Karlsruhe Institute of Technology

Formale Systeme II: Theorie

(Introduction to) Category Theory

Summer Semester 2024

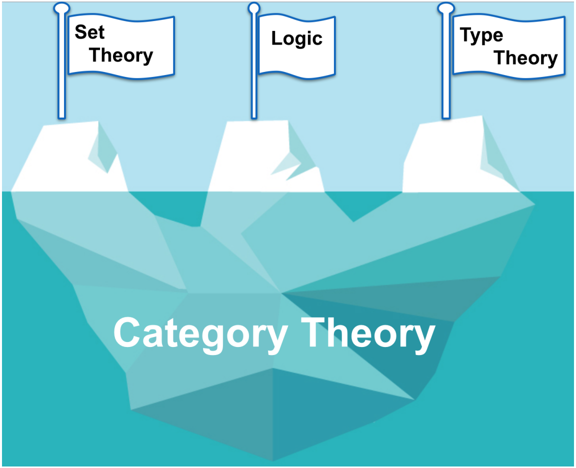
Prof. Dr. Bernard Beckert · Dr. Romain Pascual · Dr. Mattias Ulbrich

Motivation

Why study category theory?

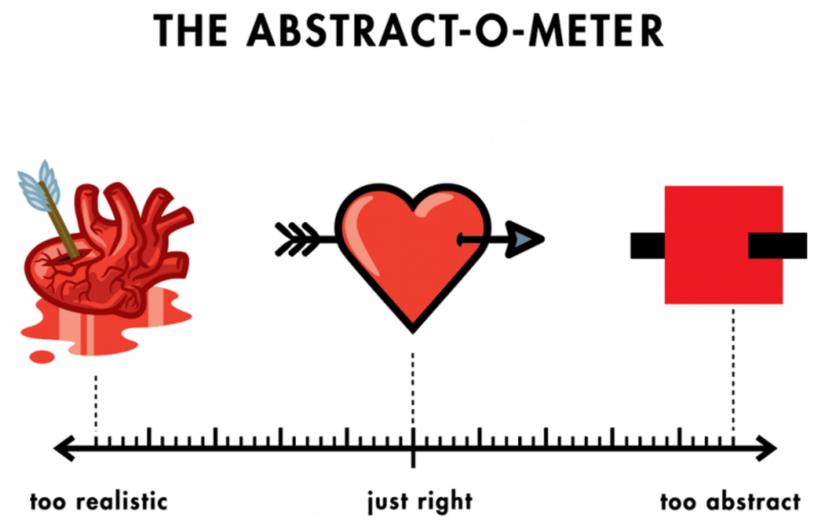
Why study category theory?

Category theory offers a **unified language** for mathematical structures, **generalizing** concepts across various fields.



(Some) Applications in computer science

- Categories with certain structures provide models for **type theories**.
- **Semantics of programming languages** requires an axiomatic theory of 'composable stuff', e.g., monads and functors in Haskell are rooted in category theory.
- Coalgebras generalize **automata** for reasoning on infinite data types.
- In **software engineering**, model transformations are described by triple graph grammars and bidirectional transformations.
- **Quantum computing** relies on monoidal categories and string diagrams.


THE ABSTRACT-O-METER
too realistic
just right
too abstract

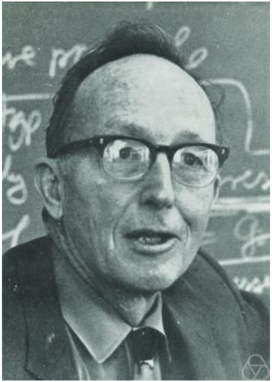
Motivation

The underlying philosophy of category theory is to replace the notions of sets and **membership relation** between sets (i.e., set theory) with an abstract notion of sets and **functions**.

In set theory, a set is defined by its internal structure. In category theory, objects are described by their relations with other objects of the same mathematical environment.

Category theory is a relational way of doing mathematics.

Eilenberg & Mac Lane



Category Theory, introduced by Samuel Eilenberg and Saunders Mac Lane in 1942-1945, provides an abstract language for expressing mathematical concepts and reasoning about them.

A bit of history

Category Theory significantly changed how mathematicians look at their subject, opening up new possibilities for important discoveries.

Computer science, linguistics, philosophy, ...

1970s

Logic (F.W. Lawvere et al.)

1960s

Algebraic geometry (A. Grothendieck et al.)

1950s

Algebraic topology and abstract algebra

1945: Eilenberg and Mac Lane's
"General theory of natural equivalences"

1940s

Further reading

- Saunders Mac Lane. **Categories for the working mathematician**. 2nd ed. Vol. 5. Springer, 1998
- Tom Leinster. **Basic Category Theory**. Dec. 29, 2016
- Bartosz Milewski. **Category Theory for Programmers**. 2018
- Benjamin C. Pierce. **Basic category theory for computer scientists**. Cambridge, MA, USA: MIT Press, 1991
- Michael Barr and Charles Wells. **Category theory for computing science**. Vol. 49. New York: Prentice Hall, 1990
- David Spivak. **Category Theory for the Sciences**. MIT Press, 2014
- Lectures by O. Caramello

Categories

What is a category?

- Objects
- A transitive reflexive relation

Some categorical definitions and results

A monad is a monoid in the category of endofunctors of some fixed category.

An elementary topos is a finitely complete Cartesian-closed category with a subobject classifier.

$$
\begin{tikzcd}
1_{\mathbb{W}\times\mathbb{D}} & & & \\
E_{\mathbb{D}}(P_v) \arrow[u, "!_{E_{\mathbb{D}}(P_v)}"] & L \arrow[ul, "!_L"'] & K \arrow[l, hook'] \arrow[r, hook] \arrow[ull, "!_K"'] & R \arrow[ulll, "!_R"'] \\
P_v \arrow[u, "E_{\mathbb{D}}"] \arrow[d, hook, "p_v"'] & E_{\mathbb{D}}(P_v) \times L \arrow[ul] \arrow[u] \arrow[d, "\pi_{\mathbb{D}}"] & E_{\mathbb{D}}(P_v) \times K \arrow[l, hook'] \arrow[r, hook] \arrow[ull] \arrow[u] \arrow[d, "\pi_{\mathbb{D}}"] & E_{\mathbb{D}}(P_v) \times R \arrow[ulll] \arrow[u] \arrow[d, "\pi_{\mathbb{D}}"] \\
\mathbb{P}(\mathcal{G}) & \iota(L, P_v) \arrow[d, hook, "m"] & \iota(K, P_v) \arrow[l, hook'] \arrow[r, hook] \arrow[d, hook] & \iota(R, P_v) \arrow[d, hook] \\
 & \mathcal{G} \arrow[ul, "\mathbb{P}"] & \mathcal{D} \arrow[l, hook'] \arrow[r, hook] & \mathcal{H}
\end{tikzcd}
$$

Some categorical definitions and results

A monad is a monoid in the category of endofunctors of some fixed category.

A monad is a wrapper around a function changing the return value.

An elementary topos is a finitely complete Cartesian-closed category with a subobject classifier.

A topos is a semantics for intuitionistic formal systems.

$$
\begin{tikzcd}
1_{\mathbb{W}\times\mathbb{D}} & & & \\
E_{\mathbb{D}}(P_v) \arrow[u, "!_{E_{\mathbb{D}}(P_v)}"] & L \arrow[ul, "!_L"'] & K \arrow[l, hook'] \arrow[r, hook] \arrow[ull, "!_K"'] & R \arrow[ulll, "!_R"'] \\
P_v \arrow[u, "E_{\mathbb{D}}"] \arrow[d, hook, "p_v"'] & E_{\mathbb{D}}(P_v)\times L \arrow[u] \arrow[ul] \arrow[d, "\pi_{\mathbb{D}}"] & E_{\mathbb{D}}(P_v)\times K \arrow[l, hook'] \arrow[r, hook] \arrow[u] \arrow[ull] \arrow[d, "\pi_{\mathbb{D}}"] & E_{\mathbb{D}}(P_v)\times R \arrow[u] \arrow[ulll] \arrow[d, "\pi_{\mathbb{D}}"] \\
\mathbb{P}(\mathcal{G}) & \iota(L,P_v) \arrow[d, hook, "m"] & \iota(K,P_v) \arrow[l, hook'] \arrow[r, hook] \arrow[d, hook] & \iota(R,P_v) \arrow[d, hook] \\
 & \mathcal{G} \arrow[ul, "\mathbb{P}"] & \mathcal{D} \arrow[l, hook'] \arrow[r, hook] & \mathcal{H}
\end{tikzcd}
$$

There is an algorithm to perform topology-agnostic rewriting on generalized maps.

Set

Set

$$\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y.$$

$$\exists y(y \in x) \rightarrow \exists y(y \in x \wedge \forall z \neg(z \in x \wedge z \in y)).$$

$$\exists y \forall z(z \in y \leftrightarrow z \in x \wedge \phi(z)).$$

for any formula ϕ not containing y.

$$\exists y \forall x(x \notin y).$$

$$\exists y \forall x(x \in y \leftrightarrow x = z_1 \vee x = z_2).$$

$$\exists y \forall z(z \in y \leftrightarrow \forall u(u \in z \rightarrow u \in x)).$$

$$\exists y \forall z(z \in y \leftrightarrow \exists u(z \in u \wedge u \in x)).$$

$$\begin{aligned} \exists w(\emptyset \in w \wedge \quad & \forall x(x \in w \rightarrow \\ & \exists z(z \in w \wedge \\ & \forall u(u \in z \leftrightarrow u \in x \vee u = x)))). \end{aligned}$$

$$\begin{aligned} & \forall x, y, z(\psi(x, y) \wedge \psi(x, z) \rightarrow y = z) \rightarrow \\ & \exists u \forall w_1(w_1 \in u \leftrightarrow \exists w_2(w_2 \in a \wedge \psi(w_2, w_1))). \end{aligned}$$

$$\begin{aligned} & \forall x(x \in z \rightarrow x \neq \emptyset \wedge \\ & \forall y(y \in z \rightarrow x \cap y = \emptyset \vee x = y)) \\ & \rightarrow \\ & \exists u \forall x \exists v(x \in z \rightarrow u \cap x = \{v\}). \end{aligned}$$

Set

$$\forall z(z \in x \leftrightarrow z \in y) \to x = y.$$

$$\exists y(y \in x) \to \exists y(y \in x \land \forall z \neg(z \in x \land z \in y)).$$

$$\exists y \forall z(z \in y \leftrightarrow z \in x \land \phi(z)).$$

for any formula ϕ not containing y.

$$\exists y \forall x(x \notin y).$$

$$\exists y \forall x(x \in y \leftrightarrow x = z_1 \lor x = z_2).$$

$$\exists y \forall z(z \in y \leftrightarrow \forall u(u \in z \to u \in x)).$$

$$\exists y \forall z(z \in y \leftrightarrow \exists u(z \in u \land u \in x)).$$

$$\begin{aligned} \exists w(\emptyset \in w \land \quad & \forall x(x \in w \to \\ & \exists z(z \in w \land \\ & \forall u(u \in z \leftrightarrow u \in x \lor u = x)))). \end{aligned}$$

$$\begin{aligned} &\forall x, y, z(\psi(x, y) \land \psi(x, z) \to y = z) \to \\ &\exists u \forall w_1(w_1 \in u \leftrightarrow \exists w_2(w_2 \in a \land \psi(w_2, w_1))). \end{aligned}$$

$$\begin{aligned} &\forall x(x \in z \to x \neq \emptyset \land \\ &\forall y(y \in z \to x \cap y = \emptyset \lor x = y)) \\ &\to \\ &\exists u \forall x \exists v(x \in z \to u \cap x = \{v\}). \end{aligned}$$

Category theory tells us that to understand sets, we should **not look at the membership relation** $\in$ but at **functions** $\to$.

We write $f : A \to B$ for a function from a set A to a set B.

We can understand a set by the functions from and to it.

Set (composition)

Let $f : A \to B$ and $g : B \to C$ be two functions.

- Then we have a function $g \circ f : A \to C$ such that for all a in A,

$$(g \circ f)(a) = g(f(a))$$

- This composition of functions is associative, i.e.

$$(h \circ g) \circ f = h \circ (g \circ f)$$

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "g \circ f"'] & B \arrow[d, "g"] \arrow[dr, "h \circ g"] & \\
& C \arrow[r, "h"'] & D
\end{tikzcd}$$

Set (identities)

Let $f\colon A \to B$ be a function.

- Each set A has an identity function $\mathrm{id}_A\colon A \to A$ such that for all a in A,

$$\mathrm{id}_A(a) = a$$

- Identities are 'unit' for the composition in the sense of abstract algebra, i.e., the following diagram commutes

$$\begin{tikzcd}
A \arrow[r, "\mathrm{id}_A"] \arrow[dr, "f\circ\mathrm{id}_A"'] & A \arrow[d, "f"] \arrow[dr, "\mathrm{id}_B\circ f"] & \\
 & B \arrow[r, "\mathrm{id}_B"'] & B
\end{tikzcd}$$

Category (definition)

A category $\mathcal{C}$ consists of

- A collection of **objects** $\mathcal{O}(\mathcal{C})$,
- For each pair of objects a, b, a collection of **morphisms** (also called **maps** or **arrows**) $\mathrm{Hom}_{\mathcal{C}}(a, b)$,
- For each pair of morphisms f in $\mathrm{Hom}_{\mathcal{C}}(a, b)$ and g in $\mathrm{Hom}_{\mathcal{C}}(b, c)$, an morphism $g \circ_{\mathcal{C}} f$ called the **composite** or the **composition**.

Category (definition) – continued

Such that

Associativity law

- for any morphisms f in $\mathrm{Hom}_{\mathcal{C}}(a, b)$, g in $\mathrm{Hom}_{\mathcal{C}}(b, c)$ and h in $\mathrm{Hom}_{\mathcal{C}}(c, d)$

$$h \circ (g_{\mathcal{C}} \circ_{\mathcal{C}} f) = (h \circ_{\mathcal{C}} g) \circ_{\mathcal{C}} f$$

Unit laws

- for any object a, there exist an **identity** morphism id_a in $\mathrm{Hom}_{\mathcal{C}}(a, a)$ such that for all f in $\mathrm{Hom}_{\mathcal{C}}(a, b)$ and g in $\mathrm{Hom}_{\mathcal{C}}(b, a)$

$$f \circ_{\mathcal{C}} \mathrm{id}_a = f \text{ and } \mathrm{id}_a \circ_{\mathcal{C}} g = g$$

Alternative definition

An alternative definition (e.g., by A. Grothendieck) considers a collection of morphisms such that each morphism f as a **domain** (or **source**), written $\mathrm{dom}(f)$, and a **codomain** (or **target**), written $\mathrm{cod}(f)$.

Alternative definition

An alternative definition (e.g., by A. Grothendieck) considers a collection of morphisms such that each morphism f as a **domain** (or **source**), written $\mathrm{dom}(f)$, and a **codomain** (or **target**), written $\mathrm{cod}(f)$.

Then, the composition $g \circ_{\mathcal{C}} f$ is well-defined if and only if

$$\mathrm{cod}(f) = \mathrm{dom}(g)$$

This point of view adheres to the construction of categories from graphs.

Remarks

We write

- $a \in \mathcal{C}$ or $a \in \mathcal{O}(\mathcal{C})$ to say that a is an object of $\mathcal{C}$
- $a \xrightarrow{f} b$ or $f : a \to b$ to say that f is a morphism of $\mathrm{Hom}_{\mathcal{C}}(a, b)$
- $\mathcal{C}(a, b)$ instead of $\mathrm{Hom}_{\mathcal{C}}(a, b)$
- gf or $g \circ f$ instead of $g \circ_{\mathcal{C}} f$

The definition of categories ensures that for any sequence of morphisms

$$A_0 \xrightarrow{f_1} A_1 \xrightarrow{f_2} \cdots \xrightarrow{f_n} A_n$$

there is a unique morphism

$$A_0 \xrightarrow{f_n \cdots f_2 f_1} A_n$$

Collection?

Mathematicians sometimes use the word **collection** to denote a bunch of 'things' without prejudice as to whether those things form a **set**, a proper **class**, or some other formal notion of collection, such as a type.

The size issue

A category is

- **Locally small** if each collection of morphism is a set.
- **Small** if it is locally small and the collection of objects is a set.
- **Large** otherwise.

Remark: Most definitions will be given in the framework of small categories

Explicit examples of categories

One can define a category by explicitly providing its objects, morphisms, identities, and compositions.

The **empty category** $\varnothing$ is the category without object or morphism.

The **trivial category** $*$ (also written $\mathbf{1}$) contains a unique object $*$ and a unique morphism id_*, the identity on $*$.

Diagrammatic examples of category

One can 'draw' categories (omitting the identities)

The trivial category $*$ can be drawn as

$$\bullet$$

Assuming that composition is defined in the only possible way, we can easily create more categories

$$\begin{tikzcd} \bullet \arrow[r, shift left] \arrow[r, shift right] & \bullet \end{tikzcd} \qquad \begin{tikzcd} & \bullet \arrow[dr] & \\ \bullet \arrow[ur] \arrow[rr] & & \bullet \end{tikzcd}$$

Diagrammatic examples of category

One can 'draw' categories (omitting the identities)

The trivial category $*$ can be drawn as

$$*$$

Assuming that composition is defined in the only possible way, we can easily create more categories

$$\begin{tikzcd} a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \end{tikzcd} \qquad \begin{tikzcd} & b \arrow[dr, "g"] & \\ a \arrow[ur, "f"] \arrow[rr, "g \circ f"'] & & c \end{tikzcd}$$

First-order axiomatisation

Categories naturally admit a first-order axiomatization with a signature having

- two sorts **O** and **M**
- two unary function symbols dom and cod with profile $\mathbf{M} \to \mathbf{O}$
- a unary function symbol id
- a ternary predicate symbol of sort **M** formalizing the composition

Question: Why do we need a ternary predicate for composition (and not a function)?

First-order axiomatisation

Categories naturally admit a first-order axiomatization with a signature having

- two sorts **O** and **M**
- two unary function symbols dom and cod with profile $\mathbf{M} \to \mathbf{O}$
- a unary function symbol id
- a ternary predicate symbol of sort **M** formalizing the composition

Question: Why do we need a ternary predicate for composition (and not a function)? Because any pair of morphisms can not be composed.

Duality

The concept of a category is self-dual: the definition holds by formally reserving all arrows while keeping the objects.

The **dual** or **opposite** category of $\mathcal{C}$, written $\mathcal{C}^{op}$ is defined by

- $\mathcal{O}(\mathcal{C}^{op}) = \mathcal{O}(\mathcal{C})$
- $\mathrm{Hom}_{\mathcal{C}}^{op}(a, b) = \mathrm{Hom}_{\mathcal{C}}(b, a)$
- any $f \in \mathrm{Hom}_{\mathcal{C}}^{op}(a, b)$, $g \in \mathrm{Hom}_{\mathcal{C}}^{op}(b, c)$

$$g \circ_{\mathcal{C}^{op}} f = f \circ_{\mathcal{C}} g$$

In particular, $(\mathcal{C}^{op})^{op} = \mathcal{C}$ for any category $\mathcal{C}$.

Duality principle

Every statement expressed within category theory admits a **dual statement** obtained by formally reversing the arrows.

Duality principle

Every statement expressed within category theory admits a **dual statement** obtained by formally reversing the arrows.

- A statement holds in a category $\mathcal{C}$ if and only if the dual statement holds in the dual category.

Duality principle

Every statement expressed within category theory admits a **dual statement** obtained by formally reversing the arrows.

- A statement holds in a category $\mathcal{C}$ if and only if the dual statement holds in the dual category.
- A statement holds in all categories if and only if its dual also does.

Duality principle

Every statement expressed within category theory admits a **dual statement** obtained by formally reversing the arrows.

- A statement holds in a category $\mathcal{C}$ if and only if the dual statement holds in the dual category.
- A statement holds in all categories if and only if its dual also does.
- Statements lifted to categories, i.e., proven using only the language of category theory, directly yield a dual statement according to the duality principle.

Duality principle

Every statement expressed within category theory admits a **dual statement** obtained by formally reversing the arrows.

- A statement holds in a category $\mathcal{C}$ if and only if the dual statement holds in the dual category.
- A statement holds in all categories if and only if its dual also does.
- Statements lifted to categories, i.e., proven using only the language of category theory, directly yield a dual statement according to the duality principle.

In a given category, dual statements/constructions may result in very different statements/constructions.

Properties of morphisms

A morphism $f\colon a \to b$ is

- a **monomorphism** (mono, monic) if for all morphisms $g, g'\colon x \to a$, $f \circ g = f \circ g'$ implies $g = g'$,
- an **epimorphism** (epi, epic) if for all morphisms $g, g'\colon a \to x$, $g \circ f = g' \circ f$ implies $g = g'$,
- an **isomorphism** (iso) if there exists $g\colon b \to a$ such that $f \circ g = \mathrm{id}_b$ and $g \circ f = \mathrm{id}_a$. Then g is called the **inverse** of f

Questions / Exercices

Summary

- f **mono**: $f \circ g = f \circ g' \implies g = g'$.
- f **epi**: $g \circ f = g' \circ f \implies g = g'$.
- f **iso**: there exists g s.t. $f \circ g = \mathrm{id}_b$ and $g \circ f = \mathrm{id}_a$.

1. What is the relation between monomorphisms and epimorphisms?

2. What is the relation between monomorphisms, epimorphisms, and isomorphisms?

3. What are the monomorphisms, epimorphisms, and isomorphisms in **Set**?

4. Show that the inverse of a morphism is unique (if it exists).

Questions / Exercices

Summary

- f **mono**: $f \circ g = f \circ g' \implies g = g'$.
- f **epi**: $g \circ f = g' \circ f \implies g = g'$.
- f **iso**: there exists g s.t. $f \circ g = \mathrm{id}_b$ and $g \circ f = \mathrm{id}_a$.

1. What is the relation between monomorphisms and epimorphisms?
   Duality.
2. What is the relation between monomorphisms, epimorphisms, and isomorphisms?
   Iso implies both mono and epi, but the converse does not hold.
3. What are the monomorphisms, epimorphisms, and isomorphisms in **Set**?
   Injections, surjections, and bijections.
4. Show that the inverse of a morphism is unique (if it exists).

Categories of mathematical objects

- **Set** is the category of sets and functions
- **Fin** is the category of finite sets and functions
- **Pos** is the category of partially ordered sets and isotone maps
- **Top** is the category of topological spaces and continuous maps
- **Mon** is the category of monoids and monoid homomorphisms
- **Grp** is the category of groups and group homomorphisms
- **Rng** is the category of rings and ring homomorphisms
- **Alg**(Σ) is the category of Σ-algebras and algebra homomorphisms
- **Graph** is the category of graphs and graphs morphisms

Categories of mathematical objects

- **Set** is the category of sets and functions
- **Fin** is the category of finite sets and functions
- **Pos** is the category of partially ordered sets and isotone maps
- **Top** is the category of topological spaces and continuous maps
- **Mon** is the category of monoids and monoid homomorphisms
- **Grp** is the category of groups and group homomorphisms
- **Rng** is the category of rings and ring homomorphisms
- **Alg**(Σ) is the category of Σ-algebras and algebra homomorphisms
- **Graph** is the category of graphs and graphs morphisms

Given a first-order theory $\mathbb{T}$, the category $\mathbb{T}\text{-}\mathrm{mod}(\mathbf{Set})$ has set-based models of $\mathbb{T}$ as objects and structure-preserving maps between them.

Rel

Let S, T and U be three sets, let $r \subseteq S \times T$ and $r' \subseteq T \times U$ be two **relations**.

The **composite** $r' \circ r$ is defined as

$$r' \circ r := \{(s, u) \in S \times U \mid \exists t \in T,\ (s, t) \in r \wedge (t, u) \in r'\}$$

This notion of composition defines the category **Rel** of sets and (binary) relations.

What are the **identities**?

Rel

Let S, T and U be three sets, let $r \subseteq S \times T$ and $r' \subseteq T \times U$ be two **relations**.

The **composite** $r' \circ r$ is defined as

$$r' \circ r := \{(s,u) \in S \times U \mid \exists t \in T,\ (s,t) \in r \wedge (t,u) \in r'\}$$

This notion of composition defines the category **Rel** of sets and (binary) relations.

What are the **identities**?
The diagonal relations $\Delta_S = \{(s,s) \in S \times S \mid s \in S\}$.

Mathematical objects as categories

- A **set** is a **discrete category**, i.e., a category with only identities
- A **preorder** is a category with at most one arrow between any pair of objects
- A **monoid** is a category with a unique object
- A **group** is a category with a unique object where all arrows are isomorphisms

Functors

Functors

Looking for a relational understanding of mathematics, one should ask what the ‘meaningful’ relations between categories are.

Functors

Looking for a relational understanding of mathematics, one should ask what the 'meaningful' relations between categories are.

Functors are structure-preserving maps between categories.

Functor (definition)

A **functor** $F\colon \mathcal{C} \to \mathcal{D}$ from $\mathcal{C}$ to $\mathcal{D}$ consists of

- A function $F\colon\ \mathcal{O}(\mathcal{C}) \to \mathcal{O}(\mathcal{D})$,
- Functions $F\colon\ \mathrm{Hom}_{\mathcal{C}}(a, b) \to \mathrm{Hom}_{\mathcal{D}}(F(a), F(b))$ for all $a, b \in \mathcal{O}(\mathcal{C})$

such that

Preservation of identities

- $F(\mathrm{id}_a) = \mathrm{id}_{F(a)}$ for all objects $a \in \mathcal{O}(\mathcal{C})$

Preservation of compositions

- $F(g \circ f) = (F(g) \circ F(f))$ for all $f\colon a \to b$ and $g\colon b \to c$

Exercice: Show that functors preserve isomorphisms.

Examples

- The **identity** functor $\mathrm{id}_{\mathcal{C}}$ on a category $\mathcal{C}$ maps the objects and morphisms to themselves.
- The functors between monoids regarded as one-object categories are the monoid morphisms.
- If $\mathcal{C}$ is a discrete category, then $F\colon \mathcal{C} \to \mathcal{D}$ is just a family $(F(c))_{c \in \mathcal{C}}$.
- The functors between preorders (or posets) regarded as categories are the order-preserving maps.

Group action

A **group** G is a non-empty set with an associative binary operation with a neutral element e such that each element is invertible.

A (left) **group action** of G on a set S is a function $\alpha \colon G \times S \to S$ s.t. for all $g, h \in G$ and all $s \in S$,

- $\alpha(e, s) = s$
- $\alpha(g, \alpha(h, s)) = \alpha(g \cdot h, s)$

Exercice: Show that a functor $F \colon \mathcal{G} \to \mathbf{Set}$ where $\mathcal{G}$ is the group G viewed as a one-object category is a (left) **group action**.

Diagram

A **diagram** in a category $\mathcal{C}$ consists of some objects of $\mathcal{C}$ connected by some morphisms of $\mathcal{C}$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\ D \arrow[r, "k"'] & C \end{tikzcd}$$

Diagram

A **diagram** in a category $\mathcal{C}$ consists of some objects of $\mathcal{C}$ connected by some morphisms of $\mathcal{C}$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\ D \arrow[r, "k"'] & C \end{tikzcd}$$

Formally, a **diagram** D of shape $\mathcal{J}$ in $\mathcal{C}$ is a functor $D \colon \mathcal{J} \to \mathcal{C}$.

$$\left(\begin{tikzcd} \bullet \arrow[r] \arrow[d] & \bullet \arrow[d] \\ \bullet \arrow[r] & \bullet \end{tikzcd}\right) \to \mathcal{C}$$

Properties and examples of diagrams

Given a diagram $D\colon \mathcal{J} \to \mathcal{C}$, $\mathcal{J}$ is called the **index** category or the **scheme** of the diagram.

A diagram is said to be **small** or **finite** when the index is.

A diagram is said to **to commute** when the index is a finite poset.

Examples:

- For any object c in $\mathcal{C}$ and any index category $\mathcal{J}$, there is a constant diagram $\delta(c)$ that maps all objects to c and all morphisms to id_c.
- A diagram on $\bullet \longleftarrow \bullet \longrightarrow \bullet$ is called a **span**.
- A diagram on $\bullet \longrightarrow \bullet \longleftarrow \bullet$ is called a **cospan**.

Powerset functors

Given a set S, the **powerset** of S is the set $\mathscr{P}(S)$ of the subsets of S.

The **covariant powerset functor** $\mathscr{P}_* \colon \mathbf{Set} \to \mathbf{Set}$ sends a function $f \colon S \to T$ to its image function $f_* \colon \mathscr{P}(S) \to \mathscr{P}(T); X \mapsto f(X)$.

The **contravariant powerset functor** $\mathscr{P}^* \colon \mathbf{Set}^{op} \to \mathbf{Set}$ sends a function $f \colon S \to T$ to its preimage function $f^* \colon \mathscr{P}(T) \to \mathscr{P}(S); X \mapsto \{s \in S \mid f(s) \in X\}$.

Forgetful functors

- $U\colon \mathbf{Mon} \to \mathbf{Set}$ forgets the monoid structure and maps monoids to the underlying set and monoid homomorphisms to the associated function.
- Similarly from **Pos**, **Top**, or **Grp** to **Set**.
- $\mathbf{Rng} \to \mathbf{Mon}$ forgets the additive structure and only remembers the underlying multiplicative monoid

Free monoid and the list monad

If S is a set, the free monoid on S is the set S^* of all finite sequences of elements of S.

There is a **free** functor $F\colon \mathbf{Set} \to \mathbf{Mon}$ that maps S to S^* (the binary operation is the concatenation).

'Freedom' is to be understood as not adding anything but what is required by the axioms of the structure.

Applying the forgetful $U\colon \mathbf{Mon} \to \mathbf{Set}$, we obtain a functor $U \circ F\colon \mathbf{Set} \to \mathbf{Set}$ that provides the lists (or strings) of elements of a set.

Presheafs

Given a base category $\mathcal{C}$, a **presheaf** on $\mathcal{C}$ is a functor $\mathcal{C}^{op} \to \mathbf{Set}$.

- Constant presheaf on a set S: assigns any object of $\mathcal{C}$ to S and any morphism to id_C.
- The hom-functors of locally small categories $\mathcal{C}$ are $h_x := \mathrm{Hom}_{\mathcal{C}}(\text{-}, x)$ sends an object a to $h_x := \mathrm{Hom}_{\mathcal{C}}(a, x)$ and a morphism $f: a \to b$ to the function that maps $b \to x$ to the composite $a \xrightarrow{f} b \to x$.

Exercice: A graph is a presheaf $\mathcal{C} \to \mathbf{Set}$. What is the base category $\mathcal{C}$?

Presheafs

Given a base category $\mathcal{C}$, a **presheaf** on $\mathcal{C}$ is a functor $\mathcal{C}^{op} \to \mathbf{Set}$.

- Constant presheaf on a set S: assigns any object of $\mathcal{C}$ to S and any morphism to id_C.
- The hom-functors of locally small categories $\mathcal{C}$ are $h_x := \mathrm{Hom}_{\mathcal{C}}(\text{-}, x)$ sends an object a to $h_x := \mathrm{Hom}_{\mathcal{C}}(a, x)$ and a morphism $f : a \to b$ to the function that maps $b \to x$ to the composite $a \xrightarrow{f} b \to x$.

Exercice: A graph is a presheaf $\mathcal{C} \to \mathbf{Set}$. What is the base category $\mathcal{C}$?

$$V \underset{t}{\overset{s}{\rightrightarrows}} E$$

Properties of functors

A functor $F\colon \mathcal{C} \to \mathcal{D}$ is **faithful** if for all objects a, b in $\mathcal{A}$,

$$\begin{array}{ccc} \mathrm{Hom}_{\mathcal{C}}(a,b) & \to & \mathrm{Hom}_{\mathcal{C}}(F(a),F(b)) \\ f & \mapsto & F(f) \end{array}$$

is injective.

The functor is **full** if the mapping is surjective for all objects.

The functor is **(essentially) surjective** if all objects in $\mathcal{D}$ are (isomorphic to some) of the form $F(c)$ for some object c in $\mathcal{C}$.

A **subcategory** $\mathcal{D}$ of a category $\mathcal{C}$ is a category s.t. $\mathcal{O}(\mathcal{D}) \subseteq \mathcal{O}(\mathcal{D})$ and $\mathrm{Hom}_{\mathcal{D}}(a,b) \subseteq \mathrm{Hom}_{\mathcal{C}}$ for any objects a, b with composition and identity induced from $\mathcal{C}$. It is a **full subcategory** if the inclusion functor $i\colon \mathcal{D} \to \mathcal{C}$ is full.

Cat

Given two functors $F\colon \mathcal{C} \to \mathcal{D}$ and $G\colon \mathcal{D} \to \mathcal{E}$ between small categories $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{E}$ ...

Cat

Given two functors $F\colon \mathcal{C} \to \mathcal{D}$ and $G\colon \mathcal{D} \to \mathcal{E}$ between small categories $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{E}$ ...

... the composition $G \circ F$ is defined component-wise on objects and morphisms,

Cat

Given two functors $F\colon \mathcal{C} \to \mathcal{D}$ and $G\colon \mathcal{D} \to \mathcal{E}$ between small categories $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{E}$ ...

... the composition $G \circ F$ is defined component-wise on objects and morphisms,

... it is associative.

Cat

Given two functors $F\colon \mathcal{C} \to \mathcal{D}$ and $G\colon \mathcal{D} \to \mathcal{E}$ between small categories $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{E}$ ...

... the composition $G \circ F$ is defined component-wise on objects and morphisms,

... it is associative.

The identity functors are units for the composition.

Cat

Given two functors $F\colon \mathcal{C} \to \mathcal{D}$ and $G\colon \mathcal{D} \to \mathcal{E}$ between small categories $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{E}$ ...

...the composition $G \circ F$ is defined component-wise on objects and morphisms,

...it is associative.

The identity functors are units for the composition.

(Small) Categories and functors from a (large) category **Cat**.

Natural transformations

Natural transformations

Looking for a relational understanding of mathematics, one should ask what the ‘meaningful’ relations between functors are.

Given two functors $F, G \colon \mathcal{C} \to \mathcal{D}$ between two categories $\mathcal{C}$ and $\mathcal{D}$, how do we compare the values that the functors take at the objects of the source category?

Natural transformations (definition)

Let $\mathcal{C}$ and $\mathcal{D}$ be two categories, and $F, G\colon \mathcal{C} \to \mathcal{D}$ be two functors.

A **natural transformation** $\alpha\colon F \to G$ is a function assigning to each object $a \in \mathcal{O}(\mathcal{C})$, an arrow $\alpha_a\colon F(a) \to G(a)$ in $\mathcal{D}$ such that for all morphisms $f\colon a \to b$ in $\mathcal{C}$, the following diagram commutes

$$\begin{tikzcd} F(a) \arrow[r, "\alpha_a"] \arrow[d, "F(f)"'] & G(a) \arrow[d, "G(f)"] \\ F(b) \arrow[r, "\alpha_b"'] & G(b) \end{tikzcd}$$

We often write $\alpha(a)$ for α_a, and $\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{D} \end{tikzcd}$ to mean that α is a natural transformation from F to G.

Examples

- If $\mathcal{C}$ is a discrete category and F, G are functors $: \mathcal{C} \to \mathcal{D}$, then a natural transformation is a family of maps $(F(c) \xrightarrow{\alpha_c} G(c))_{c \in \mathcal{C}}$ in $\mathcal{D}$.
- The morphisms of presheaves are natural transformations.

Functor category

Let $\mathcal{C}$ and $\mathcal{D}$ be two categories. The functor category $[\mathcal{C}, \mathcal{D}]$ (also written $[\mathcal{C}; \mathcal{D}]$ or $\mathcal{D}^{\mathcal{C}}$) is the category with

- Functors $\mathcal{C} \to \mathcal{D}$ as objects
- Natural transformations as arrows

Universal properties

Universal properties

How do I define the empty set, singleton sets, or the Cartesian products of sets if I cannot talk about the elements of the set?

Universal properties

How do I define the empty set, singleton sets, or the Cartesian products of sets if I cannot talk about the elements of the set?

We have to rediscover everything in terms of morphisms and compositions.

Singleton sets

Any singleton set 1 with one element satisfies the following property

Singleton sets

Any singleton set 1 with one element satisfies the following property

For all sets S, there is a unique function $S \to 1$.

Singleton sets

Any singleton set 1 with one element satisfies the following property

For all sets S, there is a unique function $S \to 1$.

One has to show that

- there is **at least** one such function
- there is **at most** one such function

Singleton sets

Any singleton set 1 with one element satisfies the following property

For all sets S, there is a unique function $S \to 1$.

One has to show that

- there is **at least** one such function
- there is **at most** one such function

Then 1 is **unique** up to (unique) isomorphisms.

Singleton sets

Any singleton set 1 with one element satisfies the following property

> For all sets S, there is a unique function $S \to 1$.

One has to show that

- there is **at least** one such function
- there is **at most** one such function

Then 1 is **unique** up to (unique) isomorphisms.

A **universal property** is a definition of the form
"There is a unique X s.t. Y"

Initial and terminal objects

An object $1_{\mathcal{C}}$ in $\mathcal{C}$ is **terminal** if for all objects c in $\mathcal{C}$, there exists a unique morphism $c \to 1_{\mathcal{C}}$.

By duality, an object $\varnothing_{\mathcal{C}}$ (or $0_{\mathcal{C}}$) in $\mathcal{C}$ is **initial** if for all objects c in $\mathcal{C}$, there exists a unique morphism $\varnothing_{\mathcal{C}} \to c$.

If they exist, initial and terminal objects are unique up to unique isomorphism.

Questions / Exercices

1. What is the initial object in **Set**?

2. What is the initial object in **Rng**?

3. What are the initial and terminal objects in preorder (or posets) viewed as categories?

Questions / Exercices

1. What is the initial object in **Set**?
The empty set.

2. What is the initial object in **Rng**?
$\mathbb{Z}$.

3. What are the initial and terminal objects in preorder (or posets) viewed as categories?
The least and greatest elements.

Products

Products

The **product** of a and b in $\mathcal{C}$ is a span

$$A \xleftarrow{p_A} P \xrightarrow{p_B} B$$

which is universal with this property, i.e., for any other span $A \xleftarrow{f_A} X \xrightarrow{f_B} B$, there exists a unique morphism $f : X \to P$ s.t. the following diagram commutes

$$\begin{tikzcd} & X \arrow[dl, "f_A"'] \arrow[d, dotted, "\exists! f"] \arrow[dr, "f_B"] & \\ A & P \arrow[l, "p_A"] \arrow[r, "p_B"'] & B \end{tikzcd}$$

We obtain **coproducts** by duality.

Examples

The product of two sets X and Y in **Set** is the Cartesian product $X \times Y$ with the standard projections.

Exercice:

1. What is the coproduct of two sets?

2. What are the products and coproducts in preorders (or posets) viewed as categories?

3. Give an example of a category without products.

Examples

The product of two sets X and Y in **Set** is the Cartesian product $X \times Y$ with the standard projections.

Exercice:

1. What is the coproduct of two sets?

The disjoint union.

2. What are the products and coproducts in preorders (or posets) viewed as categories?

The meet (supremum) and join (infimum) operations.

3. Give an example of a category without products.

Any discrete category with at least two objects.

Pullbacks

The **pullback** of a cospan $A \xrightarrow{g} C \xleftarrow{h} B$ is a span $A \xleftarrow{p_A} P \xrightarrow{p_B} B$ s.t. the following diagram commutes

$$\begin{tikzcd} P \arrow[r, "p_A"] \arrow[d, "p_B"'] & A \arrow[d, "g"] \\ B \arrow[r, "h"'] & C \end{tikzcd}$$

and is universal with this property, i.e., for any span $A \xleftarrow{f_A} X \xrightarrow{f_B} B$ forming a commutative square with $A \xrightarrow{g} C \xleftarrow{h} B$, there is a unique morphism $f\colon X \to P$ s.t. the following diagram commutes

$$\begin{tikzcd} X \arrow[drr, bend left, "f_A"] \arrow[ddr, bend right, "f_B"'] \arrow[dr, dotted, "\exists! f"] & & \\ & P \arrow[r, "p_A"] \arrow[d, "p_B"'] & A \arrow[d, "g"] \\ & B \arrow[r, "h"'] & C \end{tikzcd}$$

Pushouts in Set

The pushout of $A \xleftarrow{g} C \xrightarrow{h} B$ in **Set** is

$$P = (A \sqcup B)/{\sim}$$

- $A \sqcup B$ is the disjoint union of A and B, i.e., $(\{0\} \times A) \cup (\{1\} \times B)$
- $\sim$ is the equivalence relation s.t.

$$g(c) \sim h(c) \text{ for all } c \in C$$

The **coprojections** $p_A \colon A \to P$ and $p_B \colon B \to P$ map $a \in A$ and $b \in B$ to their equivalence classes in P.

Cone

Given a small diagram $D\colon \mathcal{J} \to \mathcal{C}$, a **cone** on D is an object c of $\mathcal{C}$, the **vertex** (or **apex**) of the cone together with a familly of maps $(f_j : c \to D(j))_{j\in\mathcal{J}}$ s.t. for all morphisms $u\colon i \to j$, the following triangle commutes

$$\begin{tikzcd} & c \arrow[dl, "f_i"'] \arrow[dr, "f_j"] & \\ D(j) \arrow[rr, "D(u)"'] & & D(j) \end{tikzcd}$$

Viewing c as the constant diagram $\Delta(c)$, a cone on D is a natural transformation $\alpha\colon \Delta(c) \to D$.

Limit

A **limit** of D is cone $(p_j : l \to D(j))_{j \in \mathcal{J}}$ which is universal with that property, i.e., for any other cone on D, there exists a unique morphism $f : c \to l$ s.t. $p_j \circ f = f_j$ for all j in $\mathcal{J}$.

We obtain **cocones** and **colimits** by duality.

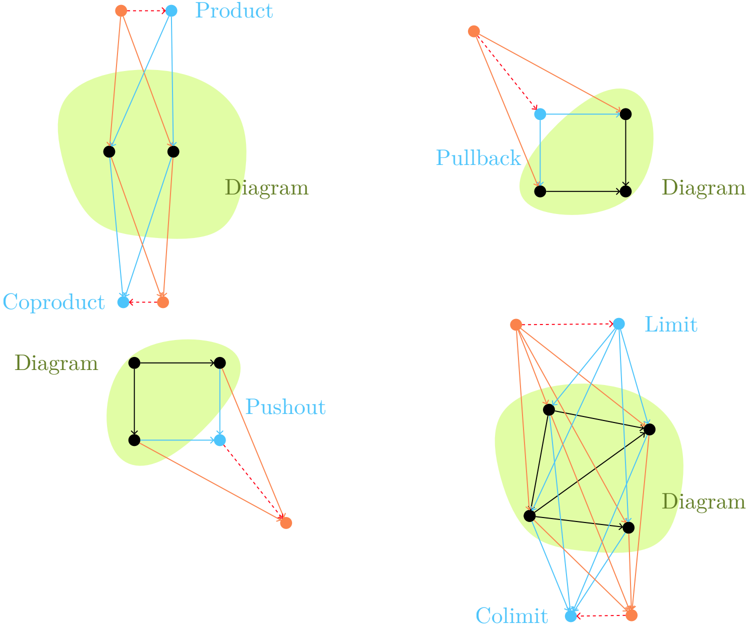
Example

The limit of a diagram $D\colon \mathcal{J} \to \mathcal{C}$ is (if it exists)

- a terminal object if $\mathcal{J}$ is the empty category $\varnothing$,
- a product if $\mathcal{J}$ is a discrete category,
- a pullback if $\mathcal{J}$ is a cospan,
- an equalizer if if $\mathcal{J}$ is $\bullet \rightrightarrows \bullet$

Exercice: Show that $f\colon a \to b$ is monic iff the following diagram is a pullback.

$$\begin{tikzcd} a \arrow[r, "\mathrm{id}_a"] \arrow[d, "\mathrm{id}_a"'] & a \arrow[d, "f"] \\ a \arrow[r, "f"'] & b \end{tikzcd}$$


Product
Diagram
Coproduct
Pullback
Diagram
Diagram
Pushout
Limit
Diagram
Colimit

Summary

Elementary algebra is the abstraction of **numbers**
Group theory is the abstraction of **symmetry**
Ring theory is the abstraction of **arithmetics**

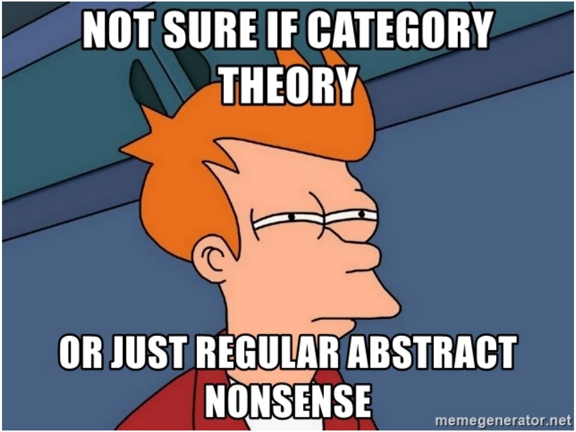
Summary

Karlsruhe Institute of Technology

Elementary algebra is the abstraction of **numbers**
Group theory is the abstraction of **symmetry**
Ring theory is the abstraction of **arithmetics**

Category theory is the abstraction of **composition**

Look at ‘things’ in context, i.e., w.r.t. similar ‘things’


NOT SURE IF CATEGORY THEORY
OR JUST REGULAR ABSTRACT NONSENSE
memegenerator.net