Abstract category theory

Abstract category theory

Concrete computations in computer algebra

“natural transformation”

Abstract category theory $\Longrightarrow$ Concrete computations in computer algebra

$$\text{Abstract category theory} \xRightarrow{\text{“natural transformation”}} \text{Concrete computations in computer algebra}$$

Categorical abstraction

“natural transformation”

Abstract category theory $\Longrightarrow$ Concrete computations in computer algebra

Categorical abstraction is a powerful

“natural transformation”

Abstract category theory $\Longrightarrow$ Concrete computations in computer algebra

Categorical abstraction is a powerful organizing principle

“natural transformation”

Abstract category theory $\Longrightarrow$ Concrete computations in computer algebra

Categorical abstraction is a powerful organizing principle and computational tool.

CAP: Categories, algorithms, programming

Sebastian Posur

August 17, 2019

Outline

Categorical abstraction is a powerful organizing principle and computational tool.

1. What is categorical abstraction?

Outline

Categorical abstraction is a powerful organizing principle and computational tool.

1. What is categorical abstraction?
2. How can it be used as an organizing priniciple?

Outline

Categorical abstraction **is a powerful** **organizing principle** **and** **computational tool.**

1. What is categorical abstraction?
2. How can it be used as an organizing priniciple?
3. Why is it a computational tool?

Outline

Categorical abstraction is a powerful organizing principle and computational tool.

1. What is categorical abstraction?
2. How can it be used as an organizing priniciple?
3. Why is it a computational tool?

Categories

Definition

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

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$

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

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$
- **Morphisms**: a family of sets $\mathrm{Hom}_{\mathcal{A}}(A, B)$, where $A, B \in \mathrm{Obj}_{\mathcal{A}}$

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

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$
- **Morphisms**: a family of sets $\mathrm{Hom}_{\mathcal{A}}(A, B)$, where $A, B \in \mathrm{Obj}_{\mathcal{A}}$

$$A \xrightarrow{\alpha} B \qquad\qquad C$$

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$
- **Morphisms**: a family of sets $\mathrm{Hom}_{\mathcal{A}}(A, B)$, where $A, B \in \mathrm{Obj}_{\mathcal{A}}$

$$A \xrightarrow{\alpha} B \xrightarrow{\beta} C$$

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$
- **Morphisms**: a family of sets $\mathrm{Hom}_{\mathcal{A}}(A, B)$, where $A, B \in \mathrm{Obj}_{\mathcal{A}}$
- **Composition**: for $A, B, C \in \mathrm{Obj}_{\mathcal{A}}$, a function
  $\circ : \mathrm{Hom}_{\mathcal{A}}(B, C) \times \mathrm{Hom}_{\mathcal{A}}(A, B) \to \mathrm{Hom}_{\mathcal{A}}(A, C)$ (assoc.)

$$A \xrightarrow{\alpha} B \xrightarrow{\beta} C$$

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_\mathcal{A}$
- **Morphisms**: a family of sets $\mathrm{Hom}_\mathcal{A}(A, B)$, where $A, B \in \mathrm{Obj}_\mathcal{A}$
- **Composition**: for $A, B, C \in \mathrm{Obj}_\mathcal{A}$, a function
  $\circ : \mathrm{Hom}_\mathcal{A}(B, C) \times \mathrm{Hom}_\mathcal{A}(A, B) \to \mathrm{Hom}_\mathcal{A}(A, C)$ (assoc.)

$$\begin{tikzcd} A \arrow[r, "\alpha"] \arrow[rr, bend left, "\beta \circ \alpha"] & B \arrow[r, "\beta"] & C \end{tikzcd}$$

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$
- **Morphisms**: a family of sets $\mathrm{Hom}_{\mathcal{A}}(A, B)$, where $A, B \in \mathrm{Obj}_{\mathcal{A}}$
- **Composition**: for $A, B, C \in \mathrm{Obj}_{\mathcal{A}}$, a function
  $\circ : \mathrm{Hom}_{\mathcal{A}}(B, C) \times \mathrm{Hom}_{\mathcal{A}}(A, B) \to \mathrm{Hom}_{\mathcal{A}}(A, C)$ (assoc.)
- **Neutral elements**: $\mathrm{id}_A \in \mathrm{Hom}_{\mathcal{A}}(A, A)$ (neutral w.r.t. composition)

$$\begin{tikzcd} A \arrow[r, "\alpha"] \arrow[rr, bend left, "\beta \circ \alpha"] & B \arrow[r, "\beta"] & C \end{tikzcd}$$

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$
- **Morphisms**: a family of sets $\mathrm{Hom}_{\mathcal{A}}(A, B)$, where $A, B \in \mathrm{Obj}_{\mathcal{A}}$
- **Composition**: for $A, B, C \in \mathrm{Obj}_{\mathcal{A}}$, a function
  $\circ : \mathrm{Hom}_{\mathcal{A}}(B, C) \times \mathrm{Hom}_{\mathcal{A}}(A, B) \to \mathrm{Hom}_{\mathcal{A}}(A, C)$ (assoc.)
- **Neutral elements**: $\mathrm{id}_A \in \mathrm{Hom}_{\mathcal{A}}(A, A)$ (neutral w.r.t. composition)

$$\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[rr, bend left, "\beta \circ \alpha"] \arrow[loop below] & B \arrow[r, "\beta"] \arrow[loop below] & C \arrow[loop below]
\end{tikzcd}$$

Categories

Definition

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

- **Objects**: a class $\mathrm{Obj}_{\mathcal{A}}$
- **Morphisms**: a family of sets $\mathrm{Hom}_{\mathcal{A}}(A, B)$, where $A, B \in \mathrm{Obj}_{\mathcal{A}}$
- **Composition**: for $A, B, C \in \mathrm{Obj}_{\mathcal{A}}$, a function
  $\circ : \mathrm{Hom}_{\mathcal{A}}(B, C) \times \mathrm{Hom}_{\mathcal{A}}(A, B) \to \mathrm{Hom}_{\mathcal{A}}(A, C)$ (assoc.)
- **Neutral elements**: $\mathrm{id}_A \in \mathrm{Hom}_{\mathcal{A}}(A, A)$ (neutral w.r.t. composition)

$$\begin{tikzcd} A \arrow[r, "\alpha"] \arrow[rr, bend left, "\beta \circ \alpha"] \arrow[loop below] & B \arrow[r, "\beta"] \arrow[loop below] & C \arrow[loop below] \end{tikzcd}$$

Isomorphisms

$$\begin{tikzcd} A \arrow[rr, bend left, "\alpha"] & & B \end{tikzcd}$$

Isomorphisms

$$\begin{tikzcd} A \arrow[rr, bend left, "\alpha"] & & B \arrow[ll, bend left, "\beta"] \end{tikzcd}$$

Isomorphisms

$$\begin{tikzcd} A \arrow[loop left, "\mathsf{id}_A = \beta \circ \alpha"] \arrow[r, bend left, "\alpha"] & B \arrow[l, bend left, "\beta"] \end{tikzcd}$$

Isomorphisms

$$\begin{tikzcd} \mathrm{id}_A = \beta \circ \alpha \arrow[r, phantom] & A \arrow[loop left] \arrow[rr, bend left, "\alpha"] & & B \arrow[ll, bend left, "\beta"] \arrow[loop right] & \alpha \circ \beta = \mathrm{id}_B \end{tikzcd}$$

Isomorphisms

$$\begin{tikzcd} A \arrow[loop left, "\mathrm{id}_A = \beta \circ \alpha"] \arrow[r, bend left, "\alpha"] & B \arrow[l, bend left, "\beta"] \arrow[loop right, "\alpha \circ \beta = \mathrm{id}_B"] \end{tikzcd}$$

- A and B are isomorphic.

Isomorphisms

$$\begin{tikzcd} A \arrow[loop left, "\mathrm{id}_A = \beta \circ \alpha"] \arrow[rr, bend left, "\alpha"] & & B \arrow[ll, bend left, "\beta"] \arrow[loop right, "\alpha \circ \beta = \mathrm{id}_B"] \end{tikzcd}$$

- A and B are isomorphic.
- Notation: $A \cong B$ or $A \simeq B$.

Isomorphisms

$$\begin{tikzcd} A \arrow[r, bend left, "\alpha"] \arrow[loop left, "\mathrm{id}_A = \beta \circ \alpha"] & B \arrow[l, bend left, "\beta"] \arrow[loop right, "\alpha \circ \beta = \mathrm{id}_B"] \end{tikzcd}$$

- A and B are isomorphic.
- Notation: $A \cong B$ or $A \simeq B$.
- α is an isomorphism with inverse β.

Isomorphisms

$$\begin{tikzcd}[column sep=huge] A \arrow[loop left, "\mathrm{id}_A = \beta \circ \alpha"] \arrow[r, bend left, "\alpha"] & B \arrow[l, bend left, "\beta"] \arrow[loop right, "\alpha \circ \beta = \mathrm{id}_B"] \end{tikzcd}$$

- A and B are isomorphic.
- Notation: $A \cong B$ or $A \simeq B$.
- α is an isomorphism with inverse β.
- β is an isomorphism with inverse α.

Isomorphisms

$$
\begin{tikzcd}
A \arrow[r, bend left, "\alpha"] \arrow[loop left, "\mathrm{id}_A = \beta \circ \alpha"] & B \arrow[l, bend left, "\beta"] \arrow[loop right, "\alpha \circ \beta = \mathrm{id}_B"]
\end{tikzcd}
$$

- A and B are isomorphic.
- Notation: $A \cong B$ or $A \simeq B$.
- α is an isomorphism with inverse β.
- β is an isomorphism with inverse α.

Probably the most important notion in category theory.

Example: Sets

Sets

Example: Sets

Sets

- $\mathrm{Obj}_{\mathrm{Sets}} := \{\text{all sets}\}$

Example: Sets

$$\mathrm{Sets}$$

- $\mathrm{Obj}_{\mathrm{Sets}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Sets}}(M, N) := \{\text{maps from } M \text{ to } N\}$

Example: Sets

$$\mathrm{Sets}$$

- $\mathrm{Obj}_{\mathrm{Sets}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Sets}}(M, N) := \{\text{maps from } M \text{ to } N\}$
- $\circ$ given by composition of maps

Example: Sets

$$\mathbf{Sets}$$

- $\mathrm{Obj}_{\mathbf{Sets}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathbf{Sets}}(M, N) := \{\text{maps from } M \text{ to } N\}$
- $\circ$ given by composition of maps
- $\mathrm{id}_M : M \longrightarrow M : m \mapsto m$

Example: Sets

$$\mathrm{Sets}$$

- $\mathrm{Obj}_{\mathrm{Sets}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Sets}}(M, N) := \{\text{maps from } M \text{ to } N\}$
- $\circ$ given by composition of maps
- $\mathrm{id}_M : M \longrightarrow M : m \mapsto m$

isomorphisms

Example: Sets

$$\mathrm{Sets}$$

- $\mathrm{Obj}_{\mathrm{Sets}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Sets}}(M, N) := \{\text{maps from } M \text{ to } N\}$
- $\circ$ given by composition of maps
- $\mathrm{id}_M : M \longrightarrow M : m \mapsto m$

$$\text{isomorphisms} = \text{bijections}$$

Example: Relations

Rels

Example: Relations

$$\mathrm{Rels}$$

- $\mathrm{Obj}_{\mathrm{Rels}} := \{\text{all sets}\}$

Example: Relations

$$\mathrm{Rels}$$

- $\mathrm{Obj}_{\mathrm{Rels}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Rels}}(M, N) := \{\text{relations from } M \text{ to } N\text{, i.e., subsets of } M \times N\}$

Example: Relations

$$\mathrm{Rels}$$

- $\mathrm{Obj}_{\mathrm{Rels}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Rels}}(M, N) := \{\text{relations from } M \text{ to } N\text{, i.e., subsets of } M \times N\}$
- $\circ$ given by composition of relations

Example: Relations

$$\mathbf{Rels}$$

- $\mathrm{Obj}_{\mathrm{Rels}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Rels}}(M, N) := \{\text{relations from } M \text{ to } N\text{, i.e., subsets of } M \times N\}$
- $\circ$ given by composition of relations

$$A \xrightarrow[f \subseteq A \times B]{} B \xrightarrow[g \subseteq B \times C]{} C$$

Example: Relations

$$\mathrm{Rels}$$

- $\mathrm{Obj}_{\mathrm{Rels}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Rels}}(M, N) := \{\text{relations from } M \text{ to } N\text{, i.e., subsets of } M \times N\}$
- $\circ$ given by composition of relations

$$g \circ f := \{(a, c) \in A \times C \mid \exists b \in B : (a, b) \in f, (b, c) \in g\}$$

$$\begin{tikzcd} A \arrow[r, "f \subseteq A \times B"'] \arrow[rr, bend left] & B \arrow[r, "g \subseteq B \times C"'] & C \end{tikzcd}$$

Example: Relations

$$\mathrm{Rels}$$

- $\mathrm{Obj}_{\mathrm{Rels}} := \{\text{all sets}\}$
- $\mathrm{Hom}_{\mathrm{Rels}}(M, N) := \{\text{relations from } M \text{ to } N\text{, i.e., subsets of } M \times N\}$
- $\circ$ given by composition of relations
- $\mathrm{id}_M := \{(m, m) \mid m \in M\} \subset M \times M$

$$g \circ f := \{(a, c) \in A \times C \mid \exists b \in B : (a, b) \in f, (b, c) \in g\}$$

$$\begin{tikzcd} A \arrow[r, "f \subseteq A \times B"'] \arrow[rr, bend left] & B \arrow[r, "g \subseteq B \times C"'] & C \end{tikzcd}$$

Example: Groups as categories

Let G be a group.

Example: Groups as categories

Let G be a group.

- $\mathrm{Obj}_G := \{*\}$ (any singleton)

Example: Groups as categories

Let G be a group.

- $\mathrm{Obj}_G := \{*\}$ (any singleton)
- $\mathsf{Hom}_G(*, *) := G$

Example: Groups as categories

Let G be a group.

- $\mathrm{Obj}_G := \{*\}$ (any singleton)
- $\mathrm{Hom}_G(*, *) := G$
- $\circ$ given by group multiplication

Example: Groups as categories

Let G be a group.

- $\mathrm{Obj}_G := \{*\}$ (any singleton)
- $\mathrm{Hom}_G(*, *) := G$
- $\circ$ given by group multiplication
- $\mathrm{id}_* :=$ identity element in G

Example: Groups as categories

Let G be a group.

- $\mathrm{Obj}_G := \{*\}$ (any singleton)
- $\mathsf{Hom}_G(*, *) := G$
- $\circ$ given by group multiplication
- $\mathsf{id}_* :=$ identity element in G

isomorphisms

Example: Groups as categories

Let G be a group.

- $\mathrm{Obj}_G := \{*\}$ (any singleton)
- $\mathrm{Hom}_G(*, *) := G$
- $\circ$ given by group multiplication
- $\mathrm{id}_* :=$ identity element in G

isomorphisms $=$ all elements in G

Example: Finite dimensional vector spaces

$$\mathrm{vec}_{\mathbb{Q}}$$

Example: Finite dimensional vector spaces

$$\mathrm{vec}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}} := \{$all finite dimensional $\mathbb{Q}$-vector spaces$\}$

Example: Finite dimensional vector spaces

$$\mathrm{vec}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}} := \{$all finite dimensional $\mathbb{Q}$-vector spaces$\}$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(V, W) := \{\mathbb{Q}$-linear maps from V to $W\}$

Example: Finite dimensional vector spaces

$$\mathrm{vec}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}} := \{$all finite dimensional $\mathbb{Q}$-vector spaces$\}$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(V, W) := \{\mathbb{Q}$-linear maps from V to $W\}$
- $\circ$ given by composition of maps

Example: Finite dimensional vector spaces

$$\mathrm{vec}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}} := \{$all finite dimensional $\mathbb{Q}$-vector spaces$\}$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(V, W) := \{\mathbb{Q}$-linear maps from V to $W\}$
- $\circ$ given by composition of maps
- $\mathrm{id}_M : M \longrightarrow M : m \mapsto m$

Example: Finite dimensional vector spaces

$$\mathrm{vec}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}} := \{$all finite dimensional $\mathbb{Q}$-vector spaces$\}$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(V, W) := \{\mathbb{Q}$-linear maps from V to $W\}$
- $\circ$ given by composition of maps
- $\mathrm{id}_M : M \longrightarrow M : m \mapsto m$

isomorphisms

Example: Finite dimensional vector spaces

$$\mathrm{vec}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}} := \{$all finite dimensional $\mathbb{Q}$-vector spaces$\}$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(V, W) := \{\mathbb{Q}$-linear maps from V to $W\}$
- $\circ$ given by composition of maps
- $\mathrm{id}_M : M \longrightarrow M : m \mapsto m$

isomorphisms $=$ $\mathbb{Q}$-linear bijections

Example: Matrices

$$\mathrm{mat}_{\mathbb{Q}}$$

Example: Matrices

$$\mathrm{mat}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} := \mathbb{N}_0$

Example: Matrices

$$\mathrm{mat}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} := \mathbb{N}_0$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(m, n) := \mathbb{Q}^{m \times n}$

Example: Matrices

$$\mathrm{mat}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} := \mathbb{N}_0$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(m, n) := \mathbb{Q}^{m \times n}$
- $A \circ B := B \cdot A$

Example: Matrices

$$\mathrm{mat}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} := \mathbb{N}_0$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(m, n) := \mathbb{Q}^{m \times n}$
- $A \circ B := B \cdot A$
- id_m : the $m \times m$ identity matrix

Example: Matrices

$$\mathrm{mat}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} := \mathbb{N}_0$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(m, n) := \mathbb{Q}^{m \times n}$
- $A \circ B := B \cdot A$
- id_m : the $m \times m$ identity matrix

isomorphisms

Example: Matrices

$$\mathrm{mat}_{\mathbb{Q}}$$

- $\mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} := \mathbb{N}_0$
- $\mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(m, n) := \mathbb{Q}^{m \times n}$
- $A \circ B := B \cdot A$
- id_m : the $m \times m$ identity matrix

isomorphisms $=$ invertible matrices

Equivalences

When are two categories “the same” in a categorical way?

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \qquad\qquad\qquad \mathcal{B}$$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \qquad\qquad\qquad \mathcal{B}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

A

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$A \quad \longmapsto \quad FA$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$A \quad \longmapsto \quad FA$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$\begin{array}{c} A \\ \alpha \Big\downarrow \\ A' \end{array}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$

Equivalences

When are two categories “the same” in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$\begin{tikzcd} A \arrow[d, "\alpha"'] & \longmapsto & FA \arrow[d, "F\alpha"] \\ A' & & FA' \end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$\begin{tikzcd} A \arrow[d, "\alpha"'] & \longmapsto & FA \arrow[d, "F\alpha"] \\ A' & & FA' \end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)

Equivalences

When are two categories “the same” in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$\begin{tikzcd} A \arrow[d, "\alpha"'] \\ A' \end{tikzcd} \quad \longmapsto \quad \begin{tikzcd} FA \arrow[d, "F\alpha"] \\ FA' \end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B}$

Equivalences

When are two categories “the same” in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B}$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$B$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B}$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$B$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B} \quad \exists A \in \mathcal{A}$:

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$A \qquad\qquad B$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B} \quad \exists A \in \mathcal{A} :$

Equivalences

When are two categories “the same” in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$\begin{array}{ccc} & & FA \\ & \longmapsto & \\ A & & B \end{array}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B} \qquad \exists A \in \mathcal{A} :$

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$\begin{array}{ccc} & & FA \\ & \longmapsto & \wr\! \| \\ A & & B \end{array}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B} \quad \exists A \in \mathcal{A}$:

Equivalences

When are two categories "the same" in a categorical way?

$$\mathcal{A} \xrightarrow{\quad F \quad} \mathcal{B}$$

$$\begin{array}{ccc} & & FA \\ & \longmapsto & \wr\vert \\ A & & B \end{array}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B} \quad \exists A \in \mathcal{A} : \quad FA \simeq B$

Equivalences

When are two categories "the same" in a categorical way?

$$\begin{tikzcd} \mathcal{A} \arrow[rr, "F"] & & \mathcal{B} \end{tikzcd}$$

$$\begin{tikzcd} & & FA \arrow[d, phantom, "\wr|"] \\ A \arrow[r, mapsto] & & B \end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathcal{A}} \longrightarrow \mathrm{Obj}_{\mathcal{B}}$
- $\mathrm{Hom}_{\mathcal{A}}(A, A') \longrightarrow \mathrm{Hom}_{\mathcal{B}}(FA, FA')$ (respects id and $\circ$, bijection)
- $\forall B \in \mathcal{B} \quad \exists A \in \mathcal{A} : \quad FA \simeq B$ (essentially surjective)

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$\mathrm{mat}_{\mathbb{Q}}$ $\qquad\qquad$ $\mathrm{vec}_{\mathbb{Q}}$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$\mathrm{mat}_{\mathbb{Q}}$ $\qquad\qquad$ $\mathrm{vec}_{\mathbb{Q}}$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

m

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\begin{array}{ccc} \mathrm{mat}_{\mathbb{Q}} & \xrightarrow{\quad F \quad} & \mathrm{vec}_{\mathbb{Q}} \\ & & \\ m & \longmapsto & \mathbb{Q}^{1\times m} \end{array}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$m \longmapsto \mathbb{Q}^{1\times m}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$\begin{tikzcd} m \arrow[d, "\left(a_{ij}\right)_{ij}"'] \\ n \end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$\begin{array}{c} m \\ \left(a_{ij}\right)_{ij} \Big\downarrow \\ n \end{array} \quad \longmapsto \quad \begin{array}{c} \mathbb{Q}^{1\times m} \\ \Big\downarrow\, v \mapsto v \cdot \left(a_{ij}\right)_{ij} \\ \mathbb{Q}^{1\times n} \end{array}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$\begin{tikzcd} m \arrow[d, "(a_{ij})_{ij}"'] \\ n \end{tikzcd} \quad \longmapsto \quad \begin{tikzcd} \mathbb{Q}^{1\times m} \arrow[d, "v \mapsto v \cdot (a_{ij})_{ij}"] \\ \mathbb{Q}^{1\times n} \end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$\begin{tikzcd} m \arrow[d, "(a_{ij})_{ij}"'] \\ n \end{tikzcd} \qquad \longmapsto \qquad \begin{tikzcd} \mathbb{Q}^{1\times m} \arrow[d, "v \mapsto v \cdot (a_{ij})_{ij}"] \\ \mathbb{Q}^{1\times n} \end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}}$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}}$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$B$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}}$

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

B

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}} \quad \exists m \in \mathrm{mat}_{\mathbb{Q}}$:

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$m \qquad\qquad\qquad\qquad B$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}} \quad \exists m \in \mathrm{mat}_{\mathbb{Q}}$:

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$\mathbb{Q}^{1\times m}$$

$$m \longmapsto \qquad B$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}} \quad \exists m \in \mathrm{mat}_{\mathbb{Q}}$:

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$m \longmapsto \begin{array}{c} \mathbb{Q}^{1\times m} \\ \wr| \\ B \end{array}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}} \quad \exists m \in \mathrm{mat}_{\mathbb{Q}}$:

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\begin{tikzcd}
\mathrm{mat}_{\mathbb{Q}} \arrow[rr, "F"] & & \mathrm{vec}_{\mathbb{Q}} \\
m \arrow[rr, mapsto] & & \begin{array}{c}\mathbb{Q}^{1\times m} \\ \wr\| \\ B\end{array}
\end{tikzcd}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m,n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}} \quad \exists m \in \mathrm{mat}_{\mathbb{Q}} : \quad \mathbb{Q}^{1\times m} \simeq B$

Equivalences

Let's compare $\mathrm{mat}_{\mathbb{Q}}$ and $\mathrm{vec}_{\mathbb{Q}}$.

$$\mathrm{mat}_{\mathbb{Q}} \xrightarrow{\quad F \quad} \mathrm{vec}_{\mathbb{Q}}$$

$$m \longmapsto \begin{array}{c} \mathbb{Q}^{1\times m} \\ \wr\| \\ B \end{array}$$

- $F : \mathrm{Obj}_{\mathrm{mat}_{\mathbb{Q}}} \longrightarrow \mathrm{Obj}_{\mathrm{vec}_{\mathbb{Q}}}$
- $\mathrm{Hom}_{\mathrm{mat}_{\mathbb{Q}}}(m, n) \longrightarrow \mathrm{Hom}_{\mathrm{vec}_{\mathbb{Q}}}(\mathbb{Q}^{1\times m}, \mathbb{Q}^{1\times n})$ (respects id and $\circ$)
- $\forall B \in \mathrm{vec}_{\mathbb{Q}} \quad \exists m \in \mathrm{mat}_{\mathbb{Q}} : \quad \mathbb{Q}^{1\times m} \simeq B \quad m = \dim(B)$

Categorical abstraction

$$\mathrm{mat}_{\mathbb{Q}} \simeq \mathrm{vec}_{\mathbb{Q}}$$

Categorical abstraction

$$\mathrm{mat}_{\mathbb{Q}} \simeq \mathrm{vec}_{\mathbb{Q}}$$

For a category theorist, these two categories look the same.

Categorical abstraction

Categorical abstraction

Q: What is a finite dimensional $\mathbb{Q}$-vector space?

Categorical abstraction

Q: What is a finite dimensional $\mathbb{Q}$-vector space?

A: An object in the **category** of finite dim. $\mathbb{Q}$-vector spaces.

Categorical abstraction

Q: What is a finite dimensional $\mathbb{Q}$-vector space?

A: An object in the **category** of finite dim. $\mathbb{Q}$-vector spaces.

Q: What is the **category** of finite dimensional $\mathbb{Q}$-vector spaces?

Categorical abstraction

Q: What is a finite dimensional $\mathbb{Q}$-vector space?

A: An object in the **category** of finite dim. $\mathbb{Q}$-vector spaces.

Q: What is the **category** of finite dimensional $\mathbb{Q}$-vector spaces?

$$\mathrm{mat}_{\mathbb{Q}} \simeq \mathrm{vec}_{\mathbb{Q}}$$

Categorical abstraction

Q: What is a finite dimensional $\mathbb{Q}$-vector space?

A: An object in the **category** of finite dim. $\mathbb{Q}$-vector spaces.

Q: What is the **category** of finite dimensional $\mathbb{Q}$-vector spaces?

$$\mathrm{mat}_{\mathbb{Q}} \simeq \mathrm{vec}_{\mathbb{Q}}$$

$\mathrm{mat}_{\mathbb{Q}}$ is a **computerfriendly** model of $\mathrm{vec}_{\mathbb{Q}}$.

Categorical abstraction

Q: What is a finite dimensional $\mathbb{Q}$-vector space?

A: An object in the **category** of finite dim. $\mathbb{Q}$-vector spaces.

Q: What is the **category** of finite dimensional $\mathbb{Q}$-vector spaces?

$$\mathrm{mat}_{\mathbb{Q}} \simeq \mathrm{vec}_{\mathbb{Q}}$$

$\mathrm{mat}_{\mathbb{Q}}$ is a **computerfriendly** model of $\mathrm{vec}_{\mathbb{Q}}$.

We want to use categories to model **computational contexts** instead of “isolated” objects.

Outline

Categorical abstraction is a powerful organizing principle and computational tool.

1. What is categorical abstraction?
2. How can it be used as an organizing priniciple?
3. Why is it a computational tool?

Computable categories

Computable categories

A category becomes computable through

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

1 2 1

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

$$1 \qquad\qquad 2 \qquad\qquad 1$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

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

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

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

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

$$\begin{tikzcd}
1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[rr, bend left, "\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)"] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] & 1
\end{tikzcd}$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

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

$$\begin{tikzcd} 1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[rr, bend left] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] & 1 \end{tikzcd}$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

$$
\begin{tikzcd}
1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[loop below, "(1)"] \arrow[rr, bend left, "\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)"] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] \arrow[loop below, "\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}"] & 1 \arrow[loop below, "(1)"]
\end{tikzcd}
$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

$\mathrm{mat}_{\mathbb{Q}}$

$$
\begin{tikzcd}[column sep=huge]
1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[rr, bend left, "\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)"] \arrow[loop below, "(1)"] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] \arrow[loop below, "\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}"] & 1 \arrow[loop below, "(1)"]
\end{tikzcd}
$$

$\mathrm{vec}_{\mathbb{Q}}$ and $\mathrm{mat}_{\mathbb{Q}}$ are examples of abelian categories.

The language of category theory

$\mathrm{vec}_{\mathbb{Q}}$ and $\mathrm{mat}_{\mathbb{Q}}$ are examples of abelian categories.

Some categorical operations in abelian categories

The language of category theory

$\mathrm{vec}_{\mathbb{Q}}$ and $\mathrm{mat}_{\mathbb{Q}}$ are examples of abelian categories.

Some categorical operations in abelian categories

- $\oplus : \mathrm{Obj} \times \mathrm{Obj} \to \mathrm{Obj}$

The language of category theory

$\mathrm{vec}_{\mathbb{Q}}$ and $\mathrm{mat}_{\mathbb{Q}}$ are examples of abelian categories.

Some categorical operations in abelian categories

- $\oplus : \mathrm{Obj} \times \mathrm{Obj} \to \mathrm{Obj}$
- $+, - : \mathsf{Hom}(A, B) \times \mathsf{Hom}(A, B) \to \mathsf{Hom}(A, B)$

The language of category theory

$\mathrm{vec}_{\mathbb{Q}}$ and $\mathrm{mat}_{\mathbb{Q}}$ are examples of abelian categories.

Some categorical operations in abelian categories

- $\oplus : \mathrm{Obj} \times \mathrm{Obj} \to \mathrm{Obj}$
- $+, - : \mathrm{Hom}(A, B) \times \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, B)$
- $\ker : \mathrm{Hom}(A, B) \to \mathrm{Obj}$

The language of category theory

$\mathrm{vec}_{\mathbb{Q}}$ and $\mathrm{mat}_{\mathbb{Q}}$ are examples of abelian categories.

Some categorical operations in abelian categories

- $\oplus : \mathrm{Obj} \times \mathrm{Obj} \to \mathrm{Obj}$
- $+, - : \mathsf{Hom}(A, B) \times \mathsf{Hom}(A, B) \to \mathsf{Hom}(A, B)$
- $\mathsf{ker} : \mathsf{Hom}(A, B) \to \mathrm{Obj}$

Categorical description of the kernel

What do we want from a **kernel**?

Categorical description of the kernel

What do we want from a **kernel**?

Given $\alpha : V \longrightarrow W$ in $\mathrm{vec}_{\mathbb{Q}}$.

What do we want from a **kernel**?

Given $\alpha : V \longrightarrow W$ in $\mathrm{vec}_{\mathbb{Q}}$.

$$\ker(\alpha) = \left\{ v \in V \mid \alpha(v) = 0 \right\}$$

Categorical description of the kernel

What do we want from a **kernel**?

Given $\alpha : V \longrightarrow W$ in $\mathrm{vec}_{\mathbb{Q}}$.

$$\ker(\alpha) = \{ v \in V \mid \alpha(v) = 0 \}$$

Complete understanding about what is mapped to 0.

Categorical description of the kernel

Complete understanding about what is mapped to 0.

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$m \xrightarrow{\left(a_{ij}\right)_{ij}} n$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$\dim\big(\ker(a_{ij})\big)$$

$$m \xrightarrow{\;\;\left(a_{ij}\right)_{ij}\;\;} n$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$\dim\left(\ker(a_{ij})\right)$

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

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$2$$

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

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$\dim\left(\,\ker(a_{ij})\right)$$

$$3 \xrightarrow{\begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}} 2$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$2$$

$$3 \xrightarrow{\begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}} 2$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$2$$

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

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$
\begin{tikzcd}
2 \arrow[dr, "\begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}"'] & \\
& 3 \arrow[r, "\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}"] & 2
\end{tikzcd}
$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$
\begin{tikzcd}
2 \arrow[dr, "\begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}"'] \arrow[drr, bend left, "0"] & & \\
& 3 \arrow[r, "\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}"] & 2
\end{tikzcd}
$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$
\begin{tikzcd}
2 \arrow[dr, "\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}"'] \arrow[drr, bend left, "0"] & & \\
& 3 \arrow[r, "\begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}"] & 2
\end{tikzcd}
$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$
\begin{tikzcd}
2 \arrow[dr, "\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}"'] \arrow[drr, bend left, "0"] & & \\
& 3 \arrow[r, "\begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}"] & 2 \\
1 \arrow[ur, "\begin{pmatrix} 2 & 1 & -1 \end{pmatrix}"] & &
\end{tikzcd}
$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$
\begin{tikzcd}
2 \arrow[dr, "\begin{pmatrix}1 & 1 & 0\\ 0 & 1 & 1\end{pmatrix}"'] \arrow[drr, bend left, "0"] & & \\
& 3 \arrow[r, "\begin{pmatrix}1 & -1\\ -1 & 1\\ 1 & -1\end{pmatrix}"] & 2 \\
1 \arrow[ur, "\begin{pmatrix}2 & 1 & -1\end{pmatrix}"] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}
$$

Categorical description of the kernel

Complete understanding about what is mapped to 0.

$$
\begin{tikzcd}
2 \arrow[dr, "{\begin{pmatrix}1 & 1 & 0\\ 0 & 1 & 1\end{pmatrix}}"'] \arrow[drr, bend left, "0"] & & \\
& 3 \arrow[r, "{\begin{pmatrix}1 & -1\\ -1 & 1\\ 1 & -1\end{pmatrix}}"] & 2 \\
1 \arrow[uu, dashed, bend left, "{\begin{pmatrix}2 & -1\end{pmatrix}}"] \arrow[ur, "{\begin{pmatrix}2 & 1 & -1\end{pmatrix}}"] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}
$$

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$.

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$.

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

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of $\varphi \ldots$

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

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

... one needs an object $\mathrm{KernelObject}(\varphi)$,

$$\mathrm{KernelObject}(\varphi) \qquad A \xrightarrow{\varphi} B$$

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

... one needs an object KernelObject(φ),
its embedding $\kappa =$ KernelEmbedding(φ),

$$\begin{tikzcd} \text{KernelObject}(\varphi) \arrow[r, hook, "\kappa"] & A \arrow[r, "\varphi"] & B \end{tikzcd}$$

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

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

$$\begin{tikzcd}
\mathrm{KernelObject}(\varphi) \arrow[dr, hook, "\kappa"] \arrow[drr, bend left, "0"] & & \\
& A \arrow[r, "\varphi"] & B \\
T \arrow[ur, "\tau"] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}$$

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

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

$$\begin{tikzcd}
\mathrm{KernelObject}(\varphi) \arrow[dr, hook, "\kappa"] \arrow[drr, bend left, "0"] & & \\
& A \arrow[r, "\varphi"] & B \\
T \arrow[uu, dashed, "\lambda"] \arrow[ur, "\tau"] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}$$

Categorical description of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

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

$$
\begin{tikzcd}
\mathrm{KernelObject}(\varphi) \arrow[dr, hook, "\kappa"] \arrow[drr, bend left, "0"] & & \\
\circlearrowleft & A \arrow[r, "\varphi"] & B \\
T \arrow[uu, dashed, "\lambda"] \arrow[ur, "\tau"'] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}
$$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathsf{Hom}\,(m,n) := \mathbb{Q}^{m \times n}$$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathrm{Hom}\,(m, n) := \mathbb{Q}^{m \times n}$$

$$m \xrightarrow{(a_{ij})_{ij}} n$$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathrm{Hom}\,(m, n) := \mathbb{Q}^{m \times n}$$

KernelObject$((a_{ij})_{ij})$

$$m \xrightarrow{(a_{ij})_{ij}} n$$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathsf{Hom}\,(m,n) := \mathbb{Q}^{m\times n}$$

$\mathsf{KernelObject}((a_{ij})_{ij})$

$$m \xrightarrow{(a_{ij})_{ij}} n$$

Compute

- $\mathsf{KernelObject}((a_{ij})_{ij}) := m - \mathrm{rank}((a_{ij})_{ij})$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathrm{Hom}\,(m, n) := \mathbb{Q}^{m \times n}$$

$$\begin{tikzcd} \mathrm{KernelObject}((a_{ij})_{ij}) \arrow[dr, hook, "\kappa"] & & \\ & m \arrow[r, "(a_{ij})_{ij}"] & n \end{tikzcd}$$

Compute

- $\mathrm{KernelObject}((a_{ij})_{ij}) := m - \mathrm{rank}((a_{ij})_{ij})$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathrm{Hom}\,(m, n) := \mathbb{Q}^{m \times n}$$

$$\begin{tikzcd}
\mathrm{KernelObject}((a_{ij})_{ij}) \arrow[dr, hook, "\kappa"] & & \\
& m \arrow[r, "(a_{ij})_{ij}"] & n
\end{tikzcd}$$

Compute

- $\mathrm{KernelObject}((a_{ij})_{ij}) := m - \mathrm{rank}((a_{ij})_{ij})$
- $\kappa :=$ matrix whose rows form a basis of solutions of $X \cdot (a_{ij})_{ij} = 0$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathrm{Hom}\,(m, n) := \mathbb{Q}^{m \times n}$$

$$\begin{tikzcd}
\mathrm{KernelObject}((a_{ij})_{ij}) \arrow[dr, hook, "\kappa"] & & \\
& m \arrow[r, "(a_{ij})_{ij}"] & n \\
t \arrow[ur, "\tau"] & &
\end{tikzcd}$$

Compute

- $\mathrm{KernelObject}((a_{ij})_{ij}) := m - \mathrm{rank}((a_{ij})_{ij})$
- $\kappa :=$ matrix whose rows form a basis of solutions of $X \cdot (a_{ij})_{ij} = 0$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathrm{Hom}\,(m, n) := \mathbb{Q}^{m \times n}$$

$$\begin{tikzcd}
\mathrm{KernelObject}((a_{ij})_{ij}) \arrow[dr, hook, "\kappa"] & & \\
& m \arrow[r, "(a_{ij})_{ij}"] & n \\
t \arrow[uu, dashed, "\lambda"] \arrow[ur, "\tau"'] & &
\end{tikzcd}$$

Compute

- $\mathrm{KernelObject}((a_{ij})_{ij}) := m - \mathrm{rank}((a_{ij})_{ij})$
- $\kappa :=$ matrix whose rows form a basis of solutions of $X \cdot (a_{ij})_{ij} = 0$

Implementation of the kernel

$\mathrm{mat}_{\mathbb{Q}}$

$$\mathrm{Obj} := \mathbb{N}_0, \qquad \mathrm{Hom}\,(m,n) := \mathbb{Q}^{m\times n}$$

$$\begin{tikzcd}
\mathrm{KernelObject}((a_{ij})_{ij}) \arrow[rr, hook, "\kappa"] & & m \arrow[r, "(a_{ij})_{ij}"] & n \\
t \arrow[u, dashed, "\lambda"] \arrow[urr, "\tau"'] & & &
\end{tikzcd}$$

Compute

- $\mathrm{KernelObject}((a_{ij})_{ij}) := m - \mathrm{rank}((a_{ij})_{ij})$
- $\kappa :=$ matrix whose rows form a basis of solutions of $X \cdot (a_{ij})_{ij} = 0$
- $\lambda :=$ the unique solution of $X \cdot \kappa = \tau$

The language of category theory

The language of category theory

Given a diagram of vector spaces:

The language of category theory

Given a diagram of vector spaces:

$$\begin{tikzcd} \ker \arrow[r, hook] & A' \arrow[r] \arrow[d, "\alpha"'] & B' \arrow[d] \\ \ker \arrow[r, hook] & A \arrow[r] & B \end{tikzcd}$$

The language of category theory

Given a diagram of vector spaces:

$$\begin{tikzcd}
\textcolor{blue}{\ker} \arrow[r, hook] \arrow[d, dashed] & A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\textcolor{blue}{\ker} \arrow[r, hook] & A \arrow[r, blue] & B
\end{tikzcd}$$

The language of category theory

Given a diagram of vector spaces:

$$\begin{tikzcd}
x \in \ker \arrow[r, hook] \arrow[d, dashed] & A' \arrow[r] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\ker \arrow[r, hook] & A \arrow[r] & B
\end{tikzcd}$$

The language of category theory

Given a diagram of vector spaces:

$$\begin{tikzcd} x \in \textcolor{blue}{\ker} \arrow[r, hook] \arrow[d, dashed] & x \in A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\ \textcolor{blue}{\ker} \arrow[r, hook] & A \arrow[r, blue] & B \end{tikzcd}$$

The language of category theory

Given a diagram of vector spaces:

$$
\begin{tikzcd}
x \in \textcolor{blue}{\ker} \arrow[r, hook] \arrow[d, dashed] & x \in A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\textcolor{blue}{\ker} \arrow[r, hook] & \alpha(x) \in A \arrow[r, blue] & B
\end{tikzcd}
$$

The language of category theory

Given a diagram of vector spaces:

$$\begin{tikzcd}
x \in \color{blue}{\ker} \arrow[r, hook] \arrow[d, dashed] & x \in A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\color{blue}{\ker} \arrow[r, hook] & \alpha(x) \in A \arrow[r, blue] & 0 \in B
\end{tikzcd}$$

The language of category theory

Given a diagram of vector spaces:

$$
\begin{tikzcd}
x \in {\color{blue}\ker} \arrow[r, hook] \arrow[d, dashed] & x \in A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\alpha(x) \in {\color{blue}\ker} \arrow[r, hook] & \alpha(x) \in A \arrow[r, blue] & 0 \in B
\end{tikzcd}
$$

The language of category theory

Given a diagram of vector spaces:

$$\begin{tikzcd}
x \in \textcolor{blue}{\ker} \arrow[r, hook] \arrow[d, dashed] & x \in A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\alpha(x) \in \textcolor{blue}{\ker} \arrow[r, hook] & \alpha(x) \in A \arrow[r, blue] & 0 \in B
\end{tikzcd}$$

The language of category theory

The same example in the language of category theory:

$$\begin{tikzcd}
\color{blue}{\ker} \arrow[r, "\kappa'"] \arrow[d, dashed] & A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\color{blue}{\ker} \arrow[r] & A \arrow[r, blue, "\color{blue}{\varphi}"] & B
\end{tikzcd}$$

The language of category theory

The same example in the language of category theory:

$$\begin{tikzcd}
\color{blue}{\mathrm{ker}} \arrow[r, "\kappa'"] \arrow[d, dashed] & A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\color{blue}{\mathrm{ker}} \arrow[r] & A \arrow[r, blue, "\color{blue}{\varphi}"] & B
\end{tikzcd}$$

$\downarrow$

The language of category theory

The same example in the language of category theory:

$$\begin{tikzcd}
\color{blue}{\mathrm{ker}} \arrow[r, "\kappa'"] \arrow[d, dashed] & A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\color{blue}{\mathrm{ker}} \arrow[r] & A \arrow[r, blue, "\color{blue}{\varphi}"] & B
\end{tikzcd}$$

$$\begin{tikzcd} {} \arrow[d, dashed] \\ {} \end{tikzcd} =$$

The language of category theory

The same example in the language of category theory:

$$\begin{tikzcd}
\color{blue}{\ker} \arrow[r, "\kappa'"] \arrow[d, dashed] \arrow[dr, red] & A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\color{blue}{\ker} \arrow[r] & A \arrow[r, blue, "\varphi"] & B
\end{tikzcd}$$

$$\downarrow \; = \qquad \color{red}{\alpha \circ \kappa'}$$

The language of category theory

The same example in the language of category theory:

$$\begin{tikzcd}
\mathrm{ker} \arrow[r, "\kappa'"] \arrow[d, dashed] \arrow[dr, color=red] & A' \arrow[r, color=blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\mathrm{ker} \arrow[r] & A \arrow[r, color=blue, "\varphi"] & B
\end{tikzcd}$$

$$\downarrow \; = \mathsf{KernelLift}(\varphi, \alpha \circ \kappa')$$

The same example in the language of category theory:

$$\begin{tikzcd}
\mathrm{ker} \arrow[r, "\kappa'"] \arrow[d, dashed] \arrow[dr, red] & A' \arrow[r, blue] \arrow[d, "\alpha"'] & B' \arrow[d] \\
\mathrm{ker} \arrow[r] & A \arrow[r, blue, "\varphi"] & B
\end{tikzcd}$$

$$\downarrow \; = \mathrm{KernelLift}(\varphi, \alpha \circ \kappa')$$

This term may be interpreted in other contexts as well.

The language of category theory

$\mathrm{vec}_{\mathbb{Q}}$

$\mathrm{vec}_{\mathbb{Q}}$ $\mathrm{mat}_{\mathbb{Q}}$

The language of category theory

$$\mathrm{vec}_{\mathbb{Q}} \qquad \simeq \qquad \mathrm{mat}_{\mathbb{Q}}$$

The language of category theory

$$\underbrace{\mathrm{vec}_{\mathbb{Q}} \qquad \simeq \qquad \mathrm{mat}_{\mathbb{Q}}}_{\text{categorical abstraction}}$$

The language of category theory

$$\underbrace{\mathrm{vec}_{\mathbb{Q}} \qquad \simeq \qquad \mathrm{mat}_{\mathbb{Q}}}_{\text{categorical abstraction}}$$

$\vdots$

KernelObject

KernelEmbedding

KernelLift

$\vdots$

The language of category theory

$$\underbrace{\mathrm{vec}_{\mathbb{Q}} \quad \simeq \quad \mathrm{mat}_{\mathbb{Q}}}_{\text{categorical abstraction}} \quad \circlearrowleft \quad \begin{array}{l} \vdots \\ \mathsf{KernelObject} \\ \mathsf{KernelEmbedding} \\ \mathsf{KernelLift} \\ \vdots \end{array}$$

The language of category theory

$$\underbrace{\mathrm{vec}_{\mathbb{Q}} \quad \simeq \quad \mathrm{mat}_{\mathbb{Q}}}_{\text{categorical abstraction}} \quad \circlearrowleft \quad \underbrace{\begin{array}{l} \vdots \\ \mathsf{KernelObject} \\ \mathsf{KernelEmbedding} \\ \mathsf{KernelLift} \\ \vdots \end{array}}_{\text{categorical language}}$$

The language of category theory

$$\mathcal{A} \quad\quad \circlearrowleft \quad\quad \underbrace{\begin{array}{l} \vdots \\ \mathsf{KernelObject} \\ \mathsf{KernelEmbedding} \\ \mathsf{KernelLift} \\ \vdots \end{array}}_{\text{categorical language}}$$

An introduction to finitely presented modules

Let R be a ring.

Definition

A (left) R-module M is called **finitely presented** if

$$M \cong \frac{R^{1\times n}}{\langle r_1, \dots, r_m \rangle}$$

for $n, m \in \mathbb{N}_0$, $r_1, \dots, r_m \in R^{1\times n}$.

Examples

$$M \cong \frac{R^{1\times n}}{\langle r_1,\ldots,r_m\rangle}$$

Examples

$M \cong \frac{R^{1\times n}}{\langle r_1,\ldots,r_m \rangle}$

$R = \mathbb{Q}[x, y, z]$

Examples

$M \cong \frac{R^{1 \times n}}{\langle r_1, \ldots, r_m \rangle}$

$R = \mathbb{Q}[x, y, z]$

$$\frac{\mathbb{Q}[x, y, z]}{\langle x^2 + y^2 + z^2 - 1 \rangle}$$

Examples

$M \cong \frac{R^{1\times n}}{\langle r_1,\ldots,r_m\rangle}$

$R = \mathbb{Q}[x, y, z]$

$$\frac{\mathbb{Q}[x, y, z]}{\langle x^2 + y^2 + z^2 - 1\rangle}$$

$$\frac{\mathbb{Q}[x, y, z]^{1\times 2}}{\langle \begin{pmatrix} x & -y \end{pmatrix}, \begin{pmatrix} y & -x \end{pmatrix} \rangle}$$

Examples

$M \cong \frac{R^{1\times n}}{\langle r_1,\ldots,r_m\rangle}$

$R = \mathbb{Q}[x, y, z]$

$$\frac{\mathbb{Q}[x,y,z]}{\langle x^2+y^2+z^2-1\rangle}$$

$$\frac{\mathbb{Q}[x,y,z]^{1\times 2}}{\langle \begin{pmatrix} x & -y \end{pmatrix}, \begin{pmatrix} y & -x \end{pmatrix}\rangle}$$

$$\frac{\mathbb{Q}[x,y,z]^{1\times 6}}{\left\langle \text{Rows of} \begin{pmatrix} 0 & 0 & 0 & 0 & xz & -z^2 \\ 0 & 0 & 0 & 0 & xy & -yz \\ 0 & -x^2z+xyz+xz^2 & y^2z & -xz+yz & x-y & 0 \\ 0 & 0 & 0 & 0 & x^2 & -xz \\ -xy & -x^3+x^2y+x^2z & xy^2 & -x^2+xy & 0 & x-y \\ z & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle}$$

Examples

$M \cong \frac{R^{1\times n}}{\langle r_1,\ldots,r_m\rangle}$

$R = \mathbb{Q}[x,y,z]$

$$\frac{\mathbb{Q}[x,y,z]}{\langle x^2+y^2+z^2-1\rangle}$$

$$\frac{\mathbb{Q}[x,y,z]^{1\times 2}}{\langle \begin{pmatrix} x & -y \end{pmatrix}, \begin{pmatrix} y & -x \end{pmatrix}\rangle}$$

$$\frac{\mathbb{Q}[x,y,z]^{1\times 6}}{\left\langle \begin{pmatrix} 0 & 0 & 0 & 0 & xz & -z^2 \\ 0 & 0 & 0 & 0 & xy & -yz \\ 0 & -x^2z+xyz+xz^2 & y^2z & -xz+yz & x-y & 0 \\ 0 & 0 & 0 & 0 & x^2 & -xz \\ -xy & -x^3+x^2y+x^2z & xy^2 & -x^2+xy & 0 & x-y \\ z & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle}$$

Category of finitely presented modules

Finitely presented R-modules form a category

Category of finitely presented modules

Finitely presented R-modules form a category

$$\mathrm{mod}_R$$

Category of finitely presented modules

Finitely presented R-modules form a category

$$\mathrm{mod}_R$$

with R-linear maps as morphisms.

Category of finitely presented modules

Finitely presented R-modules form a category

$$\mathrm{mod}_R$$

with R-linear maps as morphisms.

Computerfriendly model?

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelObject
   - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
    - objects
    - morphisms
2. **Algorithms**
    - composition
    - identities
    - KernelObject
    - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelObject
   - $\dots$

Data structures: objects

$$\frac{\mathbb{Q}[x,y,z]^{1\times 6}}{\left\langle \text{Rows of} \begin{pmatrix} 0 & 0 & 0 & 0 & xz & -z^2 \\ 0 & 0 & 0 & 0 & xy & -yz \\ 0 & -x^2z+xyz+xz^2 & y^2z & -xz+yz & x-y & 0 \\ 0 & 0 & 0 & 0 & x^2 & -xz \\ -xy & -x^3+x^2y+x^2z & xy^2 & -x^2+xy & 0 & x-y \\ z & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle}$$

Data structures: objects

$$\frac{\mathbb{Q}[x,y,z]^{1\times 6}}{\left\langle \begin{pmatrix} 0 & 0 & 0 & 0 & xz & -z^2 \\ 0 & 0 & 0 & 0 & xy & -yz \\ 0 & -x^2z+xyz+xz^2 & y^2z & -xz+yz & x-y & 0 \\ 0 & 0 & 0 & 0 & x^2 & -xz \\ -xy & -x^3+x^2y+x^2z & xy^2 & -x^2+xy & 0 & x-y \\ z & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle}$$

Data structures: objects

$$\frac{\mathbb{Q}[x,y,z]^{1\times 6}}{\left\langle \begin{pmatrix} 0 & 0 & 0 & 0 & xz & -z^2 \\ 0 & 0 & 0 & 0 & xy & -yz \\ 0 & -x^2z+xyz+xz^2 & y^2z & -xz+yz & x-y & 0 \\ 0 & 0 & 0 & 0 & x^2 & -xz \\ -xy & -x^3+x^2y+x^2z & xy^2 & -x^2+xy & 0 & x-y \\ z & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle}$$

Data structures: objects

$$\frac{\mathbb{Q}[x,y,z]^{1\times 6}}{\left\langle \begin{pmatrix} 0 & 0 & 0 & 0 & xz & -z^2 \\ 0 & 0 & 0 & 0 & xy & -yz \\ 0 & -x^2z+xyz+xz^2 & y^2z & -xz+yz & x-y & 0 \\ 0 & 0 & 0 & 0 & x^2 & -xz \\ -xy & -x^3+x^2y+x^2z & xy^2 & -x^2+xy & 0 & x-y \\ z & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle}$$

Idea: a matrix $M \in R^{m\times n}$ can represent the module $\frac{R^{1\times n}}{\langle M\rangle}$.

Data structures: objects

$$\frac{\mathbb{Q}[x,y,z]^{1\times 6}}{\left\langle \begin{pmatrix} 0 & 0 & 0 & 0 & xz & -z^2 \\ 0 & 0 & 0 & 0 & xy & -yz \\ 0 & -x^2z+xyz+xz^2 & y^2z & -xz+yz & x-y & 0 \\ 0 & 0 & 0 & 0 & x^2 & -xz \\ -xy & -x^3+x^2y+x^2z & xy^2 & -x^2+xy & 0 & x-y \\ z & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle}$$

Idea: a matrix $M \in R^{m\times n}$ can represent the module $\frac{R^{1\times n}}{\langle M\rangle}$.

Objects

$$\mathrm{Obj}_{\mathrm{fpres}_R} := \biguplus_{m,n\in\mathbb{N}_0} R^{m\times n}$$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelObject
   - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelObject
   - ...

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects ✓
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelObject
   - ...

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects ✓
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelObject
   - ...

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\frac{R^{1 \times n}}{\langle M \rangle} \qquad\qquad \frac{R^{1 \times n'}}{\langle M' \rangle}$$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\frac{R^{1 \times n}}{\langle M \rangle} \longrightarrow \frac{R^{1 \times n'}}{\langle M' \rangle}$$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\frac{R^{1 \times n}}{\langle M \rangle} \longrightarrow \frac{R^{1 \times n'}}{\langle M' \rangle}$$

$$\overline{e_i}$$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$
\begin{array}{ccc}
\frac{R^{1 \times n}}{\langle M \rangle} & \longrightarrow & \frac{R^{1 \times n'}}{\langle M' \rangle} \\[1em]
\overline{e_i} & \longmapsto & \overline{r_i}
\end{array}
$$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\begin{array}{ccc}
\frac{R^{1 \times n}}{\langle M \rangle} & \xrightarrow{\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} & \frac{R^{1 \times n'}}{\langle M' \rangle} \\
 & & \\
\overline{e_i} & \longmapsto & \overline{r_i}
\end{array}$$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\begin{array}{ccc} \frac{R^{1\times n}}{\langle M \rangle} & \xrightarrow{\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} & \frac{R^{1\times n'}}{\langle M' \rangle} \\ & & \\ \overline{e_i} & \longmapsto & \overline{r_i} \end{array}$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\begin{array}{ccc} \frac{R^{1\times n}}{\langle M \rangle} & \xrightarrow{A := \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} & \frac{R^{1\times n'}}{\langle M' \rangle} \\ \overline{e_i} & \longmapsto & \overline{r_i} \end{array}$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

$A \in R^{n \times n'}$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$A := \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

$$\begin{array}{ccc} \frac{R^{1\times n}}{\langle M \rangle} & \xrightarrow{\hspace{3em}} & \frac{R^{1\times n'}}{\langle M' \rangle} \\ & & \\ \overline{e_i} & \longmapsto & \overline{r_i} \end{array}$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

$A \in R^{n \times n'}$ such that

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\frac{R^{1\times n}}{\langle M \rangle} \xrightarrow{A := \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} \frac{R^{1\times n'}}{\langle M' \rangle}$$

$$\overline{e_i} \longmapsto \overline{r_i}$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

$A \in R^{n \times n'}$ such that $\{\text{Rows of } M \cdot A\} \subseteq \langle M' \rangle$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$
\begin{array}{ccc}
\frac{R^{1\times n}}{\langle M \rangle} & \xrightarrow{A := \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} & \frac{R^{1\times n'}}{\langle M' \rangle} \\
 & & \\
\overline{e_i} & \longmapsto & \overline{r_i}
\end{array}
$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

$$A \in R^{n \times n'} \quad \text{such that} \quad \exists X \in R^{m \times m'} : M \cdot A = X \cdot M'$$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\begin{array}{ccc} \frac{R^{1\times n}}{\langle M \rangle} & \xrightarrow{A := \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} & \frac{R^{1\times n'}}{\langle M' \rangle} \\[2em] \overline{e_i} & \longmapsto & \overline{r_i} \end{array}$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

$$A \in R^{n \times n'} \quad \text{such that} \quad \exists X \in R^{m \times m'} : M \cdot A = X \cdot M'$$

A defines the 0 morphism

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\begin{array}{ccc} \frac{R^{1 \times n}}{\langle M \rangle} & \xrightarrow{A := \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} & \frac{R^{1 \times n'}}{\langle M' \rangle} \\ \\ \overline{e_i} & \longmapsto & \overline{r_i} \end{array}$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

$$A \in R^{n \times n'} \quad \text{such that} \quad \exists X \in R^{m \times m'} : M \cdot A = X \cdot M'$$

A defines the 0 morphism iff $\exists X \in R^{n \times m'} : A = X \cdot M'$

Data structures: morphisms

Given: $M \in R^{m \times n}$ and $M' \in R^{m' \times n'}$.

$$\frac{R^{1\times n}}{\langle M \rangle} \xrightarrow{A := \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}} \frac{R^{1\times n'}}{\langle M' \rangle}$$

$$\overline{e_i} \longmapsto \overline{r_i}$$

$\mathrm{Hom}_{\mathrm{fpres}_R}(M, M') :=$

$$A \in R^{n \times n'} \quad \text{such that} \quad \exists X \in R^{m \times m'} : M \cdot A = X \cdot M'$$

A defines the 0 morphism iff $\exists X \in R^{n \times m'} : A = X \cdot M'$

Data structures: morphisms

Deciding **well-definedness** and **being zero** of morphisms in fpres_R requires finding particular solutions of inhomogeneous linear systems.

Data structures: morphisms

Deciding **well-definedness** and **being zero** of morphisms in fpres_R requires finding particular solutions of inhomogeneous linear systems.

Ring	Algorithms

Data structures: morphisms

Deciding **well-definedness** and **being zero** of morphisms in fpres_R requires finding particular solutions of inhomogeneous linear systems.

Ring	Algorithms
$\mathbb{Q}$	Gauss

Data structures: morphisms

Deciding **well-definedness** and **being zero** of morphisms in fpres_R requires finding particular solutions of inhomogeneous linear systems.

Ring	Algorithms
$\mathbb{Q}$	Gauss
$\mathbb{Z}$	Hermite

Data structures: morphisms

Deciding **well-definedness** and **being zero** of morphisms in fpres_R requires finding particular solutions of inhomogeneous linear systems.

Ring	Algorithms
$\mathbb{Q}$	Gauss
$\mathbb{Z}$	Hermite
$\mathbb{Q}[x, y, z]$	Buchberger

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects ✓
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelEmbedding
   - ...

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects $\checkmark$
   - morphisms
2. **Algorithms**
   - composition
   - identities
   - KernelEmbedding
   - $\ldots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures**
   - objects ✓
   - morphisms ✓
2. **Algorithms**
   - composition
   - identities
   - KernelEmbedding
   - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** ✓
   - objects ✓
   - morphisms ✓
2. **Algorithms**
   - composition
   - identities
   - KernelEmbedding
   - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** $\checkmark$
   - objects $\checkmark$
   - morphisms $\checkmark$
2. **Algorithms**
   - composition
   - identities
   - KernelEmbedding
   - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** $\checkmark$
   - objects $\checkmark$
   - morphisms $\checkmark$
2. **Algorithms**
   - composition
   - identities
   - KernelEmbedding
   - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** ✓
   - objects ✓
   - morphisms ✓
2. **Algorithms**
   - composition
   - identities
   - KernelEmbedding
   - ...

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** ✓
   - objects ✓
   - morphisms ✓
2. **Algorithms**
   - composition ✓
   - identities
   - KernelEmbedding
   - ...

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** $\checkmark$
   - objects $\checkmark$
   - morphisms $\checkmark$
2. **Algorithms**
   - composition $\checkmark$
   - identities
   - KernelEmbedding
   - $\dots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** $\checkmark$
   - objects $\checkmark$
   - morphisms $\checkmark$
2. **Algorithms**
   - composition $\checkmark$
   - identities
   - KernelEmbedding
   - $\ldots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** $\checkmark$
   - objects $\checkmark$
   - morphisms $\checkmark$
2. **Algorithms**
   - composition $\checkmark$
   - identities $\checkmark$
   - KernelEmbedding
   - $\ldots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** $\checkmark$
   - objects $\checkmark$
   - morphisms $\checkmark$
2. **Algorithms**
   - composition $\checkmark$
   - identities $\checkmark$
   - KernelEmbedding
   - $\ldots$

Applying the organizing principle

Goal: create computerfriendly model fpres_R of mod_R.

What we need

1. **Data structures** $\checkmark$
   - objects $\checkmark$
   - morphisms $\checkmark$
2. **Algorithms**
   - composition $\checkmark$
   - identities $\checkmark$
   - KernelEmbedding
   - $\ldots$

Kernels

We cannot expect kernels to exist in fpres_R for all rings:

Kernels

We cannot expect kernels to exist in fpres_R for all rings:

Example

$R := k[Z, X_i \mid i \in \mathbb{N}]/\langle ZX_i \mid i \in \mathbb{N}\rangle$.

Kernels

We cannot expect kernels to exist in fpres_R for all rings:

Example

$R := k[Z, X_i \mid i \in \mathbb{N}]/\langle ZX_i \mid i \in \mathbb{N}\rangle$.Then

$$\alpha : R \longrightarrow R : r \mapsto \overline{Z} \cdot r$$

Kernels

We cannot expect kernels to exist in fpres_R for all rings:

Example

$R := k[Z, X_i \mid i \in \mathbb{N}]/\langle ZX_i \mid i \in \mathbb{N}\rangle$.Then

$$\alpha : R \longrightarrow R : r \mapsto \overline{Z} \cdot r$$

has

$$\ker(\alpha) = \langle \overline{X_i} \mid i \in \mathbb{N}\rangle.$$

Kernels

Rings for which fpres_R has kernels are called **coherent.**

Kernels

Rings for which fpres_R has kernels are called **coherent.**

Examples

$\mathbb{Q}$

Kernels

Rings for which fpres_R has kernels are called **coherent.**

Examples

$\mathbb{Q}, \quad \mathbb{Z}$

Kernels

Rings for which fpres_R has kernels are called **coherent.**

Examples

$\mathbb{Q}, \quad \mathbb{Z}, \quad \mathbb{Q}[x, y, z]$

Kernels

Rings for which fpres_R has kernels are called **coherent.**

Examples

$\mathbb{Q}, \quad \mathbb{Z}, \quad \mathbb{Q}[x, y, z], \quad \mathbb{Q}[X_i \mid i \in \mathbb{N}]$

Kernels

Theorem

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

- **Input**: $A \in R^{m \times n}$.

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

- **Input**: $A \in R^{m \times n}$.
- **Output**: $S \in R^{s \times m}$ s.t. the rows of S generate the row kernel of A.

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

- **Input**: $A \in R^{m \times n}$.
- **Output**: $S \in R^{s \times m}$ s.t. the rows of S generate the row kernel of A.

Then we have an algorithm for computing

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

- **Input**: $A \in R^{m \times n}$.
- **Output**: $S \in R^{s \times m}$ s.t. the rows of S generate the row kernel of A.

Then we have an algorithm for computing

- KernelObject

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

- **Input**: $A \in R^{m \times n}$.
- **Output**: $S \in R^{s \times m}$ s.t. the rows of S generate the row kernel of A.

Then we have an algorithm for computing

- KernelObject
- KernelEmbedding

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

- **Input**: $A \in R^{m \times n}$.
- **Output**: $S \in R^{s \times m}$ s.t. the rows of S generate the row kernel of A.

Then we have an algorithm for computing

- KernelObject
- KernelEmbedding

in fpres_R.

Kernels

Theorem

Let R be a ring equipped with the following algorithm:

- **Input**: $A \in R^{m \times n}$.
- **Output**: $S \in R^{s \times m}$ s.t. the rows of S generate the row kernel of A.

Then we have an algorithm for computing

- KernelObject
- KernelEmbedding

in fpres_R.

Can you prove this theorem?

Outline

Categorical abstraction **is a powerful** **organizing principle** **and** **computational tool**.

1. What is categorical abstraction?
2. How can it be used as an organizing priniciple?
3. Why is it a computational tool?

Computing the intersection

Let $M_1 \subseteq N$ and $M_2 \subseteq N$ subobjects in an abelian category.

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category.

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category. Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
M_1 \arrow[dr, hook, "\iota_1"] & \\
& N \\
M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
 & M_1 \oplus M_2 & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category. Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] & & N \\
 & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
& M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] \arrow[dr, two heads, "\pi_2"'] & & N \\
& M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category. Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
& M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] & & N \\
& M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category. Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$
- $\kappa := \mathrm{KernelEmbedding}\left(\varphi\right)$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category. Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$
\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}
$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i \right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$
- $\kappa := \mathrm{KernelEmbedding}\left(\varphi \right)$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects in an abelian category. Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$
- $\kappa := \mathrm{KernelEmbedding}(\varphi)$
- $\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$$\pi_i := \text{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right),\ i = 1, 2$$

$$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$$

$$\kappa := \text{KernelEmbedding}\left(\varphi\right)$$

$$\gamma := \iota_1 \circ \pi_1 \circ \kappa$$

Translation to CAP

$\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right),\ i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

$\kappa := \mathrm{KernelEmbedding}\left(\varphi\right)$

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right),\ i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

```
lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );
```

$\kappa := \mathrm{KernelEmbedding}\left(\varphi\right)$

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$\pi_i := \text{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right),\ i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

```
lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );
```

$\kappa := \text{KernelEmbedding}\left(\varphi\right)$

```
kappa := KernelEmbedding( phi );
```

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right),\ i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

```
lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );
```

$\kappa := \mathrm{KernelEmbedding}\left(\varphi\right)$

```
kappa := KernelEmbedding( phi );
```

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

```
gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );

kappa := KernelEmbedding( phi );

gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );
kappa := KernelEmbedding( phi );
gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
IntersectionSubobjects := function( iota1, iota2 )

  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );

  kappa := KernelEmbedding( phi );

  gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
IntersectionSubobjects := function( iota1, iota2 )

  M1 := Source( iota1 );
  M2 := Source( iota2 );

  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );

  kappa := KernelEmbedding( phi );

  gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
IntersectionSubobjects := function( iota1, iota2 )

  M1 := Source( iota1 );
  M2 := Source( iota2 );

  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );

  kappa := KernelEmbedding( phi );

  gamma := PostCompose( lambda, kappa );

  return gamma;
end;
```

Translation to CAP

```
IntersectionSubobjects := function( iota1, iota2 )
  local M1, M2, pi1, pi2, lambda, phi, kappa, gamma;
  M1 := Source( iota1 );
  M2 := Source( iota2 );
  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );
  kappa := KernelEmbedding( phi );
  gamma := PostCompose( lambda, kappa );
  return gamma;
end;
```

Computing the intersection

Compute the intersection in $\mathrm{mat}_{\mathbb{Q}}$ of

$$\begin{tikzcd}
M_1 \arrow[r, hook, "\iota_1 := \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}"] \arrow[d, equal] & N \arrow[d, equal] & M_2 \arrow[l, hook', "\iota_2 := \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}"'] \arrow[d, equal] \\
2 & 3 & 2
\end{tikzcd}$$

Computing the intersection

Compute the intersection in $\mathrm{mat}_{\mathbb{Q}}$ of

$$
\begin{array}{ccccc}
M_1 & \xhookrightarrow{\iota_1 := \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}} & N & \xhookleftarrow{\iota_2 := \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}} & M_2 \\
\| & & \| & & \| \\
2 & & 3 & & 2
\end{array}
$$

```
gap> gamma := IntersectionOfSubobject( iota1, iota2 );
<A morphism in the category of matrices over Q>
```

Computing the intersection

Compute the intersection in $\mathrm{mat}_{\mathbb{Q}}$ of

$$\begin{array}{ccccc} M_1 & \xhookrightarrow{\iota_1 := \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}} & N & \xhookleftarrow{\iota_2 := \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}} & M_2 \\ \| & & \| & & \| \\ 2 & & 3 & & 2 \end{array}$$

```
gap> gamma := IntersectionOfSubobject( iota1, iota2 );
<A morphism in the category of matrices over Q>

gap> Display( gamma );
[ [  1,  1,  0 ] ]

A morphism in the category of matrices over Q
```

Computing the intersection

The same algorithm can be applied in fpres_R

Computing the intersection

The same algorithm can be applied in fpres_R (your turn).

Goals of the hands-on session

After the hands-on session, you will be able to

Goals of the hands-on session

After the hands-on session, you will be able to

- define abelian categories constructively,

Goals of the hands-on session

After the hands-on session, you will be able to

- define abelian categories constructively,
- compute with f.d. vector spaces and f.p. modules only using categorical operations,

Goals of the hands-on session

After the hands-on session, you will be able to

- define abelian categories constructively,
- compute with f.d. vector spaces and f.p. modules only using categorical operations,
- write short generic code that works on every instance of an abelian category.

`https://github.com/sebastianpos/cap-aachen2018`

Goals of the hands-on session

After the hands-on session, you will be able to

- define abelian categories constructively,
- compute with f.d. vector spaces and f.p. modules only using categorical operations,
- write short generic code that works on every instance of an abelian category,

Goals of the hands-on session

After the hands-on session, you will be able to

- define abelian categories constructively,
  - DirectSum, ProjectionInFactorOfDirectSum, InjectionOfCofactorOfDirectSum,
  - KernelObject, KernelEmbedding, KernelLift
  - CokernelObject, CokernelProjection, CokernelColift,
  - LiftAlongMonomorphism, ColiftAlongEpimorphism
- compute with f.d. vector spaces and f.p. modules only using categorical operations,
- write short generic code that works on every instance of an abelian category,

Goals of the hands-on session

After the hands-on session, you will be able to

- define abelian categories constructively,
  - DirectSum, ProjectionInFactorOfDirectSum, InjectionOfCofactorOfDirectSum,
  - KernelObject, KernelEmbedding, KernelLift
  - CokernelObject, CokernelProjection, CokernelColift,
  - LiftAlongMonomorphism, ColiftAlongEpimorphism
- compute with f.d. vector spaces and f.p. modules only using categorical operations,
  - model for vector spaces: category of matrices,
  - model for modules: category of (left) presentations,
- write short generic code that works on every instance of an abelian category,

Goals of the hands-on session

After the hands-on session, you will be able to

- define abelian categories constructively,
  - DirectSum, ProjectionInFactorOfDirectSum, InjectionOfCofactorOfDirectSum,
  - KernelObject, KernelEmbedding, KernelLift
  - CokernelObject, CokernelProjection, CokernelColift,
  - LiftAlongMonomorphism, ColiftAlongEpimorphism
- compute with f.d. vector spaces and f.p. modules only using categorical operations,
  - model for vector spaces: category of matrices,
  - model for modules: category of (left) presentations,
- write short generic code that works on every instance of an abelian category,
  - functoriality of kernels,
  - intersection of subobjects,
  - addition of subobjects.

Take-home message

Categorical abstraction is a powerful **organizing principle** and **computational tool**.