TITLE

Daniel Sinderson

May 2024

Introduction

Category Theory Basics

Categories

Category theory provides a mathematical structure for understanding the abstract concept of composition. Categories themselves are fairly simple to define.

Definition 1 (Category)**.** A category $\mathcal{C}$ is defined by the following:

1. $\mathcal{C}$ contains a collection of objects $ob(\mathcal{C})$
2. For any two objects $a, b \in ob(\mathcal{C})$ there is a collection of morphisms between those objects $\mathcal{C}(a, b)$ called the homset. We'll denote an element $f \in \mathcal{C}(a, b)$ using function notation $f : a \to b$.
3. Every object $a \in ob(\mathcal{C})$ has a morphism to itself $1_a : a \to a$.
4. For every two morphisms $f : a \to b$ and $g : b \to c$ there's a third morphism $g \circ f : a \to c$ that's their composition.

These objects and morphisms are then under the following two constraints:

1. (Unitality) Any morphism $f : a \to b$ can be composed with the identity morphisms of a and b such that $f \circ 1_a = 1_b \circ f = f$.
2. (Associativity) For any morphisms $f : a \to b$, $g : b \to c$, and $h : c \to d$, $h \circ (g \circ f) = (h \circ g) \circ f$.

It turns out that categories are a fantastically powerful abstraction. Since composition lies at the heart of most of mathematics and many other disciplines such as physics, engineering, and linguistics, an abstraction over composition has applications in a stunningly wide range of fields. Categories can even model categories.

Functors and Categories of Categories

Since categories are a mathematical structure, like groups or vector spaces, it makes sense to ask if there exists a kind of structure-preserving map between them, much like a homomorphism between groups or a linear transformation between vector spaces. For categories, such a map would have to preserve identities, composition, unitality, and associativity. We call such a map a functor.

Definition 2 (Functor)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ is a map between categories $\mathcal{C}$ and $\mathcal{D}$ such that the following hold:

1. For any object $a \in ob(\mathcal{C})$ there is an object $Fa \in ob(\mathcal{D})$.
2. For any morphism $f : a \to b$ between objects a and b in $\mathcal{C}$ there is a morphism $Ff : Fa \to Fb$ between objects Fa and Fb in $\mathcal{D}$.
3. For all objects $a \in ob(\mathcal{C})$ and $Fa \in ob(\mathcal{D})$, $F1_a = 1_{Fa}$.
4. For any composition of morphisms $g \circ f$ in $\mathcal{C}$, $Fg \circ Ff = F(g \circ f)$ in $\mathcal{D}$.

With functors, it's possible to construct a category where the objects themselves are categories and the morphism between them are functors. There are some technicalities for such constructions involving formal notions of local and global size that are necessary to avoid the Russel-esque paradox of the category of categories containing itself, but we'll ignore these in this paper.

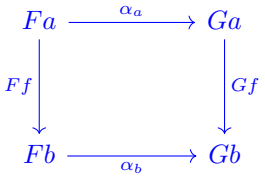
Functor Categories and Natural Transformations

Taking one more step into the rarefied air, it's also possible to create a category whose objects are functors and whose morphisms are structure-preserving maps called natural transformations.

Definition 3 (Natural Transformation)**.** A natural transformation $\alpha : F \Rightarrow G$ is a map between functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{C} \to \mathcal{D}$ such that the following holds:

1. For each object $a \in Ob(\mathcal{C})$, there is a morphism $\alpha_a : Fa \to Ga$ in $\mathcal{D}$. This is called the c-component of α.
2. For every morphism $f : a \to b$ in $\mathcal{C}$, $\alpha_b \circ Ff = Gf \circ \alpha_a$. This is called the naturality condition, and it preserves functoriality.

This means that the diagram below commutes.



Structure within Categories

Kinds of Morphisms

NOTE: Isomorphism in the category **Set** are bijective functions.

Definition 4 (Isomorphism)**.** We say that two objects in a category are isomorphic if there exists a morphism $f : a \to b$ for which there is another morphism $f^{-1} : b \to a$ such that $g \circ f = 1_b$ and $f \circ g = 1_a$. Such a morphism is called an isomorphism.

NOTE: Monomorphism in the category **Set** are injective functions.

Definition 5 (Monomorphism)**.** A morphism $f : b \to c$ is a monomorphism if for any pair of parallel morphisms $g, h : a \to b$, $f \circ g = f \circ h$ implies that $g = h$.

NOTE: Epimorphisms in the category **Set** are surjective functions.

Definition 6 (Epimorphism)**.** A morphism $f : b \to c$ is an epimorphism if for any pair of parallel morphisms $g, h : c \to d$, $g \circ f = h \circ f$ implies that $g = h$.

Universal Properties

Definition 7 (Terminal Object)**.** Let $t \in Ob(\mathcal{C})$ be given. Then t is a terminal object in $\mathcal{C}$ if the following hold:

1. For all $x \in Ob(\mathcal{C})$, there is exactly one morphism $f : x \to t$.
2. For any other object t' in $\mathcal{C}$ that meets condition (1), there exists an unique isomorphism $g : t \to t'$.

Dual to terminal objects are initial objects.

Definition 8 (Initial Object)**.** Let $i \in Ob(\mathcal{C})$ be given. Then i is a initial object in $\mathcal{C}$ if the following hold:

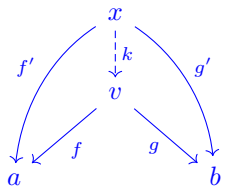
1. For all $x \in Ob(\mathcal{C})$, there is exactly one morphism $f : i \to x$.
2. For any other object i' in $\mathcal{C}$ that meets condition (1), there exists an unique isomorphism $g : i \to i'$.

In category theory there's a universal property that generalizes notions of multiplication called the product.

Definition 9 (Product)**.** Let $a, b, v \in Ob(\mathcal{C})$ and morphisms $f : v \to a$ and $g : v \to b$ be given. The object v is a product of a and b if the following holds:

1. For any other object x with morphisms $f' : x \to a$ and $g' : x \to b$, there is a unique isomorphism $k : x \to v$ such that $f \circ k = f'$ and $g \circ k = g'$.

This means the following diagram commutes:



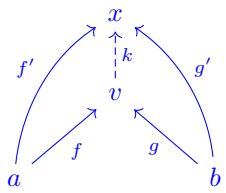
This conditions ensures that the object v is the universal or canonical object satisfying the pattern, with the morphisms of any other such object being able to be written as a sort of embellishment of the morphisms from v.

Dual to products are coproducts, which generalize the notion of addition.

Definition 10 (Coproduct)**.** Let $a, b, v \in Ob(\mathcal{C})$ and morphisms $f : a \to v$ and $g : b \to v$ be given. The object v is a coproduct of a and b if the following holds:

1. For any other object x with morphisms $f' : a \to x$ and $g' : b \to x$, there is a unique isomorphism $k : v \to x$ such that $k \circ f = f'$ and $k \circ g = g'$.

This means the following diagram commutes:

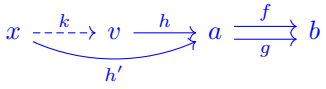


NOTE: In **Set**, equalizers pick out the subset of a set a that is the intersection of the preimages $f^{-1} : b \to a$ and $g^{-1} : b \to a$.

Definition 11 (Equalizer)**.** Let $a, b, v \in Ob(\mathcal{C})$ and let morphisms $f : a \to b$, $g : a \to b$, and $h : v \to a$ be given. The object v is an equalizer if the following hold:

1. $f \circ h = g \circ h$.
2. For any other object $x \in Ob(\mathcal{C})$ with morphism $h' : x \to a$ such that $f \circ h' = g \circ h'$, there exists a unique morphism $k : x \to v$ such that $h' = h \circ k$.

This means the following diagram commutes:

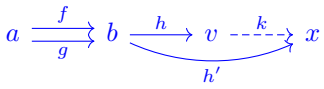


NOTE: In **Set**, coequalizers pick out the subset of a set b that is the intersection of the images $f : a \to b$ and $g : a \to b$.

Definition 12 (Coequalizer)**.** Let $a, b, v \in Ob(\mathcal{C})$ and let morphisms $f : a \to b$, $g : a \to b$, and $h : b \to v$ be given. The object v is a coequalizer if the following hold:

1. $h \circ f = h \circ g$.
2. For any other object $x \in Ob(\mathcal{C})$ with morphism $h' : b \to x$ such that $h' \circ f = h' \circ g$, there exists a unique morphism $k : v \to x$ such that $h' = k \circ h$.

This means that the following diagram commutes:

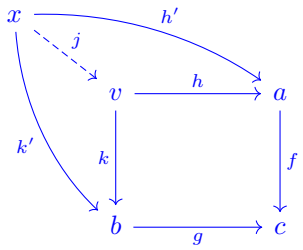


NOTE: In **Set**, pullbacks are the fiber product of two sets.

Definition 13 (Pullback)**.** Let $a, b, c, v \in Ob(\mathcal{C})$ and let morphisms $f : a \to c$, $g : b \to c$, $h : v \to a$, and $k : v \to b$ be given. The object v is a pullback if the following hold:

1. $f \circ h = g \circ k$.
2. For any other object $x \in Ob(\mathcal{C})$ with morphisms $h' : x \to a$, and $k' : x \to b$ such that $f' \circ h = g' \circ k$, there exists a unique morphism $j : x \to v$ such that $f' = f \circ j$ and $g' = g \circ j$.

This means that the following diagram commutes:



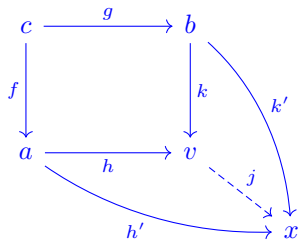
NOTE: In **Set**, pushouts are...

Definition 14 (Pushout)**.** Let $a, b, c, v \in Ob(\mathcal{C})$ and let morphisms $f : c \to a$, $g : c \to b$, $h : a \to v$, and $k : b \to v$ be given. The object v is a pushout if the following hold:

1. $h \circ f = k \circ g$.

2. For any other object $x \in Ob(\mathcal{C})$ with morphisms $h' : a \to x$, and $k' : b \to x$ such that $h' \circ f = k' \circ g$, there exists a unique morphism $j : v \to x$ such that $h' = j \circ h$ and $k' = j \circ k$.

This means that the following diagram commutes:



In this last part of the section we're going to discuss a generalization of all of the previous universal properties called cones and cocones. A cone consists of any finite collection of objects and morphisms between them, combined with an additional object v called the vertex of the cone and morphisms from v to all other objects in the collection. Such a cone is universal if for all other cones of that type in the category there is a unique morphisms from their vertex to v: i.e, $k : v' \to v$ exists and is unique.

A cocone consists of any finite collection of objects and morphisms between them, combined with an additional object v called the vertex of the cocone and morphisms from all other objects in the collection to v. Such a cocone is universal if for all other cocones of that type in the category there is a unique morphisms from v to their vertex: i.e, $k :\to v'$ exists and is unique.

Limits and Colimits

Given any cone, if there is an instance of that cone for which all other instances have a unique morphism to it, that cone is the limit. Dually for cocones and colimits. In fact, it's possible to construct a category whose objects are the cones and whose morphisms are these unique factorizations. In this category, the limit is a terminal object. Dually for cocones and colimits again, except the colimit is an initial object in the constructed category of cocones instead of a terminal object.

Structure Between Categories

Kinds of Functors

Definition 15 (Full Functor)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ is full if for each $x, y \in Ob(\mathcal{C})$, the map $\mathcal{C}(x, y) \to \mathcal{D}(Fx, Fy)$ is surjective.

Definition 16 (Faithful Functor)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ is faithful if for each $x, y \in Ob(\mathcal{C})$, the map $\mathcal{C}(x, y) \to \mathcal{D}(Fx, Fy)$ is injective.

Definition 17 (Essentially Surjective On Objects)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ is essentially surjective on objects if for each $y \in Ob(\mathcal{D})$, there is some $x \in Ob(\mathcal{C})$ such that y is isomorphic to Fc.

Definition 18 (Injective on Objects)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ is injective on objects if for each $x \in Ob(\mathcal{C})$, there is a unique $y \in Ob(\mathcal{D})$ such that $y = Fx$.

NOTE: Play a huge role in category theory: i.e in the Yoneda lemma
NOTE: Representable functors require that the source category $\mathcal{C}$ be locally small so that the collections of morphisms between objects in the category are sets.

Definition 19 (Covariant Representable)**.** A functor $F : \mathcal{C} \to \mathbf{Set}$ is representable if there is an object $c \in Ob(\mathcal{C})$ such that the functor $\mathcal{C}(c, -) : \mathcal{C} \to \mathbf{Set}$ is naturally isomorphic to F.

Definition 20 (Contravariant Representable)**.** A functor $F : \mathcal{C}^{op} \to \mathbf{Set}$ is representable if there is an object $c \in Ob(\mathcal{C})$ such that the functor $\mathcal{C}(-, c) : \mathcal{C}^{op} \to \mathbf{Set}$ is naturally isomorphic to F.

Definition 21 (Forgetful)**.** A forgetful functor $F : \mathcal{C} \to \mathbf{Set}$ is a functor that sends objects in a category of structures-on-sets, like groups or preorders, to the underlying sets themselves. They "forget" the additional structure in the source category.

Definition 22 (Free)**.** A free functor $F : \mathbf{Set} \to \mathcal{C}$ is a functor that sends sets to the free construction on that set, like the free group on the set. They construct the desired structure on the set in a way that requires no choices.

Comparing Categories

Definition 23 (Subcategory)**.** A category $\mathcal{C}$ with a functor $F : \mathcal{C} \to \mathcal{D}$ that is both faithful and injective on objects is a subcategory of the domain $\mathcal{D}$. A functor F of this type is called an **embedding** of $\mathcal{C}$ into $\mathcal{D}$.

Definition 24 (Full Subcategory)**.** A category $\mathcal{C}$ with a functor $F : \mathcal{C} \to \mathcal{D}$ that is full, faithful, and injective on objects is a full subcategory of the domain $\mathcal{D}$. A functor F of this type is called a **full embedding** of $\mathcal{C}$ into $\mathcal{D}$.

Definition 25 (Natural Isomorphism)**.** A natural isomorphism is a natural tranformation $\alpha : F \to G$ where all of the c-components of the transformation $\alpha_c : Fc \to Gc$ are isomorphisms. This means there exists some other natural transformation $\eta : G \to F$ such that $\eta_c \circ \alpha_c = 1_{Fc}$ and $\alpha_c \circ \eta_c = 1_{Gc}$.

Definition 26 (Adjunctions)**.** An adjunction is a pair of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ such that for any object $c \in Ob(\mathcal{C})$ and any object $d \in Ob(\mathcal{D})$ there is an isomorphism between the sets of functions between them and their images under the functors that is natural. In other words, that for every $c \in Ob(\mathcal{C})$ and $d \in Ob(\mathcal{D})$ there is a natural isomorphism $\mathcal{C}(c, Gd) \cong \mathcal{D}(Fc, d)$.

In this case, the functor F is the left adjoint functor and the functor G is the right adjoint functor.

The Yoneda Lemma

Definition 27. The Yoneda lemma states that for any functor $F : \mathcal{C} \to \mathbf{Set}$ where $\mathcal{C}$ is locally small and any object $c \in Ob(\mathcal{C})$, there is a bijection between the collection of natural transformations going from the functor $\mathcal{C}(c, -)$ to F and the object $Fc \in Ob(\mathbf{Set})$.

Stated another way, this means that $Hom(\mathcal{C}(c, -), F) \cong Fc$ for any functor $F : \mathcal{C} \to \mathbf{Set}$ and all objects $c \in Ob(\mathcal{C})$.

References

[Che22] Eugenia Cheng. *The joy of abstraction: an exploration of math, category theory, and life*. Cambridge University Press, 2022.

> *The Joy of Abstraction* provides a gentle yet rigorous tour of category theory by first building up some intuition with non-mathematical concepts, then surveying common mathematical structures from ungraduate mathematics as categories, and then presenting the real (formal) definitions of the basic concepts of the theory. This book is a perfect introduction and companion volume to the richer, more advanced texts.

[FS19] Brendan Fong and David I Spivak. *An invitation to applied category theory: seven sketches in compositionality*. Cambridge University Press, 2019.

> This is the go-to book for applications. It's also a fantastic source for building intuition and finding detailed examples of core categorical concepts. It's been core to my research.

[Gol14] Robert Goldblatt. *Topoi: the categorial analysis of logic*. Elsevier, 2014.

> This is a large book covering much more than basic category theory, but the first three chapters provide an excellent primer on the fundamentals. It's a bit older so some of the notation is different and the topics covered are slightly different. This is a good thing though! The inclusion of arrow categories and comma categories in chapter two gives a different perspective on functor categories and categories of (co)cones that I found useful.

[Lei14] Tom Leinster. *Basic category theory*, volume 143. Cambridge University Press, 2014.

> This is a good second presentation of ideas. Its scope is smaller than *Category Theory in Context* so it can spend more time on explanations. Otherwise it's written at a similar level.

[LR03] F William Lawvere and Robert Rosebrugh. *Sets for mathematics*. Cambridge University Press, 2003.

> *Sets for Mathematics* has been good for its longer explanations of concepts and the fact that it's all happening in **Set**, which is familiar from last term. It does use different terminology sometimes. This can be bad because it's confusing, but it can also be a good test of my understanding of the

underlying math instead of my understanding of terminology.

[Mil18] Bartosz Milewski. *Category theory for programmers*. Blurb, 2018.

This is another helpful book for acquiring intuition when the more rigorous and mathematical texts confused me. The level of rigor is around the same as (or even less than) *The Joy of Abstraction* though and the examples are only useful if you have experience programming. Otherwise they'll just add to your confusion.

[ML13] Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.

This book was immediately overwhelming. The overall level of expected mathematical maturity was simply too high (graduate level). I put it down after the first couple of pages.

[Mye22] David Jaz Myers. Categorical systems theory. *Manuscript in preparation*, 2022.

no annotation yet

[Rie17] Emily Riehl. *Category theory in context*. Courier Dover Publications, 2017.

This book is not for the faint of heart. It is, without hesitation, the most difficult book I've ever read. Riehl's definitions are detailed, clear, and rigorous, but she expects a lot of mathematical maturity from the reader. This really becomes apparent in her examples, within which she assumes a deep understanding of all of undergraduate mathematics. This book is the syllabus and the test: it tells me what I need to know and tells me, in no uncertain terms, if I actually know it. The other books are the lectures where I actually learn the concepts.

[Ros22] Daniel Rosiak. *Sheaf theory through examples*. MIT Press, 2022.

no annotation yet

[Vak17] Ravi Vakil. The rising sea: Foundations of algebraic geometry. *preprint*, 2017.

no annotation yet