Constructive Category Theory and Tilting Equivalences via Strong Exceptional Sequences

Kamal Saleh

University of Siegen

March 09, 2022

Overview

Overview

Overview

Section 1

Constructive Categories and Towers of Categories

Categories

Definition

A category $\mathcal{C}$ consists of the following data:

Categories

Definition

A category $\mathcal{C}$ consists of the following data:

1. Obj

$A \qquad B \qquad C$

Categories

Definition

A category $\mathcal{C}$ consists of the following data:

1. Obj
2. $\mathrm{Hom}(A, B)$

$$A \longrightarrow B \qquad\qquad C$$

Categories

Definition

A category $\mathcal{C}$ consists of the following data:

1. Obj
2. $\mathrm{Hom}(A, B)$

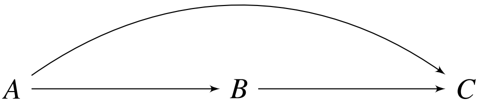
$$A \longrightarrow B \longrightarrow C$$

Categories

Definition

A category $\mathcal{C}$ consists of the following data:

1. Obj
2. $\mathrm{Hom}(A, B)$
3. $\bullet : \mathrm{Hom}(A, B) \times \mathrm{Hom}(B, C) \to \mathrm{Hom}(A, C)$

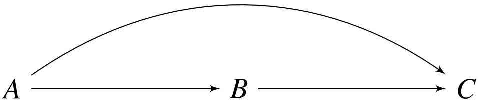


Categories

Definition

A category $\mathcal{C}$ consists of the following data:

1. Obj
2. $\mathrm{Hom}(A, B)$
3. $\bullet : \mathrm{Hom}(A, B) \times \mathrm{Hom}(B, C) \to \mathrm{Hom}(A, C)$
4. Associative property

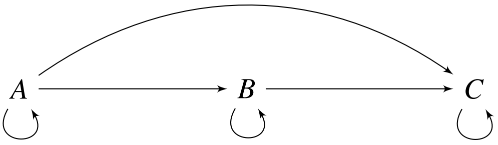


Categories

Definition

A category $\mathcal{C}$ consists of the following data:

1. Obj
2. $\mathrm{Hom}(A, B)$
3. $\bullet : \mathrm{Hom}(A, B) \times \mathrm{Hom}(B, C) \to \mathrm{Hom}(A, C)$
4. Associative property
5. neutral element: $\mathrm{id}_A \in \mathrm{Hom}(A, A)$



Constructive categories

Constructive categories

A category becomes **constructive** through

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

1 2 1

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

1 2 1

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

$$1 \xrightarrow[\begin{pmatrix} 1 & 2 \end{pmatrix}]{} 2 \xrightarrow[\begin{pmatrix} 3 \\ 4 \end{pmatrix}]{} 1$$

Constructive categories

A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

$$1 \xrightarrow[\begin{pmatrix} 1 & 2 \end{pmatrix}]{} 2 \xrightarrow[\begin{pmatrix} 3 \\ 4 \end{pmatrix}]{} 1$$

Constructive categories

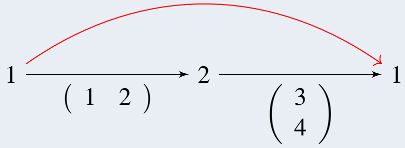
A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$



Constructive categories

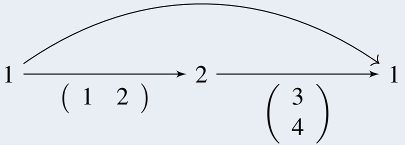
A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$



Constructive categories

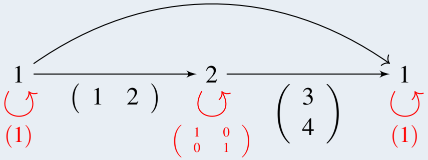
A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$



Constructive categories

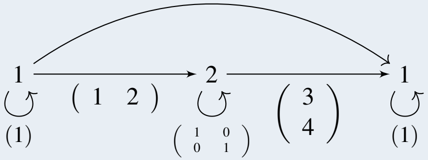
A category becomes **constructive** through

- data structures for *objects* and *morphisms*,
- algorithms to compute the *composition* of morphisms and *identity morphisms* of objects.

A constructive category **has decidable equality of morphisms** through an algorithm which for A, B and $\alpha, \beta \in \mathrm{Hom}(A, B)$ decides whether $\alpha = \beta$.

Example: $\mathbb{Q}$-mat

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$



Categorical Doctrines

- We use the term **categorical doctrine** to describe categories with specified additional properties or structures.

Categorical Doctrines

- We use the term **categorical doctrine** to describe categories with specified additional properties or structures.

Example: The doctrine of pre-Abelian Categories

The doctrine of pre-Abelian categories consists of categories in which

1. finite biproduct exists,
2. each morphism has an additive inverse,
3. kernels and cokernels exists.

Categorical Doctrines

- We use the term **categorical doctrine** to describe categories with specified additional properties or structures.

Example: The doctrine of pre-Abelian Categories

The doctrine of pre-Abelian categories consists of categories in which

1. finite biproduct exists,
2. each morphism has an additive inverse,
3. kernels and cokernels exists.

- A category $\mathcal{P}$ is **constructive as an instance** of a doctrine $\mathcal{D}$ if all disjunctions ($\vee$) and all existential quantifiers ($\exists$) in the defining axioms of the doctrine $\mathcal{D}$ are realized in $\mathcal{P}$ by algorithms.

Categorical Doctrines

- We use the term **categorical doctrine** to describe categories with specified additional properties or structures.

Example: The doctrine of pre-Abelian Categories

The doctrine of pre-Abelian categories consists of categories in which

1. finite biproduct exists,
2. each morphism has an additive inverse,
3. kernels and cokernels exists.

- A category $\mathcal{P}$ is **constructive as an instance** of a doctrine $\mathcal{D}$ if all disjunctions ($\vee$) and all existential quantifiers ($\exists$) in the defining axioms of the doctrine $\mathcal{D}$ are realized in $\mathcal{P}$ by algorithms.

The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

$$A \xrightarrow{\varphi} B$$

The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

. . . one have an object $\ker \varphi$,

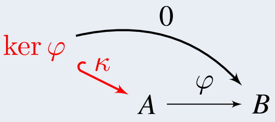
$$\ker \varphi$$

$$A \xrightarrow{\varphi} B$$

The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

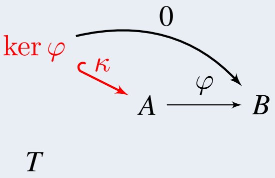
... one have an object $\ker \varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,



The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

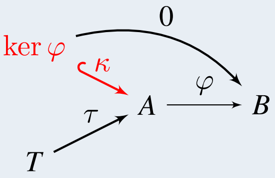
... one have an object $\ker \varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ



The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

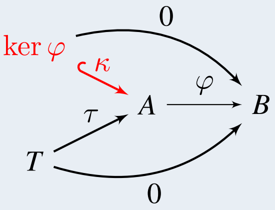
... one have an object $\ker \varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ



The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

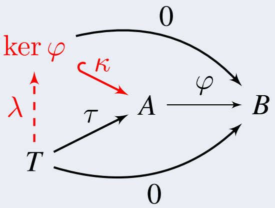
... one have an object $\ker\varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ



The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

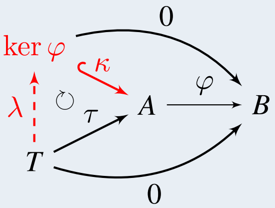
… one have an object $\ker\varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ
a unique morphism $\lambda = \mathrm{KernelLift}(\varphi, \tau)$



The existential quantifiers of “kernels”

Let $\varphi \in \mathrm{Hom}(A, B)$. The existence of a kernel to φ means

… one have an object $\ker \varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ
a unique morphism $\lambda = \mathrm{KernelLift}(\varphi, \tau)$, such that



Category Constructors in CAP

CAP stands for *Categories, Algorithms and Programming*, is a software project for constructive category theory written in the computer algebra system GAP.

Category Constructors in CAP

CAP stands for *Categories, Algorithms and Programming*, is a software project for constructive category theory written in the computer algebra system GAP.
Two of the main design features of CAP are its support for:

Category Constructors in CAP

CAP stands for *Categories, Algorithms and Programming*, is a software project for constructive category theory written in the computer algebra system GAP.

Two of the main design features of CAP are its support for:

- defining *categorical doctrines*;

Category Constructors in CAP

CAP stands for *Categories, Algorithms and Programming*, is a software project for constructive category theory written in the computer algebra system GAP.

Two of the main design features of CAP are its support for:

- defining *categorical doctrines*;
- building *category constructors* which create instances of such doctrines.

Category Constructors in CAP

CAP stands for *Categories, Algorithms and Programming*, is a software project for constructive category theory written in the computer algebra system GAP.
Two of the main design features of CAP are its support for:

- defining *categorical doctrines*;
- building *category constructors* which create instances of such doctrines.

A **category constructor**, as the name suggests, is a procedure which outputs a category $\mathcal{P}$ as an instance of a doctrine $\mathbf{D}$. The constructor

Category Constructors in CAP

CAP stands for *Categories, Algorithms and Programming*, is a software project for constructive category theory written in the computer algebra system GAP.
Two of the main design features of CAP are its support for:

- defining *categorical doctrines*;
- building *category constructors* which create instances of such doctrines.

A **category constructor**, as the name suggests, is a procedure which outputs a category $\mathcal{P}$ as an instance of a doctrine $\mathbf{D}$. The constructor

- specifies the data structure of the objects and morphisms of the output category,

Category Constructors in CAP

CAP stands for *Categories, Algorithms and Programming*, is a software project for constructive category theory written in the computer algebra system GAP.

Two of the main design features of CAP are its support for:

- defining *categorical doctrines*;
- building *category constructors* which create instances of such doctrines.

A **category constructor**, as the name suggests, is a procedure which outputs a category $\mathcal{P}$ as an instance of a doctrine $\mathbf{D}$. The constructor

- specifies the data structure of the objects and morphisms of the output category,
- expresses all defining categorical operations of the doctrine $\mathbf{D}$ (of its output category) as algorithms written in terms of the provided input data.

Examples: An acyclic quiver $\mathfrak{q}$ and a field k

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.

$\mathfrak{q}$ k

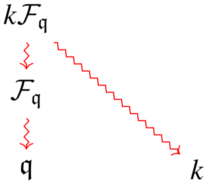
Examples: The free category over $\mathfrak{q}$

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.

$$\begin{array}{ccc} \mathcal{F}_{\mathfrak{q}} & & \\ \big\downarrow & & \\ \mathfrak{q} & & k \end{array}$$

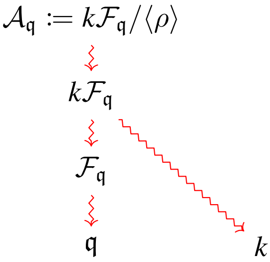
Examples: The k-linear closure category

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



Examples: The quotient category by two-sided ideal

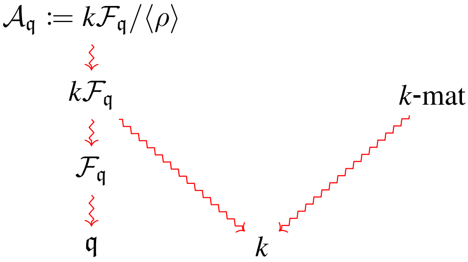
Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



ρ is a finite set in $\mathrm{Mor}(k\mathcal{F}_{\mathfrak{q}})$.

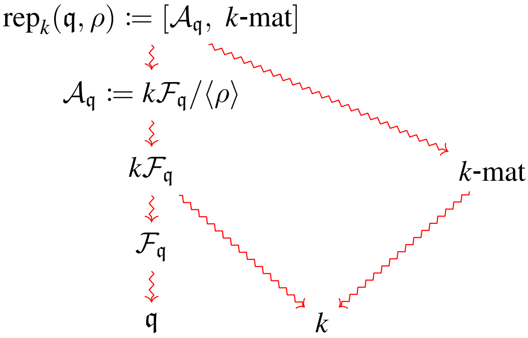
Examples: The category of matrices

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



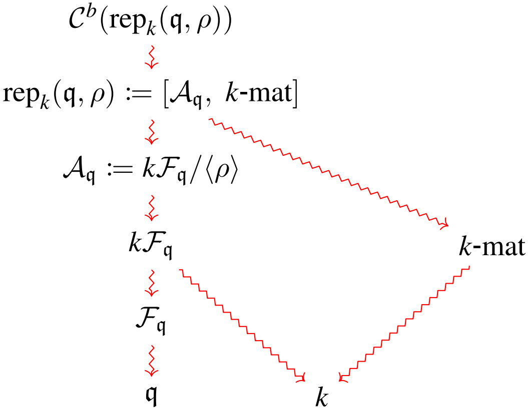
Examples: The f.d. ρ-bounded quiver representations

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



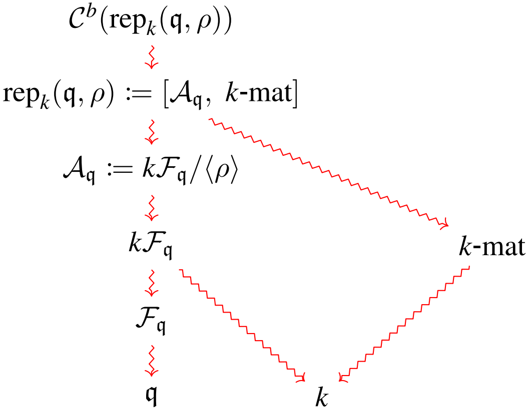
Examples: The category of bounded complexes

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



Examples: The category of bounded complexes

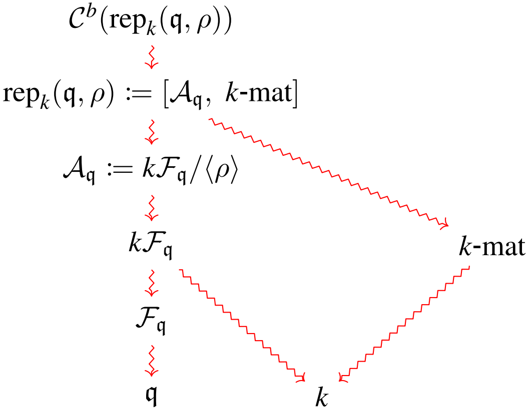
Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



They all have decidable equality of morphisms.

Examples: The bounded homotopy category?

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



What about the bounded homotopy category $\mathcal{K}^b(\mathrm{rep}_k(\mathfrak{q}, \rho))$?

The bounded homotopy category

- Let $\mathcal{P}$ be an additive category and let $\varphi\colon A \to B$ be in $\mathcal{C}^b(\mathcal{P})$.

The bounded homotopy category

- Let $\mathcal{P}$ be an additive category and let $\varphi\colon A \to B$ be in $\mathcal{C}^b(\mathcal{P})$.

$$\cdots \longrightarrow A^{i-1} \xrightarrow{\partial_A^{i-1}} A^i \xrightarrow{\partial_A^i} A^{i+1} \longrightarrow \cdots$$

The bounded homotopy category

- Let $\mathcal{P}$ be an additive category and let $\varphi\colon A \to B$ be in $\mathcal{C}^b(\mathcal{P})$.

$$\cdots \longrightarrow A^{i-1} \xrightarrow{\partial_A^{i-1}} A^{i} \xrightarrow{\partial_A^{i}} A^{i+1} \longrightarrow \cdots$$

$$\cdots \longrightarrow B^{i-1} \xrightarrow[\partial_B^{i-1}]{} B^{i} \xrightarrow[\partial_B^{i}]{} B^{i+1} \longrightarrow \cdots$$

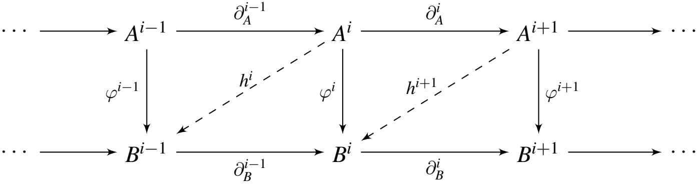
The bounded homotopy category

- Let $\mathcal{P}$ be an additive category and let $\varphi\colon A \to B$ be in $\mathcal{C}^b(\mathcal{P})$.

$$
\begin{array}{ccccccccc}
\cdots & \longrightarrow & A^{i-1} & \xrightarrow{\partial_A^{i-1}} & A^i & \xrightarrow{\partial_A^{i}} & A^{i+1} & \longrightarrow & \cdots \\
 & & \Big\downarrow{\scriptstyle \varphi^{i-1}} & \circlearrowleft & \Big\downarrow{\scriptstyle \varphi^{i}} & \circlearrowleft & \Big\downarrow{\scriptstyle \varphi^{i+1}} & & \\
\cdots & \longrightarrow & B^{i-1} & \xrightarrow[\partial_B^{i-1}]{} & B^i & \xrightarrow[\partial_B^{i}]{} & B^{i+1} & \longrightarrow & \cdots
\end{array}
$$

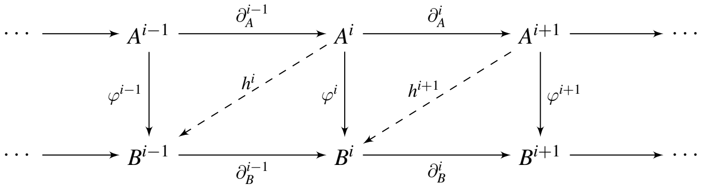
The bounded homotopy category

- Let $\mathcal{P}$ be an additive category and let $\varphi\colon A \to B$ be in $\mathcal{C}^b(\mathcal{P})$. Then φ is called **null-homotopic** if there exists a family of morphisms $\left(h^i\colon A^i \to B^{i-1}\right)_{i\in\mathbb{Z}}$ such that $\partial_A^i \cdot h^{i+1} + h^i \cdot \partial_B^{i-1} = \varphi^i$ for all $i \in \mathbb{Z}$.



The bounded homotopy category

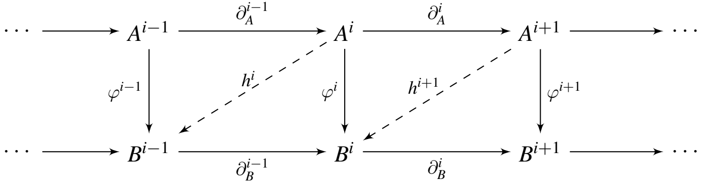
- Let $\mathcal{P}$ be an additive category and let $\varphi\colon A \to B$ be in $\mathcal{C}^b(\mathcal{P})$. Then φ is called **null-homotopic** if there exists a family of morphisms $\left(h^i\colon A^i \to B^{i-1}\right)_{i\in\mathbb{Z}}$ such that $\partial_A^i \bullet h^{i+1} + h^i \bullet \partial_B^{i-1} = \varphi^i$ for all $i \in \mathbb{Z}$.



- The **homotopy category**, denoted by $\mathcal{K}^b(\mathcal{P})$, is the quotient of $\mathcal{C}^b(\mathcal{P})$ by the (two sided) ideal of null-homotopic morphisms.

The bounded homotopy category

- Let $\mathcal{P}$ be an additive category and let $\varphi\colon A \to B$ be in $\mathcal{C}^b(\mathcal{P})$. Then φ is called **null-homotopic** if there exists a family of morphisms $\left(h^i\colon A^i \to B^{i-1}\right)_{i\in\mathbb{Z}}$ such that $\partial_A^i \bullet h^{i+1} + h^i \bullet \partial_B^{i-1} = \varphi^i$ for all $i \in \mathbb{Z}$.



- The **homotopy category**, denoted by $\mathcal{K}^b(\mathcal{P})$, is the quotient of $\mathcal{C}^b(\mathcal{P})$ by the (two sided) ideal of null-homotopic morphisms.
- That is, two morphisms $\varphi, \psi : A \to B$ in $\mathcal{K}^b(\mathcal{P})$ are equal if and only if there exists a solution to the linear system

$$\partial_A^i \bullet \chi^{i+1} + \chi^i \bullet \partial_B^{i-1} = \varphi^i - \psi^i$$

where i varies over a finite set of values.

Section 2

Homomorphism Structures

Homomorphism structures

Let $\mathcal{P}$ and $\mathcal{D}$ be categories.

Homomorphism structures

Let $\mathcal{P}$ and $\mathcal{D}$ be categories. A $\mathcal{D}$-**homomorphism structure** for $\mathcal{P}$ consists of the following data:

Homomorphism structures

Let $\mathcal{P}$ and $\mathcal{D}$ be categories. A $\mathcal{D}$-**homomorphism structure** for $\mathcal{P}$ consists of the following data:

- A distinguished object $1 \in \mathcal{D}$

Homomorphism structures

Let $\mathcal{P}$ and $\mathcal{D}$ be categories. A $\mathcal{D}$-**homomorphism structure** for $\mathcal{P}$ consists of the following data:

- A distinguished object $1 \in \mathcal{D}$
- A functor $H(-,-)\colon \mathcal{P}^{\mathrm{op}} \times \mathcal{P} \to \mathcal{D}$

Homomorphism structures

Let $\mathcal{P}$ and $\mathcal{D}$ be categories. A $\mathcal{D}$-**homomorphism structure** for $\mathcal{P}$ consists of the following data:

- A distinguished object $1 \in \mathcal{D}$
- A functor $H(-,-)\colon \mathcal{P}^{\mathrm{op}} \times \mathcal{P} \to \mathcal{D}$
- An isomorphism $\nu\colon \mathrm{Hom}_{\mathcal{P}}(B,C) \cong \mathrm{Hom}_{\mathcal{D}}(1, H(B,C))$ natural in B, C,

Homomorphism structures

Let $\mathcal{P}$ and $\mathcal{D}$ be categories. A $\mathcal{D}$-**homomorphism structure** for $\mathcal{P}$ consists of the following data:

- A distinguished object $1 \in \mathcal{D}$
- A functor $H(-,-)\colon \mathcal{P}^{\mathrm{op}} \times \mathcal{P} \to \mathcal{D}$
- An isomorphism $\nu\colon\ \mathrm{Hom}_{\mathcal{P}}(B,C) \cong \mathrm{Hom}_{\mathcal{D}}(1, H(B,C))$ natural in B, C,

Homomorphism structures

Let $\mathcal{P}$ and $\mathcal{D}$ be categories. A $\mathcal{D}$-**homomorphism structure** for $\mathcal{P}$ consists of the following data:

- A distinguished object $1 \in \mathcal{D}$
- A functor $H(-,-) \colon \mathcal{P}^{\mathrm{op}} \times \mathcal{P} \to \mathcal{D}$
- An isomorphism $\nu \colon \operatorname{Hom}_{\mathcal{P}}(B, C) \cong \operatorname{Hom}_{\mathcal{D}}(1, H(B, C))$ natural in B, C, i.e.,

  $$\nu(\alpha \bullet \chi \bullet \beta) = \nu(\chi) \bullet H(\alpha, \beta)$$

  for all compatible triples α, χ, β.

Example

- Let k be a field.

Example

- Let k be a field.
- $\mathcal{P} = \mathcal{D} = k\text{-mat}$

Example

- Let k be a field.
- $\mathcal{P} = \mathcal{D} = k\text{-mat}$
- The distinguished object is $1 \in k\text{-mat}$

Example

- Let k be a field.
- $\mathcal{P} = \mathcal{D} = k\text{-mat}$
- The distinguished object is $1 \in k\text{-mat}$
- H on objects: $H(m, n) = mn$

Example

- Let k be a field.
- $\mathcal{P} = \mathcal{D} = k\text{-mat}$
- The distinguished object is $1 \in k\text{-mat}$
- H on objects: $H(m,n) = mn$
- H on morphisms: $H(A,B) = A^{\mathrm{tr}} \otimes B$

Example

- Let k be a field.
- $\mathcal{P} = \mathcal{D} = k\text{-mat}$
- The distinguished object is $1 \in k\text{-mat}$
- H on objects: $H(m, n) = mn$
- H on morphisms: $H(A, B) = A^{\mathrm{tr}} \otimes B$
- For a morphism $A : m \to n$ we define $\nu(A) : 1 \to mn$ by the vectorization $\mathbf{vec}(A)$ of A, i.e., the one-row matrix defined by the concatenation of all rows of A.

Example

- Let k be a field.
- $\mathcal{P} = \mathcal{D} = k\text{-mat}$
- The distinguished object is $1 \in k\text{-mat}$
- H on objects: $H(m,n) = mn$
- H on morphisms: $H(A,B) = A^{\mathrm{tr}} \otimes B$
- For a morphism $A : m \to n$ we define $\nu(A) : 1 \to mn$ by the vectorization $\mathbf{vec}(A)$ of A, i.e., the one-row matrix defined by the concatenation of all rows of A.
- The induced equality:

$$\mathbf{vec}(M \bullet U \bullet N) = \mathbf{vec}(U) \bullet (M^T \otimes N)$$

is known as the Kronecker product trick for solving two-sided matrix equations.

Two-sided linear systems

Two-sided linear systems

- Let $\mathcal{P}$ be an additive category. A **linear system** in $\mathcal{P}$ is a collection of morphisms α_{ij}, β_{ij} and γ_i of the following form:

$$
\begin{array}{ccccc}
\alpha_{11} \bullet X_1 \bullet \beta_{11} & + \cdots + & \alpha_{1n} \bullet X_n \bullet \beta_{1n} & = & \gamma_1 \\
\vdots & \vdots & \vdots & & \vdots \\
\alpha_{m1} \bullet X_1 \bullet \beta_{m1} & + \cdots + & \alpha_{mn} \bullet X_n \bullet \beta_{mn} & = & \gamma_m
\end{array}
$$

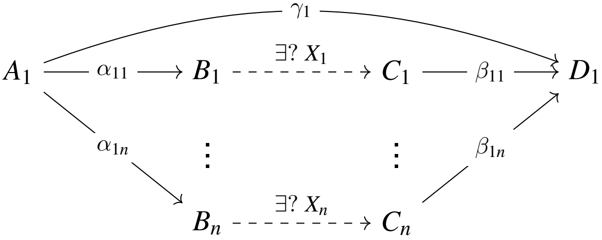
where the X_j are unknown morphisms.

Two-sided linear systems

- Let $\mathcal{P}$ be an additive category. A **linear system** in $\mathcal{P}$ is a collection of morphisms α_{ij}, β_{ij} and γ_i of the following form:

$$\begin{array}{ccccccc} \alpha_{11} \bullet X_1 \bullet \beta_{11} & + \cdots + & \alpha_{1n} \bullet X_n \bullet \beta_{1n} & = & \gamma_1 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_{m1} \bullet X_1 \bullet \beta_{m1} & + \cdots + & \alpha_{mn} \bullet X_n \bullet \beta_{mn} & = & \gamma_m \end{array}$$

where the X_j are unknown morphisms.

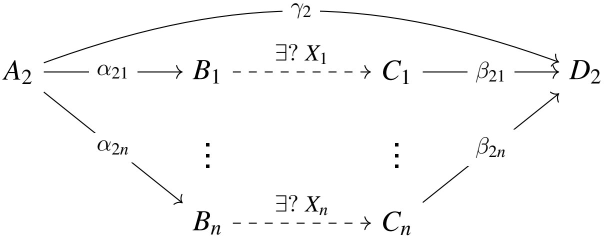


Two-sided linear systems

- Let $\mathcal{P}$ be an additive category. A **linear system** in $\mathcal{P}$ is a collection of morphisms α_{ij}, β_{ij} and γ_i of the following form:

$$\begin{array}{ccccc} \alpha_{11} \bullet X_1 \bullet \beta_{11} & + \cdots + & \alpha_{1n} \bullet X_n \bullet \beta_{1n} & = & \gamma_1 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_{m1} \bullet X_1 \bullet \beta_{m1} & + \cdots + & \alpha_{mn} \bullet X_n \bullet \beta_{mn} & = & \gamma_m \end{array}$$

where the X_j are unknown morphisms.

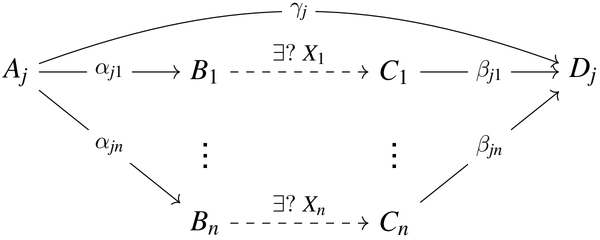


Two-sided linear systems

- Let $\mathcal{P}$ be an additive category. A **linear system** in $\mathcal{P}$ is a collection of morphisms α_{ij}, β_{ij} and γ_i of the following form:

$$\begin{array}{ccccc} \alpha_{11} \bullet X_1 \bullet \beta_{11} & + \cdots + & \alpha_{1n} \bullet X_n \bullet \beta_{1n} & = & \gamma_1 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_{m1} \bullet X_1 \bullet \beta_{m1} & + \cdots + & \alpha_{mn} \bullet X_n \bullet \beta_{mn} & = & \gamma_m \end{array}$$

where the X_j are unknown morphisms.

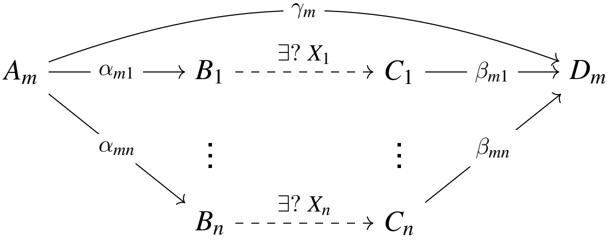


Two-sided linear systems

- Let $\mathcal{P}$ be an additive category. A **linear system** in $\mathcal{P}$ is a collection of morphisms α_{ij}, β_{ij} and γ_i of the following form:

$$\begin{array}{ccccc} \alpha_{11} \bullet X_1 \bullet \beta_{11} & + \cdots + & \alpha_{1n} \bullet X_n \bullet \beta_{1n} & = & \gamma_1 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_{m1} \bullet X_1 \bullet \beta_{m1} & + \cdots + & \alpha_{mn} \bullet X_n \bullet \beta_{mn} & = & \gamma_m \end{array}$$

where the X_j are unknown morphisms.



Two-sided linear systems

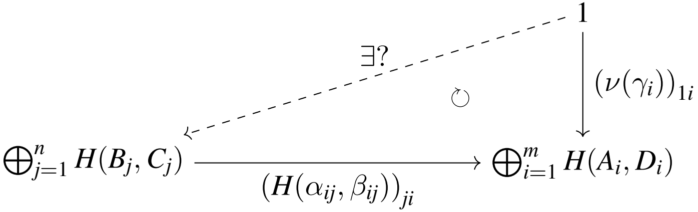
- Let $\mathcal{D}$ be an additive category.

Two-sided linear systems

- Let $\mathcal{D}$ be an additive category.
- Assume $\mathcal{P}$ is equipped with a $\mathcal{D}$-homomorphism structure.

Two-sided linear systems

- Let $\mathcal{D}$ be an additive category.
- Assume $\mathcal{P}$ is equipped with a $\mathcal{D}$-homomorphism structure.
- There exists a solution in $\mathcal{P}$ for the previous linear system iff there exists a solution in $\mathcal{D}$ to the lift problem:



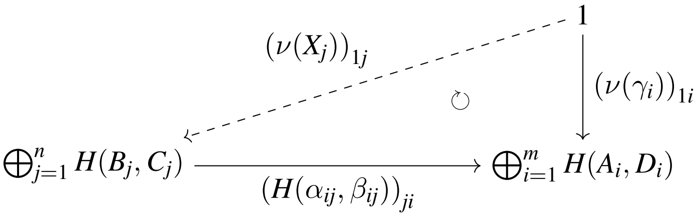
Two-sided linear systems

- Let $\mathcal{D}$ be an additive category.
- Assume $\mathcal{P}$ is equipped with a $\mathcal{D}$-homomorphism structure.
- There exists a solution in $\mathcal{P}$ for the previous linear system iff there exists a solution in $\mathcal{D}$ to the lift problem:

$$\begin{array}{ccc} & & 1 \\ & \overset{(\nu(X_j))_{1j}}{\swarrow} \quad \circlearrowleft & \downarrow (\nu(\gamma_i))_{1i} \\ \bigoplus_{j=1}^{n} H(B_j, C_j) & \xrightarrow[(H(\alpha_{ij}, \beta_{ij}))_{ji}]{} & \bigoplus_{i=1}^{m} H(A_i, D_i) \end{array}$$

Two-sided linear systems

- Let $\mathcal{D}$ be an additive category.
- Assume $\mathcal{P}$ is equipped with a $\mathcal{D}$-homomorphism structure.
- There exists a solution in $\mathcal{P}$ for the previous linear system iff there exists a solution in $\mathcal{D}$ to the lift problem:



- That is, we can transfer a two-sided linear system in $\mathcal{P}$ to a one-sided linear equation in $\mathcal{D}$.

Equality of morphisms in homotopy categories

Theorem

Let $\mathcal{P}$ be an additive category equipped with a $\mathcal{D}$-homomorphism structure $(1, H(-,-), \nu)$ such that $\mathcal{D}$ has decidable lifts, i.e., we can decide in $\mathcal{D}$ the solvability of the one-sided equation $\mathcal{Y} \bullet \beta = \gamma$. Then

Equality of morphisms in homotopy categories

Theorem

Let $\mathcal{P}$ be an additive category equipped with a $\mathcal{D}$-homomorphism structure $(1, H(-,-), \nu)$ such that $\mathcal{D}$ has decidable lifts, i.e., we can decide in $\mathcal{D}$ the solvability of the one-sided equation $\mathcal{Y} \bullet \beta = \gamma$. Then

1. $\mathcal{K}^b(\mathcal{P})$ has decidable equality of morphisms,

Equality of morphisms in homotopy categories

Theorem

Let $\mathcal{P}$ be an additive category equipped with a $\mathcal{D}$-homomorphism structure $(1, H(-,-), \nu)$ such that $\mathcal{D}$ has decidable lifts, i.e., we can decide in $\mathcal{D}$ the solvability of the one-sided equation $\mathcal{Y} \bullet \beta = \gamma$. Then

1. $\mathcal{K}^b(\mathcal{P})$ has decidable equality of morphisms,
2. If $\mathcal{D}$ is Abelian and $1 \in \mathcal{D}$ is a projective object, then $\mathcal{C}^b(\mathcal{P})$ and $\mathcal{K}^b(\mathcal{P})$ can be equipped with a $\mathcal{D}$-homomorphism structure.

Example: An acyclic quiver $\mathfrak{q}$ and a field k

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.

$\mathfrak{q}$ k

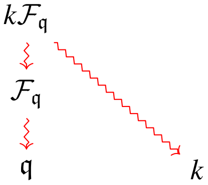
Example: The free category over $\mathfrak{q}$

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.

$$\begin{array}{ccc} \mathcal{F}_{\mathfrak{q}} & & \\ \rightsquigarrow & & \\ \mathfrak{q} & & k \end{array}$$

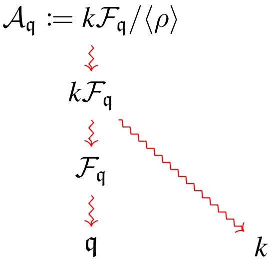
Example: The k-linear closure category

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



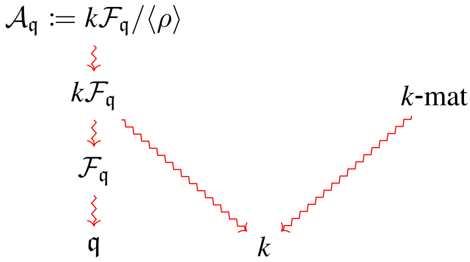
Example: The quotient category by two-sided ideal

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



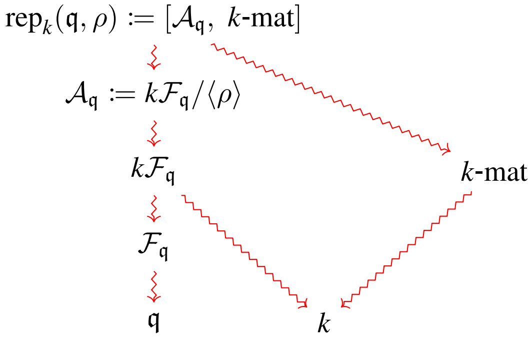
Example: The category of matrices

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



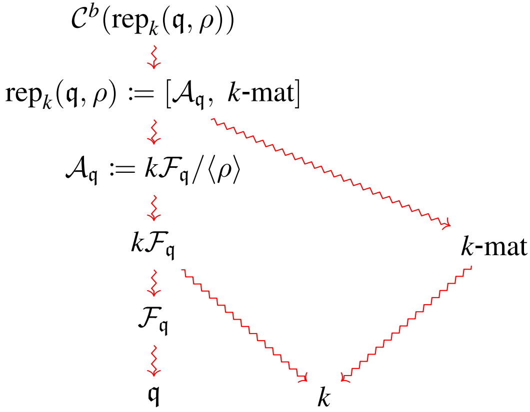
Example: The f.d. ρ-bounded quiver representations

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



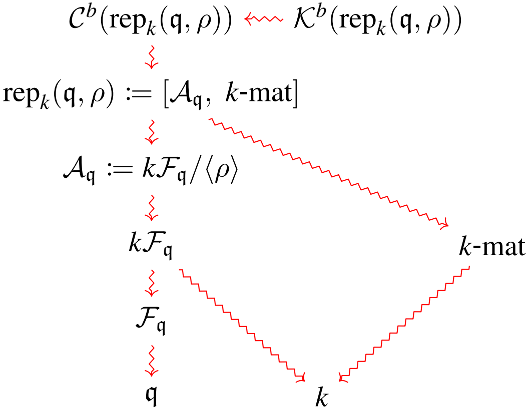
Example: The category of bounded complexes

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



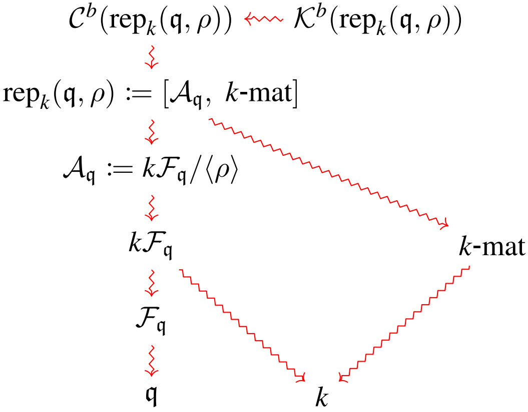
Example: The bounded homotopy category

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



Example: The bounded homotopy category

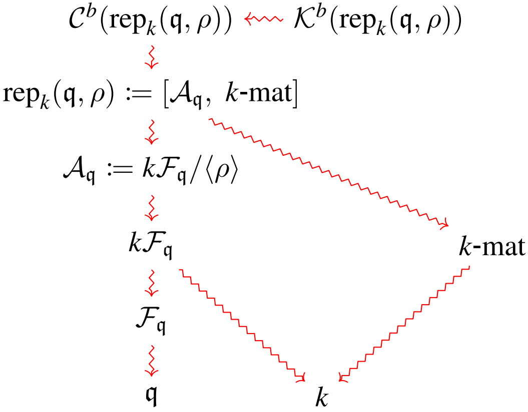
Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



They all have decidable equality of morphisms.

Example: The bounded homotopy category

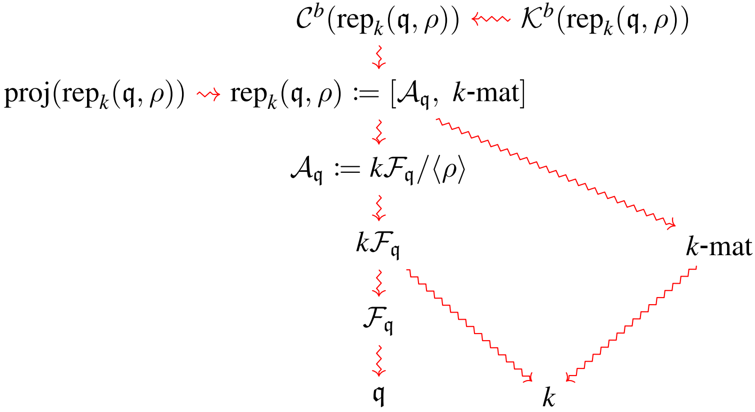
Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



What about the bounded derived category $\mathcal{D}^b(\mathrm{rep}_k(\mathfrak{q},\rho))$?

Example: The subcategory of projective objects

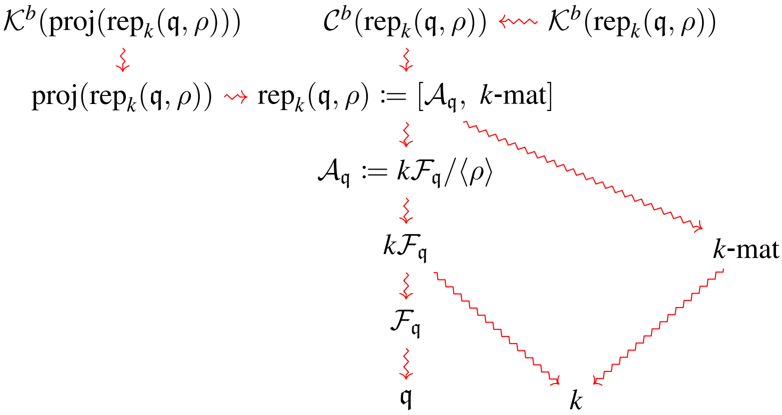
Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



What about the bounded derived category $\mathcal{D}^b(\mathrm{rep}_k(\mathfrak{q},\rho))$?

Example: The bounded homotopy category

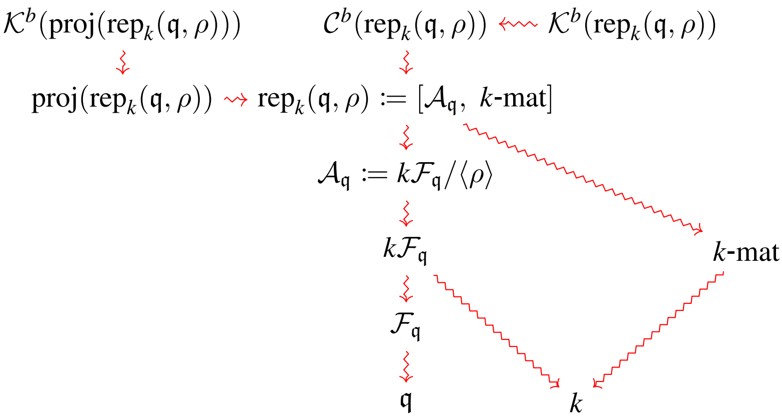
Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



What about the bounded derived category $\mathcal{D}^b(\mathrm{rep}_k(\mathfrak{q},\rho))$?

Example: The bounded derived category

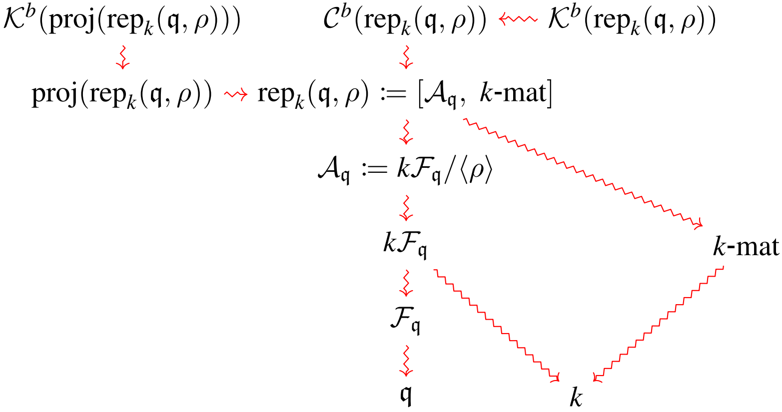
Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



If $\mathrm{rep}_k(\mathfrak{q},\rho)$ has a f.g. dimension then $\mathcal{D}^b(\mathrm{rep}_k(\mathfrak{q},\rho)) \simeq \mathcal{K}^b(\mathrm{proj}(\mathrm{rep}_k(\mathfrak{q},\rho)))$.

Example: The bounded derived category

Let k be a field and let $\mathfrak{q}$ be an acyclic quiver.



They all have decidable equality of morphisms.

Section 3

Tilting Equivalences in Bounded Homotopy Categories

Triangles

A **category with shift functor** is a pair $(\mathfrak{T}, \Sigma)$ given by

Triangles

A **category with shift functor** is a pair $(\mathfrak{T}, \Sigma)$ given by

- An additive category $\mathfrak{T}$.

Triangles

A **category with shift functor** is a pair $(\mathfrak{T}, \Sigma)$ given by

- An additive category $\mathfrak{T}$.
- An autoequivalence $\Sigma : \mathfrak{T} \to \mathfrak{T}$ called **shift**.

Triangles

A **category with shift functor** is a pair $(\mathfrak{T}, \Sigma)$ given by

- An additive category $\mathfrak{T}$.
- An autoequivalence $\Sigma : \mathfrak{T} \to \mathfrak{T}$ called **shift**.

A **triangle** over $(\mathfrak{T}, \Sigma)$ is a sequence

$$A \xrightarrow{\alpha} B \xrightarrow{\iota} C \xrightarrow{\pi} \Sigma A.$$

Triangles

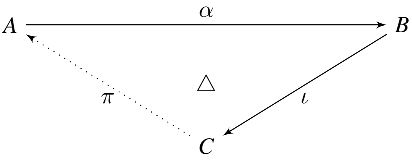
A **category with shift functor** is a pair $(\mathfrak{T}, \Sigma)$ given by

- An additive category $\mathfrak{T}$.
- An autoequivalence $\Sigma : \mathfrak{T} \to \mathfrak{T}$ called **shift.**

A **triangle** over $(\mathfrak{T}, \Sigma)$ is a sequence

$$A \xrightarrow{\alpha} B \xrightarrow{\iota} C \xrightarrow{\pi} \Sigma A.$$

A triangle may be depicted by



Triangulated categories

A **triangulated category** is defined by the following data:

Triangulated categories

A **triangulated category** is defined by the following data:

- a category with shift functor $(\mathfrak{T}, \Sigma)$

Triangulated categories

A **triangulated category** is defined by the following data:

- a category with shift functor $(\mathfrak{T}, \Sigma)$
- distinguished class of triangles $\triangle$, called the **exact triangles**

subject to many Axioms:

Triangulated categories

A **triangulated category** is defined by the following data:

- a category with shift functor $(\mathfrak{T}, \Sigma)$
- distinguished class of triangles $\triangle$, called the **exact triangles**

subject to many Axioms:

- Any morphism $\alpha\colon A \to B$

Triangulated categories

A **triangulated category** is defined by the following data:

- a category with shift functor $(\mathfrak{T}, \Sigma)$
- distinguished class of triangles $\triangle$, called the **exact triangles**

subject to many Axioms:

- Any morphism $\alpha\colon A \to B$ can be completed to an exact triangle

$$A \xrightarrow{\alpha} B \xrightarrow{\iota(\alpha)} \mathrm{Cone}(\alpha) \xrightarrow{\pi(\alpha)} \Sigma A \ \in \ \triangle.$$

Triangulated categories

A **triangulated category** is defined by the following data:

- a category with shift functor $(\mathfrak{T}, \Sigma)$
- distinguished class of triangles $\triangle$, called the **exact triangles**

subject to many Axioms:

- Any morphism $\alpha : A \to B$ can be completed to an exact triangle

$$A \xrightarrow{\alpha} B \xrightarrow{\iota(\alpha)} \mathrm{Cone}(\alpha) \xrightarrow{\pi(\alpha)} \Sigma A \ \in \ \triangle.$$

Triangulated categories

A **triangulated category** is defined by the following data:

- a category with shift functor $(\mathfrak{T}, \Sigma)$
- distinguished class of triangles $\triangle$, called the **exact triangles**

subject to many Axioms:

- Any morphism $\alpha : A \to B$ can be completed to an exact triangle
$$A \xrightarrow{\alpha} B \xrightarrow{\iota(\alpha)} \mathrm{Cone}(\alpha) \xrightarrow{\pi(\alpha)} \Sigma A \ \in \ \triangle.$$
- The rotation axiom, the octahedral axiom, etc ...

Example

(Bounded) homotopy and derived categories are triangulated.

Strong exceptional sequences

Let k be a field and let $(\mathfrak{T}, \Sigma, \triangle)$ be a k-linear triangulated category.

Strong exceptional sequences

Let k be a field and let $(\mathfrak{T}, \Sigma, \triangle)$ be a k-linear triangulated category.

- A **strong exceptional sequence** is a sequence of objects $\mathcal{E} = (E_1, \dots, E_m)$ in $\mathfrak{T}$ such that

Strong exceptional sequences

Let k be a field and let $(\mathfrak{T}, \Sigma, \triangle)$ be a k-linear triangulated category.

- A **strong exceptional sequence** is a sequence of objects $\mathcal{E} = (E_1, \ldots, E_m)$ in $\mathfrak{T}$ such that
  1. $\operatorname{Hom}_{\mathfrak{T}}(E_i, E_i) \simeq k$ for $i = 1, \ldots, m$.

Strong exceptional sequences

Let k be a field and let $(\mathfrak{T}, \Sigma, \triangle)$ be a k-linear triangulated category.

- A **strong exceptional sequence** is a sequence of objects $\mathcal{E} = (E_1, \ldots, E_m)$ in $\mathfrak{T}$ such that
  1. $\mathrm{Hom}_{\mathfrak{T}}(E_i, E_i) \simeq k$ for $i = 1, \ldots, m$.
  2. $\mathrm{Hom}_{\mathfrak{T}}(E_i, E_j) = 0$ for all $i > j$ (no backward morphisms).

Strong exceptional sequences

Let k be a field and let $(\mathfrak{T}, \Sigma, \triangle)$ be a k-linear triangulated category.

- A **strong exceptional sequence** is a sequence of objects $\mathcal{E} = (E_1, \ldots, E_m)$ in $\mathfrak{T}$ such that
  1. $\mathrm{Hom}_{\mathfrak{T}}(E_i, E_i) \simeq k$ for $i = 1, \ldots, m$.
  2. $\mathrm{Hom}_{\mathfrak{T}}(E_i, E_j) = 0$ for all $i > j$ (no backward morphisms).
  3. $\mathrm{Hom}_{\mathfrak{T}}(E_i, \Sigma^{\ell} E_j) = 0$ for all i, j and all $\ell \neq 0$ (no higher morphisms).

Strong exceptional sequences

Let k be a field and let $(\mathfrak{T}, \Sigma, \triangle)$ be a k-linear triangulated category.

- A **strong exceptional sequence** is a sequence of objects $\mathcal{E} = (E_1, \ldots, E_m)$ in $\mathfrak{T}$ such that
  1. $\mathrm{Hom}_{\mathfrak{T}}(E_i, E_i) \simeq k$ for $i = 1, \ldots, m$.
  2. $\mathrm{Hom}_{\mathfrak{T}}(E_i, E_j) = 0$ for all $i > j$ (no backward morphisms).
  3. $\mathrm{Hom}_{\mathfrak{T}}(E_i, \Sigma^{\ell} E_j) = 0$ for all i, j and all $\ell \neq 0$ (no higher morphisms).
- An exceptional sequence is called **full** if the smallest triangulated subcategory generated by $\mathcal{E}$, denoted by $\mathcal{E}^{\triangle}$, equals $\mathfrak{T}$.

Theorem: Equivalences induced by strong exceptional sequences

Let k be a field, $\mathcal{P}$ a k-linear Hom-finite category and $\mathcal{E}$ is strong exceptional sequence in $\mathcal{K}^b(\mathcal{P})$. Then, there exists an acyclic quiver $\mathfrak{q}_{\mathcal{E}}$ and set of relations $\rho \subset \mathrm{Mor}(k\mathcal{F}_{\mathfrak{q}_{\mathcal{E}}})$ such that

$$\mathcal{D}^b(\mathrm{rep}_k(\mathfrak{q}_{\mathcal{E}}, \rho)) \simeq \mathcal{K}^b(\mathcal{E}^{\oplus}) \simeq \mathcal{E}^{\triangle} \subseteq \mathcal{K}^b(\mathcal{P})$$

where $\mathcal{E}^{\oplus}$ is the full subcategory of $\mathcal{K}^b(\mathcal{P})$ generated by all finite direct sums of objects in $\mathcal{E}$.

The convolution functor in one slide

The convolution functor is the functor $\mathcal{K}^b(\mathcal{E}^{\oplus}) \xrightarrow{F} \mathcal{E}^{\triangle} \subseteq \mathcal{K}^b(\mathcal{P})$. Let U be an object in $\mathcal{K}^b(\mathcal{E}^{\oplus})$ with lower bound m and upper bound n. Then $F(U)$ can be computed as follows:

$$\cdots \xrightarrow{\partial^{n-4}} U^{n-3} \xrightarrow{\partial^{n-3}} U^{n-2} \xrightarrow{\partial^{n-2}} U^{n-1} \xrightarrow{\partial^{n-1}} U^{n} \longrightarrow 0$$

The convolution functor in one slide

The convolution functor is the functor $\mathcal{K}^b(\mathcal{E}^{\oplus}) \xrightarrow{F} \mathcal{E}^{\triangle} \subseteq \mathcal{K}^b(\mathcal{P})$. Let U be an object in $\mathcal{K}^b(\mathcal{E}^{\oplus})$ with lower bound m and upper bound n. Then $F(U)$ can be computed as follows:

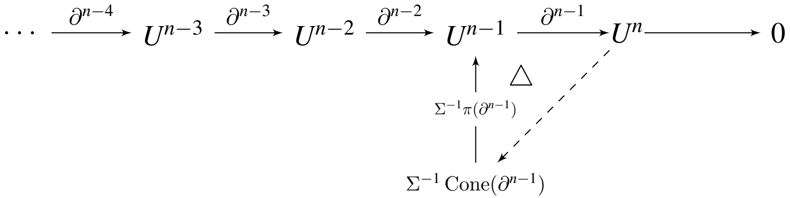
- If $m = n$, then $F(U) := \Sigma^{-m} U^m \in \mathcal{E}^{\triangle}$.

$$\cdots \xrightarrow{\partial^{n-4}} U^{n-3} \xrightarrow{\partial^{n-3}} U^{n-2} \xrightarrow{\partial^{n-2}} U^{n-1} \xrightarrow{\partial^{n-1}} U^n \longrightarrow 0$$

The convolution functor in one slide

The convolution functor is the functor $\mathcal{K}^b(\mathcal{E}^{\oplus}) \xrightarrow{F} \mathcal{E}^{\triangle} \subseteq \mathcal{K}^b(\mathcal{P})$. Let U be an object in $\mathcal{K}^b(\mathcal{E}^{\oplus})$ with lower bound m and upper bound n. Then $F(U)$ can be computed as follows:

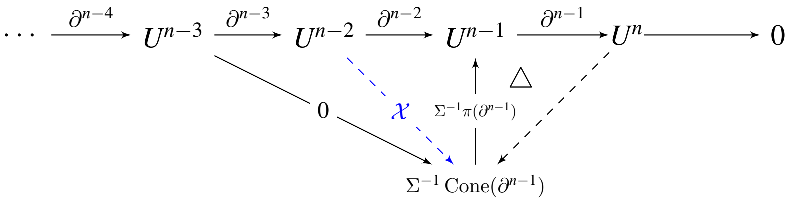
- If $m = n$, then $F(U) := \Sigma^{-m} U^m \in \mathcal{E}^{\triangle}$.



The convolution functor in one slide

The convolution functor is the functor $\mathcal{K}^b(\mathcal{E}^{\oplus}) \xrightarrow{F} \mathcal{E}^{\triangle} \subseteq \mathcal{K}^b(\mathcal{P})$. Let U be an object in $\mathcal{K}^b(\mathcal{E}^{\oplus})$ with lower bound m and upper bound n. Then $F(U)$ can be computed as follows:

- If $m = n$, then $F(U) := \Sigma^{-m} U^m \in \mathcal{E}^{\triangle}$.
- Compute $\mathcal{X}$ with $\mathcal{X} \bullet \Sigma^{-1}\pi(\partial^{n-1}) = \partial^{n-2}$ and $\partial^{n-3} \bullet \mathcal{X} = 0$.



The convolution functor in one slide

The convolution functor is the functor $\mathcal{K}^b(\mathcal{E}^{\oplus}) \xrightarrow{F} \mathcal{E}^{\triangle} \subseteq \mathcal{K}^b(\mathcal{P})$. Let U be an object in $\mathcal{K}^b(\mathcal{E}^{\oplus})$ with lower bound m and upper bound n. Then $F(U)$ can be computed as follows:

- If $m = n$, then $F(U) := \Sigma^{-m} U^m \in \mathcal{E}^{\triangle}$.
- Compute $\mathcal{X}$ with $\mathcal{X} \bullet \Sigma^{-1}\pi(\partial^{n-1}) = \partial^{n-2}$ and $\partial^{n-3} \bullet \mathcal{X} = 0$.
- Repeat the procedure for the following new object U:

$$\cdots \xrightarrow{\partial^{n-4}} U^{n-3} \xrightarrow{\partial^{n-3}} U^{n-2} \xrightarrow{\mathcal{X}} \Sigma^{-1}\operatorname{Cone}(\partial^{n-1}) \longrightarrow 0 \longrightarrow 0$$

The convolution functor in one slide

The convolution functor is the functor $\mathcal{K}^b(\mathcal{E}^{\oplus}) \xrightarrow{F} \mathcal{E}^{\triangle} \subseteq \mathcal{K}^b(\mathcal{P})$. Let U be an object in $\mathcal{K}^b(\mathcal{E}^{\oplus})$ with lower bound m and upper bound n. Then $F(U)$ can be computed as follows:

- If $m = n$, then $F(U) := \Sigma^{-m} U^m \in \mathcal{E}^{\triangle}$.
- Compute $\mathcal{X}$ with $\mathcal{X} \bullet \Sigma^{-1}\pi(\partial^{n-1}) = \partial^{n-2}$ and $\partial^{n-3} \bullet \mathcal{X} = 0$.
- Repeat the procedure for the following new object U:

$$\cdots \xrightarrow{\partial^{n-4}} U^{n-3} \xrightarrow{\partial^{n-3}} U^{n-2} \xrightarrow{\mathcal{X}} \Sigma^{-1}\,\mathrm{Cone}(\partial^{n-1}) \longrightarrow 0 \longrightarrow 0$$

Convolution functor also depends on solving two-sided linear systems.

References

1. S. Posur, *A constructive approach to Freyd categories*
2. S. I. Gelfand and Y. I. Manin, *Methods of Homological Algebra*
3. CAP on GitHub:
   `https://github.com/homalg-project/CAP_project`
4. HOMALG on GitHub:
   `https://github.com/homalg-project/homalg_project`
5. HIGHERHOMOLOGICALALGEBRA on GitHub: `https://github.com/homalg-project/HigherHomologicalAlgebra`

Thank you for your attention