $\epsilon_A$  and  $\epsilon_B$  to be compatible with the natural commutativity morphisms of  $(-)^*$  with direct sums, the commutativity condition of Lemma I.3.51 is equivalent to an equation between matrices with entries given by homomorphisms in  $\bigoplus_{i \in I} k\text{-vec}$ :

$$\begin{pmatrix} \epsilon_{A_1} \\ \cdot \\ \cdot \\ \epsilon_{A_2} \end{pmatrix} \circ \left( \begin{pmatrix} \text{id}_{A_1} & \cdot \\ \cdot & \text{id}_{A_2} \end{pmatrix} \otimes \begin{pmatrix} \alpha_{11}^* & \alpha_{21}^* \\ \alpha_{12}^* & \alpha_{22}^* \end{pmatrix} \right) = \begin{pmatrix} \epsilon_{B_1} \\ \cdot \\ \cdot \\ \epsilon_{B_2} \end{pmatrix} \circ \left( \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \otimes \begin{pmatrix} \text{id}_{B_1} & \cdot \\ \cdot & \text{id}_{B_2} \end{pmatrix} \right).$$

But this holds by induction on the number of irreducible summands of  $A$  and  $B$ .  $\square$

**3.3.7. Skeletal Representation Category of Finite Groups.** We give a summary of our construction of  $\text{SRep}_k(G)$ , where the prefix ‘‘S’’ stands for skeletal.

**Definition 3.57.** A category  $\mathbf{C}$  is **skeletal** if  $A \simeq A'$  implies  $A = A'$  for all objects  $A, A' \in \mathbf{C}$ .

Let  $G$  be a finite group. Let  $k$  be a field such that

- (1)  $\text{char}(k) \nmid |G|$ ,
- (2)  $\text{char}(k) \neq 2$ ,
- (3) any irreducible representation of  $G$  over  $k$  is already absolutely irreducible.

Let  $I := \{1, \dots, |\text{Irr}(G)|\}$  be a set which is in bijection to irreducible characters of  $G$ . Then we can define on the category  $\bigoplus_{i \in I} k\text{-vec}$

- (1) a bifunctor  $\otimes$  (Subsection I.3.3.2),
- (2) an associator  $\alpha$  (Subsection I.3.3.3),
- (3) a braiding  $\gamma$  (Subsection I.3.3.4),
- (4) unitors  $\lambda, \rho$  (Subsection I.3.3.5),
- (5) duals (Subsection I.3.3.6),

such that it becomes a tensor category which we denote by  $\text{SRep}_k(G)$ . By construction, we have an equivalence

$$\text{SRep}_k(G) \simeq \text{Rep}_k(G)$$

of tensor categories. Since we constructed  $k\text{-vec}$  as a skeletal category (see Section I.2),  $\text{SRep}_k(G)$  is also skeletal.

**3.3.8. Graded Group Representations.** Studying the functor category

$$\text{Rep}_k^{\mathbb{Z}}(G) := \text{Hom}(\mathbf{B}G, \bigoplus_{d \in \mathbb{Z}} k\text{-vec})$$

means studying  **$\mathbb{Z}$ -graded representations of  $G$** . From the skeletal model  $\text{SRep}_k(G)$  of group representations, we can easily derive a skeletal model  $\text{SRep}_k^{\mathbb{Z}}(G)$  of  $\mathbb{Z}$ -graded group representations: Let  $I := \{1, \dots, |\text{Irr}(G)|\}$ .

- (1)  $\text{SRep}_k^{\mathbb{Z}}(G)$  has  $\bigoplus_{I \times \mathbb{Z}} k\text{-vec}$  as its underlying abelian category.

- (2) The tensor product of  $\chi^{i,d}$  and  $\chi^{j,d'}$  is defined as the object  $\chi^i \otimes_n \chi^j \in \text{SRep}_k(G)$  decorated with degree  $d + d'$ .
- (3) Let  $\chi^u$  denote the unit in  $\text{SRep}_k(G)$ . Then the unit in  $\text{SRep}_k^{\mathbb{Z}}(G)$  is given by  $\chi^{u,0}$ .
- (4) For  $\chi^i \in \text{SRep}_k(G)$ , let  $\chi^{i'}$  denote its dual object. Then the dual object of  $\chi^{i,d}$  in  $\text{SRep}_k^{\mathbb{Z}}(G)$  is given by  $\chi^{i',-d}$ .
- (5) The functor

$$\text{SRep}_k^{\mathbb{Z}}(G) \rightarrow \text{SRep}_k(G) : \chi^{i,d} \mapsto \chi^i$$

forgetting degrees uniquely determines the remaining structure morphisms of a tensor category.

The tensor category  $\text{SRep}_k^{\mathbb{Z}}(G)$  will be used in Section III.3 in the modeling of  $G$ -equivariant  $\mathbb{Z}$ -graded modules over the exterior algebra.

Another way to create a skeletal model for  $\text{Rep}_k^{\mathbb{Z}}(G)$  is to apply structure transport to an equivalence of abelian categories

$$F : \bigoplus_{I \times \mathbb{Z}} k\text{-vec} \xrightarrow{\sim} \text{Rep}_k^{\mathbb{Z}}(G)$$

just as we did in the case of  $\text{Rep}_k(G)$ . Structure transport is a universal tool which can be also applied to define a tensor category structure on  $\bigoplus_{i \in I} k\text{-vec}$  equivalent to the representation category of a given Lie-algebra  $\mathfrak{g}$ .

3.3.9. *Example:  $S_3$ .* In this example we are going to construct a computational model for  $\text{Rep}_k(S_3)$ , the tensor category of representations over  $k := \mathbb{Q}$  of the symmetric group  $S_3$ , using the methods of this section. As abelian categories, we have an equivalence

$$F : k\text{-vec} \oplus k\text{-vec} \oplus k\text{-vec} \xrightarrow{\sim} \text{Rep}_k(S_3)$$

where each summand corresponds to an irreducible character of  $S_3$  and thus to a simple object in  $\text{Rep}_k(S_3)$ . We start with the choice of an irreducible representation  $V^1, V^2, V^3$  for each irreducible character. Such a choice makes the functor  $F$  explicit, as described in Subsection I.3.3.1.

**Computation 3.58.** For the computation of  $V^1, V^2, V^3$ , we use the CAP package `GroupRepresentationsForCAP`, which relies on the GAP implementation of a method by Dixon [Dix93] for making an automatic choice of irreducible representations for given irreducible characters. Of course, for such a small example, we could have chosen the representations manually, but having such an automatic process is extremely handy for examples with more and higher-dimensional representations (although not too high-dimensional).

GAP provides the following character table of  $S_3 = \langle (1, 2), (1, 2, 3) \rangle$ :

Character	(1)	(1, 2)	(1, 2, 3)
$\psi^1$	1	-1	1
$\psi^2$	2	0	-1
$\psi^3$	1	1	1

Applying Dixon's method yields the following representations:



Character	Representation	(1, 2)	(1, 2, 3)
$\psi^1$	$V^1$	$\begin{pmatrix} -1 \end{pmatrix}$	$\begin{pmatrix} 1 \end{pmatrix}$
$\psi^2$	$V^2$	$\begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix}$
$\psi^3$	$V^3$	$\begin{pmatrix} 1 \end{pmatrix}$	$\begin{pmatrix} 1 \end{pmatrix}$

Note that we can make such a choice for  $V^1, V^2, V^3$  once and for all. CAP stores these representations and can make use of them in future sessions. Thus, the following code (yielding the above results) only has to be performed once, which is why it can make use of argumentless global functions.

```
gap> G := SymmetricGroup( 3 );
Sym( [ 1 .. 3 ] )
gap> InitializeGroupDataDixon( G );;
gap> DisplayInitializedGroupData();
-Representations of:
  SymmetricGroup( [ 1 .. 3 ] )
-Defined over the rationals
-Given by images of the following generators:
  [ (1,2,3), (1,2) ]
-Affording the irreducible characters:
  CT1

      2 1 1 .
      3 1 . 1

      1a 2a 3a
  2P 1a 1a 3a
  3P 1a 2a 1a

X.1    1 -1 1
X.2    2  . -1
X.3    1  1 1

-----
Representation affording character X.1:
(1,2,3)->
1
(1,2)->
-1
```

Representation affording character X.2:

(1,2,3)->

0, 1,

-1,-1

(1,2)->

1,1,

0,-1

Representation affording character X.3:

(1,2,3)->

1

(1,2)->

1

As a next step, we choose decomposition isomorphisms

$$\epsilon_{a,b} : \bigoplus_{i \in \{1,2,3\}} \bigoplus_{j=1}^{n(a,b)_i} V^i \xrightarrow{\sim} V^a \otimes V^b$$

for  $a, b = 1, 2, 3$ .

**Computation 3.59.** From the following multiplication table of irreducible characters, we can read off the decomposition numbers  $n(a, b)_i$ :

$\otimes$	$\psi^1$	$\psi^2$	$\psi^3$
$\psi^1$	$\psi^3$	$\psi^2$	$\psi^1$
$\psi^2$	$\psi^2$	$\psi^1 + \psi^2 + \psi^3$	$\psi^2$
$\psi^3$	$\psi^1$	$\psi^2$	$\psi^3$

In the following table, we list the underlying matrices of some possible choices for the isomorphisms  $\epsilon_{a,b}$ :

$a \setminus b$	1	2	3
1	(1)	$\begin{pmatrix} -1 & -2 \\ 2 & 1 \end{pmatrix}$	(1)
2	$\begin{pmatrix} -1 & -2 \\ 2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 \end{pmatrix}$	$I_2$
3	(1)	$I_2$	(1)

The following two commands of the CAP package `GroupRepresentationsForCAP` compute and display these isomorphisms  $\epsilon_{a,b}$  in the context of our current session. Note again that we can make such a choice once and for all.

```
gap> SkeletalFunctorTensorData();
gap> DisplaySkeletalFunctorTensorData();
-----

1*(X.3) -> (X.1)*(X.1):

1
-----

1*(X.2) -> (X.1)*(X.2):

-1,-2,
2, 1
-----

1*(X.1) -> (X.1)*(X.3):

1
-----

1*(X.2) -> (X.2)*(X.1):

-1,-2,
2, 1
-----

1*(X.1) + 1*(X.2) + 1*(X.3) -> (X.2)*(X.2):

0, -1,1, 0,
-1,-1,-1,0,
0, 1, 1, 1,
2, 1, 1, 2
-----
```

```

1*(X.2) -> (X.2)*(X.3):
1,0,
0,1
-----

1*(X.1) -> (X.3)*(X.1):
1
-----

1*(X.2) -> (X.3)*(X.2):
1,0,
0,1
-----

1*(X.3) -> (X.3)*(X.3):
1

```

With these isomorphisms, we are able to transport the tensor product and the associator from  $\text{Rep}_k(S_3)$  to  $k\text{-vec} \oplus k\text{-vec} \oplus k\text{-vec}$ , as described in Theorem [I.3.36](#) and Construction [I.3.38](#).

**Computation 3.60.** We are now able to work in our skeletal model  $\text{SRep}_k(S_3)$ . In the following session (based on the CAP package `GroupRepresentationsForCAP`), we first define  $\text{SRep}_k(S_3)$ .

```
Defining SRep_k(S_3)
```

```
gap> SRepG := RepresentationCategory( G );
The representation category of SymmetricGroup( [ 1 .. 3 ] )
```

For constructing objects in  $\text{SRep}_k(S_3)$ , we need to define the set of irreducible characters of  $S_3$ .

```
Defining Irr(S_3)
```

```
gap> irr := Irr( G );;
```

The simple objects  $\chi^1, \chi^2, \chi^3$  in  $\text{SRep}_k(S_3)$  correspond to the irreducible characters  $\psi^1, \psi^2, \psi^3$  of  $S_3$ . An arbitrary object in  $\text{SRep}_k(S_3)$  is given by a formal  $\mathbb{N}_0$ -linear combination of

simple objects. With the following commands, we define all simple objects in our current session.

Defining the simple object  $v_1 := 1 \cdot \chi^1$

```
gap> v1 := RepresentationCategoryObject( irr[1], SRepG );
1*(x_1)
```

Defining the simple object  $v_2 := 1 \cdot \chi^2$

```
gap> v2 := RepresentationCategoryObject( irr[2], SRepG );
1*(x_2)
```

Defining the simple object  $v_3 := 1 \cdot \chi^3$

```
gap> v3 := RepresentationCategoryObject( irr[3], SRepG );
1*(x_3)
```

We check that  $v_3$  is the tensor unit in  $\text{SRep}_k(G)$ .

```
gap> TensorUnit( SRepG );
1*(x_3)
```

Note that in this session the trivial character was labeled by the number 3.

Now, we enlist the associators of the form

$$(v_a \otimes v_b) \otimes v_c \xrightarrow{\sim} v_a \otimes (v_b \otimes v_c).$$

We omit the following special cases:

- (1) One of the objects is the tensor unit.
- (2) All objects are 1-dimensional.

In both cases, the associator is given by the identity matrix. In the first case, the associator can be built from the unitors, which are given by identity matrices. For the second case, see Example [I.3.40](#).

For reading the following table enlisting the associators, note that a morphism in  $\bigoplus_{i \in \{1,2,3\}} k\text{-vec}$  consists of three components.

Component:	$\chi^1$	$\chi^2$	$\chi^3$
$\alpha_{v_2, v_1, v_1}$	-	$(-3)$	-
$\alpha_{v_1, v_2, v_1}$	-	$(1)$	-
$\alpha_{v_1, v_1, v_2}$	-	$(-\frac{1}{3})$	-
$\alpha_{v_2, v_2, v_1}$	$(-1)$	$(-1)$	$(\frac{1}{3})$
$\alpha_{v_2, v_1, v_2}$	$(-1)$	$(1)$	$(-1)$
$\alpha_{v_1, v_2, v_2}$	$(1)$	$(-1)$	$(-3)$
$\alpha_{v_2, v_2, v_2}$	$(-1)$	$\begin{pmatrix} -\frac{1}{2} & -1 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{1}{2} & 1 & \frac{1}{2} \end{pmatrix}$	$(1)$

```
gap> Display( AssociatorLeftToRight( v2, v1, v1 ) );
Component: (x_2)
```

-3

A morphism in Category of matrices over Q

```
gap> Display( AssociatorLeftToRight( v1, v2, v1 ) );
Component: (x_2)
```

1

A morphism in Category of matrices over Q

```
gap> Display( AssociatorLeftToRight( v1, v1, v2 ) );
Component: (x_2)
```

-1/3

A morphism in Category of matrices over Q

```
gap> Display( AssociatorLeftToRight( v2, v2, v1 ) );
Component: (x_1)
```

-1

A morphism in Category of matrices over Q

```
-----
Component: (x_2)
```

```
-1
```

```
A morphism in Category of matrices over Q
```

```
-----
Component: (x_3)
```

```
1/3
```

```
A morphism in Category of matrices over Q
```

```
-----
gap> Display( AssociatorLeftToRight( v2, v1, v2 ) );
```

```
Component: (x_1)
```

```
-1
```

```
A morphism in Category of matrices over Q
```

```
-----
Component: (x_2)
```

```
1
```

```
A morphism in Category of matrices over Q
```

```
-----
Component: (x_3)
```

```
-1
```

```
A morphism in Category of matrices over Q
```

```
-----
gap> Display( AssociatorLeftToRight( v1, v2, v2 ) );
```

```
Component: (x_1)
```

```
1
```

```
A morphism in Category of matrices over Q
```

```
-----
Component: (x_2)
```

```
-1
```

```

A morphism in Category of matrices over Q
-----
Component: (x_3)

-3

A morphism in Category of matrices over Q
-----
gap> Display( AssociatorLeftToRight( v2, v2, v2 ) );
Component: (x_1)

-1

A morphism in Category of matrices over Q
-----
Component: (x_2)

-1/2,-1,1/2,
1/2, 0, 1/2,
-1/2,1, 1/2

A morphism in Category of matrices over Q
-----
Component: (x_3)

1

A morphism in Category of matrices over Q
-----

```

3.3.10. *Example:  $D_8$  and  $Q_8$ .* The dihedral group  $D_8$  of order 8 and the quaternion group  $Q_8$  have equal character tables, but nonequivalent representation categories regarded as tensor categories. Using CAP, we can demonstrate this fact in a simple way.

**Computation 3.61.** Let  $k = \mathbb{Q}(i)$ . In  $\text{Rep}_k(Q_8)$  and in  $\text{Rep}_k(D_8)$  the irreducible representation  $v$  of dimension 2 is categorically characterized up to isomorphism by being simple but not being invertible (meaning that the unit  $\eta_v$  is not an isomorphism). Our goal is to compute the object  $\wedge^2 v$  (see Definition 1.3.43) in  $\text{Rep}_k(Q_8)$  and  $\text{Rep}_k(D_8)$ .

We start with the construction in  $\text{SRep}_k(D_8)$ .



Defining  $\text{SRep}_k(D_8)$

```
gap> SRepG := RepresentationCategory( 8, 3 );
The representation category of Group( [ f1, f2, f3 ] )
```

The pair  $(8, 3)$  is the identification number of  $D_8$  in GAP's `SmallGroups` library.

```
gap> G := UnderlyingGroupForRepresentationCategory( SRepG );
<pc group of size 8 with 3 generators>
gap> StructureDescription( G );
"D8"
```

We need access to irreducible characters for defining objects in  $\text{SRep}_k(D_8)$ .

Defining  $\text{Irr}(D_8)$

```
gap> irr := Irr( G );;
```

Now, we can construct  $v$  from the 5-th irreducible character in  $\text{Irr}(D_8)$ .

Defining  $v := 1 \cdot \chi^5$

```
gap> v := RepresentationCategoryObject( irr[5], SRepG );
1*(x_5)
```

We check that this is really the simple object of dimension 2.

```
gap> Dimension( v );
2
```

Now, we can turn to the construction of  $\wedge^2 v = \text{coker}(\text{id}_v + \gamma_{v,v})$ .

Defining  $\alpha := \text{id}_v + \gamma_{v,v}$

```
gap> alpha := IdentityMorphism( TensorProductOnObjects( v, v ) )
> + Braiding( v, v );
<A morphism in The representation category of Group( [ f1, f2, f3 ] )>
```

Computing  $\text{coker}(\alpha) = 1 \cdot \chi^4$

```
gap> CokernelObject( alpha );
1*(x_4)
```

We want to compare this object to the tensor unit.

```
gap> TensorUnit( SRepG );
1*(x_1)
```

Since  $\chi^1 \neq \chi^4$ , we see that  $\wedge^2 v$  is *not* isomorphic to the tensor unit.

When we repeat the same sequence of commands in the case of  $Q_8$ , it follows that  $\wedge^2 v$  is isomorphic to the tensor unit. Thus,  $\text{SRep}_k(Q_8) \cong \text{SRep}_k(D_8)$  as tensor categories.

```

gap> SRepG := RepresentationCategory( 8, 4 );
The representation category of Group( [ f1, f2, f3 ] )
gap> G := UnderlyingGroupForRepresentationCategory( SRepG );
<pc group of size 8 with 3 generators>
gap> StructureDescription( G );
"Q8"
gap> irr := Irr( G );;
gap> v := RepresentationCategoryObject( irr[5], SRepG );
1*(x_5)
gap> Dimension( v );
2
gap> alpha := IdentityMorphism( TensorProductOnObjects( v, v ) )
> + Braiding( v, v );
<A morphism in The representation category of Group( [ f1, f2, f3 ] )>
gap> CokernelObject( alpha );
1*(x_1)
gap> TensorUnit( SRepG );
1*(x_1)

```

3.3.11. *Example: Subgroup of Order 1000 of the Automorphism Group of the Horrocks-Mumford Bundle.* In [HM73] Horrocks and Mumford constructed a  $G$ -equivariant rank 2 vector bundle on the complex projective space of dimension 4 with  $G \simeq H_5 \rtimes \mathrm{SL}_2(5)$ . Here,  $H_5$  is the Heisenberg group of order  $5^3$ . In particular, the order of  $G$  is  $15000 = 2^3 \cdot 3 \cdot 5^4$ . Using GAP, we can see that this group has 50 irreducible characters, and 4 of them are of degree 30. If we tried Construction I.3.38 for computing associators, we would have to deal with matrices of dimension  $30^3 \times 30^3 = 27000 \times 27000$  over  $\mathbb{Q}[\epsilon]$ , where  $\epsilon$  is a 5-th primitive root of unity.

We are going to construct a subgroup  $H$  of  $G$  having lower-dimensional irreducible representations, for this makes the construction of the associator more feasible on the computer. The Horrocks-Mumford bundle is still an  $H$ -equivariant vector bundle, and we will be able to compute its equivariant cohomology in the end of this thesis (see Computation III.3.11). For the construction of  $H$ , take any Sylow 2-subgroup  $S$  of  $G$ . Then  $S$  will also be a Sylow 2-subgroup of  $\mathrm{SL}_2(5)$ , and any such  $S$  is isomorphic to the quaternion group  $Q_8$ . The subgroup  $H := H_5 \rtimes Q_8$  of  $G$  is of order 1000.

**Computation 3.62.** In the following GAP session, we first construct  $G$  using the matrices described in [HM73].

```

gap> eps := E(5);;
gap> g1 := PermutationMat( (1,2,3,4,5), 5 );;
gap> g2 := DiagonalMat( [1, eps, eps^2, eps^3, eps^4] );;
gap> n1 := PermutationMat( (2,5)(3,4), 5 );;
gap> n2 := -1*PermutationMat( (2,3,5,4), 5 );;

```

```

gap> n3 := DiagonalMat( [1, eps, eps^4, eps^4, eps] );;
gap> n4 := -1*ER(5)^(-1) *
> [ [ eps, eps^2, eps^3, eps^4, 1 ],
>   [ eps^2, eps^4, eps, eps^3, 1 ],
>   [ eps^3, eps, eps^4, eps^2, 1 ],
>   [ eps^4, eps^3, eps^2, eps, 1 ],
>   [ 1, 1, 1, 1, 1 ] ];;
gap> G := Group( g1, g2, n1, n2, n3, n4 );
<matrix group with 6 generators>
gap> ConjugacyClasses(G);;

```

$$G \simeq H_5 \rtimes \mathrm{SL}_2(5)$$

```

gap> StructureDescription( G );
"((C5 x C5) : C5) : SL(2,5)"

```

Next, we create a subgroup  $H \leq G$  isomorphic to  $H_5 \rtimes Q_8$ .

```

gap> H5 := Group( [ g1, g2 ] );;
gap> genH5 := GeneratorsOfGroup( H5 );;
gap> Q8 := SylowSubgroup( G, 2 );;
gap> genQ8 := GeneratorsOfGroup( Q8 );;
gap> H := Group( Concatenation( genH5, genQ8 ) );
<matrix group with 5 generators>

```

$$H \simeq H_5 \rtimes Q_8$$

```

gap> StructureDescription( H );
"((C5 x C5) : C5) : Q8"

```

We are interested in the number and degree of the irreducible characters of  $H$ .

```

gap> Size( Irr( H ) );
28
gap> List( Irr( H ), Degree );
[ 1, 1, 1, 1, 2, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 8, 8, 8,
  10, 10, 10, 10 ]

```

Thus, the largest degree is given by 10, and an associator computation for  $H$  has to deal with  $1000 \times 1000$  matrices over  $\mathbb{Q}[\epsilon]$ , which is a feasible task with the help of the computer algebra system Magma [BCP97] (which we can access via GAP using `homalg` [hom17, BLH11]). In Computation III.3.11 we will actually work with the tensor category  $\mathrm{SRep}_k^{\mathbb{Z}}(H)$  on the computer.



## CHAPTER 2

### Constructive Homological Algebra

The goal of this chapter is to describe, in the context of an arbitrary abelian category  $\mathbf{A}$ , an algorithm for computing spectral sequences which is suitable for a direct computer implementation. This means we will only use categorical constructions provided by the axioms of an abelian category, such as the existence of kernels and cokernels, and construct the objects and differentials on the pages of a spectral sequence as a concatenation of such primitives. Not only will we achieve this goal but we will get *non-recursive* closed formulas for the objects and differentials (Construction II.2.23 and II.2.25). In addition, we realize the functoriality of spectral sequences (Construction II.2.26).

Classical descriptions of spectral sequences (e.g. [Wei94]) are based on diagram chasing, which is a tool in homological algebra used for proving properties or the existence of morphisms situated in (commutative) diagrams of prescribed shape. Famous instances are for example the five lemma or the snake lemma [ML71]. For realizing our constructive approach to spectral sequences it will suffice to render diagram chases constructive.

There are many different approaches to diagram chases, but most of them are based on the idea of chasing some kind of “element” through a given diagram. For example, if we use the (non-constructive) Freyd-Mitchell embedding theorem [Mit65], we may think of objects  $A \in \mathbf{A}$  as modules, and get access to the elements of their underlying sets. In [ML71], we chase so-called “members” of  $A$ , which are modeled by morphisms  $X \rightarrow A$  that can be adjusted by composing with an arbitrary epimorphism  $Y \twoheadrightarrow X \rightarrow A$ . In [Sta16, Tag 05PP], diagram chasing for the abelian category of sheaves on a site is said to be performed by working with global sections or at the level of stalks.

A crucial step while performing a diagram chase with elements is the choice of a preimage of an element  $a \in A$  under an epimorphism  $\alpha : X \twoheadrightarrow A$ . Usually, one picks any such preimage  $x \in X$ , defines a new term with the help of  $x$ , and in the end proves that this term is independent of the particular choice of  $x$ . As an example, consider the following diagram in the category of abelian groups:

$$\begin{array}{ccc} A & & B \\ & \nwarrow \alpha & \nearrow \beta \\ & X & \end{array}$$

For a given  $a \in A$ , we choose  $x \in \alpha^{-1}(\{a\})$  and define the term  $\beta(x)$ . It is easy to see that  $\beta(x)$  is independent of the choice of  $x$  if and only if  $\beta(\ker(\alpha)) = 0$ . In this case, the assignment  $a \mapsto \beta(x)$  gives us a well-defined group homomorphism  $A \rightarrow B$ .

A natural strategy to avoid this choice is to work with the whole preimage  $\alpha^{-1}(\{a\})$  at once, and not just to take a single element  $x \in \alpha^{-1}(\{a\})$ . For that, we define a *relation*

$$\begin{array}{ccc} A & \overset{\alpha^{-1}}{\dashrightarrow} & X \\ a & \longmapsto & \alpha^{-1}(\{a\}) \end{array}$$

and call it the *pseudo-inverse* of  $\alpha$ . The dashed line highlights that  $\alpha^{-1}$  has to be seen as a relation and not as a function. The assignment  $a \mapsto \beta(x)$  can now be conveniently written as a composition of relations:  $\beta \circ \alpha^{-1}$ . In fact, any morphism constructed by means of a diagram chase can then be expressed as a composition of *relations*.

It follows that a constructive approach to diagram chases boils down to a categorification of the concept of relations, i.e., a description of relations which only uses categorical notions (in particular, it must not use elements). Such a categorification was given by Johnstone [Joh02] in the context of creating and characterizing toposes, where he uses what we call a stable span (see Definition II.1.20) as a data structure for a relation. His approach works for arbitrary *regular categories*. A formal axiomatization of the theory of “categories with relations as morphisms” is given in [FS90] under the name *allegory*.

In this chapter we develop the theory of relations in the context of an abelian category. This approach is a special instance of Johnstone’s work, since abelian categories are particular regular categories. But in fact, we can greatly benefit from the additional abelian structure. For example, instead of being forced to work with spans  $A \leftarrow X \rightarrow B$  as a data structure for relations, we are able to choose between six different data structures (see Subsection II.1.4), including cospans  $A \rightarrow X \leftarrow B$  and 3-arrows  $A \leftrightarrow X \rightarrow Y \leftarrow B$ . Different data structures might be best suited for the different computational tasks we want to solve:

- The 3-arrow data structure provides an easy normal form for a relation (Corollary II.1.45),
- there is a 4-arrow data structure which immediately gives an epi-mono factorization of a relation (Corollary II.1.53),
- computing the composition of cospans involves only *one* pushout,
- computing the composition of spans involves only *one* pullback.

In the context of regular categories we simply lack this variety of data structures. Another advantage of developing the theory of relations for abelian categories is that we can define notions tailored for homological algebra, or specifically for diagram chases. An example of such a notion is given by the *defect* (see Definition II.1.59), which is a subobject of the source measuring how far away a relation is from being single-valued, which simply cannot be defined in an arbitrary regular category. Furthermore, our presentation of the material focuses on constructive aspects of relations which are of importance for an *effective* computer implementation. For example, whereas in [FS90], spans have to undergo a normalization process each time when they are composed, we prove that this normalization process actually commutes with span composition and thus can be deferred to the end of the

computation (see Theorem II.1.15), which in turn can lead to a substantial improvement of performance.

Due to all of these computational and conceptual advantages, we introduce the new term *generalized morphisms* for *relations* in the abelian context, following Barakat’s convention in [Bar09a], [Bar09b], and [BLH14].

The author does not claim originality for the mathematical notions and theorems presented in this chapter. The approach is rather covered by the axiomatic setup for performing diagram chases in [Pup62] and [BP69] combined with the explicit construction of the category of generalized morphisms given in [Hil66]. However, our exposition refrains from stating the axioms in [BP69] (which are designed to reach even beyond abelian categories) and instead focuses on performing all necessary constructions directly in the category of generalized morphisms. In this way, it singles out those ideas relevant for the purpose of our computer implementation.

The chapter is divided in two sections. The first section is devoted to the general theory of generalized morphisms culminating in Subsection II.1.7 where we state computations rules useful for diagram chasing. The second section applies this theory to diagram chases which, in turn, are used for a constructive treatment of spectral sequences.

## 1. Generalized Morphisms

**1.1. Additive Relations.** In this first motivational subsection we introduce a framework for performing a diagram chase in the context of  $R\text{-mod}$ , the category of left modules over a unital ring  $R$ . The goal of the subsequent subsections is a categorification of this framework, i.e., a generalization to arbitrary abelian categories.

The main idea is to replace  $R$ -module homomorphisms in the category  $R\text{-mod}$  by additive relations.

**Definition 1.1.** Let  $A, B \in R\text{-mod}$ . We call a submodule  $S \subseteq A \oplus B$  an **additive relation** (or simply relation) **from**  $A$  **to**  $B$  and denote it by  $S : A \dashrightarrow B$ . For  $a \in A, b \in B$ , we also write  $S(a, b)$  instead of the term “ $(a, b) \in S$ ”.

The categorification of an additive relation is given by an object in the category of spans (see Definition II.1.8).

**Definition 1.2.** For every homomorphism of modules  $f : A \rightarrow B$ , its **graph**

$$\Gamma_f = \{(a, f(a)) \in A \oplus B \mid a \in A\}$$

defines a relation  $\Gamma_f : A \dashrightarrow B$ .

**Definition 1.3.** Let  $S : A \dashrightarrow B$  and  $T : B \dashrightarrow C$  be relations. We define their composite as

$$T \circ S := \{(a, c) \in A \oplus C \mid \exists b \in B : S(a, b) \wedge T(b, c)\}.$$

$T \circ S$  is an additive relation from  $A$  to  $C$ . Its categorification is given by the pullback operation as described in Definition II.1.9.

**Definition 1.4.** The **category of relations**  $\text{Rel}(R\text{-mod})$  consists of the following data:

- (1) Objects are  $R$ -modules.
- (2) Morphisms are relations of  $R$ -modules.
- (3) For an object  $A \in \text{Rel}(R\text{-mod})$ , the identity of  $A$  is given by

$$\{(a, a) \in A \oplus A \mid a \in A\}.$$

- (4) Composition of relations is given as in Definition II.1.3.

In Definition II.1.17 we describe the generalized morphism category for an arbitrary abelian category  $\mathbf{A}$ , which turns out to be equivalent to  $\text{Rel}(R\text{-mod})$  in the case  $\mathbf{A} = R\text{-mod}$  (see Theorem II.1.19).

Sending an  $R$ -module homomorphism  $f$  to its graph  $\Gamma_f$  is compatible with composition, which yields an embedding of  $R\text{-mod}$  into  $\text{Rel}(R\text{-mod})$ . Its categorification is described in Lemma II.1.26. Thus, we can interpret every diagram in  $R\text{-mod}$  as a diagram in  $\text{Rel}(R\text{-mod})$ . The benefit for diagram chases is enormous: *Every* morphism can be reversed (in the sense of a pseudo-inverse).

**Definition 1.5.** Let  $S : A \dashrightarrow B$  be a relation. The relation  $S^{-1} : B \dashrightarrow A$  given by swapping components is called the **pseudo-inverse** of  $S$ . That means for all  $a \in A$ ,  $b \in B$ , we have

$$S^{-1}(b, a) \Leftrightarrow S(a, b).$$

A categorical definition of the pseudo-inverse is given in Definition II.1.27.

Let us consider a toy example of a diagram chase. Given the following diagram of  $R$ -modules:

$$\begin{array}{ccc} A & & B \\ & \swarrow \alpha & \nearrow \beta \\ & X & \end{array}$$

The goal is to find a homomorphism from  $A$  to  $B$  rendering this diagram commutative. Regarding the homomorphisms  $\alpha$  and  $\beta$  as relations, we can always form the composite relation

$$\beta \circ \alpha^{-1} = \{(a, b) \in A \times B \mid \exists x \in \alpha^{-1}(\{a\}) : \beta(x) = b\}.$$

The only question left is: When is  $\beta \circ \alpha^{-1}$  equal to the graph of an  $R$ -module homomorphism? There are exactly 2 possible obstructions:  $\beta \circ \alpha^{-1}$  might not be **single-valued**, which means that one element in  $A$  might have multiple images, or not **total**, which means that one element in  $A$  might have no image at all. These obstructions are measured by the *defect* and the *domain*.

**Definition 1.6.** Let  $S : A \dashrightarrow B$  be a relation. We introduce the following subobjects.

- The **domain** of  $S$  is defined as

$$\text{dom}(S) := \{a \in A \mid \exists b \in B : S(a, b)\} \subseteq A.$$



- The **generalized kernel of  $S$**  is defined as

$$\text{gker}(S) := \{a \in A \mid S(a, 0)\} \subseteq A.$$

- The **defect of  $S$**  is defined as

$$\text{def}(S) := \{b \in B \mid S(0, b)\} \subseteq B.$$

- The **generalized image of  $S$**  is defined as

$$\text{gim}(S) := \{b \in B \mid \exists a \in A : S(a, b)\} \subseteq B.$$

Dually, we introduce the following quotient objects.

- The **codomain of  $S$**  is defined as  $\text{codom}(S) := B/\text{def}(S)$ .
- The **generalized cokernel of  $S$**  is defined as  $\text{gcoker}(S) := B/\text{gim}(S)$ .
- The **codefect of  $S$**  is defined as  $\text{codef}(S) := A/\text{dom}(S)$ .
- The **generalized coimage of  $S$**  is defined as  $\text{gcoim} := A/\text{gker}(S)$ .

In our example,  $\text{def}(\beta \circ \alpha^{-1}) = \beta(\text{ker}(\alpha))$  and  $\text{dom}(\beta \circ \alpha^{-1}) = \text{im}(\alpha)$ . It follows that  $\beta \circ \alpha^{-1}$  equals the graph of an  $R$ -module homomorphism if and only if

$$\beta(\text{ker}(\alpha)) = 0 \quad \text{and} \quad \text{im}(\alpha) = A.$$

So, we see that the canonical subobjects and quotient objects of Definition II.1.6 are helpful tools for diagram chasing.

All these canonical subobjects and quotient objects can be described in a completely categorical way (see Definition II.1.59).

**1.2. Categorification of Additive Relations.** As a first step towards a constructive framework for diagram chases, we formulate the concept of an additive relation in the context of an arbitrary abelian category.

**Notation 1.7.** In this subsection  $\mathbf{A}$  denotes an abelian category. In this case,  $\mathbf{A}$  has pullbacks (see Construction I.2.20), i.e., is equipped with a dependent function mapping  $\alpha : A \rightarrow B$  and  $\gamma : C \rightarrow B$  to a pullback  $(A \times_B C, A \xleftarrow{\gamma^*} A \times_B C \xrightarrow{\alpha^*} C)$  of  $\alpha$  and  $\gamma$ . Dually,  $\mathbf{A}$  has pushouts, i.e., is equipped with a dependent function mapping  $\alpha : B \rightarrow A$  and  $\gamma : B \rightarrow C$  to a pushout  $(A \sqcup_B C, A \xrightarrow{\gamma_*} A \sqcup_B C \xleftarrow{\alpha_*} C)$  of  $\alpha$  and  $\gamma$ .

In  $R$ -mod an additive relation from  $A$  to  $B$  was defined as a submodule  $S \subseteq A \oplus B$ . By the universal property of  $\oplus$ , such a submodule defines homomorphisms  $A \longleftarrow S \longrightarrow B$ . Such a *span* serves as a data structure for relations.

**Definition 1.8.** Let  $A, B \in \mathbf{A}$ . The **category of spans from  $A$  to  $B$**  is defined as the dependent sum category  $\sum_{C \in \mathbf{A}} \text{Hom}_{\mathbf{A}}(C, A) \times \text{Hom}_{\mathbf{A}}(C, B)$  and denoted by  $\text{Span}_{\mathbf{A}}(A, B)$ . We depict an object  $S = (C, \alpha, \beta)$  as

$$\begin{array}{ccc} A & \overset{\text{---} \boxed{S} \text{---}}{\dashrightarrow} & B \\ & \swarrow \alpha & \nearrow \beta \\ & C & \end{array}$$

or as

$$A \xleftarrow{\alpha} C \xrightarrow{\beta} B$$

Concretely, a morphism from  $(A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  to  $(A \xleftarrow{\alpha'} D \xrightarrow{\beta'} B)$  consists of a morphism  $\gamma : C \rightarrow D$  such that  $\alpha = \alpha' \circ \gamma$  and  $\beta = \beta' \circ \gamma$ .

Now, we are going to define a category where the homomorphisms from  $A$  to  $B$  are given by spans from  $A$  to  $B$ . Since such spans are themselves arranged in a category, we would expect to obtain a 2-category. However, since we do not need these extra data, we will truncate them and just speak about equality of spans.

**Definition 1.9.** The **category of spans of  $\mathbf{A}$** , denoted by  $\text{Span}(\mathbf{A})$ , is defined by the following data:

- (1) Objects are given by  $\text{Obj}_{\mathbf{A}}$ .
- (2) Morphisms from  $A$  to  $B$  are objects in  $\text{Span}_{\mathbf{A}}(A, B)$ .
- (3) Two spans are considered to be **equal as spans** if they are isomorphic as objects in  $\text{Span}_{\mathbf{A}}(A, B)$ .
- (4) The identity of  $A$  is given by  $(A \xleftarrow{\text{id}} A \xrightarrow{\text{id}} A)$ , where  $\text{id}$  denotes the identity of  $A$  regarded as an object in  $\mathbf{A}$ .
- (5) Composition of  $(A \xleftarrow{\alpha} D \xrightarrow{\beta} B)$  and  $(B \xleftarrow{\gamma} E \xrightarrow{\delta} C)$  is given by the outer span in the following diagram:

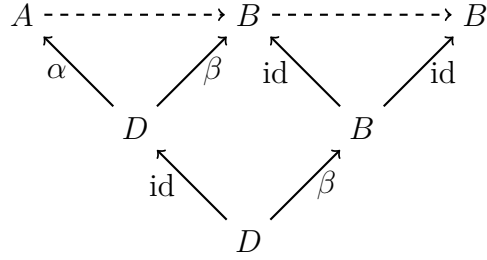
$$\begin{array}{ccccc}
 A & \text{-----} & B & \text{-----} & C \\
 & \swarrow \alpha & & \nwarrow \gamma & \\
 & D & & E & \\
 & \swarrow \beta & & \nwarrow \delta & \\
 & & & & \\
 & \swarrow \gamma^* & & \nwarrow \beta^* & \\
 & D \times_B E & & & 
 \end{array}$$

We need to check that the defining equations of a category hold for  $\text{Span}(\mathbf{A})$  up to equality of spans.

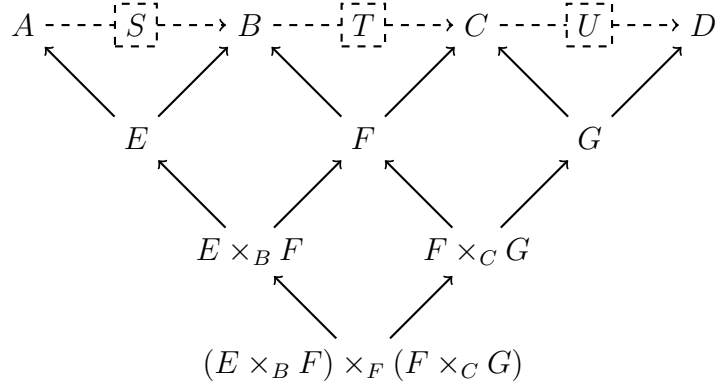
**Lemma 1.10.**

- (1) *The identity in  $\text{Span}(\mathbf{A})$  acts like a unit up to equality of spans.*
- (2) *Composition of morphisms in  $\text{Span}(\mathbf{A})$  is associative up to equality of spans.*

**PROOF.** For the first assertion, let  $(A \xleftarrow{\alpha} D \xrightarrow{\beta} B)$  be a span. Composition with  $(B \xleftarrow{\text{id}} B \xrightarrow{\text{id}} B)$  from the right yields the diagram



This proves that the identity is a right unit. An analogous argument shows that it is also a left unit. For the second assertion, consider the following diagram of consecutive pullbacks:



By transitivity of pullbacks, the rectangles with vertices  $E, B, F \times_C G, (E \times_B F) \times_F (F \times_C G)$  and  $C, G, E \times_B F, (E \times_B F) \times_F (F \times_C G)$  are also pullback squares. But this means that the outer span of the above diagram is isomorphic to both  $(U \circ T) \circ S$  and  $U \circ (T \circ S)$ .  $\square$

Every span  $(A \leftarrow C \rightarrow B)$  determines a morphism  $C \rightarrow A \oplus B$  which is *not monic* in general. Thus, in the case  $\mathbf{A} = R\text{-mod}$ , we cannot expect  $\text{Rel}(R\text{-mod})$  to be equivalent to the category of spans of  $\text{Span}(\mathbf{A})$ . This issue can be fixed by passing to an appropriate quotient category.

**Definition 1.11.** Given a span  $(A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$ , we define its **associated relation** as the image of the morphism

$$(\alpha, \beta) : C \rightarrow A \oplus B.$$

In particular, the associated relation of a span is a subobject of  $A \oplus B$ .

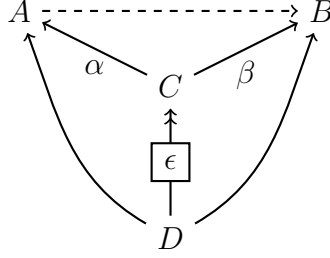
**Definition 1.12.** We say two spans from  $A$  to  $B$  are **stably equivalent** if and only if their associated relations are equal as subobjects of  $A \oplus B$ .

REMARK 1.13. Being stably equivalent is coarser than being equal as spans.

**Lemma 1.14.** Let  $\epsilon : D \rightarrow C$  be an epimorphism in  $\mathbf{A}$ . Every span of the form

$$(A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$$

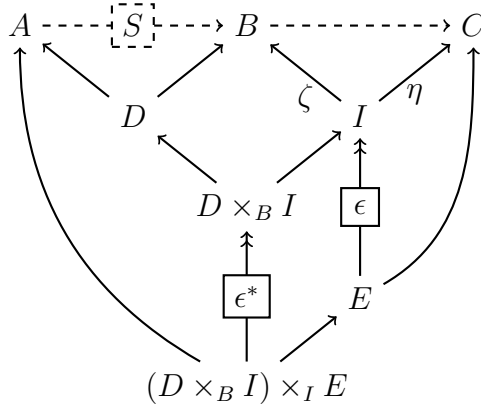
is stably equivalent to the outer span in the diagram given by composition with  $\epsilon$ :



PROOF. We have  $(\alpha \circ \epsilon, \beta \circ \epsilon) = (\alpha, \beta) \circ \epsilon$ , and in an abelian category, the image is not affected by epimorphisms. Thus,  $\text{im}((\alpha \circ \epsilon, \beta \circ \epsilon)) = \text{im}((\alpha, \beta))$ .  $\square$

**Theorem 1.15.** *Being stably equivalent defines a congruence on  $\text{Span}(\mathbf{A})$ .*

PROOF. Let  $S = (A \leftarrow D \rightarrow B)$  be a span and let  $(\zeta, \eta) : I \hookrightarrow B \oplus C$  be a monomorphism. Let  $T = (B \leftarrow E \rightarrow C)$  be a span obtained by composing  $\zeta, \eta$  with an epimorphism  $\epsilon : E \twoheadrightarrow I$ . By transitivity of the pullback, we get  $T \circ S$  as the outer span in the following diagram:



In an abelian category the pullback of an epimorphism yields an epimorphism. Thus,  $\epsilon^*$  is an epimorphism. Now, we apply Lemma II.1.14 to see that the stable equivalence class of  $T \circ S$  only depends on  $(\zeta, \eta)$ , which is the associated relation of  $T$ . Thus, if  $T$  and  $T'$  have the same associated relation, i.e., are stably equivalent, then so are  $S \circ T$  and  $S \circ T'$ . By the symmetry of the situation, a similar statement holds for stably equivalent  $S, S'$  and compositions  $S \circ T, S' \circ T$ . This shows the claim.  $\square$

REMARK 1.16. For the proof of Theorem II.1.15, we actually did not need  $\mathbf{A}$  to be abelian, but to admit an image factorization such that regular epimorphisms are preserved by pullbacks. Regular categories satisfy these properties, which is why in [Joh02], the constructions of this subsection are performed with regular categories.

Due to Theorem II.1.15, we can now define the generalized morphism category.

**Definition 1.17.** Let  $\mathbf{A}$  be an abelian category. The quotient category of  $\text{Span}(\mathbf{A})$  modulo stable equivalences is called the **generalized morphism category of  $\mathbf{A}$** , and denoted by  $\mathbf{G}(\mathbf{A})$ . Concretely, it consists of the following data:

- (1) Objects are given by  $\text{Obj}_{\mathbf{A}}$ .
- (2) Morphisms from  $A$  to  $B$  are spans from  $A$  to  $B$ .
- (3) Two spans are considered to be **equal as generalized morphisms** if and only if they are stably equivalent.
- (4) Identity and composition are given as in Definition II.1.9.

We call a span from  $A$  to  $B$  a **generalized morphism** when we regard it as a morphism in  $\mathbf{G}(\mathbf{A})$ .

REMARK 1.18. Note that due to Theorem II.1.15, composition of spans commutes with taking the associated relation. In particular, if we want to compose several generalized morphisms, we may first compose them as spans, and defer the computation of the associated relation to the end. This is an extremely useful feature w.r.t. an implementation of generalized morphisms on the computer. Note that in [Joh02] the computation of the associated relation is built into the composition process, which can be very costly in an actual implementation.

Now, we are going to show that in the case  $\mathbf{A} = R\text{-mod}$ , the concept of generalized morphisms is a formalization of additive relations in the language of category theory. For seeing this, we start by defining a functor

$$F : \mathbf{G}(R\text{-mod}) \rightarrow \text{Rel}(R\text{-mod})$$

sending a span to its associated relation. On the other hand, we can define a functor

$$G : \text{Rel}(R\text{-mod}) \rightarrow \mathbf{G}(R\text{-mod})$$

sending an additive relation  $S \subseteq A \oplus B$  to the span  $(A \xleftarrow{\pi_A} A \oplus B \leftrightarrow S \hookrightarrow A \oplus B \xrightarrow{\pi_B} B)$ .

**Theorem 1.19.**  *$F$  and  $G$  are well-defined functors giving rise to an equivalence of categories.*

PROOF. First, we show compatibility with composition. For this we will use the variable names in the following diagram depicting a composition:

$$\begin{array}{ccccc}
 A & \overset{\text{---}\{\overline{S}\}\text{---}}{\dashrightarrow} & B & \overset{\text{---}\{\overline{T}\}\text{---}}{\dashrightarrow} & C \\
 \swarrow \alpha & & \nearrow \beta & \swarrow \gamma & \nearrow \delta \\
 & D & & E & \\
 \swarrow \gamma^* & & \nearrow \beta^* & & \\
 & D \times_B E & & & 
 \end{array}$$

We can describe the associated relation of  $S$  as follows: For  $a \in A$ ,  $b \in B$ , we have

$$\begin{aligned}
 S(a, b) : & \iff (a, b) \in \text{im}((\alpha, \beta)) \\
 & \iff b \in \beta(\alpha^{-1}(\{a\})) \iff a \in \alpha(\beta^{-1}(\{b\})).
 \end{aligned}$$

For  $a \in A$ ,  $c \in C$ , we have equivalences

$$\begin{aligned}
& \exists b \in B : S(a, b) \wedge T(b, c) \\
& \iff \exists b \in B : (b \in \beta(\alpha^{-1}(\{a\}))) \wedge (b \in \gamma(\delta^{-1}(\{c\}))) \\
& \iff \exists d \in D, e \in E : (\alpha(d) = a) \wedge (\delta(e) = c) \wedge (\beta(d) = \gamma(e)) \\
& \iff \exists p \in D \times_B E : (\alpha\gamma^*(p) = a) \wedge (\delta\beta^*(p) = c).
\end{aligned}$$

This proves compatibility of  $F$  with composition. On the other hand, if we assume that  $S$  and  $T$  are morphisms lying in the image of  $G$ , we can also use the above chain of equivalences to prove compatibility of  $G$  with composition.  $F \circ G$  yields the identity.  $G \circ F$  does not change the associated relation and thus yields stably equivalent spans.  $\square$

**1.3. Computation Rules for Generalized Morphisms.** For the construction of morphisms by diagram chases, we have to learn how to compute within the generalized morphism category. In this subsection we see how we can decide equality of generalized morphisms (Remark II.1.23), interpret a morphism in  $\mathbf{A}$  as a generalized morphism (Lemma II.1.26), construct pseudo-inverses (Definition II.1.27), and simplify equations of generalized morphisms via computation rules (Theorem II.1.31 and Theorem II.1.35).

For deciding equality, we establish a normal form.

**Definition 1.20.** A span  $(A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  is called **stable** if the morphism

$$(\alpha, \beta) : C \rightarrow A \oplus B$$

is a monomorphism.

**Lemma 1.21.** *Two stable spans are equal as generalized morphisms if and only if they are equal as spans.*

PROOF. For stable spans, being equal as spans means that the associated relations are equal as subobjects.  $\square$

**Lemma 1.22.** *Any span  $S = (A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  is stably equivalent to a stable span.*

PROOF. We construct the image factorization

$$\begin{array}{ccc}
C & \xrightarrow{(\alpha, \beta)} & A \oplus B. \\
& \searrow & \nearrow \\
& I & \xrightarrow{(\zeta, \eta)}
\end{array}$$

By Lemma II.1.14,  $(I, \zeta, \eta)$  is stably equivalent to  $S$ .  $\square$

REMARK 1.23. Combining the Lemmas II.1.21 and II.1.22, we can think of stable spans as a normal form for generalized morphisms.

Here is how we can construct a normal form for a generalized morphism.

**Construction 1.24.** The image embedding of a morphism  $(\alpha, \beta) : C \rightarrow A \oplus B$  by definition is given by

$$\text{KernelEmbedding}(\text{CokernelProjection}(\alpha, \beta)).$$

It can be constructed via pullbacks and pushouts using the following fact: A sequence

$$C \xrightarrow{(\alpha, \beta)} A \oplus B \xrightarrow{\begin{pmatrix} \gamma \\ \delta \end{pmatrix}} D$$

is right/left exact if and only if the commutative diagram

$$\begin{array}{ccc} & D & \\ \gamma \nearrow & & \nwarrow -\delta \\ A & & B \\ \alpha \nwarrow & & \nearrow \beta \\ & C & \end{array}$$

is a pushout/pullback square, respectively. In particular, the image embedding of  $(\alpha, \beta)$  can be constructed by first taking the pushout of  $\alpha, \beta$ , which gives us morphisms  $\gamma, -\delta$ , and then taking the pullback of  $\gamma, -\delta$ , which yields morphisms  $\zeta, \eta$ . Then  $(\zeta, \eta) : I \rightarrow A \oplus B$  is the image embedding  $(\alpha, \beta)$ .

**REMARK 1.25.** In general, the method for computing the image embedding described in Construction II.1.24 does not work for regular categories which are not abelian. For example, let us study relations in **Set**. Every relation  $R \subseteq A \times B$  which arises as the pullback of a cospan  $(A \rightarrow D \leftarrow B)$  has the following “transitivity” property: For all  $a, a' \in A, b, b' \in B$ :

$$R(a, b) \wedge R(a', b) \wedge R(a', b') \Rightarrow R(a, b').$$

Conversely, every relation satisfying this “transitivity” property arises as the pullback of a cospan. But not every relation in **Set** satisfies the “transitivity” property, e.g., define  $R \subseteq \{1, 2, 3\}^2$  such that exactly  $R(1, 2), R(3, 2), R(3, 1)$  hold, but not  $R(1, 1)$ . The method described in Construction II.1.24 then yields a coarser relation, i.e., a relation containing  $R$ .

Since we want to interpret every diagram in **A** as a diagram in **G(A)**, we need an embedding of **A** into **G(A)**.

**Lemma 1.26.** Mapping a morphism  $\alpha : A \rightarrow B$  to the span

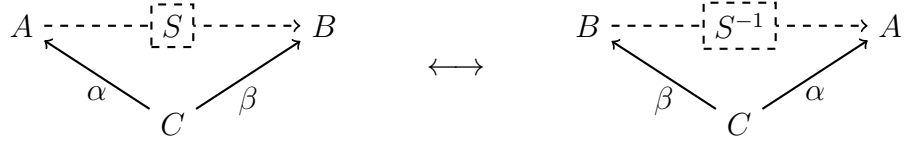
$$\begin{array}{ccc} A & \overset{[\alpha]}{\dashrightarrow} & B \\ & \swarrow \text{id}_A & \nearrow \alpha \\ & A & \end{array}$$

yields a faithful functor  $[-] : \mathbf{A} \rightarrow \mathbf{G}(\mathbf{A})$ .

PROOF. The pullback of an identity morphism can be chosen as an identity morphism. This proves functoriality. Because  $[\alpha] = (A \xleftarrow{\text{id}} A \xrightarrow{\alpha} B)$  is a stable span, it is equal to  $[\beta] = (A \xleftarrow{\text{id}} A \xrightarrow{\beta} B)$  as generalized morphisms if and only if  $\alpha = \beta$  (Lemma II.1.21). Thus,  $[-]$  is faithful.  $\square$

The most powerful feature of  $\mathbf{G}(\mathbf{A})$  regarding diagram chases is the possibility to “reverse” every arrow.

**Definition 1.27.** For a span  $S = (A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  from  $A$  to  $B$ , we call the span  $(B \xleftarrow{\beta} C \xrightarrow{\alpha} A)$  from  $B$  to  $A$  its **pseudo-inverse** and denote it by  $S^{-1}$ .



REMARK 1.28. Taking pseudo-inverses is compatible with stable equivalences. Thus, it defines an equivalence of categories

$$(-)^{-1} : \mathbf{G}(\mathbf{A})^{\text{op}} \rightarrow \mathbf{G}(\mathbf{A}).$$

REMARK 1.29. We will see in Theorem II.1.35 that pseudo-inverses are inverses in the sense of semi-groups. From this it will follow that if a generalized morphism  $S$  is an isomorphism, its inverse equals its pseudo-inverse. This justifies the notation  $S^{-1}$ .

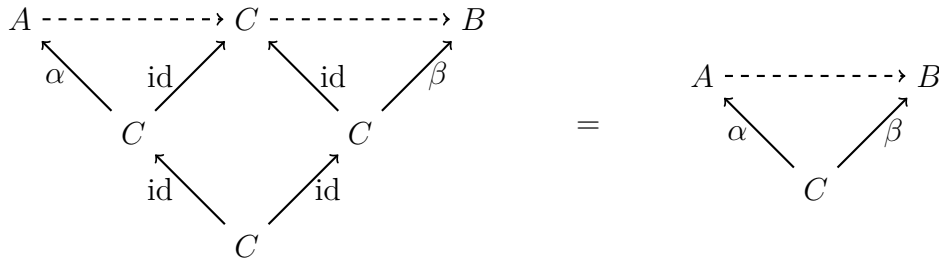
The morphisms constructed by diagram chases are often given by a finite composition of the form

$$(\dagger) \quad \cdots \circ [\alpha_i] \circ [\alpha_{i+1}]^{-1} \circ [\alpha_{i+2}] \circ [\alpha_{i+3}]^{-1} \circ [\alpha_{i+4}] \circ \cdots$$

for morphisms  $\alpha_i \in \mathbf{A}$  (as an example see the Snake Lemma II.2.1). Thus, we need to learn how to simplify terms like  $(\dagger)$ .

**Lemma 1.30.** Every span  $(A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  is equal to  $[\beta] \circ [\alpha]^{-1}$  as generalized morphisms.

PROOF. A square consisting of identities is a pullback square. Thus, we have an equation of generalized morphisms (even as spans):

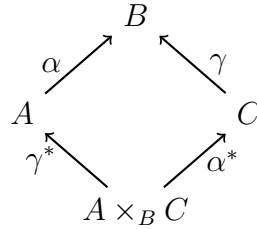


$\square$



Now, we can state computation rules for  $\mathbf{G}(\mathbf{A})$  that allow us to “swap” two consecutive morphisms in a term like  $(\dagger)$ .

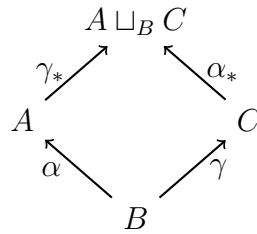
**Theorem 1.31.** *Given a pullback diagram*



the *pullback computation rule*

$$[\gamma]^{-1} \circ [\alpha] = [\alpha^*] \circ [\gamma^*]^{-1}$$

holds. Dually, given a pushout square

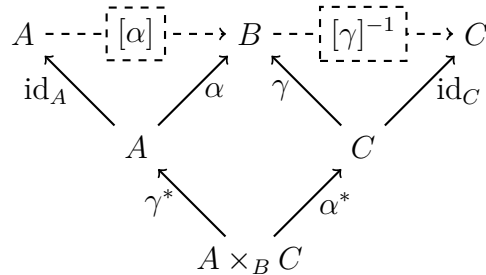


the *pushout computation rule*

$$[\gamma] \circ [\alpha]^{-1} = [\alpha_*]^{-1} \circ [\gamma_*]$$

holds.

PROOF. From the diagram



and Lemma II.1.30, we get the pullback computation rule. Now let  $\alpha_*^* : A \times_{A \sqcup_B C} C \rightarrow A$  and  $\gamma_*^* : A \times_{A \sqcup_B C} C \rightarrow C$  be the pullback projections of  $\gamma_*$ ,  $\alpha_*$ :

$$\begin{array}{ccc}
 & A \sqcup_B C & \\
 \gamma_* \nearrow & & \nwarrow \alpha_* \\
 A & & C \\
 \alpha_*^* \nwarrow & & \nearrow \gamma_*^* \\
 & A \times_{A \sqcup_B C} C &
 \end{array}$$

By the pullback computation rule, we have

$$[\alpha_*]^{-1} \circ [\gamma_*] = [\gamma_*^*] \circ [\alpha_*^*]^{-1}.$$

But taking pushout followed by taking pullback yields the image embedding (see Construction II.1.24) and thus the stable representative of  $[\gamma] \circ [\alpha]^{-1}$ . It follows that

$$[\gamma_*^*] \circ [\alpha_*^*]^{-1} = [\gamma] \circ [\alpha]^{-1}.$$

□

REMARK 1.32. Remark II.1.25 shows that in general, the pushout computation rule does not hold if  $\mathbf{A}$  was a regular category.

The next computation rule that we are going to state justifies the name of the *pseudo-inverse*. For its proof, we need to know the connection between bicartesian squares and stable spans.

**Definition 1.33.** A square

$$\begin{array}{ccc}
 & C & \\
 \nearrow & & \nwarrow \\
 A & & B \\
 \nwarrow & & \nearrow \\
 & D &
 \end{array}$$

is called **bicartesian** if it is a pullback and a pushout square.

**Lemma 1.34.** A span  $(A \xleftarrow{\alpha} C \xrightarrow{\gamma} B)$  is stable if and only if the pushout diagram

$$\begin{array}{ccc}
 & A \sqcup_C B & \\
 \gamma_* \nearrow & & \nwarrow \alpha_* \\
 A & & B \\
 \alpha \nwarrow & & \nearrow \gamma \\
 & C &
 \end{array}$$

is bicartesian.

PROOF. By Construction II.1.24,  $(\alpha, \gamma) : C \rightarrow A \oplus B$  is a monomorphism if and only if the given pushout diagram is also a pullback diagram. □

**Theorem 1.35.** *Let  $S$  be a span from  $A$  to  $B$ . Then the following computation rules hold:*

- (1)  $S \circ S^{-1} \circ S = S$ ,
- (2)  $S^{-1} \circ S \circ S^{-1} = S^{-1}$ ,

*as generalized morphisms. In particular, if  $S$  is an isomorphism in  $\mathbf{G}(\mathbf{A})$ , then its pseudo-inverse is an inverse.*

**PROOF.** The second equation follows by applying  $(-)^{-1}$  to the first equation. Thus, it suffices to prove the first claim. Using Lemma II.1.22, we may assume that the span  $S = (A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  is stable. By Lemma II.1.34, this means that the following pushout diagram is bicartesian, where  $D$  denotes  $A \sqcup_C B$ :

$$\begin{array}{ccc}
 & D & \\
 \gamma_* \nearrow & & \nwarrow \alpha_* \\
 A & & B \\
 \alpha \nwarrow & & \nearrow \gamma \\
 & C &
 \end{array}$$

Now, consider the following commutative diagram, where  $(C \times_D D, \eta, \zeta)$  denotes the pullback of  $\alpha_* \circ \gamma$  and  $\gamma_* \circ \alpha$ , and  $u$  the universal morphism associated to the source  $(C, \text{id}_C, \text{id}_C)$ :

$$\begin{array}{ccccc}
 & & \gamma_* & & \alpha_* \\
 & & \curvearrowright & & \curvearrowleft \\
 & & D & & D \\
 & & \alpha_* \nearrow & & \nwarrow \gamma_* \\
 A & \text{---} \boxed{S} \text{---} & B & \text{---} \boxed{S^{-1}} \text{---} & A & \text{---} \boxed{S} \text{---} & B \\
 \alpha \nwarrow & & \nearrow \gamma & & \nwarrow \alpha & & \nearrow \gamma \\
 & C & & & C & & \\
 & \eta \nearrow & & & \nwarrow \zeta & & \\
 & C \times_D C & & & C \times_D C & & \\
 \text{id}_C \nearrow & & & & \nwarrow \text{id}_C & & \\
 & C & & & C & & \\
 & u \uparrow & & & & & \\
 & C & & & & &
 \end{array}$$

From the pushout and the pullback computation rules, we get the equation

$$S \circ S^{-1} \circ S = [\gamma \circ \zeta] \circ [\alpha \circ \eta]^{-1}.$$

Since  $(\alpha \circ \eta, \gamma \circ \zeta) \circ u = (\alpha, \gamma)$ , we have  $\text{im}((\alpha \circ \eta, \gamma \circ \zeta)) \supseteq \text{im}((\alpha, \gamma))$ . To prove the other direction, we construct a morphism  $v : C \times_D C \rightarrow C$  as the universal morphism associated

to the source  $(C \times_D C, \alpha \circ \eta, \gamma \circ \zeta)$ , where we consider  $C$  as a pullback of  $\gamma_*, \alpha_*$  (which we can do since we assumed  $S$  to be stable):

$$\begin{array}{ccccc}
 & & D & & \\
 & \nearrow \gamma_* & & \nwarrow \alpha_* & \\
 A & & & & B \\
 \alpha \uparrow & & & & \uparrow \gamma \\
 C & & C & & C \\
 \eta \uparrow & & \uparrow v & & \uparrow \zeta \\
 & & C \times_D C & & 
 \end{array}$$

Since  $(\alpha, \gamma) \circ v = (\alpha \circ \eta, \gamma \circ \zeta)$ , we have  $\text{im}((\alpha \circ \eta, \gamma \circ \zeta)) \subseteq \text{im}((\alpha, \gamma))$ . Thus,  $[\gamma \circ \zeta] \circ [\alpha \circ \eta]^{-1}$  and  $[\gamma] \circ [\alpha]^{-1} = S$  are stably equivalent.  $\square$

REMARK 1.36. Theorem II.1.35 may not hold if  $\mathbf{A}$  is a regular category which is not abelian. For example, if  $\mathbf{A} = \mathbf{Set}$ , the relation associated to the span  $S \circ S^{-1} \circ S$  always has the “transitivity” property described in Remark II.1.25.

**1.4. Data Structures for Generalized Morphisms.** Remember that we are building a framework for effective diagram chases based on computing with generalized morphisms. Up to now, we have represented a generalized morphism by a span

$$(A \longleftarrow C \longrightarrow B)$$

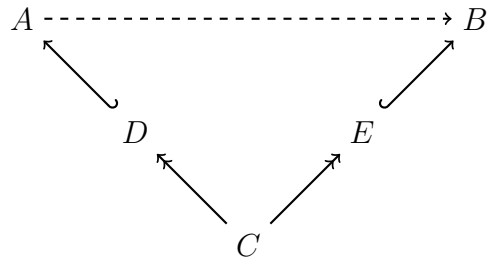
Composition of spans involves pullbacks, which is a costly operation in specific instances of abelian categories (like in the category of finitely presented  $R$ -modules introduced in the CAP project chapter). Pushouts, on the other hand, can be very cheap. Thus, for an effective implementation of generalized morphisms on the computer, alternative data structures are desirable.

In this subsection we derive six different data structures for generalized morphisms, including cospans (which can be composed via pushouts). We further characterize normal forms for each of these data structures. In addition to their computational value, these data structures clarify the abstract structure of generalized morphisms and are the key ingredient to the proof that any generalized morphism has a universal epi-mono factorization in Subsection II.1.5.

This is how we derive the six data structures for generalized morphisms: Given a span

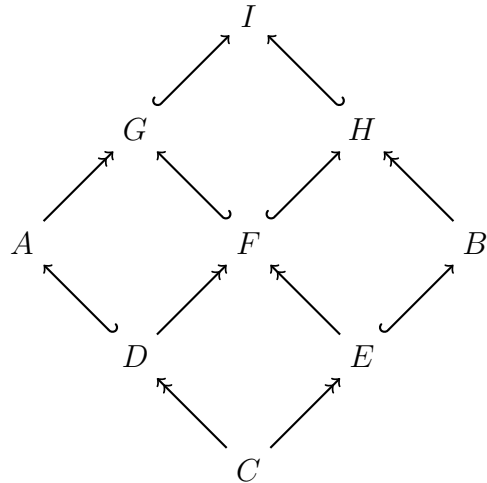
$$\begin{array}{ccc}
 A & \overset{\text{-----}}{\longrightarrow} & B \\
 \alpha \swarrow & & \searrow \beta \\
 & C & 
 \end{array}$$

we apply an image factorization to  $\alpha$  and  $\beta$ :

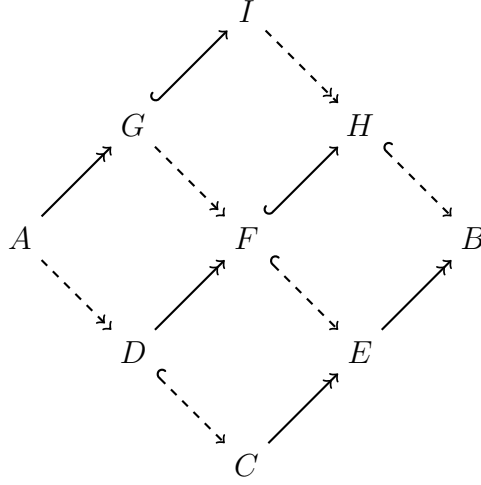


Next, we successively take pushouts to construct a diamond shaped diagram (Figure II.1). Here, the pushouts are taken in alphabetical order, i.e.,  $F, G, H, I$ .

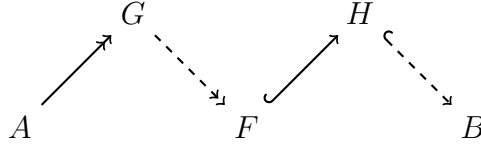
FIGURE 1. A diamond with epimorphisms and monomorphisms.



Since pushouts in any category respect epimorphisms, and pushouts in any abelian category respect monomorphisms, each arrow in the diamond is either a monomorphism or an epimorphism, the exact distribution is depicted in Figure II.1. Taking pseudo-inverses of those arrows pointing north-west, we get six paths from  $A$  to  $B$  in  $\mathbf{G}(\mathbf{A})$  (see Figure II.2). Note that since taking pseudo-inverse is an anti-equivalence of  $\mathbf{G}(\mathbf{A})$ , monomorphisms are mapped to epimorphisms and vice versa.

FIGURE 2. There exist six paths from  $A$  to  $B$ .

By the pushout computation rule II.1.31, each of those paths yields the same generalized morphism. Following for example the path



we can already see that a generalized morphism can be decomposed into an epimorphism and a monomorphism. In Corollary II.1.53, we will prove that such a factorization is universal, i.e., essentially unique.

REMARK 1.37. If  $\mathbf{A}$  was only a regular category, the six paths could differ, since the pushout computation rule may not hold (see Remark II.1.32).

Each of those paths may serve as a representation for the same generalized morphism. We already have a formal context for the representation corresponding to the lower path of Figure II.2, namely it can be seen as an object in the category  $\text{Span}(A, B)$ . We give a formal context for each of the other five representations.

**Notation 1.38.** For objects  $X, Y \in \mathbf{A}$ , we introduce the notation  $Y^X := \text{Hom}_{\mathbf{A}}(X, Y)$ .

**Definition 1.39.** Let  $A, B \in \mathbf{A}$ .

- (1) The **category of cospans from  $A$  to  $B$**  is defined as the dependent sum category

$$\text{Cospan}(A, B) := \sum_{I \in \mathbf{A}} I^A \times I^B$$

- (2) The **category of 3-arrows from  $A$  to  $B$**  is defined as the full subcategory of the dependent sum category

$$\sum_{D, H \in \mathbf{A}} A^D \times H^D \times H^B$$

generated by those objects  $(D, H, \alpha, \beta, \gamma)$  such that  $\alpha : D \hookrightarrow A$  is a monomorphism and  $\gamma : B \twoheadrightarrow H$  is an epimorphism. It is denoted by  $3\text{-Arrow}(A, B)$ .

- (3) The **category of reversed 3-arrows from  $A$  to  $B$**  is defined as the full subcategory of the dependent sum category

$$\sum_{G, E \in \mathbf{A}} G^A \times G^E \times B^E$$

generated by those objects  $(G, E, \alpha, \beta, \gamma)$  such that  $\alpha : A \twoheadrightarrow G$  is an epimorphism and  $\gamma : E \hookrightarrow B$  is a monomorphism. It is denoted by  $3\text{-Arrow}^\circ(A, B)$ .

- (4) The **category of 4-arrows from  $A$  to  $B$**  is defined as the full subcategory of the dependent sum category

$$\sum_{D, F, E \in \mathbf{A}} A^D \times F^D \times F^E \times B^E$$

generated by those objects  $(D, F, E, \alpha, \beta, \gamma, \delta)$  such that  $\alpha : D \hookrightarrow A$ ,  $\delta : E \hookrightarrow B$  are monomorphisms and  $\beta : D \twoheadrightarrow F$ ,  $\gamma : E \twoheadrightarrow F$  are epimorphisms. It is denoted by  $4\text{-Arrow}(A, B)$ .

- (5) The **category of reversed 4-arrows from  $A$  to  $B$**  is defined as the full subcategory of the dependent sum category

$$\sum_{G, F, H \in \mathbf{A}} G^A \times G^F \times H^F \times H^B$$

generated by those objects  $(G, F, H, \alpha, \beta, \gamma, \delta)$  such that  $\alpha : A \twoheadrightarrow G$ ,  $\delta : B \twoheadrightarrow H$  are epimorphisms and  $\beta : F \hookrightarrow G$ ,  $\gamma : F \hookrightarrow H$  are monomorphisms. It is denoted by  $4\text{-Arrow}^\circ(A, B)$ .

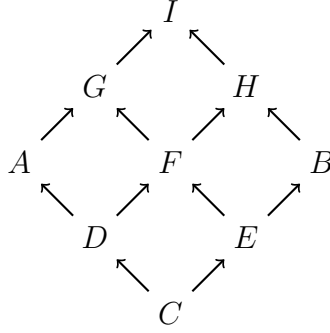
So, each of the new representations that we found for a generalized morphism can be seen as an object in one of the five categories enlisted in Definition II.1.39. We call the underlying category of a given representation its **data structure**. Two representations of the same data structure are said to be **equal as representations** if they are isomorphic as objects.

Now, we are going to study normal forms for generalized morphisms in each of these data structures. To handle all cases at one stroke, we introduce the following special type of diagram.

**Definition 1.40.** Let  $A, B \in \mathbf{A}$ . The **category of diamonds from  $A$  to  $B$**  is defined as the dependent sum category

$$\sum_{\substack{C, D, E, F, \\ G, H, I \in \mathbf{A}}} D^C \times E^C \times A^D \times F^D \times F^E \times B^E \times G^A \times G^F \times H^F \times H^B \times I^G \times I^H$$

and denoted by  $\diamond(A, B)$ . Objects in  $\diamond(A, B)$  are called **diamonds (from  $A$  to  $B$ )**. We depict a diamond by a *not necessarily* commutative diagram of the following form:



We say two diamonds from  $A$  to  $B$  are **equal as diamonds** if they are isomorphic as objects in  $\diamond(A, B)$ .

The process described in the beginning of this subsection defines a functor

$$\diamond : \text{Span}(A, B) \rightarrow \diamond(A, B),$$

since constructing an image factorization and taking pushouts are functorial operations. Given a diamond, we can forget all of its objects and morphisms but the ones in the lower span and compose consecutive morphisms. This process yields a functor

$$\vee : \diamond(A, B) \rightarrow \text{Span}(A, B)$$

such that  $\vee \circ \diamond = \text{id}_{\text{Span}(A, B)}$ . Since we want to construct normal forms for the six different data structures, we are interested in  $\diamond(S)$  for stable spans  $S$ .

**Definition 1.41.** A diamond is called **stable** if it is a diamond with epimorphisms and monomorphisms as depicted in Figure II.1 and if each of its inner squares is bicartesian.

**Theorem 1.42.** *The functors  $\diamond$  and  $\vee$  restrict to an equivalence of categories between stable spans and stable diamonds.*

**PROOF.** On the one hand, we already know that  $\vee \circ \diamond = \text{id}_{\text{Span}(A, B)}$ . On the other hand, if  $\Delta$  is a stable diamond, then  $(\diamond \circ \vee)(\Delta)$  and  $\Delta$  are equal as diamonds due to the uniqueness of pushouts and universal epi-mono factorizations. Thus, it suffices to show that  $\diamond$  and  $\vee$  restrict to stable spans and stable diamonds.

Given a stable diamond  $\Delta$ , its outer square is bicartesian due to transitivity of pullback and pushout. By Lemma II.1.34, this means that  $\vee(\Delta)$  is stable.

Now, if  $S = (A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  is a span, then  $\diamond(S)$  is a diamond with epimorphisms and monomorphisms as depicted in Figure II.1 by construction of  $\diamond$ . We will use the variable names in Figure II.1 for our notation in the rest of this proof. Since  $S$  is stable, the morphism  $(\alpha, \beta) : C \rightarrow A \oplus B$  is a monomorphism. Since  $D \hookrightarrow A$  and  $E \hookrightarrow B$  are monomorphisms, so is  $D \oplus E \hookrightarrow A \oplus B$ . Since subobjects of  $A \oplus B$  are related via monomorphisms, the morphism  $C \rightarrow D \oplus E$  is a monomorphism. Thus, by Lemma II.1.34, the square with vertices  $C, D, E, F$  is bicartesian.

Furthermore, the morphism  $D \rightarrow A \oplus F$  is a monomorphism since  $D \rightarrow A$  is. Again by Lemma II.1.34, the square with vertices  $D, A, F, G$  is bicartesian. An analogous argument



is valid for the other two inner squares. This shows stability of  $\diamond(S)$  and concludes the proof.  $\square$

**REMARK 1.43.** Likewise, functors analogous to  $\diamond$  can be also defined for the other five data structures by decomposing arrows via the image factorization, and taking successively pushouts and pullbacks until we obtain a diamond with epimorphisms and monomorphisms as depicted in Figure II.1.

Functors analogous to  $\vee$  can also be defined for the other five data structures by forgetting all of its objects and morphisms but the ones in the path corresponding to the chosen data structure, and by composing consecutive morphisms.

Again, we have  $\vee \circ \diamond = \text{id}$ .

**Corollary 1.44.** *Let  $A, B \in \mathbf{A}$ . Let  $\mathbf{C}$  denote either  $\text{Span}(A, B)$  or one of the categories defined in II.1.39. Let  $S, T \in \mathbf{C}$  such that  $\diamond(S)$  and  $\diamond(T)$  are stable diamonds. Then  $S$  and  $T$  represent the same generalized morphism if and only if  $\diamond(S)$  and  $\diamond(T)$  are equal as diamonds.*

**PROOF.** This is a direct consequence of Theorem II.1.42 and Lemma II.1.21.  $\square$

**Corollary 1.45** (Normal forms for 3-arrows and 4-arrows). *Let  $A, B \in \mathbf{A}$  and let  $\mathbf{C}$  denote either  $3\text{-Arrow}(A, B)$ ,  $3\text{-Arrow}^\circ(A, B)$ ,  $4\text{-Arrow}(A, B)$ , or  $4\text{-Arrow}^\circ(A, B)$ . Let  $S, T \in \mathbf{C}$ . Then  $S$  and  $T$  represent the same generalized morphism if and only if they are equal as representations.*

**PROOF.** First we show that  $\diamond(S)$  is stable for  $S \in \mathbf{C}$ . This follows from the fact that the pushout of two morphisms  $\alpha, \gamma$ , where  $\alpha$  is a monomorphism, yields a bicartesian square, and dually, the pullback of two morphisms  $\alpha, \gamma$ , where  $\alpha$  is an epimorphism, also yields a bicartesian square. Now, using Corollary II.1.44, we see that  $S$  and  $T$  represent the same generalized morphism if and only if  $\diamond(S) = \diamond(T)$ . Applying  $\vee$  yields  $S = T$ .  $\square$

To complete the description of normal forms, we discuss cospans.

**Definition 1.46.** A cospan  $(A \xrightarrow{\alpha} C \xleftarrow{\beta} B)$  is called **stable** if the morphism

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} : A \oplus B \rightarrow C$$

is an epimorphism.

**Lemma 1.47** (Normal forms for cospans). *Let  $A, B \in \mathbf{A}$  and let  $S, T \in \text{Cospan}(A, B)$  be stable. Then  $S, T$  represent the same generalized morphism if and only if they are equal as cospans.*

**PROOF.** Taking the pullback of the morphisms in a stable cospan yields a bicartesian square. Now,  $S, T$  represent the same generalized morphism if and only if the spans inside these bicartesian squares are equal as spans, which is the case if and only if the cospans inside these bicartesian squares are equal as cospans.  $\square$

In order to compute with each of these data structures, we have to transfer the relevant computational functions such as composition and pseudo-inversion from spans to the data structure in question. For example, a description of the composition in the three arrow calculus  $3\text{-Arrow}(A, B)$  can be found in [BLH14]. Such a transfer can be done using conversions of data structures.

**Theorem 1.48** (Conversion from spans to cospans). *Mapping a span*

$$(A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$$

to its pushout cospan

$$(A \xrightarrow{\beta_*} A \sqcup_C B \xleftarrow{\alpha_*} B)$$

defines an equivalence of categories

$$\text{Conversion} : \mathbf{G}(\mathbf{A}) \rightarrow \mathbf{G}(\mathbf{A}^{\text{op}}).$$

PROOF. Conversion maps stably equivalent spans to equal cospans due to Construction II.1.24. Furthermore, it respects composition due to the pullback and pushout computation rules. Its inverse functor is given by mapping a cospan to its pullback span.  $\square$

REMARK 1.49. Theorem II.1.48 does not hold in general if  $\mathbf{A}$  was only a regular category, since for example, if  $\mathbf{A} = \mathbf{Set}$ , not all relations arise as a pullback (see Remark II.1.25).

From Theorem II.1.48 and Remark II.1.28, we get a tool for *dualizing* in the context of generalized morphisms.

**Definition 1.50.** The equivalence of categories

$$\text{Dualize} : \mathbf{G}(\mathbf{A})^{\text{op}} \rightarrow \mathbf{G}(\mathbf{A}^{\text{op}})$$

is defined as the composition

$$\mathbf{G}(\mathbf{A})^{\text{op}} \xrightarrow{(-)^{-1}} \mathbf{G}(\mathbf{A}) \xrightarrow{\text{Conversion}} \mathbf{G}(\mathbf{A}^{\text{op}}).$$

REMARK 1.51. The functor Dualize (see Definition II.1.50) *dualizes* a generalized morphism in the following way: It first constructs the pseudo-inverse and second applies  $(-)^{\text{op}}$  to the arrows in its representation. More formally: If  $\gamma = (A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$  is a generalized morphism in  $\mathbf{G}(\mathbf{A})$ , then

$$\text{Dualize}(\gamma) = \text{Dualize}([\beta] \circ [\alpha]^{-1}) = [\alpha^{\text{op}}]^{-1} \circ [\beta^{\text{op}}].$$

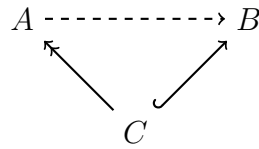
This is a direct consequence of the pushout computation rule. By abuse of notation, we also write  $\gamma^{\text{op}}$  for  $\text{Dualize}(\gamma)$ :

$$\left( \begin{array}{ccc} A & \overset{\text{---}\gamma\text{---}}{\dashrightarrow} & B \\ & \swarrow \alpha & \searrow \beta \\ & C & \end{array} \right)^{\text{op}} \quad \xleftrightarrow{\text{Dualize}} \quad \begin{array}{ccc} A & \overset{\text{---}\gamma^{\text{op}}\text{---}}{\dashrightarrow} & B \\ & \swarrow \alpha^{\text{op}} & \searrow \beta^{\text{op}} \\ & C & \end{array}$$

**1.5. Epi-Mono Factorizations of Generalized Morphisms.** In this subsection we prove that any generalized morphism admits a *universal* epi-mono factorization. In particular, any generalized morphism has an *image* and a *coimage* in the categorical sense (see Subsection I.1.3), and they are isomorphic like in abelian categories. These images and coimages are the key tool for reasoning about the morphisms we construct via diagram chases, as we will see in Subsection II.1.6.

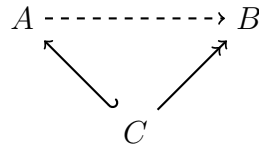
In the last subsection we saw that any generalized morphism admits an epi-mono factorization due to the 4-arrow representation. In order to see that this factorization is essentially unique, we need to understand the structure of arbitrary monomorphisms and epimorphisms in  $\mathbf{G}(\mathbf{A})$ .

**Theorem 1.52.** *Every monomorphism in  $\mathbf{G}(\mathbf{A})$  has a stable representative of the form*



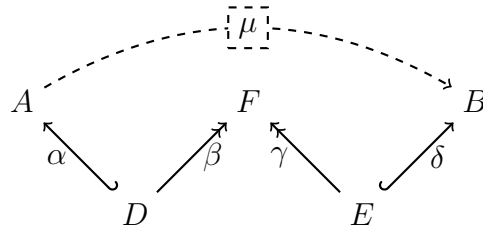
*and every morphism of this form is a monomorphism. Moreover, every monomorphism in  $\mathbf{G}(\mathbf{A})$  is split with its pseudo-inverse as a retraction.*

*Dually, every epimorphism in  $\mathbf{G}(\mathbf{A})$  has a stable representative of the form*



*and every morphism of this form is an epimorphism. Moreover, every epimorphism in  $\mathbf{G}(\mathbf{A})$  is split with its pseudo-inverse as a section.*

**PROOF.** By Remark II.1.28 it suffices to show the claim for monomorphisms. So given a monomorphism  $\mu$  in  $\mathbf{G}(\mathbf{A})$ , we represent it as a 4-arrow:



Let  $\epsilon : A \twoheadrightarrow \text{coker}(\alpha)$  denote the cokernel projection of  $\alpha$ . We compute

$$\begin{aligned} [\alpha]^{-1} \circ [\epsilon]^{-1} &= [\epsilon \circ \alpha]^{-1} \\ &= [0]^{-1} \\ &= [0 \circ \alpha]^{-1} = [\alpha]^{-1} \circ [0]^{-1}. \end{aligned}$$

From this we conclude

$$\begin{aligned} \mu \circ [\epsilon]^{-1} &= [\delta] \circ [\gamma]^{-1} \circ [\beta] \circ ([\alpha]^{-1} \circ [\epsilon]^{-1}) \\ &= [\delta] \circ [\gamma]^{-1} \circ [\beta] \circ ([\alpha]^{-1} \circ [0]^{-1}) = \mu \circ [0]^{-1}. \end{aligned}$$

Since  $\mu$  is a monomorphism, it follows that  $\epsilon$  is the zero morphism and thus  $\alpha$  is an isomorphism. So from now on we may assume that  $\mu$  is given by a three arrow

$$\begin{array}{ccc} A & \overset{\overline{\overline{\mu}}}{\dashrightarrow} & B \\ \beta \downarrow & & \uparrow \delta \\ F & \xleftarrow{\gamma} & E \end{array}$$

Let  $\kappa : \ker(\beta) \hookrightarrow A$  denote the kernel embedding of  $\beta$ . A similar computation as above proves that  $\kappa$  is the zero morphism and thus  $\beta$  an isomorphism. Thus, every monomorphism can be represented as a span of the form

$$\begin{array}{ccc} A & \overset{\dashrightarrow}{\phantom{\alpha}} & B \\ \alpha \swarrow & & \searrow \beta \\ & C & \end{array}$$

Conversely, given a monomorphism  $\beta : C \hookrightarrow B$ , the morphism  $[\beta]$  is a split monomorphism since

$$[\beta]^{-1} \circ [\beta] = [\text{id}_C] \circ [\text{id}_C]^{-1} = \text{id}_C$$

by the pullback computation rule and the fact that the pullback of two equal monomorphisms is given by identity morphisms. By the dual argumentation, given an epimorphism  $\alpha : C \twoheadrightarrow A$ ,  $[\alpha]$  is a split epimorphism. Since  $[-]^{-1}$  maps split epimorphisms to split monomorphisms, the claim follows.  $\square$

**Corollary 1.53.** *The generalized morphism category has a universal epi-mono factorization, i.e., an essentially unique epi-mono factorization (see Definition 1.1.34).*

PROOF. The 4-arrow data structure of a generalized morphism  $\gamma : A \dashrightarrow B$  is an epi-mono factorization. Now, given any other epi-mono factorization

$$A \dashrightarrow F \hookrightarrow B$$

Theorem II.1.52 shows that the epimorphism and the monomorphism in this factorization can themselves be decomposed as

$$\begin{array}{ccccc} A & \dashrightarrow & F & \hookrightarrow & B \\ \swarrow & & \nearrow & \swarrow & \nearrow \\ & D & & E & \end{array}$$

But this is a 4-arrow representation of  $\gamma$ . Thus, the claim follows from the uniqueness of the 4-arrow representation (Corollary II.1.45).  $\square$

Due to Lemma I.1.36, we are now able to work with **images and coimages of generalized morphisms**.

Here is another nice fact about diagrams involving epimorphisms and monomorphisms in  $\mathbf{A}$ , which is helpful in diagram chases (for example in the proof of the Snake Lemma II.2.1).

**Corollary 1.54.** *For every commutative diagram in  $\mathbf{A}$  of the form*

$$\begin{array}{ccc} A & & \\ \gamma \downarrow & \searrow \beta & \\ B & \xrightarrow{\alpha} & C \end{array}$$

where  $\alpha$  is a monomorphism, the identity

$$[\gamma] = [\alpha]^{-1} \circ [\beta]$$

holds in  $\mathbf{G}(\mathbf{A})$ . Dually, for every commutative diagram in  $\mathbf{A}$  of the form

$$\begin{array}{ccc} A & & \\ \gamma \uparrow & \swarrow \beta & \\ B & \xleftarrow{\alpha} & C \end{array}$$

where  $\alpha$  is an epimorphism, the identity

$$[\gamma] = [\beta] \circ [\alpha]^{-1}$$

holds in  $\mathbf{G}(\mathbf{A})$ .

**PROOF.** Due to Theorem II.1.52, in the first case  $[\alpha]^{-1}$  is a retraction which we apply to  $[\beta] = [\alpha] \circ [\gamma]$ , and in the second case,  $[\alpha]^{-1}$  is a section which we apply to  $[\beta] = [\gamma] \circ [\alpha]$ .  $\square$

## 1.6. Attributes and Properties of Generalized Morphisms.

1.6.1. *Canonical Objects in the Underlying Abelian Category.* In Subsection II.1.1 we saw that the notions *defect* and *domain* are indispensable for reasoning about the generalized morphisms that we construct via diagram chases. In this subsection we can finally define these notions in the context of an arbitrary abelian category  $\mathbf{A}$ , using the fact that every generalized morphism has an image and a coimage (see Subsection II.1.5).

The image of a generalized morphism  $\gamma : A \dashrightarrow B$  is a subobject of  $A$  (where we regard  $A$  as an object in  $\mathbf{G}(\mathbf{A})$ ). Such a subobject corresponds to a subquotient of  $A$  (where we regard  $A$  as an object in the underlying abelian category  $\mathbf{A}$ ).

**Definition 1.55.** Let  $A \in \mathbf{A}$ . A **subquotient of  $A$**  is defined as a subobject of  $A$  in  $\mathbf{G}(\mathbf{A})$ . We say two subquotients are **equal as subquotients** if they are equal as subobjects in  $\mathbf{G}(\mathbf{A})$ .

REMARK 1.56. Since the functor  $(-)^{-1}$  maps monomorphisms to epimorphisms and vice versa, we could have defined a subquotient of  $A$  equivalently as a quotient object of  $A$  in  $\mathbf{G}(\mathbf{A})$ .

REMARK 1.57. Every pair of objects  $(A'', A')$  in  $\mathbf{A}$  such that  $A'' \subseteq A' \subseteq A$  defines a subquotient

$$\text{emb}\left(\frac{A'}{A''}\right) := \begin{array}{ccc} & A'/A'' & \text{-----} \rightarrow A \\ & \swarrow & \nearrow \\ & A' & \end{array}$$

which we also call a **subquotient embedding**. Conversely, every subquotient

$$\begin{array}{ccc} B & \text{-----} \rightarrow & A \\ & \swarrow \alpha & \nearrow \\ & A' & \end{array}$$

defines such a pair  $(\ker(\alpha), A')$ . These two conversions are mutual inverses with respect to the notions of *equality as subquotients* and *being isomorphic as objects in the category*  $\sum_{A'', A' \in \mathbf{C}} \text{Hom}_{\mathbf{A}}(A'', A') \times \text{Hom}_{\mathbf{A}}(A', A)$ .

So from now on, we also call pairs  $(A'', A')$  as described in Remark II.1.57 a subquotient of  $A$  and, by abuse of notation, we also denote this subquotient simply by  $\frac{A'}{A''}$ .

REMARK 1.58. We call the pseudo-inverse of a subquotient embedding a **subquotient projection**. Given a subquotient  $\frac{A'}{A''}$  of  $A$ , we set

$$\text{proj}\left(\frac{A'}{A''}\right) := \text{emb}\left(\frac{A'}{A''}\right)^{-1} : A \text{ -----} \twoheadrightarrow A'/A''$$

Finally, we can define the canonical objects for reasoning about the generalized morphisms constructed by diagram chases.

**Definition 1.59.** Given a generalized morphism  $\gamma : A \dashrightarrow B$ , its **image** is a subquotient embedding

$$\frac{B'}{B''} \hookrightarrow \text{im}(\gamma) \text{ -----} \rightarrow B$$

and its **coimage** is a subquotient projection

$$A \text{ -----} \twoheadrightarrow \text{coim}(\gamma) \frac{A'}{A''}.$$

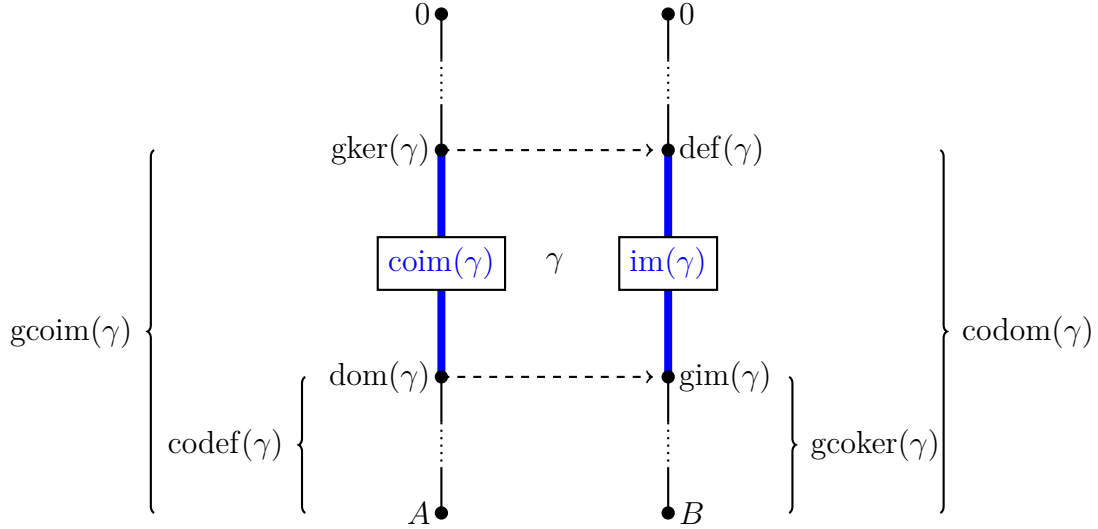
We introduce the following *canonical objects*:

- The **domain of**  $\gamma$  is defined as the subobject  $A' \subseteq A$  and denoted by  $\text{dom}(\gamma)$ .
- The **codomain of**  $\gamma$  is defined as the quotient object  $B/B''$  and denoted by  $\text{codom}(\gamma)$ .

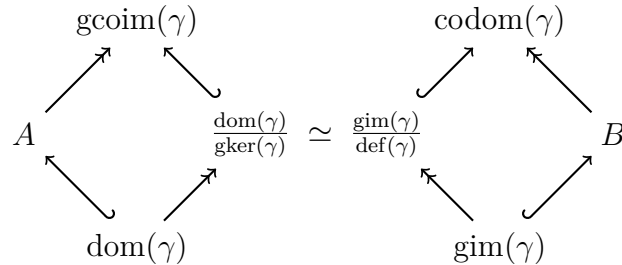
- The **generalized kernel** of  $\gamma$  is defined as the subobject  $A'' \subseteq A$  and denoted by  $\text{gker}(\gamma)$ .
- The **generalized cokernel** of  $\gamma$  is defined as the quotient object  $B/B'$  and denoted by  $\text{gcoker}(\gamma)$ .
- The **codefect** of  $\gamma$  is defined as the quotient object  $A/A'$  and denoted by  $\text{codef}(\gamma)$ .
- The **defect** of  $\gamma$  is defined as the subobject  $B'' \subseteq B$  and denoted by  $\text{def}(\gamma)$ .
- The **generalized image** of  $\gamma$  is defined as the subobject  $B' \subseteq B$  and denoted by  $\text{gim}(\gamma)$ .
- The **generalized coimage** of  $\gamma$  is defined as the quotient object  $A/A''$  and denoted by  $\text{gcoim}(\gamma)$ .

The subobjects and quotient objects of Definition II.1.59 can be nicely depicted using a Hasse diagram. An explanation of Hasse diagrams is given in [Bar09b].

FIGURE 3. A Hasse diagram depicting subobjects and quotient objects associated to a generalized morphism  $\gamma$ .



REMARK 1.60. Using the notions of Definition II.1.59, we can write the epi-mono factorization of a generalized morphism  $\gamma : A \dashrightarrow B$  as



REMARK 1.61. If  $\gamma : A \rightarrow B$  is a morphism in  $\mathbf{A}$ , then  $\text{dom}([\gamma]) = A$ ,  $\text{def}([\gamma]) = 0$ ,  $\text{codom}([\gamma]) = B$ ,  $\text{codef}([\gamma]) = 0$ . In this case, the diagram in Remark II.1.60 degenerates

to the usual epi-mono factorization in an abelian category, from which we can see that the canonical subobjects and quotient objects of  $[\gamma]$  are compatible with those of  $\gamma$ , i.e., we have the equalities

- (1)  $\text{gim}([\gamma]) = \text{im}(\gamma)$ ,
- (2)  $\text{gcoim}([\gamma]) = \text{coim}(\gamma)$ ,
- (3)  $\text{gker}([\gamma]) = \text{ker}(\gamma)$ ,
- (4)  $\text{gcoker}([\gamma]) = \text{coker}(\gamma)$ .

Remark II.1.60 suggests that given a generalized morphism, we can compute all the canonical objects by reading them off from its 4-arrow representation or its reversed 4-arrow representation. The next construction shows that there are much simpler constructions.

**Construction 1.62.** Given a generalized morphism  $\gamma : A \dashrightarrow B$  represented by a span  $S = (A \xleftarrow{\alpha} C \xrightarrow{\beta} B)$ , we can construct  $\text{gim}(\gamma)$  as the image embedding of  $\beta$ . We now prove correctness of this construction: Consider an epi-mono factorization

$$\begin{array}{ccc} C & \xrightarrow{(\alpha, \beta)} & A \oplus B. \\ & \searrow & \nearrow \\ & I. & (\zeta, \eta) \end{array}$$

Denote by  $\pi_B : A \oplus B \rightarrow B$  the natural projection. Then

$$\text{im}(\beta) = \text{im}(\pi_B \circ (\alpha, \beta)) = \text{im}(\pi_B \circ (\zeta, \eta)) = \text{im}(\eta).$$

Thus, we may assume that  $S$  is stable (see Lemma II.1.22). But then correctness is clear by the diagram in Remark II.1.60 which can be embedded into a stable diamond.

REMARK 1.63. The generalized morphism category  $\mathbf{G}(\mathbf{A})$  can be equipped with dependent functions computing all the objects described in Definition II.1.59. From Construction II.1.62, we get the generalized image. In Lemma II.1.68, we will see how to construct the other canonical subobjects using the generalized image. In Remark II.1.69, we will see how these constructions translate to constructions for the canonical quotient objects.

REMARK 1.64. If  $\mathbf{A} = R\text{-mod}$  for a ring  $R$ , the canonical subobjects and quotient objects map to the objects defined in II.1.6 via the equivalence of  $\mathbf{G}(\mathbf{A})$  and  $\text{Rel}(R\text{-mod})$  (see Theorem II.1.19). This is easy to see for the generalized image if we employ Construction II.1.62. For the other canonical subobjects, we can use Lemma II.1.68 to rewrite them in terms of the generalized image.

1.6.2. *Honest Morphisms.* In the end of a diagram chase, we have to check whether the generalized morphism that we constructed is *honest*, i.e., if it actually lies in  $\mathbf{A}$ . The defect and the codefect introduced in Definition II.1.59 precisely define the obstructions of a generalized morphism for being honest.

**Definition 1.65.** Let  $\gamma : A \dashrightarrow B$  be a generalized morphism.

- We say  $\gamma$  is **single-valued** or has **full codomain** if  $\text{def}(\gamma)$  equals 0 as subobjects. This is also the case if and only if  $\text{codom}(\gamma)$  equals  $B$  as quotient objects.



- We say  $\gamma$  is **total** or has **full domain** if  $\text{codf}(\gamma)$  equals 0 as quotient objects. This is also the case if and only if  $\text{dom}(\gamma)$  equals  $A$  as subobjects.
- We say  $\gamma$  is **honest** if it is total and single-valued.

REMARK 1.66. Every morphism  $\alpha \in \text{Hom}_{\mathbf{A}}(A, B)$  defines an honest morphism  $[\alpha] \in \text{Hom}_{\mathbf{G}(\mathbf{A})}(A, B)$ . Conversely, every honest generalized morphism from  $A$  to  $B$  represented by  $(D, H, \alpha, \beta, \gamma) \in 3\text{-Arrow}(A, B)$  defines a morphism  $\gamma^{-1} \circ \beta \circ \alpha^{-1} \in \text{Hom}_{\mathbf{A}}(A, B)$  (note that being honest means that  $\alpha$  and  $\gamma$  are invertible in  $\mathbf{A}$ ). These two assignments are mutual inverses with respect to the notions of equality for morphisms in  $\mathbf{A}$  and equality as generalized morphisms. Thus, if we want to test if a generalized morphism  $\gamma$  equals  $[\alpha]$  for a morphism  $\alpha$  in  $\mathbf{A}$ , we have to check whether the defect and codefect of  $\gamma$  equal 0.

Whenever we draw a diagram in  $\mathbf{G}(\mathbf{A})$ , we will use solid arrows ( $\longrightarrow$ ) for *honest* generalized morphisms instead of dashed ones ( $\dashrightarrow$ ).

REMARK 1.67. Given a generalized morphism  $\gamma : A \dashrightarrow B$ , the composition

$$\text{proj}(\text{codom}(\gamma)) \circ \gamma \circ \text{emb}(\text{dom}(\gamma)) : \text{dom}(\gamma) \longrightarrow \text{codom}(\gamma)$$

yields an honest morphism. This can be easily read off from the diagram in Remark II.1.60, using the pullback and pushout computation rule. We call this composition the **associated morphism of  $\gamma$** .

**1.7. Reasoning with the Canonical Objects.** This subsection provides sufficiently many rules for reasoning with the canonical objects (like the defect or codefect) in order to successfully do a diagram chase like in the Snake Lemma II.2.1 or for constructing spectral sequences (see Subsection II.2.3).

We will apply the following general strategy to compute with and to reason about the canonical subobjects and quotient objects:

- (1) Restate a given claim about the canonical subobjects and quotient objects in terms of the generalized image (a typical claim could be the vanishing of the defect or the codefect).
- (2) Apply computation rules for the generalized image for proving the restated claim.

The next lemma enlists how we can rewrite all the canonical subobjects only in terms of the generalized image.

**Lemma 1.68.** *Let  $\gamma : A \dashrightarrow B$  be a generalized morphism. Then we have the following equalities of subobjects:*

- (1)  $\text{def}(\gamma) = \text{gim}(0 \xrightarrow{[0]} A \dashrightarrow B)$ ,
- (2)  $\text{dom}(\gamma) = \text{gim}(\gamma^{-1})$ ,
- (3)  $\text{gker}(\gamma) = \text{gim}(0 \xrightarrow{[0]} B \dashrightarrow A)$ .

PROOF. We start proving the first claim. Let  $\gamma = \mu \circ \epsilon$  be the universal epi-mono factorization of  $\gamma$ . First, we prove that  $\epsilon \circ [0] = [0]$ . By Theorem II.1.52,  $\epsilon : A \dashrightarrow F$  equals

$[\beta] \circ [\alpha]^{-1}$  for a monomorphism  $\alpha$  and an epimorphism  $\beta$  in  $\mathbf{A}$ . Using Corollary II.1.54, we compute  $[\alpha]^{-1} \circ [0] = [0]$  and thus

$$[\epsilon] \circ [0] = [\beta] \circ [\alpha]^{-1} \circ [0] = [\beta] \circ [0] = [\beta \circ 0] = [0].$$

So  $\gamma \circ [0] = \mu \circ [0]$ , where  $\mu$  is represented by  $(B'/B'' \leftarrow B' \hookrightarrow B)$  again due to Theorem II.1.52. By the pullback computation rule, the lower span in the following diagram depicts the composition  $\mu \circ [0]$ :

$$\begin{array}{ccccc}
 & & B'/B'' & & B \\
 & \nearrow & & \nwarrow & \nearrow \\
 0 & & & & B' \\
 & \nwarrow & & \nearrow & \\
 & & B'' & & 
 \end{array}$$

Since  $B'' \subseteq B$  is by definition the defect of  $\gamma$ , the first claim follows.

The second claim can be read off from the diagram in Remark II.1.60. Using the first claim, the third claim is equivalent to  $\text{gker}(\gamma) = \text{def}(\gamma^{-1})$ , which can be read off from the diagram in Remark II.1.60 again.  $\square$

REMARK 1.69. Using the dualization principle stated in Remark II.1.51, we get the following dictionary between the canonical subobjects and quotient objects:

Subobjects	Quotient objects
$\text{gker}(\gamma)$	$\text{gcoker}(\gamma^{\text{op}})$
$\text{dom}(\gamma)$	$\text{codom}(\gamma^{\text{op}})$
$\text{def}(\gamma)$	$\text{codef}(\gamma^{\text{op}})$
$\text{gim}(\gamma)$	$\text{gcoim}(\gamma^{\text{op}})$

Thus, any statement about canonical quotient objects in  $\mathbf{A}$  can be restated in terms of canonical subobjects in  $\mathbf{A}^{\text{op}}$ , which in turn can be restated in terms of the generalized image by Lemma II.1.68.

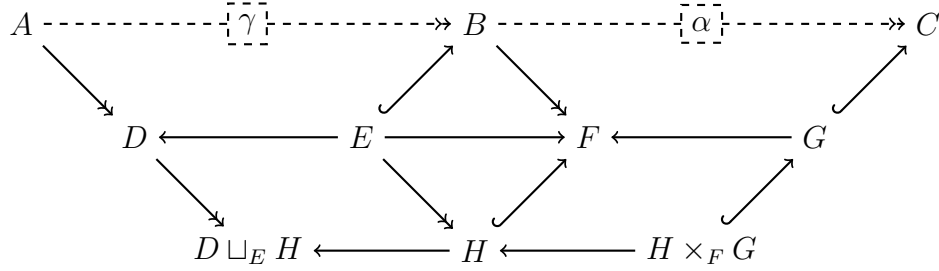
Here is a fairly general computation rule for the generalized image that we are going to employ over and over again in the next section.

**Lemma 1.70.** *Let  $\gamma, \gamma' : A \dashrightarrow B$  be generalized morphisms such that  $\text{gim}(\gamma) \subseteq \text{gim}(\gamma')$ . Then for every generalized morphism  $\alpha : B \dashrightarrow C$ , we have*

$$\text{gim}(\alpha \circ \gamma) \subseteq \text{gim}(\alpha \circ \gamma')$$

*as subobjects of  $C$ . In particular, if  $\text{gim}(\gamma) = \text{gim}(\gamma')$ , then  $\text{gim}(\alpha \circ \gamma) = \text{gim}(\alpha \circ \gamma')$ .*

PROOF. The generalized image can be read off from the monomorphism in an reversed 3-arrow representation. The composition of two reversed 3-arrows  $\gamma = (D, E, A \twoheadrightarrow D, E \rightarrow D, E \hookrightarrow B)$  and  $\alpha = (F, G, B \twoheadrightarrow F, G \rightarrow F, G \hookrightarrow C)$  can be depicted in a commutative diagram



where we can read off the resulting reversed 3-arrow from the lower morphisms. The morphisms  $E \twoheadrightarrow H$  and  $H \hookrightarrow F$  are an epi-mono factorization of  $E \rightarrow F$ . We see that  $\text{gim}(\alpha \circ \gamma) = (H \times_F G \hookrightarrow G \hookrightarrow C)$  only depends on  $\alpha$  and  $(E \hookrightarrow B) = \text{gim}(\gamma)$ . Furthermore, the association

$$(E \hookrightarrow B) \mapsto (H \times_F G \hookrightarrow G \hookrightarrow C)$$

between subobjects of  $B$  and  $C$  is functorial since all steps involved in its construction are functorial. Thus,  $\text{gim}(\gamma) \subseteq \text{gim}(\gamma')$  implies  $\text{gim}(\alpha \circ \gamma) \subseteq \text{gim}(\alpha \circ \gamma')$ , which is the claim.  $\square$

We conclude this section with an example of our strategy to reason about canonical subobjects: We restate and prove a claim about the defect in terms of the generalized image. Furthermore, this lemma simplifies the proof of the Snake Lemma II.2.1.

**Lemma 1.71.** *Let  $\gamma : A \dashrightarrow B$  be a generalized morphism. For every single-valued generalized morphism  $\sigma : C \dashrightarrow A$ , we have*

$$\text{def}(\gamma) = \text{def}(\gamma \circ \sigma)$$

as subobjects of  $C$ .

PROOF. Using Lemma II.1.68, we restate the claim as

$$\text{gim}(\gamma \circ [0]) = \text{gim}(\gamma \circ \sigma \circ [0]).$$

Due to Lemma II.1.70, it suffices to show

$$\text{gim}([0]) = \text{gim}(\sigma \circ [0]).$$

But this is true since

$$\text{gim}(\sigma \circ [0]) = \text{def}(\sigma) = 0 = \text{im}(0) = \text{gim}([0]),$$

where we again use Lemma II.1.68 and  $\sigma$  being single-valued.  $\square$

We summarize the computation rules of this subsection in the following theorem.

**Theorem 1.72.** *Let  $\gamma : A \dashrightarrow B$  be a generalized morphism.*

(1) **Rewriting canonical subobjects:**

$$(a) \text{def}(\gamma) = \text{gim}(0 \xrightarrow{[0]} A \dashrightarrow B),$$

$$(b) \text{dom}(\gamma) = \text{gim}(\gamma^{-1}),$$

$$(c) \text{gker}(\gamma) = \text{gim}(0 \xrightarrow{[0]} B \dashrightarrow A).$$

(2) **Computation rule for the generalized image:** Let  $\gamma' : A \dashrightarrow B$  be another generalized morphism such that  $\text{gim}(\gamma) \subseteq \text{gim}(\gamma')$ . Then for every generalized morphism  $\alpha : B \dashrightarrow C$ , we have

$$\text{gim}(\alpha \circ \gamma) \subseteq \text{gim}(\alpha \circ \gamma').$$

(3) **Computation rule for the defect:** For every single-valued generalized morphism  $\sigma : C \dashrightarrow A$ , we have

$$\text{def}(\gamma) = \text{def}(\gamma \circ \sigma).$$

PROOF. Lemma [II.1.68](#), [II.1.70](#), and [II.1.71](#). □

## 2. Diagram Chases and Spectral Sequences

**2.1. Constructive Diagram Chases.** Diagram chases in homological algebra are used for proving categorical theorems concerning properties and the existence of morphisms situated in diagrams of prescribed shape. Due to the theory of generalized morphisms, we are now able to perform diagram chases constructively, which means that we end up with explicit formulas for the morphisms whose existence is claimed. Furthermore, the reasoning techniques for generalized morphisms presented in Subsection II.1.7 are handy for proving correctness of our constructions. We illustrate such a constructive diagram chase with a refinement of the famous Snake Lemma.

**Lemma 2.1.** *Given the following (not necessarily commutative) diagram in an abelian category  $\mathbf{A}$ :*

$$\begin{array}{ccccccc}
 & & & & \ker(\gamma) & & \\
 & & & & \downarrow \eta := \text{KernelEmbedding}(\gamma) & & \\
 & & A & \xrightarrow{\delta} & B & \xrightarrow{\epsilon} & C \longrightarrow 0 \\
 & & \downarrow \alpha & & \downarrow \beta & & \downarrow \gamma \\
 0 & \longrightarrow & A' & \xrightarrow{\iota} & B' & \xrightarrow{\nu} & C' \\
 & & \downarrow \zeta := \text{CokernelProjection}(\alpha) & & & & \\
 & & \text{coker}(\alpha) & & & & 
 \end{array}$$

We define the generalized morphism

$$\sigma := [\zeta] \circ [\iota]^{-1} \circ [\beta] \circ [\epsilon]^{-1} \circ [\eta].$$

Then  $\sigma$  is single-valued if

- (1) the diagram is horizontally exact at  $A'$  and  $B$ ,
- (2) the left square is commutative, i.e.,  $\beta \circ \delta = \iota \circ \alpha$ .

Dually,  $\sigma$  is total if

- (1) the diagram is horizontally exact at  $C$  and  $B'$ ,
- (2) the right square is commutative, i.e.,  $\gamma \circ \epsilon = \nu \circ \beta$ .

In particular,  $\sigma$  is honest if the diagram is commutative and horizontally exact.

**PROOF.** We show that  $\sigma$  is single-valued under the given assumptions. By definition, being single-valued means that  $\text{def}(\sigma) = 0$ . We have

$$\text{def}(\sigma) = \text{def}([\zeta] \circ [\iota]^{-1} \circ [\beta] \circ [\epsilon]^{-1} \circ [\eta]) = \text{def}([\zeta] \circ [\iota]^{-1} \circ [\beta] \circ [\epsilon]^{-1})$$

since  $[\eta]$  is single-valued (Lemma II.1.71). Let  $0 : 0 \rightarrow C$  be the zero morphism. By Lemma II.1.68, we can compute the defect as a generalized image:

$$\text{def} \left( [\zeta] \circ [\iota]^{-1} \circ [\beta] \circ [\epsilon]^{-1} \right) = \text{gim} \left( [\zeta] \circ [\iota]^{-1} \circ [\beta] \circ [\epsilon]^{-1} \circ [0] \right).$$

We have

$$\text{gim}([\epsilon]^{-1} \circ [0]) = \text{gker}([\epsilon])$$

by Lemma II.1.68, and

$$\text{gker}([\epsilon]) = \ker(\epsilon) = \text{im}(\delta) = \text{gim}([\delta])$$

by Remark II.1.61 and the horizontal exactness at  $B$ . Using Lemma II.1.70, we conclude

$$\text{gim} \left( [\zeta] \circ [\iota]^{-1} \circ [\beta] \circ [\epsilon]^{-1} \circ [0] \right) = \text{gim} \left( [\zeta] \circ [\iota]^{-1} \circ [\beta] \circ [\delta] \right).$$

Exactness at  $A'$  means that  $\iota$  is a monomorphism. Since the left square commutes, Corollary II.1.54 applies and gives us

$$[\iota]^{-1} \circ [\beta] \circ [\delta] = [\alpha].$$

Thus, we are left with the computation of

$$\text{gim}([\zeta] \circ [\alpha]) = \text{gim}([\zeta \circ \alpha]) = \text{im}(\zeta \circ \alpha) = 0.$$

This proves the claim. By duality (see Remark II.1.51), we also proved that  $\sigma$  is total under the given assumptions.  $\square$

**2.2. Generalized Cochain Complexes.** Using the technology of generalized morphisms, we can now attack the problem given in the introduction of this chapter: To describe, in the context of an arbitrary abelian category  $\mathbf{A}$ , an algorithm for computing spectral sequences which is suitable for a direct computer implementation.

Our spectral sequence algorithm takes as input a descending filtration  $F$  (see Definition II.2.11) and outputs a cohomological spectral sequence  $E$  (see Definition II.2.10). As an intermediate step, our algorithm constructs a collection of so-called *generalized cochain complexes*  $\Delta$ .

A generalized cochain complex is like an ordinary cochain complex, but with generalized morphisms as differentials (Definition II.2.4). Each generalized cochain complex gives rise to an ordinary cochain complex (Theorem II.2.8). It will turn out that the spectral sequence  $E$  associated to  $F$  is nothing but the ordinary cochain complexes associated to the generalized cochain complexes  $\Delta$ .

In this subsection we start with a formal introduction of generalized chain and cochain complexes.

**Definition 2.2.** Let  $\mathbf{C}$  be a category. We define a category  $\mathbf{Gr}_{\mathbb{Z}}(\mathbf{C})$  as follows:

- (1) Objects are  $\mathbb{Z}$ -indexed families  $(A_g)_{g \in \mathbb{Z}}$  of objects  $A_g \in \mathbf{C}$ .
- (2) Morphisms from  $(A_g)_g$  to  $(B_g)_g$  are families  $(\alpha_g : A_g \rightarrow B_g)_{g \in \mathbb{Z}}$  of morphisms in  $\mathbf{C}$ .

Composition and identities are given componentwise. We call this category the **associated  $\mathbb{Z}$ -graded category of  $\mathbf{C}$** . Furthermore, for every  $h \in \mathbb{Z}$ , we define a functor  $[h] : \mathbf{Gr}_{\mathbb{Z}}(\mathbf{A}) \rightarrow \mathbf{Gr}_{\mathbb{Z}}(\mathbf{A})$  by  $(A_g)_{g \in \mathbb{Z}}[h] := (A_{g+h})_{g \in \mathbb{Z}}$  and  $(\alpha_g : A_g \rightarrow B_g)_{g \in \mathbb{Z}}[h] := (\alpha_{g+h} : A_{g+h} \rightarrow B_{g+h})_{g \in \mathbb{Z}}$ .

**Definition 2.3.** Let  $\mathbf{A}$  be an Ab-category.

- (1) The full subcategory of  $\sum_{A_{\bullet} \in \mathbf{Gr}_{\mathbb{Z}}(\mathbf{A})} \text{Hom}_{\mathbf{Gr}_{\mathbb{Z}}(\mathbf{A})}(A_{\bullet}, A_{\bullet}[-1])$  generated by those pairs  $(A_{\bullet}, d_{\bullet})$  such that  $d_{\bullet}^2 = 0$  is called the **category of chain complexes**. Its objects are called **chain complexes** and its morphisms are called **chain maps**. This category is denoted by  $\text{Ch}_{\bullet}(\mathbf{A})$ .
- (2) The full subcategory of  $\sum_{A^{\bullet} \in \mathbf{Gr}_{\mathbb{Z}}(\mathbf{A})} \text{Hom}_{\mathbf{Gr}_{\mathbb{Z}}(\mathbf{A})}(A^{\bullet}, A^{\bullet}[1])$  generated by those pairs  $(A^{\bullet}, d^{\bullet})$  such that  $(d^{\bullet})^2 = 0$  is called the **category of cochain complexes**. Its objects are called **cochain complexes** and its morphisms are called **cochain maps**. This category is denoted by  $\text{Ch}^{\bullet}(\mathbf{A})$ .

**Definition 2.4.** Let  $\mathbf{A}$  be an abelian category. A **generalized chain complex** consists of the following data:

- (1) An object  $A_{\bullet} \in \mathbf{Gr}_{\mathbb{Z}}(\mathbf{G}(\mathbf{A}))$ .
- (2) A morphism  $d_{\bullet} \in \text{Hom}_{\mathbf{Gr}_{\mathbb{Z}}(\mathbf{G}(\mathbf{A}))}(A_{\bullet}, A_{\bullet}[-1])$ , called the **generalized differential**.
- (3) For all  $i \in \mathbb{Z}$ , we have  $\text{gim}(d_{i+1}) \subseteq \text{gker}(d_i)$ .

Dually, a **generalized cochain complex** consists of the following data:

- (1) An object  $A^{\bullet} \in \mathbf{Gr}_{\mathbb{Z}}(\mathbf{G}(\mathbf{A}))$ .
- (2) A morphism  $d^{\bullet} \in \text{Hom}_{\mathbf{Gr}_{\mathbb{Z}}(\mathbf{G}(\mathbf{A}))}(A^{\bullet}, A^{\bullet}[1])$ , called the **generalized differential**.
- (3) For all  $i \in \mathbb{Z}$ , we have  $\text{gim}(d^{i-1}) \subseteq \text{gker}(d^i)$ .

**Definition 2.5.** Let  $A_{\bullet}$  be a generalized chain complex with generalized differential  $d_{\bullet}$  and  $i \in \mathbb{Z}$ . We define its  **$i$ -th homology** as

$$H_i(A_{\bullet}) := \frac{\text{gker}(d_i)}{\text{gim}(d_{i+1})}.$$

Dually, let  $A^{\bullet}$  be a generalized cochain complex with generalized differential  $d^{\bullet}$  and  $i \in \mathbb{Z}$ . We define its  **$i$ -th cohomology** as

$$H^i(A^{\bullet}) := \frac{\text{gker}(d^i)}{\text{gim}(d^{i-1})}.$$

**REMARK 2.6.** We depict a generalized cochain complex  $(A^{\bullet}, d^{\bullet})$  as follows:

$$\cdots \dashrightarrow A^{i-1} \dashrightarrow^{d^{i-1}} A^i \dashrightarrow^{d^i} A^{i+1} \dashrightarrow \cdots$$

Our goal is to assign to each generalized cochain complex  $(A^{\bullet}, d^{\bullet})$  an ordinary cochain complex. Since

$$\text{def}(d^{i-1}) \subseteq \text{gim}(d^{i-1}) \subseteq \text{gker}(d^i) \subseteq \text{dom}(d^i) \subseteq A^i,$$

we obtain a subquotient embedding

$$\frac{\text{dom}(d^i)}{\text{def}(d^{i-1})} \xleftarrow{\text{emb}^i} A^i$$

and its corresponding subquotient projection

$$A^i \dashrightarrow \frac{\text{dom}(d^i)}{\text{def}(d^{i-1})}.$$

**Lemma 2.7.** *Let  $(A^\bullet, d^\bullet)$  be a generalized cochain complex. Then*

$$\delta^i := \frac{\text{dom}(d^i)}{\text{def}(d^{i-1})} \xleftarrow{\text{emb}^i} A^i \dashrightarrow^{d^i} A^{i+1} \dashrightarrow^{\text{proj}^{i+1}} \frac{\text{dom}(d^{i+1})}{\text{def}(d^i)}$$

is an honest morphism for all  $i \in \mathbb{Z}$ .

PROOF. By Remark II.1.67, the composition

$$\alpha := \text{dom}(d^i) \xleftarrow{\iota} A^i \dashrightarrow^{d^i} A^{i+1} \xrightarrow{\epsilon} \frac{A^{i+1}}{\text{def}(d^i)}$$

is honest. It remains to prove that  $\alpha$  restricts properly. First, we compute

$$\begin{aligned} \text{im}(\alpha) &= \text{gim}(\epsilon \circ d^i \circ \iota) && \text{(Remark II.1.61)} \\ &\subseteq \text{gim}(\epsilon \circ d^i) && \text{(Lemma II.1.70)} \\ &\subseteq \text{gim}(\epsilon \circ \text{emb}(\text{gker}(d^{i+1}))) && (\text{gim}(d^i) \subseteq \text{gker}(d^{i+1})) \\ &= \frac{\text{gker}(d^{i+1})}{\text{def}(d^i)} \subseteq \frac{\text{dom}(d^{i+1})}{\text{def}(d^i)}. \end{aligned}$$

Second, we compute

$$\begin{aligned} \alpha(\text{def}(d^{i-1})) &= \text{gim}(\epsilon \circ d^i \circ \text{emb}(\text{def}(d^{i-1}))) \\ &\subseteq \text{gim}(\epsilon \circ d^i \circ \text{emb}(\text{gker}(d^i))) && (\text{def}(d^{i-1}) \subseteq \text{gker}(d^i)) \\ &= \text{gim}(\epsilon \circ \text{emb}(\text{def}(d^i))) && \text{(Lemma II.1.70)} \\ &= 0. \end{aligned} \quad \square$$

The following theorem is used for a crucial step in the construction of the spectral sequence associated to a descending filtration (see Construction II.2.17).

**Theorem 2.8.** *The  $\delta^i$  defined in Lemma II.2.7 give rise to a cochain complex*

$$\cdots \longrightarrow \frac{\text{dom}(d^{i-1})}{\text{def}(d^{i-2})} \xrightarrow{\delta^{i-1}} \frac{\text{dom}(d^i)}{\text{def}(d^{i-1})} \xrightarrow{\delta^i} \frac{\text{dom}(d^{i+1})}{\text{def}(d^i)} \longrightarrow \cdots$$

in  $\text{Ch}^\bullet(\mathbf{A})$ , which we call the *associated honest cochain complex of  $(A^\bullet, d^\bullet)$* .



PROOF. By Remark II.1.61, it suffices to prove  $\text{gim}([\delta^{i-1}]) \subseteq \text{gker}([\delta^i])$ . Since  $(A^\bullet, d^\bullet)$  is a generalized complex, and due to Lemma II.1.68, we have

$$(\dagger) \quad \text{gim}(d^{i-1}) \subseteq \text{gker}(d^i) = \text{gim}((d^i)^{-1} \circ [0]).$$

Now, we compute

$$\begin{aligned} \text{gim}([\delta^{i-1}]) &= \text{gim}(\text{proj}^i \circ d^{i-1} \circ \text{emb}^{i-1}) \\ &\subseteq \text{gim}(\text{proj}^i \circ d^{i-1}) && \text{(Lemma II.1.70)} \\ &\subseteq \text{gim}(\text{proj}^i \circ (d^i)^{-1} \circ [0]) && (\dagger) \\ &\subseteq \text{gim}(\text{proj}^i \circ (d^i)^{-1} \circ \text{emb}^{i+1} \circ [0]) && \text{(Lemma II.1.70)} \\ &= \text{gker}([\delta^i]). && \text{(Lemma II.1.68)} \end{aligned}$$

□

REMARK 2.9. Taking cohomology commutes with taking the associated cochain complex, since for all generalized cochain complexes  $(A^\bullet, d^\bullet)$ , we have

$$\text{def}(d^{i-1}) \subseteq \text{gim}(d^{i-1}) \subseteq \text{gker}(d^i) \subseteq \text{dom}(d^i) \subseteq A^i.$$

**2.3. Spectral Sequence of a Filtered Complex.** As described in the introduction of Subsection II.2.2, our spectral sequence algorithm constructs as an intermediate step a collection of generalized cochain complexes  $\Delta$  associated to a descending filtration  $F$ . In this subsection we will see how this construction is carried out (Definition II.2.14) and why it furthermore gives rise to a spectral sequence (Corollary II.2.18).

We recall the definition of the category of spectral sequences, which can be found in [Wei94].

**Definition 2.10.** Let  $\mathbf{A}$  be an abelian category. A **cohomological spectral sequence** starting at  $a \in \mathbb{Z}$  consists of the following data: For all  $p, q \in \mathbb{Z}$ ,  $r \geq a$ , we have

- (1) objects  $E_r^{p,q} \in \mathbf{A}$ ,
- (2) morphisms  $\partial_r^{p,q} : E_r^{p,q} \rightarrow E_r^{p+r, q-(r-1)} \in \mathbf{A}$ ,
- (3) isomorphisms  $\iota_r^{p,q} : E_{r+1}^{p,q} \xrightarrow{\sim} \frac{\ker(\partial_r^{p,q})}{\text{im}(\partial_r^{p-r, q+(r-1)})}$ ,
- (4) the equation  $\partial_r^{p+r, q-(r-1)} \circ \partial_r^{p,q} = 0$  holds.

The isomorphisms  $\iota_r^{p,q}$  identify  $E_{r+1}^{p,q}$  with a subquotient of  $E_r^{p,q}$ , whose subquotient embedding and corresponding subquotient projection are denoted by

$$E_{r+1}^{p,q} \hookrightarrow \overset{\text{emb}_r^{p,q}}{\dashrightarrow} E_r^{p,q}$$

and

$$E_r^{p,q} \dashrightarrow \overset{\text{proj}_r^{p,q}}{\twoheadrightarrow} E_{r+1}^{p,q}$$

respectively. A morphism between two cohomological spectral sequences  $(E_r^{p,q}, \partial_r^{p,q})$  and  $(E'_r{}^{p,q}, \partial'_r{}^{p,q})$  starting at  $a$  consists of the following data: For all  $p, q \in \mathbb{Z}$ ,  $r \geq a$ , we have

- (1) morphisms  $f_r^{p,q} : E_r^{p,q} \rightarrow E_r^{p,q}$ ,
- (2) the equation  $\partial_r^{p,q} \circ f_r^{p,q} = f_r^{p+r,q-(r-1)} \circ \partial_r^{p,q}$ ,
- (3) the equation of generalized morphisms

$$[f_{r+1}^{p,q}] = \text{proj}_r^{p,q} \circ [f_r^{p,q}] \circ \text{emb}_r^{p,q}.$$

**Definition 2.11.** Let  $\mathbf{C}$  be a category.

- (1) The full subcategory of  $\sum_{C \in \mathbf{Gr}_{\mathbb{Z}}(\mathbf{C})} \text{Hom}_{\mathbf{Gr}_{\mathbb{Z}}(\mathbf{C})}(C, C[1])$  generated by those pairs  $((C_p)_{p \in \mathbb{Z}}, (\iota_p)_{p \in \mathbb{Z}})$  such that  $\iota_p$  is a monomorphism for all  $p \in \mathbb{Z}$  is called the **category of ascending filtrations** and denoted by  $\text{Filt}_{\bullet}(\mathbf{C})$ . We set  $F_p C := C_p$ . An object in  $\text{Filt}_{\bullet}(\mathbf{C})$  can be depicted by a chain of successive subobjects

$$\cdots \subseteq F_{p-1} C \subseteq F_p C \subseteq F_{p+1} C \subseteq \cdots$$

- (2) The full subcategory of  $\sum_{C \in \mathbf{Gr}_{\mathbb{Z}}(\mathbf{C})} \text{Hom}_{\mathbf{Gr}_{\mathbb{Z}}(\mathbf{C})}(C, C[-1])$  generated by those pairs  $((C^p)_{p \in \mathbb{Z}}, (\iota^p)_{p \in \mathbb{Z}})$  such that  $\iota^p$  is a monomorphism for all  $p \in \mathbb{Z}$  is called the **category of descending filtrations** and denoted by  $\text{Filt}^{\bullet}(\mathbf{C})$ . We set  $F^p C := C^p$ . An object in  $\text{Filt}^{\bullet}(\mathbf{C})$  can be depicted by a chain of successive subobjects

$$\cdots \supseteq F^{p-1} C \supseteq F^p C \supseteq F^{p+1} C \supseteq \cdots$$

REMARK 2.12. Note that we defined the category of filtrations, and not the category of filtered objects. This means that we do not consider a superobject  $S$  as part of our data, i.e., an object such that  $F^p C \subseteq S$  for all  $p \in \mathbb{Z}$ . The reason is that such a superobject does not play a role for the associated spectral sequence.

REMARK 2.13. If  $\mathbf{A}$  is an additive category, then so is  $\text{Filt}^{\bullet}(\mathbf{A})$ . Furthermore, we have an equivalence of categories

$$\text{Filt}^{\bullet}(\text{Ch}^{\bullet}(\mathbf{A})) \simeq \text{Ch}^{\bullet}(\text{Filt}^{\bullet}(\mathbf{A}))$$

given by interpreting the diagram

$$\begin{array}{ccccccc} & & \vdots & & \vdots & & \\ & & \downarrow & & \downarrow & & \\ \cdots & \longrightarrow & F^{p+1} A^i & \longrightarrow & F^{p+1} A^{i+1} & \longrightarrow & \cdots \\ & & \downarrow & & \downarrow & & \\ \cdots & \longrightarrow & F^p A^i & \longrightarrow & F^p A^{i+1} & \longrightarrow & \cdots \\ & & \downarrow & & \downarrow & & \\ & & \vdots & & \vdots & & \end{array}$$

row-wise or column-wise. We simply write  $(F^{\bullet} A^{\bullet}, d^{\bullet, \bullet})$  for objects in one of these categories.

From the collection  $\Delta$  of generalized differentials in the next definition, we will later be able to read off the spectral sequence of a descending filtration.

**Definition 2.14.** Let  $\mathbf{A}$  be an abelian category,  $(F^\bullet A^\bullet, d^{\bullet, \bullet})$  be a descending filtration in  $\text{Filt}^\bullet(\text{Ch}^\bullet(\mathbf{A}))$ . For  $r, p, i \in \mathbb{Z}$ , the generalized differentials  $\Delta_r^{p,i}$  are defined by the composition

$$\frac{F^p A^i}{F^{p+1} A^i} \hookrightarrow \text{emb}^{p,i} \dashrightarrow F^s A^i \xrightarrow{d^{s,i}} F^s A^{i+1} \dashrightarrow \text{proj}^{p+r,i+1} \dashrightarrow \frac{F^{p+r} A^{i+1}}{F^{p+r+1} A^{i+1}},$$

where  $s$  is any index smaller or equal to  $p$ , and  $\text{emb}^{p,i}$  denotes the subquotient embedding,  $\text{proj}^{p+r,i+1}$  the subquotient projection.

REMARK 2.15. Since

$$\begin{array}{ccc} F^s A^i & \xrightarrow{d^{s,i}} & F^s A^{i+1} \\ \downarrow & & \downarrow \\ F^{s-1} A^i & \xrightarrow{d^{s-1,i}} & F^s A^i \end{array}$$

commutes, Corollary II.1.54 implies that the definition of  $\Delta_r^{p,i}$  is independent of the choice of  $s$ .

The following lemma is the key to the construction of spectral sequences associated to filtered cochain complexes. First, it proves that the generalized differentials  $\Delta_r^{p,i}$  give rise to generalized cochain complexes. More importantly, it states that the cohomology groups of the generalized cochain complexes  $\Delta_r$  (which are of the form  $\frac{\text{gker}}{\text{gim}}$ ) are equal to the objects in the associated honest cochain complexes of  $\Delta_{r+1}$  (which are of the form  $\frac{\text{dom}}{\text{def}}$ ).

**Lemma 2.16.** *Let  $i \in \mathbb{Z}$ . Then the following holds:*

- (1)  $\text{gim}(\Delta_r^{p,i}) \subseteq \text{gker}(\Delta_r^{p+r,i+1})$ ,
- (2)  $\text{gim}(\Delta_r^{p,i}) = \text{def}(\Delta_{r+1}^{p-1,i})$ ,
- (3)  $\text{gker}(\Delta_r^{p,i}) = \text{dom}(\Delta_{r+1}^{p,i})$ .

PROOF. For sufficiently small  $s$ , we compute the four canonical subobjects of the generalized morphism  $\Delta_r^{p,i}$ . For our computation, we will use the subquotient embeddings

$$\frac{F^p A^i}{F^{p+1} A^i} \hookrightarrow \text{emb}^{p,i} \dashrightarrow F^s A^i$$

and

$$\frac{F^{p+r} A^{i+1}}{F^{p+r+1} A^{i+1}} \hookrightarrow \text{emb}^{p+r,i+1} \dashrightarrow F^s A^{i+1}$$

and their corresponding subquotient projections  $\text{proj}^{p,i}$  and  $\text{proj}^{p+r,i+1}$ , respectively.

**Generalized image:**

$$\begin{aligned} \text{gim}(\Delta_r^{p,i}) &= \text{gim} \left( \text{proj}^{p+r,i+1} \circ [d^{s,i}] \circ \text{emb}^{p,i} \right) \\ &= \text{gim} \left( \text{proj}^{p+r,i+1} \circ (d^{s,i}(F^p A^i) \hookrightarrow F^s A^{i+1}) \right). \end{aligned} \quad (\text{Lemma II.1.70})$$

**Defect:**

$$\begin{aligned} \text{def}(\Delta_r^{p,i}) &= \text{gim}(\Delta_r^{p,i} \circ [0]) \quad (\text{Lemma II.1.68}) \\ &= \text{gim} \left( \text{proj}^{p+r,i+1} \circ [d^{s,i}] \circ \text{emb}^{p,i} \circ [0] \right) \\ &= \text{gim} \left( \text{proj}^{p+r,i+1} \circ (d^{s,i}(F^{p+1} A^i) \hookrightarrow F^s A^{i+1}) \right). \end{aligned} \quad (\text{Lemma II.1.70})$$

**Generalized kernel:**

$$\begin{aligned} \text{gker}(\Delta_r^{p,i}) &= \text{gim} \left( (\Delta_r^{p,i})^{-1} \circ [0] \right) \quad (\text{Lemma II.1.68}) \\ &= \text{gim} \left( \text{proj}^{p,i} \circ [d^{s,i}]^{-1} \circ \text{emb}^{p+r,i+1} \circ [0] \right) \\ &= \text{gim} \left( \text{proj}^{p,i} \circ [d^{s,i}]^{-1} \circ (F^{p+r+1} A^{i+1} \hookrightarrow F^s A^{i+1}) \right). \end{aligned} \quad (\text{Lemma II.1.70})$$

**Domain:**

$$\begin{aligned} \text{dom}(\Delta_r^{p,i}) &= \text{gim} \left( (\Delta_r^{p,i})^{-1} \right) \quad (\text{Lemma II.1.68}) \\ &= \text{gim} \left( \text{proj}^{p,i} \circ [d^{s,i}]^{-1} \circ \text{emb}^{p+r,i+1} \right) \\ &= \text{gim} \left( \text{proj}^{p,i} \circ [d^{s,i}]^{-1} \circ (F^{p+r} A^{i+1} \hookrightarrow F^s A^{i+1}) \right). \end{aligned} \quad (\text{Lemma II.1.70})$$

Now, we can read off the second and the third claim simply by substituting the indices. For the first claim, note that

$$(\dagger) \quad d^{s,i}(F^p A^i) \subseteq \ker(d^{s,i+1}) = (d^{s,i+1})^{-1}(0) \subseteq (d^{s,i+1})^{-1}(F^{p+2r+1} A^{i+2})$$

as subobjects of  $F^s A^{i+1}$ . We compute

$$\begin{aligned} \text{gim}(\Delta_r^{p,i}) &= \text{gim} \left( \text{proj}^{p+r,i+1} \circ (d^{s,i}(F^p A^i) \hookrightarrow F^s A^{i+1}) \right) \\ &\subseteq \text{gim} \left( \text{proj}^{p+r,i+1} \circ [d^{s,i+1}]^{-1} \circ (F^{p+2r+1} A^{i+2} \hookrightarrow F^s A^{i+2}) \right) \quad (\dagger) \\ &= \text{gker}(\Delta_r^{p+r,i+1}). \quad \square \end{aligned}$$

**Construction 2.17.** Now, we can construct the spectral sequence associated to a descending filtration  $(F^\bullet A^\bullet, d^{\bullet,\bullet})$ . First, we define

$$E_0^{p,q} := E_0^{p,q}(F^\bullet A^\bullet) := \frac{F^p A^{p+q}}{F^{p+1} A^{p+q}}.$$

For each triple  $p, q, r \in \mathbb{Z}$ , the generalized differentials  $\Delta_r^{p+\bullet, p+q+\bullet}$  (see Definition II.2.14) give rise to a generalized cochain complex

$$\dots \dashrightarrow E_0^{p,q} \dashrightarrow E_0^{p+r, q-(r-1)} \dashrightarrow E_0^{p+2r, q-2(r-1)} \dashrightarrow \dots$$

since  $\text{gim}(\Delta_r^{p,p+q}) \subseteq \text{gker}(\Delta_r^{p+r,p+q+1})$  holds by the Key Lemma II.2.16. Since  $d^\bullet$  respects the filtration,  $\Delta_0^{p,p+q}$  is honest and thus equal to  $[\partial_0^{p,q}]$  for a morphism  $\partial_0^{p,q}$  in  $\mathbf{A}$ . Note that the indices of  $\Delta_0^{p,\bullet}$  and  $\partial_0^{p,\bullet}$  differ by a shift. For  $r \geq 1$ , we take the associated honest cochain complex of  $\Delta_r^{p+\bullet,r,p+q+\bullet}$  (see Theorem II.2.8), which gives us:

$$\dots \longrightarrow E_r^{p,q} \xrightarrow{\partial_r^{p,q}} E_r^{p+r,q-(r-1)} \xrightarrow{\partial_r^{p+r,q-(r-1)}} E_r^{p+2r,q-2(r-1)} \longrightarrow \dots$$

**Corollary 2.18.** *The collection of objects  $E_r^{p,q}$  and morphisms  $\partial_r^{p,q}$  for  $p, q \in \mathbb{Z}, r \geq 0$  are the data of a cohomological spectral sequence.*

PROOF. Due to Remark II.2.9, the cohomology of the sequence  $\partial_r^{p+\bullet,r,q-\bullet(r-1)}$  equals the cohomology of the generalized cochain complex  $\Delta_r^{p+\bullet,r,p+q+\bullet}$ . Now, the claim is a direct consequence of the definition of the associated honest cochain complex (Theorem II.2.8) and the Key Lemma II.2.16.  $\square$

**Corollary 2.19.** *The association*

$$(F^\bullet A^\bullet, d^{\bullet,\bullet}) \mapsto (E_r^{p,q}, \partial_r^{p,q})_{p,q \in \mathbb{Z}, r \geq 0}$$

*defines a functor from  $\text{Filt}^\bullet(\text{Ch}^\bullet(\mathbf{A}))$  to the category of cohomological spectral sequences starting at 0.*

PROOF. A morphism  $F^\bullet \alpha^\bullet$  between descending filtered cochain complexes  $(F^\bullet A^\bullet, d^{\bullet,\bullet})$  and  $(F^\bullet B^\bullet, d^{\bullet,\bullet})$  defines honest morphisms  $\bar{\alpha}^{p,q}$  between the graded parts of the filtered complexes:

$$E_0^{p,q}(F^\bullet A^\bullet) \hookrightarrow F^p A^{p+q} \xrightarrow{F^p(\alpha^{p+q})} F^p B^{p+q} \twoheadrightarrow E_0^{p,q}(F^\bullet B^\bullet).$$

We denote by  $\Delta_r^{p,p+q}(A)$  and  $\Delta_r^{p,p+q}(B)$  the generalized differentials of  $(F^\bullet A^\bullet, d^{\bullet,\bullet})$  and  $(F^\bullet B^\bullet, d^{\bullet,\bullet})$ , respectively, which are defined in II.2.14 and whose domains and defects give rise to the spectral sequences (see Corollary II.2.18). Since  $\bar{\alpha}^{p,q}$  respects the domains of  $\Delta_r^{p,p+q}(A)$  and  $\Delta_r^{p,p+q}(B)$ , and since it furthermore respects the defects of  $\Delta_r^{p-r,p+q-1}(A)$  and  $\Delta_r^{p-r,p+q-1}(B)$ , it restricts properly to the subquotients of all higher pages. Thus,  $\bar{\alpha}^{p,q}$  gives rise to a morphism between the spectral sequences.  $\square$

REMARK 2.20. The spectral sequence  $E_r^{p,q}$  coincides with the one given in [Wei94], where we can find a definition of spectral sequences associated to filtered chain complexes in the case  $\mathbf{A} = R\text{-mod}$  for a ring  $R$ . This can be checked by using the description of the defect and domain in the special case of  $R\text{-mod}$  given in Definition II.1.6.

We close this subsection with a compatibility statement of spectral sequences with homotopy, which is needed for our investigation of so-called *spectral cohomology tables* in the next chapter.

**Definition 2.21.** A **null homotopic morphism** between descending filtered cochain complexes  $(F^\bullet A^\bullet, d_A^{\bullet,\bullet})$  and  $(F^\bullet B^\bullet, d_B^{\bullet,\bullet})$  consists of the following data:

- (1) Morphisms  $F^\bullet s^i : F^\bullet A^i \rightarrow F^\bullet B^{i-1}$  for all  $i \in \mathbb{Z}$ .

(2) A morphism  $F^\bullet \alpha^\bullet$  from  $(F^\bullet A^\bullet, d_A^{\bullet, \bullet})$  to  $(F^\bullet B^\bullet, d_B^{\bullet, \bullet})$  such that

$$F^\bullet \alpha^i = d_B^{\bullet, i-1} \circ F^\bullet s^i + F^\bullet s^{i+1} \circ d_A^{\bullet, i}$$

for all  $i \in \mathbb{Z}$ .

**Corollary 2.22.** *The association*

$$(F^\bullet A^\bullet, d^{\bullet, \bullet}) \mapsto (E_r^{p,q}, \partial_r^{p,q})_{p,q \in \mathbb{Z}, r \geq 1}$$

defines a functor from  $\text{Filt}^\bullet(\text{Ch}^\bullet(\mathbf{A}))$  modulo null homotopic morphisms to the category of cohomological spectral sequences starting at 1.

**PROOF.** We have to show that the functor of Corollary II.2.19, restricted to spectral sequences starting at 1, maps null homotopic morphisms to zero. But since the subquotients on the first page are given by the kernels and images of the differentials on the 0-th page, any morphism of the form  $d_B^{\bullet, i-1} \circ F^\bullet s^i + F^\bullet s^{i+1} \circ d_A^{\bullet, i}$  clearly becomes zero when it is restricted to these subquotients.  $\square$

**2.4. Computing Spectral Sequences.** In this subsection we describe algorithms summarizing our constructive approach to spectral sequences. We work with an arbitrary abelian category  $\mathbf{A}$ . The correctness of our constructions follows from our discussion in the preceding Subsection II.2.3.

**Construction 2.23. Input:** A descending filtration  $FA = (F^\bullet A^\bullet, d^{\bullet, \bullet}) \in \text{Filt}^\bullet(\text{Ch}^\bullet \mathbf{A})$  and integers  $p, q \in \mathbb{Z}$  and  $r \geq 0$ .

**Output:** The object  $E_r^{p,q}$  of the associated spectral sequence of  $FA$ , realized as a subquotient of  $E_0^{p,q}$  (see Definition II.1.55).

**Algorithm:**

(1) **Realize the graded objects of the filtration as subquotients:** Construct the generalized monomorphism

$$\begin{array}{ccc} \frac{F^{p+r} A^{p+q+1}}{F^{p+r+1} A^{p+q+1}} & \xrightarrow{\text{emb}^{p+r}} & F^{p-r} A^{p+q+1} \\ & \swarrow \quad \searrow & \\ & F^{p+r} A^{p+q+1} & \end{array}$$

This represents a subquotient of  $F^{p-r} A^{p+q+1}$  (see Remark II.1.57). Analogously, construct

$$\frac{F^p A^{p+q}}{F^{p+1} A^{p+q}} \xrightarrow{\text{emb}^p} F^{p-r} A^{p+q}$$

and

$$\frac{F^{p-r} A^{p+q-1}}{F^{p-r+1} A^{p+q-1}} \xrightarrow{\text{emb}^{p-r}} F^{p-r} A^{p+q-1}.$$

(2) **Construct generalized differentials:** Compute the two generalized morphisms

$$\begin{array}{ccccccc} \frac{F^{p-r}A^{p+q-1}}{F^{p-r+1}A^{p+q-1}} & \hookrightarrow & F^{p-r}A^{p+q-1} & \xrightarrow{[d^{p-r,p+q-1}]} & F^{p-r}A^{p+q} & \dashrightarrow & \frac{F^pA^{p+q}}{F^{p+1}A^{p+q}} \\ & & & & & \searrow & \\ & & & & & \Delta^{p-r} & \end{array}$$

and

$$\begin{array}{ccccccc} \frac{F^pA^{p+q}}{F^{p+1}A^{p+q}} & \hookrightarrow & F^{p-r}A^{p+q} & \xrightarrow{[d^{p-r,p+q}]} & F^{p-r}A^{p+q+1} & \dashrightarrow & \frac{F^{p+r}A^{p+q+1}}{F^{p+r+1}A^{p+q+1}} \\ & & & & & \searrow & \\ & & & & & \Delta^p & \end{array}$$

For this computation, we use composition of generalized morphisms (Definition II.1.9), pseudo-inversion of generalized morphisms (Definition II.1.27), and regarding a morphism in  $\mathbf{A}$  as a generalized morphism (Lemma II.1.26).

- (3) **Read off the result from canonical subobjects:** Construct the domain of  $\Delta^p$  and the defect of  $\Delta^{p-r}$  (Definition II.1.59) as subobjects of  $\frac{F^pA^{p+q}}{F^{p+1}A^{p+q}}$ . We have inclusions

$$\text{def}(\Delta^{p-r}) \subseteq \text{dom}(\Delta^p) \subseteq \frac{F^pA^{p+q}}{F^{p+1}A^{p+q}} =: E_0^{p,q},$$

and  $E_r^{p,q} := \frac{\text{dom}(\Delta^p)}{\text{def}(\Delta^{p-r})}$  is the result of the construction.

REMARK 2.24. Each of the three steps in Construction II.2.23 can be rewritten in terms of primitives in the abelian category  $\mathbf{A}$ .

- 1.Step: Constructs quotient objects in  $\mathbf{A}$ . Thus, we have to compute cokernels in  $\mathbf{A}$ .
- 2.Step: Uses composition in  $\mathbf{G}(\mathbf{A})$ . Thus, we need to compute pullbacks in  $\mathbf{A}$ , which in turn can be rewritten in terms of kernels and direct sums in  $\mathbf{A}$  (see Construction I.2.20).
- 3.Step: Constructs the defect and the domain of a generalized morphism. This can be derived from the computation of an image embedding in  $\mathbf{A}$  (see Construction II.1.62 and Remark II.1.63).

**Construction 2.25.** We will use the notation of Construction II.2.23. Let

$$E_r^{p,q} \hookrightarrow \overset{\iota^p}{\dashrightarrow} E_0^{p,q}$$

and

$$E_r^{p+r,q-(r-1)} \hookrightarrow \overset{\iota^{p+r}}{\dashrightarrow} E_0^{p+r,q-(r-1)}$$

denote the results of applying Construction II.2.23 to  $(p, q, r)$  and  $(p + r, q - (r - 1), r)$ . Then the differential  $\partial_r^{p,q}$  on the  $r$ -th page between  $E_r^{p,q}$  and  $E_r^{p+r, q-(r-1)}$  can be computed as the following composition of generalized morphisms:

$$E_r^{p,q} \hookrightarrow \underset{\iota^p}{\dashrightarrow} E_0^{p,q} \underset{\Delta^p}{\dashrightarrow} E_0^{p+r, q-(r-1)} \underset{(\iota^{p+r})^{-1}}{\dashrightarrow} E_r^{p+r, q-(r-1)}$$

**Construction 2.26.** We will use the notation of Construction II.2.23. Let  $FB$  be another filtered cochain complex and let  $f : FA \rightarrow FB$  be a morphism between filtered cochain complexes. Taking graded parts,  $f$  induces a morphism  $f_0^{p,q} : E_0^{p,q}(FA) \rightarrow E_0^{p,q}(FB)$  between the objects of the 0-th page of the associated spectral sequences. Then the composition of generalized morphisms given by

$$E_r^{p,q}(FA) \hookrightarrow \underset{\iota_{FA}^p}{\dashrightarrow} E_0^{p,q}(FA) \xrightarrow{f_0^{p,q}} E_0^{p+r, q-(r-1)}(FB) \underset{(\iota_{FB}^p)^{-1}}{\dashrightarrow} E_r^{p+r, q-(r-1)}(FB)$$

computes the morphism  $f_r^{p,q} : E_r^{p,q}(FA) \rightarrow E_r^{p,q}(FB)$  given by the functoriality of taking spectral sequences. Here,  $\iota_{FA}^p$  and  $\iota_{FB}^p$  are the outputs of construction II.2.23 applied to  $FA, FB$ , respectively.

Since these algorithms only use the categorical constructions directly provided by the axioms of  $\mathbf{A}$ , we can use them in any “element-free” context (see for example Computation III.3.21).



## CHAPTER 3

### Applications to Equivariant Sheaves

Cohomology tables are among the most important invariants for coherent sheaves on projective space  $\mathbb{P}_k^n$  over a field  $k$ . In this chapter we generalize them in two different ways: First, if a finite group  $G$  acts on a coherent sheaf, it also acts on its cohomology groups and we can store the characters of these actions within an *equivariant cohomology table*. As an example, we compute an excerpt of the equivariant cohomology table of the famous Horrocks-Mumford bundle (see Subsection III.3.2.2). Second, we will read off from the so-called Tate sequence of a coherent sheaf new numerical invariants. These numerical invariants complement the numbers in the cohomology table to what we call a *spectral cohomology table*, a concept that first appeared in [BLH17]. An excerpt of the spectral cohomology table of the Horrocks-Mumford bundle is given in Figure III.4.

If a coherent sheaf has supernatural cohomology, its spectral cohomology table provides the same information as the cohomology table (Theorem III.3.19). However, when we consider coherent sheaves not having supernatural cohomology, spectral cohomology tables are actually the stronger invariant. Using the Horrocks-Mumford bundle and the existence of supernatural vector bundles (thanks to Boij-Söderberg theory), we will be able to prove the existence of coherent sheaves having identical cohomology tables, but unequal spectral cohomology tables (Theorem III.3.20).

Both generalizations of cohomology tables can be nicely joined, see Figure III.5 for the *equivariant spectral cohomology table* of the Horrocks-Mumford bundle. Its computation is made possible due to our hard work in the previous chapters: The development of constructive categorical  $G$ -equivariant methods in Chapter I and the description of a constructive approach to spectral sequences in Chapter II.

Now, we describe the idea of spectral cohomology tables in full generality. Given an abelian category  $\mathbf{A}$ , there are sometimes canonical ways to equip every object  $A \in \mathbf{A}$  with a filtration which is automatically respected by all morphisms. We call such filtrations *natural*. Equipping an abelian group with its torsion subgroup is an example of a natural filtration. Whenever we have a natural filtration  $F$  for the objects in  $\mathbf{A}$ , every cochain complex  $C$  of  $\mathbf{A}$  can automatically be regarded as a filtered cochain complex, and thus gives rise to an associated spectral sequence  $S$ . Since the assignment  $C \mapsto S$  is functorial, this spectral sequence is an invariant of the cochain complex  $C$ , and whenever the cochain complex  $C$  itself arises as an invariant of some object  $A \in \mathbf{A}$ , the spectral sequence  $S$  becomes an invariant of  $A$  as well. We can depict this process diagrammatically as follows:

$$\begin{array}{ccc}
 A \in \mathbf{A} & \xrightarrow{\quad} & \text{Cochain complex } C & \xrightarrow{\quad} & \text{Associated spectral sequence } S. \\
 & \text{some functorial process} & & \text{using a natural filtration } F &
 \end{array}$$

Special instances of this process will give us spectral cohomology tables (see Subsection III.3.3) as well as spectral Betti tables (see Subsection III.1.2).

We turn to the concrete case where  $\mathbf{A}$  is the category of finitely generated graded modules over the exterior algebra  $E$  in  $n + 1$  indeterminates over  $k$ , denoted by  $E\text{-grmod}$ . In this particular case, the grading of the objects in  $E\text{-grmod}$  induces a natural filtration  $F$  (see Definition III.3.12). The functorial process (up to homotopy) we are interested in is the so-called *Tate resolution*: Given  $M \in E\text{-grmod}$ , we concatenate a minimal projective resolution  $P^\bullet \twoheadrightarrow M$  and a minimal injective resolution  $M \hookrightarrow I^\bullet$ . Following the process described above, we get a spectral sequence  $S$  associated to  $M$ .

The significance of the functorial assignment  $M \mapsto S$  can be explained best in geometric terms. Due to the Bernstein-Gel'fand-Gel'fand correspondence [BGG78], there exists an exact equivalence between the stable category  $E\text{-grmod}$  (see Definition III.3.8) and  $D^b(\mathcal{Coh}(\mathbb{P}_k^n))$ , the bounded derived category of coherent sheaves on projective space of dimension  $n$ . In particular, to every  $\mathcal{F} \in \mathcal{Coh}(\mathbb{P}_k^n)$  there corresponds an  $M \in E\text{-grmod}$  (for details see Subsection III.3.1). It follows that the spectral sequence  $S$  is also an invariant of  $\mathcal{F}$ . From the socles of the objects on the first page of  $S$ , we can read off the cohomology groups  $H^q(\mathcal{F}(p))$  (see Subsection III.3.3). However, there is clearly more to  $S$  than only its objects on the first page. The information provided by its higher pages assembles to what we call a spectral cohomology table, the main object of investigation in this chapter.

The whole setup smoothly generalizes to a  $G$ -equivariant context for a finite group  $G$ . Here,  $G$ -equivariant coherent sheaves on projective space correspond to  $G$ -equivariant graded modules over the exterior algebra, and the  $G$ -action naturally extends to the Tate resolution. So, what we need is a nice constructive model of the category of finitely generated  $G$ -equivariant modules over the exterior algebra in order to keep track of the  $G$ -actions in our computations.

For this let us recall the following notions: The  $k$ -algebra  $E$  equipped with a  $k$ -linear  $G$ -action compatible with the ring multiplication is also called a  *$G$ -equivariant  $k$ -algebra*. A  $G$ -equivariant  $E$ -module is given by an  $E$ -module  $M$  equipped with an *extra datum*, namely a  $k$ -linear action of  $G$  on  $M$  satisfying

$$g(am) = g(a)g(m)$$

for all  $g \in G, a \in E, m \in M$ . Morphisms between  $G$ -equivariant  $E$ -modules are defined as  $E$ -module homomorphisms which in addition are  $G$ -equivariant maps, and we obtain a category  $(E \rtimes G)\text{-mod}$ .

If we used these definitions for a computer implementation of  $(E \rtimes G)\text{-mod}$ , we would have to provide data structures for  $E$ -modules, and enhance them with the *extra datum* of the  $G$ -action. It seems as if computing in  $(E \rtimes G)\text{-mod}$  means to compute with  $E$ -modules while taking care of the  $G$ -action, but thanks to category theory, this does not have to be seen as a burden. By passing to a category equivalent to  $(E \rtimes G)\text{-mod}$ , we can actually benefit from the  $G$ -action: Let  $\text{Rep}_k(G)$  denote the category of  $k$ -linear representations of  $G$ , equipped with its tensor product  $\otimes := \otimes_k$  of representations. The multiplication map  $\mu : E \otimes E \rightarrow E : (a, b) \mapsto a \cdot b$  and the unit map  $\eta : k \rightarrow E : 1 \mapsto 1$  are  $G$ -equivariant and thus can be regarded as morphisms in  $\text{Rep}_k(G)$ . We can summarize these facts by the

statement that the triple

$$(E, \mu : E \otimes E \rightarrow E, \eta : k \rightarrow E)$$

is a *monoid object* in  $\text{Rep}_k(G)$ , or a *monoid internal to the monoidal category*  $(\text{Rep}_k(G), \otimes)$ . Now, there is a purely categorical Definition III.2.4 of a module over a monoid object which gives rise to a category  $(E, \mu, \eta)\text{-mod}$ . In Subsection III.2.2 we will see the following equivalence:

$$(E, \mu, \eta)\text{-mod} \simeq (E \rtimes G)\text{-mod}$$

The objects in  $(E, \mu, \eta)\text{-mod}$  do not provide easier data structures yet, so we have to proceed one step further: In Chapter I we constructed a skeletal tensor category  $\text{SRep}_k(G)$  equivalent to  $\text{Rep}_k(G)$  as tensor categories (with some restrictions for  $k$ , see Subsection I.3.3.7 for details). In particular, we can transfer the monoid  $(E, \mu, \eta)$  from  $\text{Rep}_k(G)$  to  $\text{SRep}_k(G)$ . The category of modules over this transferred monoid is still equivalent to  $(E \rtimes G)\text{-mod}$ , and we finally get nice and concise data structures for our computations, since objects in  $\text{SRep}_k(G)$  can be simply modeled by group characters.

We close this introduction with a brief summary of each section: In the first section we formally introduce the notions of a natural filtration and a spectral Betti table, a concept motivating spectral cohomology tables. The second section provides an exposition of modules over the exterior algebra  $E$  from a categorical point of view including the computation of free and cofree resolutions. In the last section we investigate spectral cohomology tables and compute examples.

## 1. (Co)homological Invariants

**Notation 1.1.** In this section  $\mathbf{A}$  denotes an abelian category.

**1.1. Natural Filtrations.** Natural filtrations provide an easy tool to equip every cochain complex with a filtration. We will apply this tool in Subsection III.3.3 to regard every Tate sequence as filtered cochain complex.

**Definition 1.2.** The **category of descending filtered objects of  $\mathbf{A}$**  is given by the following data:

- (1) Objects are pairs  $(A, F^\bullet A)$  consisting of an object  $A \in \mathbf{A}$  and a descending filtration (see Definition II.2.11)

$$\dots \supseteq F^{p-1}A \supseteq F^pA \supseteq F^{p+1}A \supseteq \dots,$$

where each  $F^pA$  is a subobject of  $A$ , and each inclusion  $F^pA \supseteq F^{p+1}A$  is an inclusion of subobjects of  $A$ .

- (2) Morphisms between  $(A, F^\bullet A)$  and  $(B, F^\bullet B)$  are given by morphisms  $\alpha : A \rightarrow B$  in  $\mathbf{A}$  such that for all  $p \in \mathbb{Z}$ , there exist restrictions  $F^p\alpha : F^pA \rightarrow F^pB$  rendering the diagram

$$\begin{array}{ccc}
A & \xrightarrow{\alpha} & B \\
\uparrow & & \uparrow \\
F^p A & \xrightarrow{F^p \alpha} & F^p B
\end{array}$$

commutative.

We denote this category by  $\text{FiltObj}^\bullet \mathbf{A}$ . Furthermore, we denote the forgetful functor mapping a pair  $(A, F^\bullet A)$  to its first component by

$$\pi : \text{FiltObj}^\bullet \mathbf{A} \rightarrow \mathbf{A}.$$

Analogously, we define the **category of ascending filtered objects of  $\mathbf{A}$** , which we denote by  $\text{FiltObj}_\bullet \mathbf{A}$ .

**REMARK 1.3.** The restrictions  $F^p \alpha$  are uniquely determined and can be computed as honest representatives of the generalized morphisms

$$F^p A \hookrightarrow A \xrightarrow{\alpha} B \dashrightarrow F^p B$$

Now, we come to the basic definition of this section.

**Definition 1.4.** A **natural descending filtration** is defined as a section of  $\pi$ , i.e., a functor

$$F : \mathbf{A} \rightarrow \text{FiltObj}^\bullet \mathbf{A}$$

such that  $\pi \circ F \simeq \text{id}_{\mathbf{A}}$ . Analogously, we define **natural ascending filtrations**.

We can think of natural filtrations as filtrations with which every object in  $\mathbf{A}$  can be equipped, and which is respected by every morphism in  $\mathbf{A}$ .

**Example 1.5.** Equipping an abelian group  $A$  with its torsion subgroup

$$A^{\text{torsion}} \subseteq A$$

defines a natural filtration, since every group homomorphism  $A \rightarrow B$  restricts to a morphism between its torsion subgroups  $A^{\text{torsion}} \rightarrow B^{\text{torsion}}$ .

**Example 1.6.** The purity (or grade) filtration of a module over a commutative noetherian ring [Bar09a] defines a natural filtration generalizing Example III.1.5.

**Example 1.7.** Let  $R$  be a ring and  $I \subseteq R$  be an ideal. Given an  $R$ -module  $M$ , the submodules  $F^p M := I^p M \subseteq M$  define a natural descending filtration.

**Example 1.8.** Let  $S$  be an  $\mathbb{N}_0$ -graded ring. For any graded module  $M = \bigoplus_{d \in \mathbb{Z}} M_d$ , the submodules defined by  $F^p M := \bigoplus_{d \geq p} M_d$  give rise to a natural descending filtration. We obtain a natural ascending filtration by setting  $F_p(M) \subseteq M$  as the submodule *generated* by  $\bigoplus_{d \leq p} M_d$ .

**Example 1.9.** This example is dual to Example III.1.8. Let  $E$  be a  $(-\mathbb{N}_0)$ -graded ring. For any graded module  $M = \bigoplus_{d \in \mathbb{Z}} M_d$ , the submodules defined by  $F_p M := \bigoplus_{d \leq p} M_d$  give rise to a natural ascending filtration. We obtain a natural descending filtration by setting  $F^p(M) \subseteq M$  as the submodule *generated* by  $\bigoplus_{d \geq p} M_d$ .

**1.2. Spectral Betti Tables.** As a motivational example for our construction of spectral cohomology tables (see Subsection III.3.3), we briefly discuss spectral Betti tables in the context of an abelian category  $\mathbf{A}$  having enough projectives or injectives. We will see in Example III.1.14 how spectral Betti tables generalize Betti tables of finitely generated modules over the graded symmetric algebra.

Every natural descending filtration  $F : \mathbf{A} \rightarrow \text{FiltObj}^\bullet \mathbf{A}$  automatically lifts to the category of cochain complexes  $\text{Ch}^\bullet(\mathbf{A})$  by applying  $F$  to the diagram

$$\begin{array}{ccccccc} \dots & \longrightarrow & A^{i-1} & \xrightarrow{d_A^{i-1}} & A^i & \xrightarrow{d_A^i} & A^{i+1} \longrightarrow \dots \\ & & \alpha^{i-1} \downarrow & & \alpha^i \downarrow & & \alpha^{i+1} \downarrow \\ \dots & \longrightarrow & B^{i-1} & \xrightarrow{d_B^{i-1}} & B^i & \xrightarrow{d_B^i} & B^{i+1} \longrightarrow \dots \end{array}$$

representing a cochain map between cochain complexes. Furthermore, the resulting functor

$$\text{Ch}^\bullet(F) : \text{Ch}^\bullet(\mathbf{A}) \rightarrow \text{FiltObj}^\bullet(\text{Ch}^\bullet(\mathbf{A}))$$

respects null homotopic maps, i.e., it sends a null homotopic map in  $\text{Ch}^\bullet(\mathbf{A})$  to a null homotopic map in the sense of Definition II.2.21. This follows from applying  $F$  to the data defining a null homotopy in  $\text{Ch}^\bullet(\mathbf{A})$ .

**Definition 1.10.** Given a natural descending filtration  $F : \mathbf{A} \rightarrow \text{FiltObj}^\bullet \mathbf{A}$  and a cochain complex  $A \in \text{Ch}^\bullet(\mathbf{A})$ , we define the **induced spectral sequence of  $F$  and  $A$**  as the spectral sequence associated to the descending filtered cochain complex  $\text{Ch}^\bullet(F)(A)$ . We can define the induced spectral sequence of a natural ascending filtration and a chain complex in a dual way.

**REMARK 1.11.** For a given  $F$ , assigning to  $A$  the induced spectral sequence of  $F$  and  $A$  is a functorial operation (see Corollary II.2.19). If we consider  $A$  up to homotopy, then the assignment becomes functorial if we discard the 0-th page and only keep those pages  $\geq 1$  (see Corollary II.2.22).

If  $\mathbf{A}$  has enough projectives, then every *chain* complex  $A$  with bounded below homology admits a quasi-isomorphism  $A \leftarrow P_\bullet$ , where  $P_\bullet$  is a bounded below chain complex consisting of projectives. Dually, if  $\mathbf{A}$  has enough injectives, then every *cochain* complex  $A$  with bounded below cohomology admits a quasi-isomorphism  $A \rightarrow I^\bullet$ , where  $I^\bullet$  is a bounded below cochain complex consisting of injectives. Mapping  $A$  to  $P_\bullet$  (or  $I^\bullet$ ) is a functorial operation modulo null homotopy. Thus, the following notion is well-defined.

**Definition 1.12.** Let  $\mathbf{A}$  be an abelian category.

- (1) Assume  $\mathbf{A}$  has enough injectives and let  $F : \mathbf{A} \rightarrow \text{FiltObj}^\bullet \mathbf{A}$  be a natural descending filtration. Let  $A \in \text{Ch}^\bullet(\mathbf{A})$  be a cochain complex with bounded below cohomology. Then we define the **spectral  $F$ -Betti table of  $A$**  as the induced spectral sequence of  $F$  and  $I^\bullet$  starting at page 1, where  $I^\bullet$  is a bounded below cochain complex consisting of injectives admitting a quasi-isomorphism  $A \rightarrow I^\bullet$ .

- (2) Assume  $\mathbf{A}$  has enough projectives and let  $F : \mathbf{A} \rightarrow \text{FiltObj}_\bullet \mathbf{A}$  be a natural ascending filtration. Let  $A \in \text{Ch}_\bullet(\mathbf{A})$  be a chain complex with bounded below homology. Then we define the **spectral  $F$ -Betti table of  $A$**  as the induced spectral sequence of  $F$  and  $P_\bullet$  starting at page 1, where  $P_\bullet$  is a bounded below chain complex consisting of projectives admitting a quasi-isomorphism  $A \leftarrow P_\bullet$ .

In both cases we denote the spectral sequence by  $\text{Betti}^F(A)$ .

The following theorem can of course be restated for an abelian category with enough projectives.

**Theorem 1.13.** *Let  $\mathbf{A}$  be an abelian category with enough injectives. The map*

$$A \mapsto \text{Betti}^F(A)$$

*defines a functor from  $D^+(\mathbf{A})$  (the bounded below derived category of  $\mathbf{A}$ ) to the category of spectral sequences starting at page 1.*

**PROOF.** It is well known that mapping a cochain complex  $A$  with bounded below cohomology to a bounded below cochain complex  $I^\bullet$  consisting of injectives which resolves  $A$  gives a functor from  $D^+(\mathbf{A})$  to  $K^+(\mathbf{A})$ , the category of bounded below cochain complexes modulo null homotopy. Now, the claim follows from Remark III.1.11.  $\square$

Since spectral Betti tables are an invariant for objects in the derived category, it is reasonable to call them a *(co)homological invariant*.

The following example shows how we reconstruct the classical notion of a Betti table from the spectral one.

**Example 1.14.** Let  $k$  be a field,  $n \in \mathbb{N}$ , and let  $S = k[x_0, \dots, x_n]$  be the  $\mathbb{N}_0$ -graded polynomial ring with  $\deg(x_i) = 1$  for all  $i = 0, \dots, n$ . Let  $F$  be the natural *ascending* filtration of Example III.1.8. For any finitely generated  $\mathbb{Z}$ -graded module  $M$ , denote the objects on the first page of  $\text{Betti}^F(M)$  by  $E_{p,q}^1$ . Then

$$E_{p,q}^1 \simeq S(-p)^{\beta_{p+q,p}},$$

where  $\beta_{i,j} \in \mathbb{N}_0$  are the graded Betti numbers of  $M$ . To see this, consider a minimal projective resolution  $M \leftarrow P_\bullet$ . Due to its minimality, the differentials on the 0-th page of the induced spectral sequence of  $F$  and  $P_\bullet$  are all zero. Thus, the objects on the first page coincide with the projective summands of the objects in  $P_\bullet$ , whose ranks give the Betti numbers.

## 2. Equivariant Modules over the Exterior Algebra

**2.1. Actions and Coactions.** As explained in the introduction of Chapter III, we want to model  $G$ -equivariant modules over the exterior algebra as modules internal to the category  $\text{SRep}_k(G)$ . In this subsection we briefly present the theory of modules internal to a monoidal category.

We have already encountered the categorification of the notion *monoid* (see Definition I.3.14). Now, we turn to its internalization.

**Definition 2.1** ([ML71]). Let  $(\mathbf{A}, \otimes, 1, \alpha, \lambda, \rho)$  be a monoidal category. A **monoid in  $\mathbf{A}$**  consists of the following data:

- (1) An object  $A \in \mathbf{A}$ .
- (2) A morphism  $\mu : A \otimes A \rightarrow A$ , called **multiplication**.
- (3) A morphism  $\eta : 1 \rightarrow A$ , called **unit**.
- (4) The associativity law holds, namely, the diagram

$$\begin{array}{ccc}
 A \otimes (A \otimes A) & \xrightarrow{\alpha} & (A \otimes A) \otimes A \\
 A \otimes \mu \downarrow & & \downarrow \mu \otimes A \\
 A \otimes A & \xrightarrow{\mu} & A \xleftarrow{\mu} A \otimes A
 \end{array}$$

commutes.

- (5) The unit constraints hold, namely, the diagram

$$\begin{array}{ccccc}
 1 \otimes A & \xrightarrow{\eta \otimes A} & A \otimes A & \xleftarrow{A \otimes \eta} & A \otimes 1 \\
 & \searrow \lambda & \downarrow \mu & \swarrow \rho & \\
 & & A & & 
 \end{array}$$

commutes.

We define the **category of monoids** as the full subcategory of

$$\sum_{A \in \mathbf{A}} \text{Hom}_{\mathbf{A}}(A \otimes A, A) \times \text{Hom}_{\mathbf{A}}(1, A)$$

generated by those objects satisfying the associativity law and the unit constraints.

For example, a monoid in the category of sets (where we set  $\otimes = \times$ ) is just an ordinary monoid. Its dual notion *comonoid* is less interesting in the context of sets (actually, being a comonoid in the category of sets is only a property every set satisfies, meaning that every set can be turned into a comonoid in a unique way). But this notion becomes very important in other categories (like  $k$ -vector spaces).

**Definition 2.2** ([ML71]). Let  $(\mathbf{A}, \otimes, 1, \alpha, \lambda, \rho)$  be a monoidal category. A **comonoid in  $\mathbf{A}$**  consists of the following data:

- (1) An object  $A \in \mathbf{A}$ .
- (2) A morphism  $\Delta : A \rightarrow A \otimes A$ , called **comultiplication**.
- (3) A morphism  $\epsilon : A \rightarrow 1$ , called **counit**.
- (4) The coassociativity law holds, namely, the diagram

$$\begin{array}{ccc}
A \otimes (A \otimes A) & \xrightarrow{\alpha} & (A \otimes A) \otimes A \\
\uparrow A \otimes \Delta & & \uparrow \Delta \otimes A \\
A \otimes A & \xleftarrow{\Delta} A \xrightarrow{\Delta} & A \otimes A
\end{array}$$

commutes.

(5) The counit constraints hold, namely, the diagram

$$\begin{array}{ccccc}
1 \otimes A & \xleftarrow{\epsilon \otimes A} & A \otimes A & \xrightarrow{A \otimes \epsilon} & A \otimes 1 \\
& \swarrow \lambda^{-1} & \uparrow \Delta & \searrow \rho^{-1} & \\
& & A & & 
\end{array}$$

commutes.

We define the **category of comonoids** as the full subcategory of

$$\sum_{A \in \mathbf{A}} \text{Hom}_{\mathbf{A}}(A, A \otimes A) \times \text{Hom}_{\mathbf{A}}(A, 1)$$

generated by those objects satisfying the coassociativity law and the counit constraints.

**REMARK 2.3.** Monoidal functors map monoids to monoids. In particular, if  $\mathbf{A}$  is a rigid symmetric monoidal category, the dualization functor maps monoids to comonoids and vice versa, giving rise to a contravariant equivalence between the category of monoids and the category of comonoids in  $\mathbf{A}$ .

The next step is an internalization of the notion of an *action* and a *module*, and their corresponding duals.

**Definition 2.4** ([ML71]). Let  $(A, \mu, \eta)$  be a monoid in a monoidal category

$$(\mathbf{A}, \otimes, 1, \alpha, \lambda, \rho)$$

and let  $M$  be an object in  $\mathbf{A}$ . A **right action of  $A$  on  $M$**  is a morphism  $\mu_M : M \otimes A \rightarrow M$  such that the diagrams

$$\begin{array}{ccc}
M \otimes (A \otimes A) & \xrightarrow{\alpha} & (M \otimes A) \otimes A \\
\downarrow M \otimes \mu & & \downarrow \mu_M \otimes A \\
M \otimes A & \xrightarrow{\mu_M} M \xleftarrow{\mu_M} & M \otimes A
\end{array}$$

and



$$\begin{array}{ccc}
 M \otimes A & \xleftarrow{M \otimes \eta} & M \otimes 1 \\
 \mu_M \downarrow & \nearrow \rho & \\
 M & & 
 \end{array}$$

commute. A **right  $A$ -module** is a pair  $(M, \mu_M)$  consisting of an object  $M \in \mathbf{A}$  and a right action  $\mu_M$  of  $A$  on  $M$ . The **category of right  $A$ -modules** is the full subcategory of  $\sum_{M \in \mathbf{A}} \text{Hom}_{\mathbf{A}}(M \otimes A, M)$  generated by the right modules. We denote this category by  $\text{mod-}A$ . The definition of the category of left modules  $A\text{-mod}$  is completely analogous.

**Definition 2.5.** Let  $(A, \Delta, \epsilon)$  be a comonoid in a monoidal category

$$(\mathbf{A}, \otimes, 1, \alpha, \lambda, \rho)$$

and let  $M$  be an object in  $\mathbf{A}$ . A **left coaction of  $A$  on  $M$**  is a morphism  $\Delta_M : M \rightarrow A \otimes M$  such that the diagrams

$$\begin{array}{ccc}
 A \otimes (A \otimes M) & \xrightarrow{\alpha} & (A \otimes A) \otimes M \\
 A \otimes \Delta_M \uparrow & & \uparrow \Delta \otimes M \\
 A \otimes M & \xleftarrow{\Delta_M} M \xrightarrow{\Delta_M} & A \otimes M
 \end{array}$$

and

$$\begin{array}{ccc}
 A \otimes M & \xrightarrow{\epsilon \otimes M} & 1 \otimes M \\
 \Delta_M \uparrow & \nearrow \lambda^{-1} & \\
 M & & 
 \end{array}$$

commute. A **left  $A$ -comodule** is a pair  $(M, \Delta_M)$  consisting of an object  $M \in \mathbf{A}$  and a left coaction  $\Delta_M$  of  $A$  on  $M$ . The **category of left  $A$ -comodules** is the full subcategory of  $\sum_{M \in \mathbf{A}} \text{Hom}_{\mathbf{A}}(M, A \otimes M)$  generated by the left comodules. We denote this category by  $A\text{-comod}$ . The definition of the category of right comodules  $\text{comod-}A$  is completely analogous.

**Example 2.6.** Consider the category of abelian groups  $\mathbf{Ab}$ . The category of monoids in  $\mathbf{Ab}$  is equivalent to the category of rings. The category of modules over such a monoid is equivalent to the category of modules over the corresponding ring. Similarly, monoids in  $k\text{-vec}$  correspond to finite dimensional  $k$ -algebras, and the corresponding notions of modules yield again equivalent categories.

In the case of a rigid symmetric monoidal category, modules and comodules are tightly related, a fact which we use in our implementation of the category of modules over the exterior algebra (see Computation III.3.11).

**Lemma 2.7.** *Let  $A$  be a monoid in a rigid symmetric monoidal category  $\mathbf{A}$  and let  $A^\vee$  be its dual comonoid (see Remark III.2.3). The adjunction*

$$\mathrm{Hom}_{\mathbf{A}}(M \otimes A, M) \simeq \mathrm{Hom}_{\mathbf{A}}(M, A^\vee \otimes M)$$

*gives rise to an equivalence of categories*

$$\mathrm{mod}\text{-}A \simeq A^\vee\text{-comod}.$$

PROOF. Being an adjunction for  $(- \otimes A) \dashv (A^\vee \otimes -)$  is equivalent to the fact that morphisms in  $\sum_{M \in \mathbf{A}} \mathrm{Hom}_{\mathbf{A}}(M \otimes A, M)$  map to morphisms in  $\sum_{M \in \mathbf{A}} \mathrm{Hom}_{\mathbf{A}}(M, A^\vee \otimes M)$ . The difficulty lies in proving that a right action is mapped to a left coaction, i.e., that the appropriate diagrams in the definitions commute. But this can be done using string diagrams.  $\square$

**2.2. Equivariant Modules.** Let  $G$  be a finite group and let  $A$  be a ring equipped with an action of  $G$  on  $A$  by ring automorphisms. In this subsection we provide three descriptions of the category of  $G$ -equivariant  $A$ -modules.

- (1) A  $G$ -equivariant  $A$ -module is given by an  $A$ -module  $M$  equipped with an additive action of  $G$  on  $M$  satisfying

$$g(am) = g(a)g(m)$$

for all  $g \in G, a \in A, m \in M$ . Morphisms between  $G$ -equivariant  $A$ -modules are  $A$ -module homomorphisms which are in addition  $G$ -equivariant maps.

- (2) We define the **crossed product ring**  $A \rtimes G$  as the ring whose underlying additive group of elements is given by  $A \otimes_{\mathbb{Z}} \mathbb{Z}[G]$ , and multiplication is given by the formula

$$(a \otimes g)(a' \otimes g') = ag(a') \otimes gg'.$$

The category of  $G$ -equivariant  $A$ -modules is now given by the category of  $A \rtimes G$ -modules.

- (3) The ring  $A$  equipped with an action of  $G$  by ring automorphisms gives rise to a monoid in the category of  $\mathbb{Z}[G]$ -modules (equipped with the usual tensor product of  $\mathbb{Z}[G]$ -modules): The multiplication map of the ring

$$\mu : A \otimes_{\mathbb{Z}} A \rightarrow A$$

actually lies in  $\mathbb{Z}[G]\text{-mod}$  since

$$\begin{aligned} \mu(g(a \otimes a')) &= \mu(ga \otimes ga') \\ &= g(a)g(a') \\ &= g(aa') = g\mu(a \otimes a') \end{aligned}$$

for all  $g \in G, a, a' \in A$ . The same holds for the unit map  $\mathbb{Z} \rightarrow A : 1 \mapsto 1$ . The category of  $G$ -equivariant  $A$ -modules can now be defined as the category of modules over the monoid in  $\mathbb{Z}[G]\text{-mod}$  given by the triple

$$(A, \mu : A \otimes_{\mathbb{Z}} A \rightarrow A, \mathbb{Z} \rightarrow A).$$

It is easy to see that all three constructions yield equivalent categories. For example, take an object in the category described in (3). It consists of a  $\mathbb{Z}[G]$ -module  $M$  together with a morphism  $\mu_M : A \otimes M \rightarrow M$  satisfying  $(aa')m = a(a'm)$  for all  $a, a' \in A, m \in M$ . So  $M$  already has a  $G$ -action and also an  $A$ -module structure. The fact that  $\mu_M$  is also a  $G$ -equivariant map yields

$$\begin{aligned} g(am) &= g(\mu_M(a \otimes m)) \\ &= \mu_M(g(a \otimes m)) \\ &= \mu_M(g(a) \otimes g(m)) \\ &= g(a)g(m) \end{aligned}$$

for all  $g \in G, a \in A, m \in M$ , which is the compatibility described in (1).

**2.3. Internal Algebra.** We have seen in Subsection III.2.1 how algebras and modules can be modeled internal to a given monoidal category. But even more well-known constructions from classical algebra can be internalized: In this subsection, we deal with an internalization of the exterior algebra, its dual coalgebra, free resolutions, and cofree resolutions. All these concepts are realized in CAP and are concretely utilized in Computation III.3.11 and III.3.21.

Let  $(\mathbf{A}, \otimes, 1, \alpha, \lambda, \rho)$  be a tensor category over a field  $k$ . Note that since  $\mathbf{A}$  is rigid,  $\otimes$  is an exact functor. We assume that  $\mathbf{A}$  admits an exact faithful braided monoidal functor

$$\text{Fib} : \mathbf{A} \rightarrow k\text{-vec}$$

which we call fiber functor. We will use  $\text{Fib}$  to reduce the verification of the commutativity of diagrams to simple familiar computations in  $k\text{-vec}$ .

2.3.1. *Exterior Algebra.* In order to define the exterior algebra internally, we will assume  $\text{char}(k) \neq 2$ .

**Construction 2.8.** Let  $V \in \mathbf{A}$ . We are going to construct objects  $\wedge^i V \in \mathbf{A}$  and morphisms  $m^{i+1} : \wedge^i V \otimes V \rightarrow \wedge^{i+1} V$  for all  $i \in \mathbb{N}_0$ . We set

$$\begin{aligned} \wedge^0 V &:= 1, \\ \wedge^1 V &:= V, \\ m^1 &:= \lambda : 1 \otimes V \rightarrow V. \end{aligned}$$

Next, we construct  $\wedge^{i+1} V$  and  $m^{i+1}$  from the data  $(m^i, \wedge^{i-1} V)$ . Denote the following morphism by  $\tau$  (standing for transposition):

$$\begin{array}{ccc} (\wedge^{i-1} V \otimes V) \otimes V & \xrightarrow{\alpha_{\wedge^{i-1} V, V, V}^{-1}} & \wedge^{i-1} V \otimes (V \otimes V) \\ & \searrow \wedge^{i-1} V \otimes \gamma_{V, V} & \\ \wedge^{i-1} V \otimes (V \otimes V) & \xrightarrow{\alpha_{\wedge^{i-1} V, V, V}} & (\wedge^{i-1} V \otimes V) \otimes V \end{array}$$

We define  $m^{i+1}$  as the cokernel projection of  $r^i := (m^i \otimes V) \circ (\tau + \text{id}_{(\wedge^{i-1}V \otimes V) \otimes V})$ . This yields an exact sequence:

$$(\wedge^{i-1}V \otimes V) \otimes V \xrightarrow{(m^i \otimes V) \circ (\tau + \text{id}_{(\wedge^{i-1}V \otimes V) \otimes V})} \wedge^i V \otimes V \xrightarrow{m^{i+1}} \wedge^{i+1}V \longrightarrow 0$$

REMARK 2.9. The objects  $\wedge^i V$  are mapped via Fib to the quotient spaces

$$(\wedge^{i-1} \text{Fib}V \otimes \text{Fib}V) / \left\langle (v_1 \wedge \cdots \wedge v_{i-2} \wedge v_{i-1}) \otimes v_i + (v_1 \wedge \cdots \wedge v_{i-2} \wedge v_i) \otimes v_{i-1} \mid v_1, \dots, v_i \in \text{Fib}V \right\rangle$$

which identify readily with the  $i$ -th exterior power of the vector space FibV (since we assumed  $\text{char}(k) \neq 2$ ). The morphisms  $m^i$  map via Fib and this identification to the multiplication

$$(v_1 \wedge \cdots \wedge v_i) \otimes v \mapsto v_1 \wedge \cdots \wedge v_i \wedge v.$$

REMARK 2.10. Deligne defines the objects  $\wedge^i V$  in [Del90] as the images of the anti-symmetrization morphisms

$$\sum_{\pi \in S_i} \text{sgn}(\pi) \pi : \bigotimes^i V \rightarrow \bigotimes^i V.$$

But the inductive Construction III.2.8 is easier to compute since it avoids the summation of  $i!$  morphisms (by using the fact that the transpositions  $(1, 2), (2, 3), \dots, (i-1, i)$  generate the group  $S_i$ ). Remark III.2.9 justifies that in our context, working with the inductive construction really yields what we are aiming for, namely an internalization of the exterior algebra.

Since  $\wedge^j \text{Fib}V \cong 0$  for  $j > \dim(\text{Fib}V)$ , the same is true for  $\wedge^j V$ . We therefore define the object  $E := \bigoplus_{i=0}^{\dim(\text{Fib}V)} \wedge^i V$  together with the 0-th summand inclusion as unit  $\eta : 1 \hookrightarrow E$ .

**Construction 2.11.** Our goal is to equip  $E$  with a multiplication  $\mu : E \otimes E \rightarrow E$  turning it into a monoid in  $\mathbf{A}$ . We proceed by induction. We set

$$\mu_E^0 := \rho_E : E \otimes 1 \rightarrow E$$

and

$$\mu_E^1 := E \otimes V \xrightarrow{\sigma_{-\otimes V}((\wedge^i V)_i)} \bigoplus_i (\wedge^i V \otimes V) \xrightarrow{\bigoplus_i m^{i+1}} E$$

where  $\sigma$  denotes the natural isomorphism of Lemma 1.2.10. Now, we construct

$$\mu_E^i : E \otimes \wedge^i V \rightarrow E$$

as a colift along an epimorphism in the following diagram:

$$\begin{array}{ccccc}
 E \otimes \wedge^i V & \xrightarrow{\mu_E^i} & E & & \\
 \uparrow E \otimes m^i & & \uparrow \mu_E^1 & & \\
 E \otimes (\wedge^{i-1} V \otimes V) & \xrightarrow{\alpha_{E, \wedge^{i-1} V, V}} & (E \otimes \wedge^{i-1} V) \otimes V & \xrightarrow{\mu_E^{i-1} \otimes V} & E \otimes V
 \end{array}$$

To see that this colift exists, note that  $E \otimes m^i$  is the cokernel of the morphism  $E \otimes r^i$  (see Construction III.2.8). The composition of the other morphisms in the above diagram yield a test morphism for  $E \otimes r^i$ , which can readily be verified after applying the fiber functor and computing with finite dimensional vector spaces.

Adding up the  $\mu_E^i$  yields a multiplication morphism  $\mu : E \otimes E \rightarrow E$ :

$$\mu_E := E \otimes E \xrightarrow{\sigma_{E \otimes -}((\wedge^i V)_i)} \bigoplus_i (E \otimes \wedge^i V) \xrightarrow{\bigoplus_i \mu_E^i} E$$

Checking the associativity law and the unit constraints can also be done after the application of Fib.

**Definition 2.12.** Given  $V \in \mathbf{A}$ , we call the monoid constructed in III.2.11 the **exterior algebra of  $V$**  and denote it by  $\wedge V$ .

2.3.2. *Dual of Exterior Algebra.* This subsection simply states the dual results of Subsection III.2.3.1 and thus can safely be skipped. The constructions provided by this subsection will be used in the computation of the Tate sequence of an  $E$ -module (see Computation III.3.11). We assume  $\text{char}(k) \neq 2$ . We are going to describe how to construct the comonoid structure of  $\bigoplus_i \wedge^i W \simeq (\bigoplus_i \wedge^i V)^* = (\wedge V)^*$  (see Remark III.2.3), where we set  $W := V^*$ .

**Construction 2.13.** This construction is the dual of Construction III.2.8. Let  $W \in \mathbf{A}$ . We are going to construct objects  $\wedge^i W \in \mathbf{A}$  and morphisms  $c^{i+1} : \wedge^{i+1} W \rightarrow W \otimes \wedge^i W$  for all  $i \in \mathbb{N}_0$ . We set

$$\begin{aligned}
 \wedge^0 W &:= 1, \\
 \wedge^1 W &:= W, \\
 c^1 &:= \rho_W^{-1} : W \rightarrow W \otimes 1.
 \end{aligned}$$

Next, we construct  $\wedge^{i+1} W$  and  $c^{i+1}$  from the data  $(c^i, \wedge^{i-1} W)$ . Denote the following morphism by  $\tau$  (standing for transposition):

$$\begin{array}{ccc}
 W \otimes (W \otimes \wedge^{i-1} W) & \xrightarrow{\alpha_{W, W, \wedge^{i-1} W}} & (W \otimes W) \otimes \wedge^{i-1} W \\
 & \searrow \gamma_{W, W} \otimes \wedge^{i-1} W & \\
 (W \otimes W) \otimes \wedge^{i-1} W & \xrightarrow{\alpha_{W, W, \wedge^{i-1} W}^{-1}} & W \otimes (W \otimes \wedge^{i-1} W)
 \end{array}$$

We define  $c^{i+1}$  as the kernel embedding of  $r^i := (\tau + \text{id}_{W \otimes (W \otimes \wedge^{i-1} W)}) \circ (W \otimes c^i)$ , which yields an exact sequence:

$$W \otimes (W \otimes \wedge^{i-1} W) \xleftarrow{(\tau + \text{id}_{W \otimes (W \otimes \wedge^{i-1} W)}) \circ (W \otimes c^i)} W \otimes \wedge^i W \xleftarrow{c^{i+1}} \wedge^{i+1} W \leftarrow 0$$

We define  $\omega_E := \bigoplus_{i=0}^{\dim(\text{Fib} W)} \wedge^i W$  together with the 0-th summand projection as counit  $\epsilon : \omega_E \rightarrow 1$ .

**Construction 2.14.** This construction is dual to Construction III.2.11. Our goal is to equip  $\omega_E$  with a comultiplication  $\Delta : \omega_E \rightarrow \omega_E \otimes \omega_E$  turning it into a comonoid in  $\mathbf{A}$ . We proceed by induction. We set

$$\Delta_{\omega_E}^0 := \lambda_{\omega_E}^{-1} : \omega_E \rightarrow 1 \otimes \omega_E$$

and

$$\Delta_{\omega_E}^1 := \omega_E \xrightarrow{\bigoplus_i c^{i+1}} \bigoplus_i (W \otimes \wedge^i W) \xrightarrow{\sigma_{W \otimes -}((\wedge^i W)_i)} W \otimes \omega_E$$

Now, we construct  $\Delta_{\omega_E}^i : \omega_E \rightarrow \wedge^i W \otimes \omega_E$  as a lift along a monomorphism in the following diagram:

$$\begin{array}{ccccc} \wedge^i W \otimes \omega_E & \xleftarrow{\Delta_{\omega_E}^i} & & & \omega_E \\ \downarrow c^i \otimes \omega_E & & & & \downarrow \Delta_{\omega_E}^1 \\ (W \otimes \wedge^{i-1} W) \otimes \omega_E & \xleftarrow{\alpha} & W \otimes (\wedge^{i-1} W \otimes \omega_E) & \xleftarrow{W \otimes \Delta_{\omega_E}^{i-1}} & W \otimes \omega_E \end{array}$$

This lift exists since the colift exists in the dual Construction III.2.11.

Adding up the  $\Delta_{\omega_E}^i$  yields a comultiplication morphism  $\Delta : \omega_E \rightarrow \omega_E \otimes \omega_E$ :

$$\Delta_{\omega_E} := \omega_E \xrightarrow{\bigoplus_i \Delta_{\omega_E}^i} \bigoplus_i (\wedge^i W \otimes \omega_E) \xrightarrow{\sigma_{-\otimes \omega_E}((\wedge^i W)_i)} \omega_E \otimes \omega_E$$

**2.3.3. Internal Free Resolutions.** The goal of this subsection is to give a purely categorical construction of free resolutions. First, we introduce free modules w.r.t a forgetful functor.

**Definition 2.15.** Let  $(A, \mu : A \otimes A \rightarrow A, \eta : 1 \rightarrow A)$  be a monoid in  $\mathbf{A}$ . The functor

$$|\cdot| : \text{mod-}A \rightarrow \mathbf{A} : (M, \mu_M) \mapsto M$$

is called the **forgetful functor**.

**Construction 2.16.** Given an object  $N \in \mathbf{A}$ , we construct the object  $N \otimes A$  equipped with the right action

$$\mu_{N \otimes A} := (N \otimes A) \otimes A \xrightarrow{\alpha_{N, A, A}^{-1}} N \otimes (A \otimes A) \xrightarrow{N \otimes \mu} (N \otimes A).$$

The verification of the action axioms can readily be checked after the application of Fib. Furthermore, mapping  $N$  to  $(N \otimes A, \mu_{N \otimes A})$  is a functorial operation.

**Definition 2.17.** The right  $A$ -modules  $(N \otimes A, \mu_{N \otimes A})$  constructed in III.2.16 are called **free modules relative to  $|\cdot|$** .

Free modules have a universal property: Given any right  $A$ -module  $(M, \mu_M)$ , there is an isomorphism

$$\mathrm{Hom}_{\mathbf{A}}(N, M) \simeq \mathrm{Hom}_{\mathrm{mod}\text{-}A}((N \otimes A, \mu_{N \otimes A}), (M, \mu_M)).$$

natural in  $N$  and  $(M, \mu_M)$ . The next construction makes this natural isomorphism explicit.

**Construction 2.18.** Given  $N \in \mathbf{A}$ ,  $(M, \mu_M) \in \mathrm{mod}\text{-}A$  and a morphism  $\phi : N \rightarrow M$  in  $\mathbf{A}$ , the morphism

$$\phi^\# := (N \otimes A) \xrightarrow{\phi \otimes A} (M \otimes A) \xrightarrow{\mu_M} M.$$

defines an  $A$ -module morphism between  $(N \otimes A, \mu_{N \otimes A})$  and  $(M, \mu_M)$  (this can be checked after the application of Fib). Conversely, given an  $A$ -module morphism  $\psi : (N \otimes A, \mu_{N \otimes A}) \rightarrow (M, \mu_M)$ , we get a morphism  $|\psi| \circ (N \otimes \eta) \circ \rho_N^{-1} : N \rightarrow M$  in  $\mathbf{A}$ . Both constructions are mutually inverse.

For computing free resolutions, we have to understand an appropriate categorical concept of generators.

**Definition 2.19.** Let  $(M, \mu_M) \in \mathrm{mod}\text{-}A$ . We say an object  $(N, \phi : N \rightarrow M)$  of the category  $\sum_{N \in \mathbf{A}} \mathrm{Hom}_{\mathbf{A}}(N, M)$  **generates**  $M$  if  $\phi^\# : N \otimes A \rightarrow M$  is an epimorphism. A dependent function mapping a module  $(M, \mu_M)$  to  $(N, \phi : N \rightarrow M)$  generating  $M$  is called a **generator function**.

Since  $\otimes$  is right exact,  $(N, \phi)$  generates  $M$  if and only if the image embedding  $(\mathrm{im}(\phi), \mathrm{im}(\phi) \hookrightarrow M)$  generates  $M$ , just as we would expect from classical module theory.

**Example 2.20.** The natural dependent function  $M \mapsto (M, \mathrm{id}_M)$  is trivially a generator function.

**Example 2.21.** Given a module  $(M, \mu_M)$  over the exterior algebra of  $V \in \mathbf{A}$  (see Definition III.2.12), consider any section  $\sigma$  of the cokernel projection of  $\mu_M^1 : M \otimes V \rightarrow M$  in  $\mathbf{A}$ . Then  $\sigma$  generates  $M$ . To see this, we apply Fib. The cokernel of  $\mu_{\mathrm{Fib}M}^1$  is then given by  $\mathrm{Fib}M / \mathrm{rad}(\mathrm{Fib}M)$ , and any section  $\mathrm{Fib}M / \mathrm{rad}(\mathrm{Fib}M) \rightarrow \mathrm{Fib}M$  yields generators of the module  $\mathrm{Fib}M$ . Note that there is no subspace of smaller dimension than  $\mathrm{Fib}M / \mathrm{rad}(\mathrm{Fib}M)$  which generates  $\mathrm{Fib}M$ . If every epimorphism in  $\mathbf{A}$  splits, then this construction gives rise to a generator function which is in general not natural since the choice of a split is in general not natural.

Now, we give a categorical construction of **free resolutions** that we are going to apply in Computation III.3.11.

**Construction 2.22.** Let

$$g : (N, \mu_N) \mapsto (gN \xrightarrow{g_N} N)$$

be a generator function for  $\text{mod-}A$  and let  $(M, \mu_M) \in \text{mod-}A$ . In the following diagram, every morphism can be constructed from its predecessor where we start at  $M$ :

$$\begin{array}{ccccc}
 & gK_1 \otimes A & & gK_0 \otimes A & & gM \otimes A & \xrightarrow{g_M^\#} & M \\
 & \nearrow & & \nearrow & & \nearrow & & \\
 \dots & & & & & & & \\
 & & g_{K_1}^\# & & g_{K_0}^\# & & & \\
 & & \searrow & & \searrow & & & \\
 & & K_1 := \ker(g_{K_0}^\#) & & K_0 := \ker(g_M^\#) & & & 
 \end{array}$$

Composing the epimorphisms with the monomorphisms yields an exact sequence

$$\dots \longrightarrow gK_1 \otimes A \longrightarrow gK_0 \otimes A \longrightarrow gM \otimes A \xrightarrow{g_M^\#} M$$

Again, exactness can be checked after the application of  $\text{Fib}$ .

**2.3.4. Internal Cofree Resolutions.** This subsection simply states the dual results of Subsection III.2.3.3 and thus can safely be skipped. The constructions provided by this subsection will be used for facilitating the construction of a cofree resolution of an  $E$ -module (needed for constructing Tate sequences, see Computation III.3.11). These constructions are all valid since their duals are.

**Definition 2.23.** Let  $(C, \Delta : C \rightarrow C \otimes C, \epsilon : C \rightarrow 1)$  be a comonoid in  $\mathbf{A}$ . The functor

$$|\cdot| : C\text{-comod} \rightarrow \mathbf{A} : (M, \Delta_M) \mapsto M$$

is called the **forgetful functor**.

**Construction 2.24.** Given an object  $N \in \mathbf{A}$ , we construct the object  $C \otimes N$  equipped with the left coaction

$$\Delta_{C \otimes N} := C \otimes N \xrightarrow{\Delta \otimes N} (C \otimes C) \otimes N \xrightarrow{\alpha^{-1}} C \otimes (C \otimes N).$$

Furthermore, mapping  $N$  to  $(C \otimes N, \Delta_{C \otimes N})$  is a functorial operation.

**Definition 2.25.** The left  $C$ -comodules  $(C \otimes N, \Delta_{C \otimes N})$  constructed in III.2.24 are called **cofree comodules relative to**  $|\cdot|$ .

Cofree comodules have a universal property: Given any left  $C$ -comodule  $(M, \Delta_M)$ , there is an isomorphism

$$\text{Hom}_{\mathbf{A}}(M, N) \simeq \text{Hom}_{C\text{-comod}}((M, \Delta_M), (C \otimes N, \Delta_{C \otimes N})).$$

natural in  $N$  and  $(M, \Delta_M)$ . The next construction makes this natural isomorphism explicit.

**Construction 2.26.** Given  $N \in \mathbf{A}$ ,  $(M, \Delta_M) \in C\text{-comod}$  and a morphism  $\phi : M \rightarrow N$  in  $\mathbf{A}$ , the morphism



$$\phi^b := M \xrightarrow{\Delta_M} C \otimes M \xrightarrow{C \otimes \phi} C \otimes N$$

defines a  $C$ -comodule morphism between  $(M, \Delta_M)$  and  $(C \otimes N, \Delta_{C \otimes N})$ . Conversely, given a  $C$ -comodule morphism  $\psi : (M, \mu_M) \rightarrow (C \otimes N, \Delta_{C \otimes N})$ , we get a morphism  $\lambda_N \circ (\epsilon \otimes N) \circ |\psi| : M \rightarrow N$  in  $\mathbf{A}$ . Both constructions are mutually inverse.

**Definition 2.27.** Let  $(M, \Delta_M) \in C\text{-comod}$ . We say an object  $(N, \phi : M \rightarrow N)$  of the category  $\sum_{N \in \mathbf{A}} \text{Hom}_{\mathbf{A}}(M, N)$  **cogenerates**  $M$  if  $\phi^b : M \rightarrow C \otimes N$  is a monomorphism. A dependent function mapping a comodule  $(M, \Delta_M)$  to  $(N, \phi : M \rightarrow N)$  cogenerating  $M$  is called a **cogenerator function**.

**Example 2.28.** Given a comodule  $(M, \Delta_M)$  over the dual of the exterior algebra of  $V \in \mathbf{A}$ , consider any retraction  $r$  of the kernel embedding of  $\Delta_M^1 : M \rightarrow V^* \otimes M$  in  $\mathbf{A}$ . Then  $r$  cogenerates  $M$ . To see this, we apply  $\text{Fib}$ . The kernel of  $\Delta_{\text{Fib}M}^1$  is then given by  $\text{soc}(\text{Fib}M)$ , and any retraction  $\text{Fib}M \rightarrow \text{soc}(\text{Fib}M)$  cogenerates the comodule  $\text{Fib}M$ .

Now, we are going to construct a **cofree resolution** of a given comodule.

**Construction 2.29.** Let

$$g : (N, \Delta_N) \mapsto (N \xrightarrow{g_N} gN)$$

be a cogenerator function for  $C$ -comod and let  $(M, \Delta_M) \in C\text{-comod}$ . In the following diagram, every morphism can be constructed from its predecessor where we start at  $M$ :

$$\begin{array}{ccccc}
 & C \otimes gK_1 & & C \otimes gK_0 & & C \otimes gM & \xleftarrow{g_M^b} & M \\
 & \swarrow & & \swarrow & & \swarrow & & \\
 \dots & & \xleftarrow{g_{K_1}^b} & & \xleftarrow{g_{K_0}^b} & & & \\
 & & K_1 := \text{coker}(g_{K_0}^b) & & K_0 := \text{coker}(g_M^b) & & & 
 \end{array}$$

Composing the epimorphisms with the monomorphisms yields an exact sequence

$$\leftarrow C \otimes gK_1 \leftarrow C \otimes gK_0 \leftarrow C \otimes gM \xleftarrow{g_M^b} M$$

### 3. Computations with Equivariant Sheaves

**3.1. BGG Correspondence.** In this subsection we give a short introduction to the Bernstein-Gel'fand-Gel'fand correspondence (BGG). We refer the reader to [BGG78], [EFS03], and the appendix of [OSS11] for the proofs.

Let  $k$  be a field,  $n \in \mathbb{N}_0$ , and  $V$  an  $(n + 1)$ -dimensional graded  $k$ -vector space concentrated in degree  $-1$  with basis  $e_0, \dots, e_n$ . We denote by  $W$  the graded  $k$ -dual of  $V$  (necessarily concentrated in degree 1), and by  $x_0, \dots, x_n$  the dual basis associated to the  $e_i$ . We further set  $E := \wedge V$ ,  $S := \text{Sym}(W)$ , and  $\omega_E := \text{Hom}_k(E, k)$  (as an  $E$ - $E$ -bimodule). We write  $E\text{-grMod}$  and  $S\text{-grMod}$  for the categories of graded  $E$ -modules and  $S$ -modules,

respectively, and denote their full subcategories generated by finitely generated graded modules by  $E\text{-grmod}$  and  $S\text{-grmod}$ , respectively. Let further  $\mathbb{P}_k^n := \text{Proj}(S)$  be the projective space and  $\mathfrak{Coh}(\mathbb{P}_k^n)$  the category of coherent sheaves on  $\mathbb{P}_k^n$ .

**Construction 3.1.** Given a graded  $S$ -module  $M = \bigoplus_{d \in \mathbb{Z}} M_d$ , we construct the following cochain complex of graded  $E$ -modules:

$$\begin{aligned} \mathbf{R}(M) : \cdots \longrightarrow M_{i-1} \otimes_k \omega_E \xrightarrow{\mu^{i-1}} M_i \otimes_k \omega_E \longrightarrow \cdots \\ m \otimes \phi \quad \mapsto \quad m \left( \sum_{i=0}^n x_i \otimes e_i \right) \phi \end{aligned}$$

Given a morphism of graded  $S$ -modules  $\alpha : M \rightarrow N$ , the family of maps  $(\alpha_i \otimes \omega_E)_{i \in \mathbb{Z}}$  defines a cochain map from  $\mathbf{R}(M)$  to  $\mathbf{R}(N)$ . Thus, we obtain a functor

$$\mathbf{R} : S\text{-grMod} \longrightarrow \text{Ch}^\bullet(E\text{-grMod})$$

**Construction 3.2.** If we are given a cochain complex of graded  $S$ -modules

$$M^\bullet : \cdots \longrightarrow M^{j-1} \xrightarrow{d^{j-1}} M^j \longrightarrow \cdots$$

then applying  $\mathbf{R}$  yields a cochain complex of cochain complexes (i.e., a double cochain complex with commutative squares)

$$\cdots \longrightarrow \mathbf{R}(M^{j-1}) \xrightarrow{\mathbf{R}(d^{j-1})} \mathbf{R}(M^j) \longrightarrow \cdots$$

By taking its total cochain complex, we can extend  $\mathbf{R}$  to a functor

$$\mathbf{R} : \text{Ch}^\bullet(S\text{-grMod}) \longrightarrow \text{Ch}^\bullet(E\text{-grMod})$$

and the objects in the resulting cochain complex can be described as

$$\mathbf{R}(M^\bullet)^l = \bigoplus_{i+j=l} M_i^j \otimes_k \omega_E$$

for  $l \in \mathbb{Z}$ .

We want to add one remark on the total cochain complex construction: In the category of *graded*  $E$ -modules, the direct product of a family of objects is given by taking the direct product in each degree. In particular, the direct product  $\prod_{i+j=l} M_i^j \otimes_k \omega_E$  taken in the category of graded  $E$ -modules is actually isomorphic to  $\bigoplus_{i+j=l} M_i^j \otimes_k \omega_E$ , a fact which is horribly wrong in the non-graded case. From this observation, it follows that it does not matter whether we apply  $\text{Tot}^\oplus$  or  $\text{Tot}^\Pi$  in our construction of  $\mathbf{R}$ .

**Theorem 3.3.** *If  $M^\bullet$  is a finite cochain complex of finitely generated graded  $S$ -modules, then  $\mathbf{R}(M^\bullet)$  is eventually exact.*

**PROOF.** This theorem holds for a finitely generated graded  $S$ -modules  $M$  due to Theorem 2.3 in [EFS03]. If  $M^\bullet$  is a finite cochain complex, then  $\mathbf{R}(M^\bullet)$  is given by a total cochain complex associated to a bicomplex with eventually exact rows (see Construction

III.3.2). Since the spectral sequence associated to this bicomplex converges to the cohomology of  $\mathbf{R}(M^\bullet)$ , and since the complex is finite, the cohomology of  $\mathbf{R}(M^\bullet)$  is eventually zero.  $\square$

**Definition 3.4.** The full subcategory of the homotopy category of cochain complexes  $\text{Ch}^\bullet(E\text{-grmod})$  generated by exact cochain complexes only consisting of free objects is called the **category of Tate sequences** and denoted by  $\mathfrak{Tate}(E)$ . Objects in this category are called **Tate sequences**. A Tate sequence is called **minimal** if tensoring with the trivial  $E$ - $E$ -bimodule  $k$  only yields zeros as differentials. Note that  $\mathfrak{Tate}(E)$  inherits the structure of a triangulated category from the homotopy category of cochain complexes.

**REMARK 3.5.** The objects of a minimal Tate sequence are uniquely determined up to isomorphism.

**Construction 3.6.** Given a finite cochain complex  $M^\bullet$  of finitely generated graded  $S$ -modules, let  $c \in \mathbb{Z}$  be an index such that  $H^i(\mathbf{R}(M^\bullet)) = 0$  for  $i > c$  (Theorem III.3.3). Let  $\mu^i$  denote the  $i$ -th differential of  $\mathbf{R}(M^\bullet)$ . Combining a free resolution of  $\ker(\mu^{c+1})$  in  $E\text{-grmod}$  with the brutal truncation  $\sigma^{>c}\mathbf{R}(M^\bullet)$  defines a Tate sequence which we denote by  $\text{Tate}(M^\bullet)$ .

**Theorem 3.7.** *Construction III.3.6 gives rise to an exact equivalence*

$$\text{Tate} : D^b(\mathfrak{Coh}(\mathbb{P}_k^n)) \simeq \mathfrak{Tate}(E)$$

Here, an object in  $D^b(\mathfrak{Coh}(\mathbb{P}_k^n))$  has to be represented by the sheafification of a finite cochain complex of finitely generated graded  $S$ -modules.

**Definition 3.8.** Let  $\mathbf{I}$  be the ideal of  $E\text{-grmod}$  given by

$$\mathbf{I}(A, B) := \{\phi \in \text{Hom}_{E\text{-grmod}}(A, B) \mid \phi \text{ factors over a free } E\text{-module}\}$$

for  $A, B \in E\text{-grmod}$ . The **stable module category** is defined as the quotient category

$$\underline{E\text{-grmod}} := E\text{-grmod}/\mathbf{I}.$$

Sending a cochain complex in  $\mathfrak{Tate}(E)$  to its 0-th syzygy object (i.e., to the kernel of its zeroth differential) yields a functor

$$\text{Syz} : \mathfrak{Tate}(E) \rightarrow \underline{E\text{-grmod}}$$

We also get a functor the other way around using the following construction.

**Construction 3.9.** Let  $M \in \underline{E\text{-grmod}}$ . Construction III.2.22 gives us a free resolution  $P^\bullet \twoheadrightarrow M$ , where we use cohomological indices. Construction III.2.29 gives us a cofree resolution  $M \hookrightarrow I^\bullet$ . Joining both resolutions with the differential

$$P^0 \twoheadrightarrow M \hookrightarrow I^0$$

yields an exact complex of free objects, thus a Tate sequence. If  $M$  is a reduced  $E$ -module, i.e., if it has no proper free module as a direct summand, then choosing minimal free and cofree resolutions yields a minimal Tate sequence.

We end up with a chain of equivalences

$$D^b(\mathfrak{Coh}(\mathbb{P}_k^n)) \simeq \mathfrak{Tate}(E) \simeq E\text{-grmod}$$

to which we also refer as the **BGG correspondence**.

Now, we describe how the BGG correspondence can be used for an effective computation of sheaf cohomology.

**Theorem 3.10.** *Under the BGG correspondence, a coherent sheaf  $\mathcal{F} \in \mathfrak{Coh}(\mathbb{P}_k^n) \subseteq D^b(\mathfrak{Coh}(\mathbb{P}_k^n))$  corresponds to a minimal Tate sequence of the form*

$$\cdots \longrightarrow \left( \bigoplus_{i=0}^n H^i(\mathcal{F}(-i)) \right) \otimes_k \omega_E \xrightarrow{d^0} \left( \bigoplus_{i=0}^n H^i(\mathcal{F}(-i+1)) \right) \otimes_k \omega_E \longrightarrow \cdots$$

where the cohomology groups  $H^i(\mathcal{F}(j))$  are regarded as graded  $k$ -vector spaces concentrated in degree  $j$ . Thus, from this cochain complex, we can read off the cohomology groups of  $\mathcal{F}$  as the socles of the objects.

**PROOF.** This is Theorem 4.1 in [EFS03]. We present an alternative proof which directly uses the BGG correspondence:

$$\begin{aligned} H^i(\mathcal{F}) &\simeq \text{Ext}_{\mathfrak{Coh}(\mathbb{P}_k^n)}^i(\mathcal{O}_{\mathbb{P}_k^n}, \mathcal{F}) \\ &\simeq \text{Hom}_{D^b(\mathfrak{Coh}(\mathbb{P}_k^n))}(\mathcal{O}_{\mathbb{P}_k^n}, \mathcal{F}[i]) \\ &\simeq \text{Hom}_{\mathfrak{Tate}(E)}(\text{Tate}(\mathcal{O}_{\mathbb{P}_k^n}), \text{Tate}(\mathcal{F}[i])) \\ &\simeq \text{Hom}_{\mathfrak{Tate}(E)}(\text{Tate}(\mathcal{O}_{\mathbb{P}_k^n}), \text{Tate}(\mathcal{F})[i]) \\ &\simeq \text{Hom}_{E\text{-grmod}}(\text{Syz}(\text{Tate}(\mathcal{O}_{\mathbb{P}_k^n})), \text{Syz}(\text{Tate}(\mathcal{F})[i])) \\ &\simeq \text{Hom}_{E\text{-grmod}}(k, \text{Syz}(\text{Tate}(\mathcal{F})[i])) \end{aligned}$$

Now, if we choose  $\text{Tate}(\mathcal{F})$  as a minimal Tate resolution, its syzygy objects are reduced  $E$ -modules. From this, we conclude

$$\begin{aligned} \text{Hom}_{E\text{-grmod}}(k, \text{Syz}(\text{Tate}(\mathcal{F})[i])) &\simeq \text{Hom}_{E\text{-grmod}}(k, \text{Syz}(\text{Tate}(\mathcal{F})[i])) \\ &\simeq \text{soc}(\text{Tate}(\mathcal{F})^i)_0 \end{aligned}$$

where we use that elements in the socle of an  $E$ -module  $M$  correspond to  $E$ -module homomorphisms  $k \rightarrow M$ . Substituting  $\mathcal{F}(j)$  for  $\mathcal{F}$  gives the result in the general case.  $\square$

### 3.2. Equivariant Cohomology Tables.

3.2.1. *Equivariant BGG Correspondence.* In the author's master's thesis [Pos13], the setup of the BGG correspondence is generalized to the case where  $V$  is equipped with a  $k$ -linear action of a finite group  $G$  such that  $\text{char}(k) \nmid |G|$ . We get a full and faithful functor

$$\mathfrak{Coh}(\mathbb{P}_k(V) \rtimes G) \rightarrow \mathfrak{Tate}(E \rtimes G)$$

sending a  $G$ -equivariant coherent sheaf  $\mathcal{F} \in \mathfrak{Coh}(\mathbb{P}_k(V) \rtimes G)$  to its corresponding  $G$ -equivariant Tate sequence. The equivalence between the stable module category and the

category of Tate sequences also generalizes to the  $G$ -equivariant setup:

$$\mathfrak{Tate}(E \rtimes G) \simeq (E \rtimes G)\text{-grmod}.$$

Forgetting the  $G$ -action yields the classical BGG correspondence. In particular, the minimal  $G$ -equivariant Tate sequence associated to  $\mathcal{F}$  is nothing but the classical Tate sequence equipped with an action of  $G$  on every object such that the differentials respect this action. By Theorem III.3.10, it follows that  $G$  acts on the cohomology groups of  $\mathcal{F}$ . The collection of  $k$ -vector spaces  $(H^i(\mathcal{F}(j)))_{ij}$  equipped with their  $G$ -actions is called the  **$G$ -equivariant cohomology table of  $\mathcal{F}$** .

### 3.2.2. Equivariant Cohomology Table of the Horrocks-Mumford Bundle.

**Computation 3.11.** In Subsection I.3.3.11 we constructed a subgroup  $H$  of the automorphism group of the Horrocks-Mumford bundle  $\mathcal{E}_{\text{HM}}$ . Now, we will see that the computation of its  $H$ -equivariant cohomology table is feasible with the constructive methods developed in this thesis. We start with the construction of  $\text{SRep}_k^{\mathbb{Z}}(H)$ , the skeletal category of  $\mathbb{Z}$ -graded representations of  $H$ , using the methods developed in Subsection I.3.3. These are implemented in the CAP package `GroupRepresentationsForCAP`.

Defining  $\text{SRep}_k^{\mathbb{Z}}(H)$

```
gap> SRepH := RepresentationCategoryZGraded( 1000, 93 );
The Z-graded representation category of Group( [ f1, f2, f3,
f4, f5, f6 ] )
```

Here, the pair (1000, 93) is the identification number of  $H$  in GAP's `SmallGroups` library.

```
gap> H := UnderlyingGroupForRepresentationCategory( RepH );
<pc group of size 1000 with 6 generators>
```

In order to be able to construct objects in  $\text{SRep}_k^{\mathbb{Z}}(H)$ , first we define the set of irreducible characters of  $H$ .

Defining  $\text{Irr}(H)$

```
gap> irr := Irr( H );;
```

Computing  $|\text{Irr}(H)|$

```
gap> Length( irr );
28
```

The set  $\text{Irr}(H)$  consists of 28 irreducible characters which we denote by  $\chi^1, \dots, \chi^{28}$ . Each simple object in  $\text{SRep}_k^{\mathbb{Z}}(H)$  is given by a pair  $(\chi^i, d)$  consisting of an irreducible character  $\chi^i$  and an (internal) degree<sup>1</sup>  $d \in \mathbb{Z}$ . We denote such an object by  $\chi_d^i$ . An arbitrary object in  $\text{SRep}_k^{\mathbb{Z}}(H)$  can be simply described as a formal  $\mathbb{N}_0$ -linear combination of simple objects.

<sup>1</sup>not to be confused with the degree of a group character, i.e., the dimension of a corresponding representation

Defining  $v := 1 \cdot \chi_{-1}^6 \in \text{SRep}_k^{\mathbb{Z}}(H)$

```
gap> v := RepresentationCategoryZGradedObject( -1, irr[6], RepH );
1*(x_[-1, 6])
```

We can ask for the dimension of the representation corresponding to  $v$ .

```
gap> Dimension( v );
5
```

We set the  $\mathbb{Z}$ -degree of  $v$  to  $-1$  because we will identify  $v$  with the 5-dimensional ( $H$ -equivariant) degree  $-1$  part  $\langle e_0, \dots, e_4 \rangle$  of the exterior algebra.

With the next command, we construct the category of right  $E = \wedge v$  modules, which is implemented in the CAP package `InternalExteriorAlgebraForCAP`.

Defining `mod- $E$`

```
gap> modE := EModuleActionCategory( v );
Module category of the internal exterior algebra modeled via
right actions of 1*(x_[-1, 6])
```

We can think of the realization of this category in CAP as follows: From the object  $v \in \text{SRep}_k^{\mathbb{Z}}(H)$ , CAP constructs  $\wedge v$  as a monoid internal to  $\text{SRep}_k^{\mathbb{Z}}(H)$ , as it is described in Subsection III.2.3.1. Then CAP models  $\wedge v$ -modules via right actions of this internal monoid, as it is described in Subsection III.2.1. In particular, since  $v$  is a  $\mathbb{Z}$ -graded and  $H$ -equivariant object, we actually end up with the category of  $\mathbb{Z}$ -graded  $H$ -equivariant modules over  $\wedge v$  in this way (see Subsection III.2.2 for the equivariance), which is exactly what we want for our purpose of computing equivariant cohomology tables via the BGG correspondence.

We need to define one more object in  $\text{SRep}_k^{\mathbb{Z}}(H)$ .

Defining  $h := 1 \cdot \chi_4^5 \in \text{SRep}_k^{\mathbb{Z}}(H)$

```
gap> h := RepresentationCategoryZGradedObject( 4, irr[5], SRepH );
1*(x_[4, 5])
```

We take a look at the dimension of the representation corresponding to  $h$ .

```
gap> Dimension( h );
2
```

So, we now have two objects  $v, h \in \text{SRep}_k^{\mathbb{Z}}(H)$  of dimension 5, 2, respectively (for a list of all the dimensions of irreducible characters of  $H$ , see Subsection I.3.3.11).

As a next step, we are going to construct the free right module  $h \otimes E$  (by means of Construction III.2.16). This object will later become part of a minimal Tate resolution.

Defining  $F := h \otimes E$

```
gap> F := FreeEModule( h, Emod );
<An object in Module category of the internal exterior
algebra modeled via actions of 1*(x_[-1, 6])>
```

Let us take a look at the head and the socle of  $h \otimes E$ .

```
gap> Head( F );
1*(x_[4, 5])
gap> Socle( F );
1*(x_[-1, 5])
```

Clearly, we have  $\text{head}(h \otimes E) = h$ . And since  $E$  is the exterior algebra in 5 indeterminates, we have  $\text{deg}(\text{soc}(h \otimes E)) = \text{deg}(\text{head}(h \otimes E)) - 5$ .

The next command displays the decomposition of  $h \otimes E$ , regarded as an object in  $\text{SRep}_k^{\mathbb{Z}}(H)$ , into simple objects.

$$h \otimes E = \chi_{-1}^5 + \chi_0^{28} + (\chi_1^8 + \chi_1^{12} + \chi_1^{16} + \chi_1^{20}) + (\chi_2^7 + \chi_2^{11} + \chi_2^{15} + \chi_2^{19}) + \chi_3^{25} + \chi_4^5$$

```
gap> ActionDomain( F );
1*(x_[-1, 5]) + 1*(x_[0, 28]) + 1*(x_[1, 8]) + 1*(x_[1, 12]) + 1*(x_[1,
16]) + 1*(x_[1, 20]) + 1*(x_[2, 7]) + 1*(x_[2, 11]) + 1*(x_[2,
15]) + 1*(x_[2, 19]) + 1*(x_[3, 25]) + 1*(x_[4, 5])
```

We learn that there are no higher multiplicities in this decomposition. With the following commands, we pick out the object  $\chi_2^7$  from  $h \otimes E$  and construct the inclusion  $c : \chi_2^7 \hookrightarrow h \otimes E$  as a morphism in  $\text{SRep}_k^{\mathbb{Z}}(H)$ .

Defining  $c : \chi_2^7 \hookrightarrow h \otimes E$

```
gap> chi := Support( ActionDomain( F ) )[7];
<x_[2, 7]>
gap> c := ComponentInclusionMorphism( ActionDomain( F ), chi );
<A morphism in The Z-graded representation category of Group( [ f1, f2,
f3, f4, f5, f6 ] )>
gap> Display( c );
Component: (x_[2, 7])
```

[1]

A morphism in Category of matrices over Q[e]

-----

Invoking the universal property of  $\chi_2^7 \otimes E$ , we can create from the homomorphism of graded representations  $c : \chi_2^7 \hookrightarrow h \otimes E$  an  $E$ -module homomorphism  $\chi_2^7 \otimes E \rightarrow h \otimes E$  (by means of Construction III.2.18).

Defining  $u : \chi_2^7 \otimes E \rightarrow h \otimes E$

```
gap> u := UniversalMorphismFromFreeModule( F, c );
<A morphism in Module category of the internal exterior
algebra modeled via actions of 1*(x_[-1, 6])>
gap> Display( UnderlyingMorphism( u ) );
Component: (x_[-1, 5])
```

[1/2]

A morphism in Category of matrices over Q[e]

-----  
Component: (x\_[0, 28])

```
[1/4*(-e^7 + e^5 - e^3 + 2*e)]
[1/4*(-e^7 + e^6 - e^4 + e^3)]
[                               1/2]
```

A morphism in Category of matrices over Q[e]

-----  
Component: (x\_[1, 12])

```
[1/2*(-e^7 + e^5 - e^3 + 2*e)]
```

A morphism in Category of matrices over Q[e]

-----  
Component: (x\_[1, 16])

[-1]

A morphism in Category of matrices over Q[e]

-----  
Component: (x\_[1, 20])

```
[1/2*(-e^7 + e^3)]
```

A morphism in Category of matrices over Q[e]

-----  
Component: (x\_[2, 7])

[1]

A morphism in Category of matrices over Q[e]

-----



So, the whole morphism  $u : \chi_2^7 \otimes E \rightarrow h \otimes E$  regarded without its  $E$ -action can be encoded in 5 matrices of size  $1 \times 1$  and 1 matrix of size  $3 \times 1$  over  $\mathbb{Q}[e]$ , where  $e$  is a 20-th root of unity. From the display command we can read off the *non-zero* components of  $u$ :

Component	Dimension	Matrix
$\chi_{-1}^5$	2	$\left(\frac{1}{2}\right)$
$\chi_0^{28}$	10	$\begin{pmatrix} \frac{1}{4}(-e^7 + e^5 - e^3 + 2e) \\ \frac{1}{4}(-e^7 + e^6 - e^4 + e^3) \\ \frac{1}{2} \end{pmatrix}$
$\chi_1^{12}$	5	$\left(\frac{1}{2}(-e^7 + e^5 - e^3 + 2e)\right)$
$\chi_1^{16}$	5	$(-1)$
$\chi_1^{20}$	5	$\left(\frac{1}{2}(-e^7 + e^3)\right)$
$\chi_2^7$	5	$(1)$

This rather compact data structure for such a morphism is due to the  $H$ -equivariance and the  $\mathbb{Z}$ -grading. For example, the  $3 \times 1$  matrix of the  $\chi_0^{28}$  component encodes a  $30 \times 10$  matrix (since  $\chi_0^{28}$  has a 10-dimensional underlying vector space). Omitting the  $H$ -equivariance and the  $\mathbb{Z}$ -grading, the underlying vector space homomorphism of  $u$  has to be given by a single matrix with  $\dim(\chi_2^7 \otimes E) = 5 \cdot 2^5 = 160$  rows and  $\dim(h \otimes E) = 2 \cdot 2^5 = 64$  columns. So, we can really see how the very rigid structure in our situation helps us to encode a single big matrix with the help of a few small ones. This effect becomes even more important for the next steps of our computation, where we are going to compute stepwise a minimal free resolution of  $\ker(u)$  (by means of Construction III.2.22).

Step of a minimal free resolution  $u_2 : G(\ker(u)) \otimes E \rightarrow \ker(u) \hookrightarrow \chi_2^7 \otimes E$

```
gap> u2 := StepOfMinimalFreeResolutionOfKernel( u );
<A morphism in Module category of the internal exterior
algebra modeled via actions of 1*(x_[-1, 6])>
```

The source of  $u_2$  is the next object to the left in our Tate sequence. We compute its head, socle, and its socle's dimension.

```
gap> Head( Source( u2 ) );
1*(x_[1, 27])
gap> Socle( Source( u2 ) );
1*(x_[-4, 27])
gap> Dimension( last );
10
```

It follows that our Tate sequence is so far of the form

$$\chi_1^{27} \otimes E \xrightarrow{u_2} \chi_2^7 \otimes E \xrightarrow{u} h \otimes E$$

Enlisting the socle dimensions in a table gives us an excerpt of a cohomology table (Theorem III.3.10):

$H^4 :$				?	?	?	?
$H^3 :$	?	10	5		?	?	?
$H^2 :$	?	?			2	?	?
$H^1 :$	?	?	?				?
$H^0 :$	?	?	?	?			
$p$	-5	-4	-3	-2	-1	0	1

We see that it coincides with an excerpt of the cohomology table of the Horrocks-Mumford bundle. We resolve one step further.

Step of a minimal free resolution  $u_3 : G(\ker(u_2)) \otimes E \rightarrow \ker(u_2) \hookrightarrow \chi_1^{27} \otimes E$

```
gap> u3 := StepOfMinimalFreeResolutionOfKernel( u2 );
<A morphism in Module category of the internal exterior
algebra modeled via actions of 1*(x_[-1, 6])>
```

We compute the socle of  $\text{Source}(u_3)$  for the next diagonal entries in the cohomology table.

```
gap> soc3 := Socle( Source( u3 ) );
1*(x_[-6, 1]) + 1*(x_[-6, 2]) + 1*(x_[-6, 3]) + 1*(x_[-6, 4])
+ 1*(x_[-5, 28])
```

The socle is supported in degree  $-6$  and  $-5$ . We compute its dimension in degree  $-6$ .

```
gap> DegreePart( soc3, -6 );
1*(x_[-6, 1]) + 1*(x_[-6, 2]) + 1*(x_[-6, 3]) + 1*(x_[-6, 4])
gap> Dimension( last );
4
```

We compute its dimension in degree  $-5$ .

```
gap> DegreePart( soc3, -5 );
1*(x_[-5, 28])
gap> Dimension( last );
10
```

So due to the  $\mathbb{Z}$ -grading, we could properly read off the dimensions of the corresponding cohomology groups and now enlist them in the next diagonal of our table:

$H^4$ :	4				?	?	?	?
$H^3$ :	?	10	10	5		?	?	?
$H^2$ :	?	?				2	?	?
$H^1$ :	?	?	?					?
$H^0$ :	?	?	?	?				
$p$	-6	-5	-4	-3	-2	-1	0	1

Now, we would like to fill in the entries of the table to the right. Using CAP, we can fulfill this task in the following way: First, we interpret the morphism  $u : \chi_2^7 \otimes E \rightarrow h \otimes E$  of the Tate sequence not as an  $E$ -module morphism, but as an  $\omega_E$ -comodule morphism, where  $\omega_E$  is the coalgebra  $E^*$  (see Subsection III.2.3.2). This transformation of data structures is justified by the duality of modules and comodules in the context of rigid symmetric monoidal categories (see Lemma III.2.7).

Defining  $u_{co} \in \omega_E\text{-comod}$ , the morphism corresponding to  $u \in \text{mod-}E$

```
gap> u_co := EModuleActionCategoryMorphismAsCoactionCategoryMorphism( u );
<A morphism in Module category of the internal exterior
algebra modeled via left coactions of 1*(x_[1, 9])>
```

Note that the coacting object is now given by  $v^* = (\chi_{-1}^6)^* = \chi_1^9$ . In the context of comodules, we can speak about minimal cofree resolutions, and we are in the position to compute them by employing Construction III.2.29.

Step of a minimal cofree resolution  $(u_{co})_2 : h \otimes E \twoheadrightarrow \text{coker}(u) \hookrightarrow \omega_E \otimes G(\text{coker}(u))$

```
gap> u_co2 := StepOfMinimalCofreeResolutionOfCokernel( u_co );
<A morphism in Module category of the internal exterior
algebra modeled via coactions of 1*(x_[1, 9])>
```

We investigate the socle of  $\text{Range}((u_{co})_2)$ .

```
gap> Socle( Range( u_co2 ) );
1*(x_[1, 8])
gap> Dimension( last );
5
```

So, we can fill in the next diagonal to the right.

$H^4$ :	4					?	?	?	?
$H^3$ :	?	10	10	5			?	?	?
$H^2$ :	?	?				2		?	?
$H^1$ :	?	?	?					5	?
$H^0$ :	?	?	?	?					
$p$	-6	-5	-4	-3	-2	-1	0	1	2

We compute one more step of the cofree resolution.

```

gap> u_co3 := StepOfMinimalCofreeResolutionOfCokernel( u_co2 );
<A morphism in Module category of the internal exterior
algebra modeled via coactions of 1*(x_[1, 9])>
gap> Socle( Range( u_co3 ) );
1*(x_[2, 26])
gap> Dimension( last );
10

```

This computation yields the next diagonal.

$H^4$ :	4						?	?	?	?
$H^3$ :	?	10	10	5				?	?	?
$H^2$ :	?	?				2			?	?
$H^1$ :	?	?	?					5	10	?
$H^0$ :	?	?	?	?						
$p$	-6	-5	-4	-3	-2	-1	0	1	2	3

Since we worked all the way in an  $H$ -equivariant context, we can also enlist characters of  $H$  instead of dimensions in our table, giving an example of an excerpt of an equivariant cohomology table:

$H^4$ :	$\sum_{i=1}^4 \chi_i$						?	?	?	?
$H^3$ :	?	$\chi_{28}$	$\chi_{27}$	$\chi_7$				?	?	?
$H^2$ :	?	?				$\chi_5$			?	?
$H^1$ :	?	?	?					$\chi_8$	$\chi_{26}$	?
$H^0$ :	?	?	?	?						
$p$	-6	-5	-4	-3	-2	-1	0	1	2	3

### 3.3. Spectral Cohomology Tables.

3.3.1. *Definition.* In this subsection we work with a modification of the natural descending filtration defined in Example III.1.9.

**Definition 3.12.** We define a natural descending filtration  $F$  on  $E\text{-grmod}$  as follows: for  $M = \bigoplus_{d \in \mathbb{Z}} M_d \in E\text{-grmod}$ , we set  $F^p(M)$  as the submodule of  $M$  generated by  $\bigoplus_{d \geq p} M_{d+n+1}$ .

REMARK 3.13. The shift by  $n+1$  in Definition III.3.12 is necessary since the cohomology groups of a coherent sheaf are naturally encoded in the socles of the objects in the Tate resolution, and not in the heads.

Let  $\mathcal{F}$  be a coherent sheaf on  $\mathbb{P}_k^n$  and let  $T^\bullet$  be its corresponding Tate sequence. The induced spectral sequence of  $F$  (the natural descending filtration defined above) and  $T^\bullet$  (see Definition III.1.10) starting at page 1 is an invariant of coherent sheaves, since its assignment is a functorial process (see Remark III.1.11).

Using the same argumentation as in Example III.1.14, we see that the objects on the first page are the summands of the minimal Tate sequence described in Theorem III.3.10:

$$E_1^{p,q} \simeq H^q(\mathcal{F}(p)) \otimes_k \omega_E$$

So in particular, these objects encode the cohomology table of  $\mathcal{F}$  as

$$\left( \dim_k(\text{soc}(E_1^{p,q})) \right)_{p,q}$$

But since the whole spectral sequence is an invariant of  $\mathcal{F}$ , the objects on all of its pages are invariants as well, and in particular their associated Hilbert series:

**Definition 3.14.** Let  $M = \bigoplus_{d \in \mathbb{Z}} M_d \in E\text{-grmod}$ . Its **Hilbert series** is given by the finite sum

$$\text{HS}(M) := \sum_{d \in \mathbb{Z}} \dim_k(M_d)t^d \in \mathbb{Z}[t, t^{-1}]$$

So, we get more numerical invariants by considering additionally

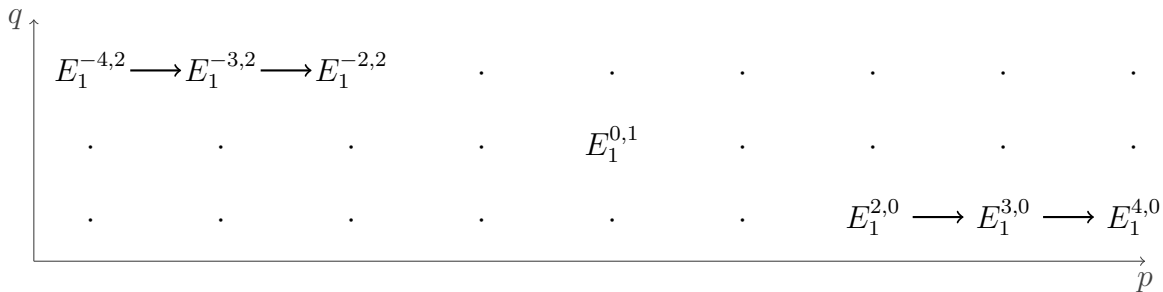
$$\left( \text{HS}(E_r^{p,q}) \right)_{r > 1, p, q}$$

Enhancing the cohomology table of  $\mathcal{F}$  with these new invariants is what we call the **spectral cohomology table of  $\mathcal{F}$** .

3.3.2. *Spectral Cohomology Table of  $\Omega_{\mathbb{P}_k^2}$ .* We are going to answer the question whether the spectral cohomology table really carries more information than the cohomology table, i.e., if there exists coherent sheaves  $\mathcal{F}$  and  $\mathcal{G}$  having the same cohomology tables but not the same spectral cohomology tables. We start our investigation with an example for which the spectral cohomology table does not provide new information, namely the cohomology table of the cotangent bundle  $\Omega_{\mathbb{P}_k^2}$  on projective space of dimension 2:

$H^2 :$	48	35	24	15	8	3									
$H^1 :$							1								
$H^0 :$										3	8	15	24	35	48
$p$	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

The shape of the first page of the spectral sequence from which we compute the spectral cohomology table of  $\Omega_{\mathbb{P}_k^2}$  can be read off from the cohomology table:



Since the differentials pointing to and going from  $E_1^{0,1} \simeq H^1(\Omega_{\mathbb{P}_k^2}) \otimes \omega_E$  are zero, this object won't change by passing to the next page, and its Hilbert series simply is a shifted

multiple of the Hilbert series of the exterior algebra. Furthermore, we know that the Tate sequence is exact, and thus the spectral sequence converges to 0. Since the differentials pointing to and going from  $E_r^{p,2}$  for  $r > 1$  and  $p < -2$  are all zero, we conclude that these objects have to be zero. The same is true for  $E_r^{p,0}$  with  $r > 1$  and  $p > 2$ . So, we are left with the following task: Given the truncated Tate sequence

$$\begin{array}{ccccccc} \dots & \rightarrow & H^2(\Omega_{\mathbb{P}_k^2}(-4)) \otimes_k \omega_E & \longrightarrow & H^2(\Omega_{\mathbb{P}_k^2}(-3)) \otimes_k \omega_E & \longrightarrow & H^2(\Omega_{\mathbb{P}_k^2}(-2)) \otimes_k \omega_E \longrightarrow 0 \\ & & \wr \downarrow & & \wr \downarrow & & \wr \downarrow & \parallel \\ \dots & \longrightarrow & E_1^{-4,2} & \longrightarrow & E_1^{-3,2} & \longrightarrow & E_1^{-2,2} & \longrightarrow 0, \end{array}$$

compute the Hilbert series of its cohomology. In our example the Hilbert series of each object are given as follows:

$$\begin{array}{ccccccc} \dots & \rightarrow & \omega_E \otimes_k H^2(\Omega_{\mathbb{P}_k^2}(-4)) & \longrightarrow & \omega_E \otimes_k H^2(\Omega_{\mathbb{P}_k^2}(-3)) & \longrightarrow & \omega_E \otimes_k H^2(\Omega_{\mathbb{P}_k^2}(-2)) \longrightarrow 0 \\ \text{HS :} & & 15t^{-4} + 45t^{-3} + 45t^{-2} + 15t^{-1} & & 8t^{-3} + 24t^{-2} + 24t^{-1} + 8 & & 3t^{-2} + 9t^{-1} + 9 + 3t \end{array}$$

Taking the alternating sum of these Hilbert series for at least  $i \geq 2$  summands, we get  $3t + 1$  plus a term which is of degree  $-i + 1$ . Due to Theorem III.3.16 (stated and proven in the next section), we can conclude that  $3t + 1$  is the Hilbert series of the cohomology. So, the second page of the spectral cohomology table has to look as follows:

$E_2^{pq} :$	2		$3t + 1$					
	1			$t^3 + 3t^2 + 3t + 1$				
	0					$t^3 + 3t^2$		
$q/p$	-3	-2	-1	0	1	2	3	

The other term  $t^3 + 3t^2$  can be either computed in the same way as  $3t + 1$ , or by using the fact that the third page has to become 0. So, we were really able to compute the spectral cohomology table of  $\Omega_{\mathbb{P}_k^2}$  only using its cohomology table.

3.3.3. *Hilbert Series of Unbounded Cochain Complexes.* We now use the standard formalism of topological groups in commutative algebra (see for example [AM69]) which allows us to reason with infinite alternating sums of Hilbert series (as it was needed in Subsection III.3.3.2).

**Definition 3.15.** Let  $\mathbb{Z}[t, t^{-1}]$  denote the abelian group of Laurent polynomials, i.e., the underlying abelian group of the localization of the polynomial ring  $\mathbb{Z}[t]$  at  $t$ . We define a sequence of subgroups

$$\mathbb{Z}[t, t^{-1}] = \mathbb{Z}[t, t^{-1}]_0 \supseteq \mathbb{Z}[t, t^{-1}]_1 \supseteq \dots \supseteq \mathbb{Z}[t, t^{-1}]_n \supseteq \dots$$

by setting  $\mathbb{Z}[t, t^{-1}]_n := \{\sum_{i \in \mathbb{Z}} a_i t^i \in \mathbb{Z}[t, t^{-1}] \mid a_i = 0 \text{ for } |i| < n\}$ . Taking these subgroups as a fundamental system of neighborhoods of 0 defines a topological group structure on  $\mathbb{Z}[t, t^{-1}]$ , whose completion is denoted by  $\widehat{\mathbb{Z}[t, t^{-1}]}$ . Since  $\bigcap_{n \geq 0} \mathbb{Z}[t, t^{-1}]_n = 0$ , which implies that  $\mathbb{Z}[t, t^{-1}]$  is Hausdorff, the natural map

$$\mathbb{Z}[t, t^{-1}] \hookrightarrow \widehat{\mathbb{Z}[t, t^{-1}]}$$

is injective. Thus, we can identify a Hilbert series of a finitely generated graded  $E$ -module with an element of  $\widehat{\mathbb{Z}[t, t^{-1}]}$ .

**Theorem 3.16.** *Let  $(C^\bullet, d^\bullet) \in \text{Ch}^\bullet(E\text{-grmod})$  be a cochain complex with the following properties.*

- (1) *There exists an  $l \in \mathbb{N}$  such that for all  $i > l$  and  $i < -l$ , we have  $\ker(d^i) \subseteq \text{rad}(C^i)$ .*
- (2) *It has bounded cohomology.*

*Then the finite alternating sum  $\sum_{i \in \mathbb{Z}} (-1)^i \text{HS}(H^i(C^\bullet))$  is equal to*

$$\sum_{i \in \mathbb{Z}} (-1)^i \text{HS}(C^i) := \lim_{j \rightarrow \infty} \sum_{|i| < j} (-1)^i \text{HS}(C^i) \in \widehat{\mathbb{Z}[t, t^{-1}]}.$$

PROOF. For any finite cochain complex  $D^\bullet$ , it is well known that

$$\sum_{i \in \mathbb{Z}} (-1)^i \text{HS}(H^i(D^\bullet)) = \sum_{i \in \mathbb{Z}} (-1)^i \text{HS}(D^i).$$

So, we analyze the partial sums  $\sum_{|i| < j} (-1)^i \text{HS}(C^i)$  and see that they equal

$$(-1)^{-(j-1)} \text{HS}(\ker(d^{-(j-1)})) + \sum_{i \in \mathbb{Z}} (-1)^i \text{HS}(H^i(C^\bullet)) + (-1)^{j-1} \text{HS}(\text{coker}(d^{j-2}))$$

for  $j$  sufficiently large, since  $C^\bullet$  has bounded cohomology. Thus, it suffices to prove

$$\lim_{j \rightarrow \infty} \text{HS}(\ker(d^{-(j-1)})) = 0 \quad \text{and} \quad \lim_{j \rightarrow \infty} \text{HS}(\text{coker}(d^{j-2})) = 0.$$

To prove the first limit, we introduce the following quantity: For  $M \in E\text{-grmod}$ , we define  $\nu(M) := \max\{d \in \mathbb{Z} \mid M_d \neq 0\}$ , where we set  $\max(\emptyset) = -\infty$ , so that we have  $\nu(M) = -\infty \Leftrightarrow M \simeq 0$ . Furthermore, we have  $\nu(M) = \nu(M/\text{rad}(M))$ , and if  $M \neq 0$ , then  $\nu(\text{rad}(M)) < \nu(M)$ .

Now, we use our first assumption on  $C^\bullet$  to compute for all  $i < -l$ :

$$\begin{aligned} \nu(C^{i-1}) &= \nu(C^{i-1}/\text{rad}(C^{i-1})) \\ &= \nu(C^{i-1}/\ker(d^{i-1})) \\ &= \nu(\text{im}(d^{i-1})) \\ &\leq \nu(\text{rad}(C^i)) \leq \nu(C^i), \end{aligned}$$

where the last inequality is strict if and only if  $C^i \not\cong 0$ . Thus, either the cochain complex is bounded to the left, or the highest degree of each  $C^i$  strictly decreases when  $i$  decreases. Since the sequence  $(\nu(\ker d^i))_{i < -l}$  is bounded by  $(\nu(C^i))_{i < -l}$ , it also approaches  $-\infty$ . And since  $\text{HS}(\ker d^i) \in \mathbb{Z}[t, t^{-1}]_{\nu(\ker d^i)}$ , we conclude  $\lim_{j \rightarrow \infty} \text{HS}(\ker(d^{-(j-1)})) = 0$ . Dualizing  $C^\bullet$  yields the claim for the second limit. This completes the proof.  $\square$

**3.3.4. Spectral Cohomology Tables of Supernatural Sheaves.** We are going to generalize the example computation of Subsection III.3.3.2 to the case of sheaves with supernatural cohomology. The following definitions come from Boij-Söderberg theory, in which vector bundles with supernatural cohomology play a crucial role (for a survey of Boij-Söderberg theory, see [Flø12]).

**Definition 3.17.** For  $s \in \mathbb{N}_0$ , a **root sequence of length  $s$**  is a sequence of strictly decreasing integers

$$\mathbf{z} : z_1 > z_2 > \cdots > z_s.$$

For our convenience, we will expand every such sequence by  $z_i := \infty$  for  $i < 1$  and  $z_i := -\infty$  for  $i > s$ . We set its associated **Hilbert polynomial** as

$$\text{HP}^{\mathbf{z}}(t) := \frac{1}{s!} \prod_{i=1}^s (t - z_i)$$

and its associated **cohomology table** as

$$\gamma^{\mathbf{z}}(p, q) := \begin{cases} |\text{HP}^{\mathbf{z}}(p)| & z_q > p > z_{q+1} \\ 0 & \text{otherwise} \end{cases}$$

**Definition 3.18.** Let  $\mathbf{z}$  be a root sequence. A coherent sheaf  $\mathcal{F} \in \mathfrak{Coh}(\mathbb{P}_k^n)$  has **supernatural cohomology of type  $\mathbf{z}$**  if the cohomology table of  $\mathcal{F}$  is given by

$$H^q(\mathcal{F}(p)) = \text{Degree}(\mathcal{F}) \cdot \gamma^{\mathbf{z}}(p, q)$$

In particular, the Hilbert polynomial of  $\mathcal{F}$  equals  $\text{Degree}(\mathcal{F}) \cdot \text{HP}^{\mathbf{z}}(t)$ .

**Theorem 3.19.** *Assume  $\mathcal{F} \in \mathfrak{Coh}(\mathbb{P}_k^n)$  has supernatural cohomology. Then the spectral cohomology table of  $\mathcal{F}$  is determined by its cohomology table.*

**PROOF.** Let  $(E_r^{p,q})_{r \geq 1}$  be the family of objects of the spectral sequence defining the spectral cohomology table of  $\mathcal{F}$ . Let further  $\mathbf{z} = z_1 > \cdots > z_s$  be the root sequence of  $\mathcal{F}$ . If  $s = 0$ , then the pages  $r > 1$  are all zero. If  $s > 0$ , let  $\mathcal{P} := (\{(p, q) \mid E_2^{p,q} \neq 0\}, <)$ , where  $(p, q) < (p', q') := \Leftrightarrow p < p'$ . By supernaturality,  $<$  is a well-order with least element given by  $(z_s - 1, s)$ . For each  $(p, q) \in \mathcal{P}$ , there is exactly one page  $\tau_{pq} \in \mathbb{Z}$  such that the differential pointing to  $E_{\tau_{pq}}^{p,q}$  is non-trivial, and exactly one page  $\phi_{pq} \in \mathbb{Z}$  such that the differential going from  $E_{\phi_{pq}}^{p,q}$  is non-trivial.

We start determining  $\text{HS}(E_r^{z_s-1, s})$  for all  $r > 1$ , i.e., the Hilbert series of the objects in the least position. We can compute

$$\text{HS}(E_2^{z_s-1, s}) = \sum_{p \in \mathbb{Z}} (-1)^p \text{HS}(E_1^{p, s})$$

as a limit due to Theorem III.3.16. Furthermore,

$$\text{HS}(E_r^{z_s-1, s}) = \begin{cases} \text{HS}(E_2^{z_s-1, s}) & r = 2, \dots, \text{out}(z_s - 1, s) \\ 0 & \text{otherwise.} \end{cases}$$

Thus the Hilbert series of all objects at  $(z_s - 1, s)$  are determined. This was the base case.

Now we do the inductive step: Let  $(p, q) \in \mathcal{P}$  such that for all predecessors  $(p', q') \in \mathcal{P}$ , we have already computed  $\text{HS}(E_r^{p', q'})$  for all  $r > 1$ . Then  $E_r^{p, q} \simeq E_1^{p, q}$  for  $r = 2, \dots, \min\{\tau_{pq}, \phi_{pq}\}$ . We further distinguish three cases.

**First case** ( $\tau_{pq} = \phi_{pq}$ ): Then  $E_r^{p, q} \simeq 0$  for  $r > \tau_{pq}$ , since the spectral sequence converges to 0.

**Second case** ( $\tau_{pq} < \phi_{pq}$ ): Let



$$0 \longrightarrow E_{\tau_{pq}}^{p-lr, q+l(r-1)} \longrightarrow \dots \longrightarrow E_{\tau_{pq}}^{p-r, q+(r-1)} \longrightarrow E_{\tau_{pq}}^{p, q} \longrightarrow 0$$

be the cochain complex  $C^\bullet$  on the  $\tau_{pq}$ -th page (of length  $l$ ). We are interested in the Hilbert series of the cohomology at the right border position.  $C^\bullet$  is everywhere exact except for the left and right border position. So when we compute  $\text{HS}(C^\bullet)$  as an alternating sum of the Hilbert series of the objects in  $C^\bullet$ , we get  $(-1)^l \text{HS}(E_{\tau_{pq+1}}^{p-lr, q+l(r-1)}) + \text{HS}(E_{\tau_{pq+1}}^{p, q})$ . Since by the inductive hypothesis, we know the value of  $\text{HS}(E_{\tau_{pq+1}}^{p-lr, q+l(r-1)})$ , we can also compute  $\text{HS}(E_{\tau_{pq+1}}^{p, q})$ . Again, this value stays constant until it gets 0 for  $r > \phi_{pq}$ .

**Third case** ( $\tau_{pq} > \phi_{pq}$ ): Let

$$0 \longrightarrow E_{\tau_{pq}}^{p-lr, q+l(r-1)} \longrightarrow \dots \longrightarrow E_{\tau_{pq}}^{p-2r, q+2(r-1)} \longrightarrow E_{\tau_{pq}}^{p-r, q+(r-1)} \longrightarrow 0$$

be the cochain complex  $C^\bullet$  on the  $\tau_{pq}$ -th page (of length  $l$ ), where we omit the object  $E_{\tau_{pq}}^{p, q}$ . The object  $E_{\tau_{pq}}^{p, q} \simeq E_{\phi_{pq+1}}^{p, q}$  is isomorphic to the cohomology of  $C^\bullet$  at the right border position. For the computation of its Hilbert series, we can proceed like in the second case. This finishes the proof.  $\square$

### 3.3.5. Spectral Cohomology Tables vs. Cohomology Tables.

**Theorem 3.20.** *The spectral cohomology table is a stronger invariant than the cohomology table, i.e., there exist vector bundles  $\mathcal{F}$  and  $\mathcal{G}$  having equal cohomology tables, but unequal spectral cohomology tables.*

**PROOF.** We will use main results of Boij-Söderberg theory. The first main result states that for every root sequence  $\mathbf{z}$  of length  $n = \dim \mathbb{P}_k^n$ , there exists a vector bundle  $\mathcal{E}^{\mathbf{z}}$  having supernatural cohomology of type  $\mathbf{z}$ . The second main result states that every cohomology table of a vector bundle on  $\mathbb{P}_k^n$  is a positive rational linear combination of cohomology tables of supernatural bundles. Both main results are proven in [ES09]. For example, the cohomology table  $\gamma^{\mathcal{E}_{\text{HM}}}$  of the Horrocks-Mumford bundle can be decomposed as

$$\gamma^{\mathcal{E}_{\text{HM}}} = \frac{2}{9}\gamma^{(5,0,-2,-5)} + \frac{7}{45}\gamma^{(4,0,-2,-5)} + \frac{56}{45}\gamma^{(4,0,-2,-6)} + \frac{7}{45}\gamma^{(3,0,-2,-6)} + \frac{2}{9}\gamma^{(3,0,-2,-7)}.$$

Now, we can build up a vector bundle  $\mathcal{F}$  as an appropriate direct sum of the supernatural vector bundles  $\mathcal{E}^{(5,0,-2,-5)}$ ,  $\mathcal{E}^{(4,0,-2,-5)}$ ,  $\mathcal{E}^{(4,0,-2,-6)}$ ,  $\mathcal{E}^{(3,0,-2,-6)}$ , and  $\mathcal{E}^{(3,0,-2,-7)}$ , such that the cohomology table of  $\mathcal{F}$  is a multiple of the cohomology table of the Horrocks-Mumford bundle  $\mathcal{E}_{\text{HM}}$ . We set  $\mathcal{G} := m \cdot \mathcal{E}_{\text{HM}}$ , where  $m \in \mathbb{N}$  is that multiple. Note that in the rest of the proof, an exact value for  $m$  is irrelevant.

By definition,  $\mathcal{F}$  and  $\mathcal{G}$  have equal cohomology tables. Assume that they have equal spectral cohomology tables. Since the spectral cohomology table is additive, and since

$$\mathcal{F} = a_1\mathcal{E}^{(5,0,-2,-5)} \oplus a_2\mathcal{E}^{(4,0,-2,-5)} \oplus a_3\mathcal{E}^{(4,0,-2,-6)} \oplus a_4\mathcal{E}^{(3,0,-2,-6)} \oplus a_5\mathcal{E}^{(3,0,-2,-7)},$$

where  $(a_1, a_2, a_3, a_4, a_5)$  is a multiple of  $(\frac{2}{9}, \frac{7}{45}, \frac{56}{45}, \frac{7}{45}, \frac{2}{9})$ , we can use the constructive proof of Theorem III.3.19 for computing the second page of  $\mathcal{F}$ . By assumption, dividing the

elements on this page by  $m$  has to yield the second page of the Horrocks-Mumford bundle. But an excerpt of the result of this computation is

1	$2t^4 + 10t^3 + 14t^2 + 5t$	$\frac{146}{135}t^7 + \frac{2533}{540}t^6 + \frac{781}{108}t^5 + 4t^4$
0		
$q/p$	1	2

which is absurd since the coefficients of the Hilbert series are not integral (see Figure III.4 for the actual spectral cohomology table of the Horrocks-Mumford bundle).  $\square$

3.3.6. *Spectral Cohomology Table of the Horrocks-Mumford Bundle.*

**Computation 3.21.** This is a continuation of Computation III.3.11. This time, we are aiming at the spectral cohomology table of the Horrocks-Mumford bundle  $\mathcal{E}_{\text{HM}}$ . Again, we start with the differential  $u : \chi_2^7 \otimes E \rightarrow h \otimes E$  and construct its associated Tate sequence (i.e., its extension by a free resolution of  $\ker(u)$  to the left and by a cofree resolution of  $\text{coker}(u)$  to the right). This can be done with a single command.

```
gap> T := FilteredTateResolution( u );
<An object in Cocomplex category of Descending filtered
object category of The Z-graded representation category of Group( [ f1,
f2, f3, f4, f5, f6 ] )>
```

Note that CAP outputs a descending filtered cochain complex in  $\text{SRep}_k^{\mathbb{Z}}(H)$ . The formal definition of this filtration is given in Definition III.3.12. Here is an illustration of the resulting filtration on one differential  $d$  of the Tate sequence of  $\mathcal{E}_{\text{HM}}$ :

$$\begin{array}{c}
 \text{deg} \\
 -7 \quad (k^{35}(7) \otimes \omega_E) \oplus (k^2(6) \otimes \omega_E) \xrightarrow{d} (k^4(6) \otimes \omega_E) \oplus (k^{10}(5) \otimes \omega_E) \\
 \quad \quad \quad \uparrow \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \parallel \\
 -6 \quad \quad \quad (k^2(6) \otimes \omega_E) \xrightarrow{\quad \quad \quad} (k^4(6) \otimes \omega_E) \oplus (k^{10}(5) \otimes \omega_E) \\
 \quad \quad \quad \uparrow \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \uparrow \\
 -5 \quad \quad \quad 0 \xrightarrow{\quad \quad \quad} (k^{10}(5) \otimes \omega_E) \\
 \quad \quad \quad \parallel \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \uparrow \\
 -4 \quad \quad \quad 0 \xrightarrow{\quad \quad \quad} 0.
 \end{array}$$

So, this filtration turns the Tate sequence into a decreasing filtered complex, which raises the question of how its associated spectral sequence looks like, or for which triples  $p, q, r \in \mathbb{Z}$  can we expect non-trivial objects  $E_r^{p,q}$ ? We have already seen that the objects on the first page are free  $E$ -modules whose socles encode the cohomology groups. We also know that since the Tate sequence is exact, the spectral sequence will converge to 0. Note that  $E_2^{p,q}$  can only be non-trivial if  $E_1^{p,q} \simeq H^q(\mathcal{E}_{\text{HM}}(p)) \otimes \omega_E$  is non-trivial, so we only look at those  $p, q$  such that  $H^q(\mathcal{E}_{\text{HM}}(p)) \neq 0$ . From the cohomology table of  $\mathcal{E}_{\text{HM}}$ , we can read

off all such  $p, q$ :

$H^4$ :	100	35	4												
$H^3$ :			2	10	10	5									
$H^2$ :							2								
$H^1$ :									5	10	10	2			
$H^0$ :												4	35	100	
$p$	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6

To the left and to the right, the table extends only by non-trivial values for  $H^4, H^0$ , respectively, and  $E_2^{p,q}$  for  $p < -8$  or  $p > 6$  is zero for all  $q$  (by the exactness of the Tate sequence).

For all the remaining cases, we let CAP do the computations. We start with  $E_2^{-1,2}$ .

```
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -1, 2 );
<A morphism in Generalized morphism category of The Z-graded
representation category of Group( [ f1, f2, f3, f4, f5, f6 ] )>
gap> E2_m1_2 := UnderlyingHonestObject( Source( s ) );;
```

The first command constructs the generalized morphism

$$E_2^{-1,2} \hookrightarrow E_0^{-1,2}$$

as it is described in Construction II.2.23. So, in order to get  $E_2^{-1,2}$ , we have to take the source of this generalized morphism and consider it as an object in  $\text{SRep}_k^{\mathbb{Z}}(H)$ . Next, we compute its Hilbert series, both its equivariant and its classical version.

```
gap> DegreeDecomposition( E2_m1_2 );
[ [ -1, 1*(x_[-1, 5]) ], [ 0, 1*(x_[0, 28]) ], [ 1, 1*(x_[1,
8]) + 1*(x_[1, 12]) + 1*(x_[1, 16]) + 1*(x_[1, 20]) ],
[ 2, 1*(x_[2, 7]) + 1*(x_[2, 11]) + 1*(x_[2, 15]) + 1*(x_[2,
19]) ], [ 3, 1*(x_[3, 25]) ], [ 4, 1*(x_[4, 5]) ] ]
gap> HilbertSeries( E2_m1_2 );
2*t^4+10*t^3+20*t^2+20*t+10+2*t^-1
```

As no surprise, we see the Hilbert series of a free object, since no differential pointed to or from  $E_1^{-1,2}$  on the first page. Now, we follow the table to the left and compute more and more of these Hilbert series in the same manner. The result of this and the next computation is summarized in Figure III.4.

```
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -3, 3 );
gap> E2_m3_3 := UnderlyingHonestObject( Source( s ) );
gap> HilbertSeries( E2_m3_3 );
5*t^2+15*t+10+2*t^-1
gap> s :=
```

```

> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -4, 3 );;
gap> E2_m4_3 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_m4_3 );
4*t^-1
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -5, 3 );;
gap> E2_m5_3 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_m5_3 );
15*t^-2+35*t^-3+20*t^-4
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -6, 3 );;
gap> E2_m6_3 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_m6_3 );
2*t^-6
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -6, 4 );;
gap> E2_m6_4 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_m6_4 );
4*t^-1
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -7, 4 );;
gap> E2_m7_4 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_m7_4 );
15*t^-2+35*t^-3+20*t^-4
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, -8, 4 );;
gap> E2_m8_4 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_m8_4 );
2*t^-6

```

Analogously, we can follow the table to the right.

```

gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, 1, 1 );;
gap> E2_1_1 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_1_1 );
2*t^4+10*t^3+15*t^2+5*t
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, 2, 1 );;
gap> E2_2_1 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_2_1 );
4*t^4
gap> s :=

```

```

> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, 3, 1 );;
gap> E2_3_1 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_3_1 );
20*t^7+35*t^6+15*t^5
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, 4, 1 );;
gap> E2_4_1 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_4_1 );
2*t^9
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, 4, 0 );;
gap> E2_4_0 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_4_0 );
4*t^4
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, 5, 0 );;
gap> E2_5_0 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_5_0 );
20*t^7+35*t^6+15*t^5
gap> s :=
> SpectralSequenceEntryOfDescendingFilteredCocomplex( T, 2, 6, 0 );;
gap> E2_6_0 := UnderlyingHonestObject( Source( s ) );;
gap> HilbertSeries( E2_6_0 );
2*t^9

```

From the third page on, all objects  $E_r^{p,q}$  are zero, so we are done with the computation of the spectral cohomology table. But note that since we never omitted the  $H$ -equivariant structure, we actually got an equivariant version of the spectral cohomology table, which we also wrote down in Figure III.5.

The algorithm for computing spectral sequences is implemented in CAP for arbitrary abelian categories. In particular, it can be also applied in the context of graded modules over the graded symmetric algebra  $S$  (which is also available in CAP, see [Gut17] for implementation details). So, we can also use the computational capabilities of CAP in the future for constructing and investigating spectral Betti tables, with special regard to a possible relation between spectral Betti tables and spectral cohomology tables inspired by Boij-Söderberg theory.

FIGURE 4. Second page of the spectral cohomology table of the Horrocks-Mumford bundle.

4	$2t^{-6}$	$a$	$4t^{-1}$												
3			$2t^{-6}$	$a$	$4t^{-1}$	$b$									
2							$b+c$								
1									$c$	$4t^4$	$d$	$2t^9$			
0												$4t^4$	$d$	$2t^9$	
$q/p$	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6

where

$$a := 15t^{-2} + 35t^{-3} + 20t^{-4},$$

$$b := 5t^2 + 15t + 10 + 2t^{-1},$$

$$c := 2t^4 + 10t^3 + 15t^2 + 5t,$$

$$d := 20t^7 + 35t^6 + 15t^5.$$

FIGURE 5. Second page of the equivariant spectral cohomology table of the Horrocks-Mumford bundle.

4	$\chi_5 t^{-6}$	$a$	$\sigma t^{-1}$												
3			$\chi_5 t^{-6}$	$a$	$\sigma t^{-1}$	$b$									
2							$b + c$								
1									$c$	$\sigma t^4$	$d$	$\chi_5 t^9$			
0												$\sigma t^4$	$d$	$\chi_5 t^9$	
$q/p$	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6

where

$$\sigma := \sum_{i=1}^4 \chi_i$$

$$a := (\chi_{10} + \chi_{14} + \chi_{18})t^{-2} + (2\chi_7 + \chi_{11} + \chi_{15} + \chi_{19} + \chi_{27})t^{-3} + (2\chi_{27})t^{-4},$$

$$b := \chi_7 t^2 + (\chi_{12} + \chi_{16} + \chi_{20})t + \chi_{28} t^0 + \chi_5 t^{-1},$$

$$c := \chi_5 t^4 + \chi_{25} t^3 + (\chi_{11} + \chi_{15} + \chi_{19})t^2 + \chi_8 t,$$

$$d := (2\chi_{26})t^7 + (2\chi_8 + \chi_{12} + \chi_{16} + \chi_{20})t^6 + (\chi_{13} + \chi_{17} + \chi_{21})t^5.$$





## List of Figures

1	A diamond with epimorphisms and monomorphisms.	99
2	There exist six paths from $A$ to $B$ .	100
3	A Hasse diagram depicting subobjects and quotient objects associated to a generalized morphism $\gamma$ .	109
4	Second page of the spectral cohomology table of the Horrocks-Mumford bundle.	164
5	Second page of the equivariant spectral cohomology table of the Horrocks-Mumford bundle.	165



## Bibliography

- [AM69] M. F. Atiyah and I. G. Macdonald, *Introduction to commutative algebra*, Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1969. MR 0242802 [156](#)
- [Bar09a] Mohamed Barakat, *The homomorphism theorem and effective computations*, Habilitation thesis, Department of Mathematics, RWTH-Aachen University, April 2009. [85](#), [130](#)
- [Bar09b] Mohamed Barakat, *Spectral filtrations via generalized morphisms*, submitted ([arXiv:0904.0240](#)) (v2 in preparation), 2009. [85](#), [109](#)
- [BCP97] Wieb Bosma, John J. Cannon, and Catherine Playoust, *The Magma algebra system. I. The user language*, JSC **24** (1997), no. 3–4, 235–266, Computational algebra and number theory (London, 1993). MR 1 484 478 [81](#)
- [BGG78] I. N. Bernšteĭn, I. M. Gel'fand, and S. I. Gel'fand, *Algebraic vector bundles on  $\mathbf{P}^n$  and problems of linear algebra*, Funktsional. Anal. i Prilozhen. **12** (1978), no. 3, 66–67. MR MR509387 (80c:14010a) [9](#), [128](#), [143](#)
- [BLH11] Mohamed Barakat and Markus Lange-Hegermann, *An axiomatic setup for algorithmic homological algebra and an alternative approach to localization*, J. Algebra Appl. **10** (2011), no. 2, 269–293, ([arXiv:1003.1943](#)). MR 2795737 (2012f:18022) [11](#), [81](#)
- [BLH14] Mohamed Barakat and Markus Lange-Hegermann, *Gabriel morphisms and the computability of Serre quotients with applications to coherent sheaves*, ([arXiv:1409.2028](#)), 2014. [85](#), [104](#)
- [BLH17] Mohamed Barakat and Markus Lange-Hegermann, *A constructive approach to the module of twisted global sections on relative projective spaces*, Algorithmic and Experimental Methods in Algebra, Geometry, and Number Theory, Springer, 2017, To appear, [arXiv:1409.6100](#). [127](#)
- [BP69] Hans-Berndt Brinkmann and Dieter Puppe, *Abelsche und exakte Kategorien, Korrespondenzen*, Lecture Notes in Mathematics, Vol. 96, Springer-Verlag, Berlin-New York, 1969. MR 0269713 [85](#)
- [Dab11] V. Dabbaghian, *Repsn, a gap4 package for constructing representations of finite groups, Version 3.0.2*, <http://www.sfu.ca/~vdabbagh/gap/repsn.html>, Aug 2011, Refereed GAP package. [56](#)
- [Day74] Brian Day, *On closed categories of functors. II*, Category Seminar (Proc. Sem., Sydney, 1972/1973), Springer, Berlin, 1974, pp. 20–54. Lecture Notes in Math., Vol. 420. MR 0360747 (50 #13194) [56](#)
- [Del90] P. Deligne, *Catégories tannakiennes*, The Grothendieck Festschrift, Vol. II, Progr. Math., vol. 87, Birkhäuser Boston, Boston, MA, 1990, pp. 111–195. MR 1106898 (92d:14002) [54](#), [138](#)
- [Dix93] John D. Dixon, *Constructing representations of finite groups*, Groups and computation (New Brunswick, NJ, 1991), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 11, Amer. Math. Soc., Providence, RI, 1993, pp. 105–112. MR 1235797 [70](#)
- [EFS03] David Eisenbud, Gunnar Fløystad, and Frank-Olaf Schreyer, *Sheaf cohomology and free resolutions over exterior algebras*, Trans. Amer. Math. Soc. **355** (2003), no. 11, 4397–4426 (electronic). MR MR1990756 (2004f:14031) [9](#), [143](#), [144](#), [146](#)
- [EG01] Pavel Etingof and Shlomo Gelaki, *Isocategorical groups*, Internat. Math. Res. Notices (2001), no. 2, 59–76. MR 1810480 [60](#)

- [EGNO15] Pavel Etingof, Shlomo Gelaki, Dmitri Nikshych, and Victor Ostrik, *Tensor categories*, Mathematical Surveys and Monographs, vol. 205, American Mathematical Society, Providence, RI, 2015. MR 3242743 [18](#)
- [ES09] David Eisenbud and Frank-Olaf Schreyer, *Betti numbers of graded modules and cohomology of vector bundles*, J. Amer. Math. Soc. **22** (2009), no. 3, 859–888. MR 2505303 (2011a:13024) [159](#)
- [Flø12] Gunnar Fløystad, *Boij-Söderberg theory: introduction and survey*, Progress in commutative algebra 1, de Gruyter, Berlin, 2012, pp. 1–54. MR 2932580 [157](#)
- [FS90] Peter J. Freyd and Andre Scedrov, *Categories, allegories*, North-Holland Mathematical Library, vol. 39, North-Holland Publishing Co., Amsterdam, 1990. MR 1071176 (93c:18001) [84](#)
- [GAP14] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.7.5*, 2014, (<http://www.gap-system.org>). [9](#)
- [Gut17] Sebastian Gutsche, *Constructive category theory with applications to algebraic geometry*, Ph.D. thesis, University of Siegen, 2017. [11](#), [163](#)
- [Hil66] Peter Hilton, *Correspondences and exact squares*, Proc. Conf. Categorical Algebra (La Jolla, Calif., 1965), Springer, New York, 1966, pp. 254–271. MR 0204487 [85](#)
- [HM73] G. Horrocks and D. Mumford, *A rank 2 vector bundle on  $\mathbf{P}^4$  with 15,000 symmetries*, Topology **12** (1973), 63–81. MR 0382279 (52 #3164) [80](#)
- [hom17] homalg project authors, *The homalg project – Algorithmic Homological Algebra*, (<http://homalg-project.github.io>), 2003–2017. [81](#)
- [Joh02] Peter T. Johnstone, *Sketches of an elephant: a topos theory compendium. Vol. 1*, Oxford Logic Guides, vol. 43, The Clarendon Press, Oxford University Press, New York, 2002. MR 1953060 (2003k:18005) [84](#), [90](#), [91](#)
- [Kel05] G. M. Kelly, *Basic concepts of enriched category theory*, Repr. Theory Appl. Categ. (2005), no. 10, vi+137, Reprint of the 1982 original [Cambridge Univ. Press, Cambridge; MR0651714]. MR 2177301 [17](#), [22](#), [35](#)
- [Mit65] Barry Mitchell, *Theory of categories*, Pure and Applied Mathematics, Vol. XVII, Academic Press, New York-London, 1965. MR 0202787 [83](#)
- [ML71] Saunders Mac Lane, *Categories for the working mathematician*, Graduate Texts in Mathematics, no. 5, Springer-Verlag, 1971. [19](#), [23](#), [47](#), [50](#), [51](#), [52](#), [53](#), [83](#), [133](#), [134](#)
- [OSS11] Christian Okonek, Michael Schneider, and Heinz Spindler, *Vector bundles on complex projective spaces*, Modern Birkhäuser Classics, Birkhäuser/Springer Basel AG, Basel, 2011, Corrected reprint of the 1980 edition, With an appendix by S. I. Gelfand. MR 2815674 (2012d:14073) [143](#)
- [Pos13] Sebastian Posur, *G-equivariant coherent sheaf cohomology*, Master’s thesis, RWTH Aachen University, March 2013. [146](#)
- [Pup62] Dieter Puppe, *Korrespondenzen in abelschen Kategorien*, Math. Ann. **148** (1962), 1–30. MR 0141698 [85](#)
- [Sel11] P. Selinger, *A survey of graphical languages for monoidal categories*, New structures for physics, Lecture Notes in Phys., vol. 813, Springer, Heidelberg, 2011, pp. 289–355. MR 2767048 (2012j:18011) [53](#), [65](#)
- [SR72] Neantro Saavedra Rivano, *Catégories Tannakiennes*, Lecture Notes in Mathematics, Vol. 265, Springer-Verlag, Berlin-New York, 1972. MR 0338002 (49 #2769) [50](#), [51](#), [58](#), [61](#), [63](#)
- [Sta16] The Stacks Project Authors, *stacks project*, <http://stacks.math.columbia.edu>, 2016. [83](#)
- [Uni13] The Univalent Foundations Program, *Homotopy type theory: Univalent foundations of mathematics*, <http://homotopytypetheory.org/book>, Institute for Advanced Study, 2013. [17](#)
- [Wei94] Charles A. Weibel, *An introduction to homological algebra*, Cambridge Studies in Advanced Mathematics, vol. 38, Cambridge University Press, Cambridge, 1994. MR MR1269324 (95f:18001) [15](#), [83](#), [119](#), [123](#)

## Index

- $G$ -equivariant cohomology table of  $\mathcal{F}$ , 147
- $\mathbb{Z}$ -graded representations of  $G$ , 69
- CAP category, 13
  
- Ab-category, 34
- Ab-functor, 34
- Ab-natural transformation, 34
- Ab-product category, 45
- abelian category, 39
- additive category, 35
- additive relation, 85
- associated  $\mathbb{Z}$ -graded category of  $\mathbf{C}$ , 117
- associated honest cochain complex of  $(A^\bullet, d^\bullet)$ , 118
- associated morphism of  $\gamma$ , 111
- associated relation, 89
- associator, 48
  
- BGG correspondence, 146
- bicartesian, 96
- bifunctors, 44
- bilinear bifunctor, 44
- braided monoidal functor, 52
- braiding, 51
  
- category, 19
- category having cokernels, 12
- category of 3-arrows from  $A$  to  $B$ , 100
- category of 4-arrows from  $A$  to  $B$ , 101
- category of ascending filtered objects of  $\mathbf{A}$ , 130
- category of ascending filtrations, 120
- category of chain complexes, 117
- category of cochain complexes, 117
- category of comonoids, 134
- category of cospans from  $A$  to  $B$ , 100
- category of descending filtered objects of  $\mathbf{A}$ , 129
- category of descending filtrations, 120
- category of diamonds from  $A$  to  $B$ , 101
- category of elements, 24
- category of left  $A$ -comodules, 135
- category of monoids, 133
- category of relations, 86
- category of reversed 3-arrows from  $A$  to  $B$ , 101
- category of reversed 4-arrows from  $A$  to  $B$ , 101
- category of right  $A$ -modules, 135
- category of spans from  $A$  to  $B$ , 87
- category of spans of  $\mathbf{A}$ , 88
- category of Tate sequences, 145
- category with bifunctor, 44
- chain complexes, 117
- chain maps, 117
- closed category, 53
- cochain complexes, 117
- cochain maps, 117
- codefect of  $\gamma$ , 109
- codomain of  $\gamma$ , 108
- cofree resolution, 143
- cogenerator function, 143
- cohomological spectral sequence, 119
- coimage, 27
- cokernel of  $\alpha$ , 37
- colift of  $\tau$  along  $\epsilon$ , 39
- comma category, 25
- comonoid in  $\mathbf{A}$ , 133
- compact closed category, 53
- components of a natural transformation, 20
- coproduct category, 41
- coslice category, 25
- crossed product ring, 136
  
- defect of  $\gamma$ , 109
- dependent function, 19
- dependent sum category, 24
- dependent type, 19, 22
- diagonal difference, 40
- dinatural transformation, 23
- direct sum, 35

- discrete natural family, 21
- domain of  $\gamma$ , 108
- dual object, 53
- dual object of  $A$ , 53
- epi factorization, 27
- epi-mono factorization of  $\alpha : A \rightarrow B$ , 27
- epimorphism, 26
- equivalence, 21
- equivalence of categories with bifunctor, 45
- exact functor, 38
- exact pairing, 53
- exterior algebra of  $V$ , 139
- forgetful functor, 140, 142
- free resolutions, 141
- full codomain, 110
- full domain, 111
- functor, 20
- functor between categories with bifunctor, 44
- generalized chain complex, 117
- generalized cochain complex, 117
- generalized coimage of  $\gamma$ , 109
- generalized cokernel of  $\gamma$ , 109
- generalized differential, 117
- generalized image of  $\gamma$ , 109
- generalized kernel of  $\gamma$ , 109
- generalized morphism, 91
- generalized morphism category of  $\mathbf{A}$ , 90
- generator function, 141
- graph, 85
- Hilbert series, 155
- honest, 111
- horizontal composition, 21
- image, 26
- induced spectral sequence of  $F$  and  $A$ , 131
- initial object, 23
- injection, 35
- interchange law, 44
- internal Hom, 53
- isomorphism, 20
- kernel of  $\alpha$ , 37
- left  $A$ -comodule, 135
- left coaction of  $A$  on  $M$ , 135
- left unitor, 48
- lift of  $\tau$  along  $\iota$ , 39
- limit of  $D$ , 29
- mono factorization of  $\alpha$ , 26
- monoid in  $\mathbf{A}$ , 133
- monoidal category, 47
- monoidal equivalence, 50
- monoidal functor, 49
- monoidal natural transformation, 50
- monomorphism, 26
- natural ascending filtrations, 130
- natural dependent function, 22
- natural descending filtration, 130
- natural isomorphism, 21
- natural transformation, 20
- null homotopic morphism, 123
- pre-abelian category, 38
- projection, 35
- pseudo-inverse, 86, 94
- pullback computation rule, 95
- pushout computation rule, 95
- quotient object of  $A$ , 28
- representation category of  $G$ , 55
- right  $A$ -module, 135
- right action of  $A$  on  $M$ , 134
- right unitor, 48
- rigid symmetric monoidal category, 53
- root sequence of length  $s$ , 158
- row convention, 36
- single-valued, 110
- skeletal, 69
- slice category, 25
- source, 29
- spectral  $F$ -Betti table of  $A$ , 131, 132
- spectral cohomology table of  $\mathcal{F}$ , 155
- stable cospan, 103
- stable diamonds, 102
- stable module category, 145
- stable span, 92
- stably equivalent, 89
- string diagrams, 53
- subobject of  $A$ , 28
- subquotient embedding, 108
- subquotient of  $A$ , 107
- subquotient projection, 108
- supernatural cohomology of type  $\mathbf{z}$ , 158
- symmetric monoidal category, 53
- Tate sequences, 145

tensor category over  $k$ , 54  
tensor unit, 47  
total, 111  
type, 19, 22  
  
universal epi-mono factorization of  $\alpha : A \rightarrow B$ ,  
27  
  
vertical composition, 20  
  
zero object, 35