Category Theory 101

Mathematical Structures in Computer Science

Oisín Flynn-Connolly

September 26, 2025

Motivation for computer scientists

1. **Convenient language:** (Higher) category theory is a very general organizational principle for mathematics. Sometimes one notices that the same construction appears in multiple contexts. Category theory allows you to formalize this via **universal constructions.** Most of modern mathematics[1] is written in the language of category theory.
2. **Compositionality.** Sometimes one wants to break a program down into smaller parts that can each be analysed independently.
3. **Logic.** There is a correspondence with logic via the Curry–Howard correspondence. Take Henning's course during master for more!

[1] or at least algebra and topology since the 1960s

Motivation for computer scientists

1. **Convenient language:** (Higher) category theory is a very general organizational principle for mathematics. Sometimes one notices that the same construction appears in multiple contexts. Category theory allows you to formalize this via **universal constructions.** Most of modern mathematics[1] is written in the language of category theory.
2. **Compositionality.** Sometimes one wants to break a program down into smaller parts that can each be analysed independently.
3. **Logic.** There is a correspondence with logic via the Curry–Howard correspondence. Take Henning's course during master for more!

[1] or at least algebra and topology since the 1960s

Motivation for computer scientists

1. **Convenient language:** (Higher) category theory is a very general organizational principle for mathematics. Sometimes one notices that the same construction appears in multiple contexts. Category theory allows you to formalize this via **universal constructions.** Most of modern mathematics[1] is written in the language of category theory.
2. **Compositionality.** Sometimes one wants to break a program down into smaller parts that can each be analysed independently.
3. **Logic.** There is a correspondence with logic via the Curry–Howard correspondence. Take Henning's course during master for more!
4. **Concrete applications**
   1. **Functional programming**: Category theory underlies the theory and syntax of functional programming languages such as OCaml and Haskell.
   2. **Linguistics:** Category theory has been used to study human language.
   3. **Processes in CS:** To analyse all manner of processes in CS; from optimization algorithms to attack trees.

[1] or at least algebra and topology since the 1960s

Direct sum of vector spaces

Definition

Given two vector spaces V, W the direct sum is defined as:

$$V + W = \{(v, w) \mid v \in V, w \in W\}$$

To define the vector space structure

$$(v_1, w_1) + (v_2, w_2) = (v_1 + v_2, w_1 + v_2) \qquad v_1, v_2 \in V, w_1, w_2 \in W$$

$$\lambda \cdot (v, w) = (\lambda v, \lambda w)$$

There are **canonical inclusions**

$$i_V : V \to V + W, \qquad v \mapsto (v, 0)$$

$$i_V : W \to V + W \qquad w \mapsto (0, w)$$

Disadvantages of this definition

- Rather ad hoc and messy. Need to define the vector space structure by hand.
- Hides the role of the canonical inclusions.
- Doesn't generalise well.

Universal constructions: Direct sum of vector spaces

You saw the direct sum of vector spaces two weeks ago. But there is a different way to construct it:

Definition

Given two vector spaces V, W the direct sum is defined to be the **unique** vector space $V + W$ equipped with maps $i_V : V \to V + W$ and $i_W : W \to V + W$ such that for any vector space U, with maps $j_V : V \to U$ and $j_W : W \to U$ there is a unique map

$$V + W \to U$$

making the following diagram commute:

$$\begin{tikzcd} & U & \\ V \arrow[ur, "j_V"] \arrow[r, "i_V"'] & V + W \arrow[u, dotted] & W \arrow[ul, "j_W"'] \arrow[l, "i_W"] \end{tikzcd}$$

Universal constructions: Coproducts

Definition

Given two vector spaces V, W the direct sum is defined to be the **unique** vector space $V + W$ equipped with maps $i_V : V \to V + W$ and $i_W : W \to V + W$ such that for any vector space U, with maps $j_V : V \to U$ and $j_W : W \to U$ there is a unique map

$$V + W \to U$$

making the following diagram commute:

$$\begin{tikzcd}
& U & \\
V \arrow[ur, "j_V"] \arrow[r, "i_V"'] & V + W \arrow[u, dotted] & W \arrow[ul, "j_W"'] \arrow[l, "i_W"]
\end{tikzcd}$$

This is an example of a very general construction: the **categorical coproduct.**

Universal constructions: Free vector space construction

Definition

Given a finite set S, we can build a vector space from it.

$$\mathbb{R}[S] = \{f : S \to \mathbb{R}\}\,.$$

- The addition is $(f+g)(s) = f(s) + g(s)$ for $s \in S$ and $f, g : S \to \mathbb{R}$.
- The scalar multiplication is given by $c \cdot f = c \cdot f(s)$ for $s \in S$ and $f : S \to \mathbb{R}$.
- This is isomorphic to $\mathbb{R}^{\#S}$ where $\#S$ is the number of elements in S. For example, $\mathbb{R}\{e_1, e_2\}$ is isomorphic to $\mathbb{R}^2$.
- There is a canonical map

$$i : S \to \mathbb{R}[S]$$

$$i(s)(t) = \begin{cases} 1 & \text{if } s = t \\ 0 & \text{if } s \neq t. \end{cases}$$

Universal property of free vector space construction

Lemma

Let V be a vector space and $f : S \to V$ be any function. Then, there is a unique linear map

$$\overline{f} : \mathbb{R}[S] \to V.$$

making the diagram below commute.

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "i"] & V \\ \mathbb{R}[S] \arrow[ur, dotted, "\overline{f}"] & \end{tikzcd}$$

This is called a **unique extension property.**

Data of a category

Definition

A **category** $\mathcal{C}$ consists of

1. a collection of **objects** $|\mathcal{C}|$
2. for any two objects $A, B \in |\mathcal{C}|$ a collection of **morphisms** $\mathcal{C}(A, B)$
3. For any three objects $A, B, C \in |\mathcal{C}|$, a binary operation called **composition**

$$\circ_{A,B,C} : \mathcal{C}(A, B) \times \mathcal{C}(B, C) \to \mathcal{C}(A, C)$$

$$(f, g) \mapsto g \circ f$$

Data of a category

Definition

A **category** $\mathcal{C}$ consists of

1. a collection of **objects** $|\mathcal{C}|$
2. for any two objects $A, B \in |\mathcal{C}|$ a collection of **morphisms** $\mathcal{C}(A, B)$
3. For any three objects $A, B, C \in |\mathcal{C}|$, a binary operation called **composition**

$$\circ_{A,B,C} : \mathcal{C}(A, B) \times \mathcal{C}(B, C) \to \mathcal{C}(A, C)$$

$$(f, g) \mapsto g \circ f$$

Shorthand: We often write $A \in \mathcal{C}$ instead of $A \in |\mathcal{C}|$ and $f : A \to B$ instead of $f \in \mathcal{C}(A, B)$

Data of a category

Definition

A **category** $\mathcal{C}$ consists of

1. a collection of **objects** $|\mathcal{C}|$
2. for any two objects $A, B \in |\mathcal{C}|$ a collection of **morphisms** $\mathcal{C}(A, B)$
3. For any three objects $A, B, C \in |\mathcal{C}|$, a binary operation called **composition**

$$\circ_{A,B,C} : \mathcal{C}(A, B) \times \mathcal{C}(B, C) \to \mathcal{C}(A, C)$$

$$(f, g) \mapsto g \circ f$$

Shorthand: We often write $A \in \mathcal{C}$ instead of $A \in |\mathcal{C}|$ and $f : A \to B$ instead of $f \in \mathcal{C}(A, B)$
We call A the **domain** of f and B the **codomain**.

Axioms of composition

Composition is supposed to model function composition. This boils down to the following two axioms.

- The operation $\circ$ is associative. In other words, if $h : A \to B$, $g : B \to C$ and $f : C \to D$ for $A, B, C, D \in \mathcal{C}$, we have that:

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

  as a function from $A \to D$.
- For $A \in \mathcal{C}$, the collection $\mathcal{C}(A, A)$ contains an identity element $\mathrm{id}_A : A \to A$, such that for any $f : A \to B$, we have

$$\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A$$

Category of sets

Definition (Category of sets)

The category of sets **Set** is defined by

1. The collection of objects are all the sets.
2. For any two sets A, B, define

$$\mathbf{Set}(A, B) = \{f : A \to B \mid f \text{ is a function}\}$$

3. Composition is defined by function composition. Ie. $f \circ g(x) = f(g(x))$ for $x \in A$ and $g : A \to B$ and $f : B \to C$.
4. The identity map $\mathrm{id}_A : A, \to A$ is the identity map on A ie. $f(x) = x$ for $x \in A$.

Another perspective: Think of each morphism as a program.

Category of pointed sets

Definition (Category of pointed sets)

The category of pointed sets $\mathbf{Set}_\bullet$ is defined by

1. The objects are all non-empty sets (A, a_0) with one distinguished element $a_0 \in A$.
2. For any two pointed sets $(A, a_0), (B, b_0)$, define

$$\mathbf{Set}_\bullet((A, a_0), (B, b_0)) = \{f : A \to B \mid f \text{ is a function such that } f(a_0) = b_0\}$$

3. Composition is defined by function composition. This preserves the base point.
4. The identity map $\mathrm{id}_{(A,a_0)} : (A, a_0) \to (A, a_0)$ is the identity map on A. This preserves the base point.

This category appears naturally when one reasons about programs.

Other Examples

Most mathematical structures that you have seen in previous lectures organize into categories.

Examples

- Sets and functions. We call this category **Set**.
- Vector spaces and

Other Examples

Most mathematical structures that you have seen in previous lectures organize into categories.

Examples

- Sets and functions. We call this category **Set**.
- Vector spaces and linear maps. We call this category **Vect**.
- Banach spaces and

Other Examples

Most mathematical structures that you have seen in previous lectures organize into categories.

Examples

- Sets and functions. We call this category **Set**.
- Vector spaces and linear maps. We call this category **Vect**.
- Banach spaces and bounded linear maps. We call this category **Ban**.
- Metric spaces and

Other Examples

Most mathematical structures that you have seen in previous lectures organize into categories.

Examples

- Sets and functions. We call this category **Set**.
- Vector spaces and linear maps. We call this category **Vect**.
- Banach spaces and bounded linear maps. We call this category **Ban**.
- Metric spaces and continuous maps (or short maps). We call this category **Met**.
- Topological spaces and continuous maps. We call this category **Top**. You will encounter it next week.
- Let $(P, \preceq)$ be a poset, for example $(\mathbb{R}, \leq)$. Then P is a category, where the objects are elements of the poset and

$$P(x, y) = \begin{cases} \{f\} & \text{if } x \preceq y \\ \emptyset & \text{otherwise.} \end{cases}$$

Universal constructions in a categories

Definition

Let $\mathcal{C}$ be a category. Given $V, W \in \mathcal{C}$ the coproduct is defined to be the **unique** (if it exists) $V + W$ equipped with maps $i_V : V \to V + W$ and $i_W : W \to V + W$ such that for $U \in \mathcal{C}$, with maps $j_V : V \to U$ and $j_W : W \to U$ there is a unique map

$$V + W \to U$$

making the following diagram commute:

$$\begin{tikzcd} & U & \\ V \arrow[ur, "j_V"] \arrow[r, "i_V"'] & V + W \arrow[u, dotted] & W \arrow[ul, "j_W"'] \arrow[l, "i_W"] \end{tikzcd}$$

Functors

Given two categories, we can define "functions" between them.

Definition

Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A functor F consists of a function

$$|\mathcal{C}| \to |\mathcal{D}|$$

along with a map

$$\mathcal{C}(A, B) \to \mathcal{D}(F(A), F(B))$$

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

Functors

Given two categories, we can define ‘functions’ between them.

Definition

Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A functor F consists of a function

$$F_0|\mathcal{C}| \to |\mathcal{D}|$$

along with a function

$$F_{A,B} : \mathcal{C}(A, B) \to \mathcal{D}(F(A), F(B))$$

for all $A, B \in \mathcal{C}$. This map should have some properties with respect to composition.

Shorthand: For readability, we generally omit this subscripts for F_0 and $F_{A,B}$ and write $F(A)$ and $F(f)$. Sometimes we even omit the brackets.

Functors should preserve the compositional structure of the category. In other words:

- **Composition is conserved:** $F(f \circ g) = F(f) \circ F(g)$ for $g : A \to B$ and $f : B \to C$.
- **Identity maps are conserved:** $F(\mathrm{id}_A) = \mathrm{id}_{F(A)}$

Remark

The axioms above means that functors preserve commutative diagrams. In other words:

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow{\scriptstyle g} & & \downarrow{\scriptstyle h} \\ C & \xrightarrow{j} & D \end{array} \Rightarrow \begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ \downarrow{\scriptstyle Fg} & & \downarrow{\scriptstyle Fh} \\ FC & \xrightarrow{Fj} & FD \end{array}$$

Pointed sets

There is a **forgetful functor**

$$U : \mathbf{Set}_\bullet \to \mathbf{Set}$$

$$U(A, a_0) = A \text{ and } U(f) = f.$$

Examples of functors

- There is a **forgetful functor**

$$|-|: \mathbf{Vect} \to \mathbf{Set}$$

$$X \mapsto |X|.$$

  It assigns to every vector space its underlying category of points. It assigns to every morphism $f : V \to W$, the map $|f| : |V| \to |W|$, that sends x to $f(x)$. The map **forgets** the fact that X is a vector space.

- Similarly there are forgetful functors from metric, Banach and topological spaces to sets defined in the same way. There are other forgetful functors from Banach to Hilbert spaces, and from metric to topological spaces. You will encounter them on your homework.

Other examples of functors

- There is a **free functor**

$$\mathbb{R}[-] : \mathbf{Set} \to \mathbf{Vect}$$

$$\mathbb{R}[X] = \{f : X \to \mathbb{R} \mid \operatorname{Supp} f \text{ is finite.}\}.$$

where $\operatorname{Supp} f = \{x \in X \mid f(x) \neq 0\}$. For example, $\mathbb{R}\{e_1, e_2\}$ is isomorphic to $\mathbb{R}^2$. It sends a map $f : X \to Y$ to the map

$$\mathbb{R}[f] : \mathbb{R}[X] \to \mathbb{R}[Y]$$

$$\sum_{x \in X} a_x i_X(x) \mapsto \sum_{x \in X} a_x i_Y(f(x))$$

For example, it sends $e_1 \to e_2$ and $e_2 \to e_1$ to the linear map

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} : \mathbb{R}^2 \to \mathbb{R}^2.$$

Dual categories

Given any category $\mathcal{C}$, we can construct its **dual category** $\mathcal{C}^{op}$.

- The objects of $\mathcal{C}^{op}$ are the same as $\mathcal{C}^{op}$. In other words

$$|\mathcal{C}^{op}| = |\mathcal{C}|$$

- We have

$$\mathcal{C}^{op}(A, B) = \mathcal{C}(B, A)$$

  In other words, if there is a map $f : A \to B$ in $\mathcal{C}$, there is a map $f^{op} : B \to A$.

In fact $(-)^{op}$ is a functor.

Returning to the extension functor

Observe that every vector space V is of the form $\mathbb{R}[T]$ for some T. We can state the lemma from earlier in the language of category theory as:

Lemma

For all $S \in \mathbf{Set}$ and maps $i_S : S \to U(\mathbb{R}[S])$, such that for all $V = \mathbb{R}[T] \in \mathbf{FVect}$ and all maps $f : S \to U(V)$, there exists a **unique** *linear map $\overline{f} : \mathbb{R}[S] \to V$ making the following diagram commute in the category of sets:*

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "i_X"] & V = U(\mathbb{R}[T]) \\ U(\mathbb{R}(S)) \arrow[ur, dotted, "\overline{f}"] & \end{tikzcd}$$

Defining $\overline{f}$

- How do we define $\overline{f}$?

Defining $\overline{f}$

- How do we define $\overline{f}$?
- Let $S = \{s_1, \ldots, s_n\}$ and let $T = \{t_1, \cdots, t_m\}$. Then we can write

$$f(s_j) = a_{1,j} i(t_1) + \cdots + a_{m,j} i(t_m)$$

As a matrix, it is exactly

$$\begin{pmatrix} a_{1,1} & a_{2,1} & \cdots & a_{n,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{mn,2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1,m} & a_{2,m} & \cdots & a_{n,m} \end{pmatrix}$$

- This is the transition matrix you saw a few weeks ago!

Exercises on categories 1

1. Do sets and **injective** maps form a category? What about sets and strict inclusions?
2. Consider the set of **bracketed** words on some alphabet $\{e, a_1, a_2, \cdots a_n\}$ where, for any word w, we have that $we = ew = w$. Define

$$\mathbf{Words}(w_1, w_2) = \{u \mid (w_1)(u) = w_2\}$$

   Define $u \circ v = (v)(u)$ Does this form a category? What about **unbracketed** words?
3. Show that a vector space V and all linear maps $V \to V$ form a 1-object category.
4. A **unital monoid** is a set X equipped with a binary associative operation $X \times X \to X$ that is associative and which has an identity element. Show that there is a one-to-one correspondence between unital monoids and categories with one element.

Exercises on universal constructions

1. Is the forgetful functor **Vect** $\to$ **Set faithful**? Can you obtain every map of sets this way?
2. What linear transformations are of the form $F(f)$ for $f : X \to Y$ for the free functor $F : \text{Set} \to \text{Vect}$. Hint: Express it in terms of matrices.

Exercises on functors

1. Is the forgetful functor **Vect** $\to$ **Set faithful**? Can you obtain every map of sets this way?
2. What linear transformations are of the form $F(f)$ for $f : X \to Y$ for the free functor $F : \mathrm{Set} \to \mathrm{Vect}$. Hint: Express it in terms of matrices.

Further Reading

We have only touched the very tip of the iceberg in your categorical journey. Here are some references to learn more.

1. Course reference. Chapter 2 of Henning's notes for logic course. Available here.
2. A classic reference : Saunders Mac Lane. *Categories for the Working Mathematician*
3. An opinionated but motivational reference : Joseph A. Goguen. *A Categorical Manifesto*
4. A modern reference : Emily Riehl. *Category Theory in Context*