

# Sha-Categories & Quasi-isomorphisms

Thomas Wiskow Thorbjørnsen

November 18, 2021

## **Abstract**

*Fill inn abstract*

## **Sammendrag**

*Fyll inn sammendraget*

## **Acknowledgements**

*Thank the people in your life who has made this journey easier :D*

# Contents



# Chapter 1

## Bar and Cobar Construction

[../thesis.tex]subfiles

### 1.1 Algebras, Coalgebras and Twisting Morphisms

In this section we will look at a result of associative algebras over a field  $\mathbb{K}$ . Given a coassociative conilpotent coalgebra  $C$  and an associative algebra  $A$ , we say that a linear transformation  $\alpha : C \rightarrow A$  is twisting if it satisfies the Maurar-Cartan equation:

$$\partial \alpha + \alpha \star \alpha = 0.$$

Let  $Tw(C, A)$  be the set of twisting morphisms, then considering it as a functor  $Tw : Coalg^{op} \times Alg \rightarrow Ab$  we want to show that it is represented in both arguments. Moreover, this representation give rise to an adjoint pair of functors, called the Bar and Cobar construction.

$$\begin{array}{ccc} & B & \\ \text{Alg} & \xrightarrow{\quad} & \text{Conil} \\ & \Upsilon & \text{Coalg} \\ & \xleftarrow{\quad} & \\ & \Omega & \end{array}$$

To obtain this result we need to define a twisting morphism. Thus this section will define algebras, coalgebras and convolution algebras before we state the result of the Bar and Cobar construction.

#### 1.1.1 Algebras

In this subsection we will define different types of associative algebras. The class of our main concerns are the unital associative algebras, and these will be referred to as algebras. The collection of algebras together with homomorphisms between them form the category  $Alg$  of algebras. Other types of algebras such as augmented and tensor algebras will be defined as well.

**Definition 1.1.1** (Algebra). Let  $\mathbb{K}$  be a field. An algebra  $A$  is a  $\mathbb{K}$ -module with structure morphisms called multiplication and unit,

$$\begin{aligned} (\cdot_A) : A \otimes_{\mathbb{K}} A &\rightarrow A \\ \nu_A : \mathbb{K} &\rightarrow A, \end{aligned}$$

satisfying the associativity and identity laws.

$$\begin{aligned} (\text{associativity}) \quad (a \cdot_A b) \cdot_A c &= a \cdot_A (b \cdot_A c) \\ (\text{unitality}) \quad \nu_A(1) \cdot_A a &= a = a \cdot_A \nu_A(1) \end{aligned}$$

*Remark 1.1.2.* Whenever  $A$  does not possess a unit morphism, we will call  $A$  a non-unital algebra. Only the associativity law must hold.

Alternatively, instead of using equations, we may represent the laws with commutative diagrams.

$$\begin{array}{ccc} & A \otimes_{\mathbb{K}} A \otimes_{\mathbb{K}} A & \xrightarrow{(\cdot_A) \otimes id_{\mathbb{K}}} A \otimes_{\mathbb{K}} A \\ (\text{associativity}) & \downarrow id_{\mathbb{K}} \otimes (\cdot_A) & \downarrow (\cdot_A) \\ & A \otimes_{\mathbb{K}} A & \xrightarrow{(\cdot_A)} A \\ (\text{unitality}) & A \otimes_{\mathbb{K}} \mathbb{K} \xrightarrow{id_A \otimes \nu_A} A \otimes_{\mathbb{K}} A \xleftarrow{\nu_A \otimes id_A} \mathbb{K} \otimes_{\mathbb{K}} A & \\ & \searrow \simeq \downarrow (\cdot_A) \swarrow \simeq & \\ & A & \end{array}$$

We may also present the structure of algebras by electric circuits. Such circuits are read from top to bottom, where morphisms are composed by lines. Morphisms in such diagrams may be highlighted with figures, conjunctions or twistings. E.g. The multiplication operator may be represented as a converging fork, and the unit as a source.

$$\begin{array}{ccc} (\text{Multiplication}) & \text{Diagram: circle with } \cdot_A \text{ and two input lines} & = \text{Diagram: converging fork} \\ (\text{Unit}) & \text{Diagram: circle with } \nu_A \text{ and one output line} & = \text{Diagram: source (vertical line with a dot)} \end{array}$$

With these operators we obtain the electric laws for an algebra.

$$\begin{array}{ccc} (\text{Associativity}) & \text{Diagram: fork then join} & = \text{Diagram: join then fork} \\ (\text{unitality}) & \text{Diagram: source then join} & = \text{Diagram: vertical line} = \text{Diagram: join then source} \end{array}$$

**Definition 1.1.3** (Algebra homomorphisms). Let  $A$  and  $B$  be algebras. Then  $f : A \rightarrow B$  is an algebra homomorphism if

1.  $f$  is  $\mathbb{K}$ -linear

2.  $f(ab) = f(a)f(b)$
3.  $f(\nu_A) = \nu_B$

Whenever  $A$  and  $B$  are non-unital, we only require 1 and 2 for a homomorphism of non-unital algebras.

**Definition 1.1.4** (Categories of algebras). • Let  $Alg$  denote the category of algebras. Its objects consists of every algebra  $A$ , and the morphisms are algebra homomorphisms. The sets of morphisms between  $A$  and  $B$  are denoted as  $Alg(A, B)$ .  
• Let  $nAlg$  denote the category of non-unital algebras. Its objects consists of every non-unital algebra  $A$ , and the morphisms are non-unital algebra homomorphisms. The sets of morphisms between  $A$  and  $B$  are denoted as  $nAlg(A, B)$ .

**Definition 1.1.5** (Augmented algebras). Let  $A$  be an algebra. It is called augmented if there is an algebra homomorphism  $\varepsilon : A \rightarrow \mathbb{K}$ .

As a module  $\mathbb{K}$  is free, so if  $A$  is an augmented algebra, then it decomposes into  $\mathbb{K} \oplus Ker \varepsilon$  as a module. The kernel of  $\varepsilon$  is called the augmentation ideal and we will denote it as  $\bar{A}$ . This kernel is an equivalence of categories between augmented algebras and non-unital algebras, with unitization as the quasi-inverse.

**Definition 1.1.6.** Define tensor algebra

Observe that the tensor algebra is augmented. Define the reduced tensor algebra as the augmentation ideal.

**Proposition 1.1.7.** *Tensor algebra is free algebra, reduced tensor algebra is free non-unital algebra.*

**Definition 1.1.8.** Define a left-module over an algebra

**Proposition 1.1.9.** *Find free modules*

## 1.1.2 Coalgebras

### 1.1.3 Derivations, Coderivations and Convolution Algebras

### 1.1.4 Twisting Morphisms

## 1.2 Strongly Homotopy Associative Algebras, Coalgebras and Twisting Morphisms

### 1.2.1 Sha Algebras

### 1.2.2 Sha Coalgebras

### 1.2.3 Twisting Sha Morphisms