# Category theory notes

December 28, 2018

They say you don't understand something until you can explain it, so here is my exercise in explaining category theory. I make no claims to any authority on this subject and the document is incomplete in proportion to my incomplete understanding.

## 1 Why

The first question I have had to explain, to myself and every single person I've spoken to about these things: why study category theory at all? This is forward looking, but here are my impressions halfway through a few textbooks:

- By *category*, we mean *context*, and we're looking for patterns across contexts. When two mathematical pursuits do the same thing, can we formalize exactly how they are or are not identical? There is a promise that even theoretical work in real-world contexts (i.e., my actual job) will benefit from better formalization of how contexts relate.
- I'm hoping it'll be a shortcut to learning more math. If I get this, then algebraic topology should be metaphorical cake.
- There is a trend in programming languages toward languages whose theory is based on a category of types. Out-of-fashion languages are written in terms of procedures: plug 4 into the square functions and we do some things and return 16. Languages oriented toward mathematical functions encourage a thinking that is far more still: the square of 4 is 16 and always has been and always will be.
- From what I've read, a lot of philosophers work on it—there's some presumption that it's not just about symbol shunting, but about finding 'natural' and 'universal' properties of those patterns across disparate things. The relationship between these categories/contexts has

always been there and we only need a way to see them. Here's Robert Penn Warren on the matter:

Here is the shadow of truth, for only the shadow is true.
And the line where the incoming swell from the sunset Pacific
First leans and staggers to break will tell all you need to know
About submarine geography, and your father's death rattle
Provides all biographical data required for the *Who's Who* of the
dead.

. . . In the distance, in *plaza, piazza, place, platz,* and square,
Boot heels, like history being born, on cobbles bang.

Everything seems an echo of something else.

. . . I watched the sheep huddling. Their eyes
Stared into nothingness. In that mist-diffused light their eyes
Were stupid and round like the eyes of fat fish in muddy water,
Or of a scholar who has lost faith in his calling. . . .

You would think that nothing would ever again happen.

That may be a way to love God.

OK, now that we've covered the *why* question, let's start from simple element-to-element mappings and work up.

## 2 Leveling up

A function $f : A \to B$, aka a mapping, assigns a value to all of the elements in the set $A$ to some element(s) in the set $B$. Note that if $A = \{\emptyset\}$ then you still have a function, though it is vacuous.

We can draw this function via arrows, maybe

$$
\begin{tikzcd}
a_1 \arrow[dr] & b_1 \\
a_2 \arrow[r] & b_2 \\
a_3 \arrow[r] & b_3
\end{tikzcd}
$$

Here, the nodes in the graph are single elements in the sets, and the arrows describe where each element goes.

The function notation $f : A \to B$ is itself a tiny graph. If there were also a function $g : C \to B$, we could construct a diagram that looks like the above element-by-element graph, but with sets at the nodes instead of single elements:

$$\begin{tikzcd} A \arrow[r, "f"] & B \\ C \arrow[ur, "g"] & \end{tikzcd}$$

So we've gone up a level: each arrow at this level could be cracked open to reveal a bundle of arrows, its own element-by-element diagram like the one we had above.

A closed function $f : A \to A$ maps from a set to itself, so we'd draw it as a loop:

$$A \circlearrowleft$$

I think a lot of math in the early- to mid-1900s worked at this level, defining as much as possible via sets and mappings between sets. But we can bundle further. In the last diagram we had a single function. But consider a collection of mappings (aka homomorphisms) from set $A$ to set $B$, $\mathrm{Hom}_{\mathbf{set}}(\mathrm{A}, \mathrm{B})$. The first diagram with three elements in $A$ and three in $B$ could be rewired nine different ways, but it's our choice and we can include one to nine functions as desired. We would draw these bundles as

$$A \xrightarrow{\mathrm{Hom}_{\mathbf{set}}} B$$

I think of this arrow as a bundle of quivering wires, each with its own power. If this were anime they'd be tentacles. But there is indeed a lot of power there: with $\mathbb{R}$ representing the real numbers, we can now summarize all real one-variable functions with the diagram

$$\mathbb{R} \circlearrowleft$$

We are approaching the level at which category theory speaks. But we can go one step further, and put this into a box as well, and now it is a single node in our graph. Let the above loop of all functions $f : \mathbb{R} \to \mathbb{R}$ be $\mathbf{R}$, and let's keep its company with the set of all functions on the natural numbers, $g : \mathbb{N} \to \mathbb{N}$, notated $\mathbf{N}$.

$$\mathbf{N} \xrightarrow{F} \mathbf{R}$$

Here, the arrow is a *functor*, which is a bundle of two sets of arrows at once: one to map elements of $\mathbb{N}$ to elements of $\mathbb{R}$, and one to map the functions $\mathbb{N} \to \mathbb{N}$ to functions $\mathbb{R} \to \mathbb{R}$. [There are additional restrictions; see below.] Custom is to write these with capital letters, to contrast with typically lower-case functions.

All these diagrams look alike, so for any diagram you come across in a textbook, it's worth checking where we are in the pyramid:

- Node is a single element; arrow represents a mapping from one element to another
- Node is a set; arrow represents a set of mappings from one set to another, aka a function
- Node is a set; arrow represents a set of functions from one set to another
- Node is a set of objects with their own sets of functions; arrow represents a functor from one context to another

## 3 Bundles

We're used to thinking in terms of sets comprising atomic elements, but mathematical objects are typically a bundle of parts. Non-mathematicians think about the non-negative real numbers $\mathbb{R}^+$, but we have to start thinking about ($\mathbb{R}^+$ paired with addition) and ($\mathbb{R}^+$ paired with multiplication) as different things. Some bundles:

- *Magma*: A set $S$ and a function $f : S \to S$.
- *Preorder*: A set $S$ and a binary relation $\leq$. (See flash cards for the formal definition)
- *Graph*: A collection $(V, A, src, tgt)$:
  - $V$ a set (vertices)
  - $A$ a set (arrows)
  - $src : A \to V$ a function giving arrow sources
  - $tgt : A \to V$ a function giving arrow targets

Authors who write in object-oriented programming languages will recognize this use of *object* to mean a structure of elements, some of which could be verbs.

## 3.1 Constraints

These objects often have constraints attached. For example, what we name the pair $(S, f)$ depends on how restrictive we want to be about the function:

- *Magma*: A set $S$ and a function $f : S \to S$.
- *Semigroup*: A magma where $f$ is associative: $f(a, f(b, c)) = f(f(a, b), c)$.
- *Monoid*: A semigroup with an identity element for which $f(id, s) = s$, for all $s \in S$.
- *Group*: A monoid where every element has an inverse: $\forall x \in S$, $\exists x_{inv}$ such that $f(x, x_{inv}) = id$.

As an example, if we allow zero in $\mathbb{R}^+$, then $(\mathbb{R}^+, +)$ is a monoid (but no negative numbers, so no inverses) and $(\mathbb{R}^+, \cdot)$ is a group.

Associativity and identity are going to be part of the definition of categories, so there won't be a lot of discussion of things on the list before monoids. Groups are reserved for contexts where the sense of reversibility or symmetry make sense.

## 3.2 Category constraints

A category is a bundle of elements as per the end of the chain of arrow abstractions. Bundle together some things that will be nodes (herein, objects), and the sets of mappings between some or all of them.

You get to pick which mappings you will put in the set of mappings, with a few simple conditions. You don't have to include every mapping (herein, morphism) from node to node, which as above may cover all of one-variable high school algebra with a single arrow. But for every set we do have to include the identity morphism $id_A : A \to A$ mapping every element of an object to itself.

A category also has a composition function, so given $F : A \to B$ and $G : B \to C$, we have to include $G \circ F : A \to C$. If the composition function is the usual $g(f(a)) = c$, so this is not difficult, but in case you get creative, the textbooks also require that $id_b \circ f(a) = f(a)$ and $g \circ id(b) = g(b)$.[1]

Having these additional rules means we can't just throw out any old set of nodes and edges and call it a category, but it's not hard to add these extra mappings as needed, to the point that people usually don't even bother drawing the loops to mark the identities.

[1] Every author makes a big deal of how $G \circ F$ looks backward, but all you have to do is read $\circ$ as *of*, as in $g$ *of* $f$, which is exactly how people read $g(f(x))$. Really not a big deal at all.

Functors, as above, are a bundle of mappings of objects to objects plus mappings from functions to functions. We want them to represent a sensible transformation from one context to another, so a functor $F$ has to have these constraints on how it maps functions and objects:

- $F(f : A \to B) = F(f) : F(A) \to F(B)$
- $F(id_A) = id_{F(A)}$
- $f \circ g = F(f) \circ F(g)$

# 4 Starting to map

The nodes in all the graphs above could be anything, including the bundles of elements from Section 3.

Let **PrO** be the set of all preorders and **Graph** be the set of graphs. A single preorder is a set of relations of the form $a \leq b$, and we could replace the $\leq$ symbol with an arrow, like $a \leftarrow b$, and we have another little inline graph. Doing the same for all of the elements in the preorder would generate a list of (nodes, arrows, sources, targets) in **Graph**. That is, $f : \mathbf{PrO} \to \mathbf{Graph}$ makes as much sense as $f : \mathbb{R} \to \mathbb{R}$. Set up such a mapping for every element of **PrO**, and we have a category diagram with two objects and one arrow representing all homomorphisms mapping all partial orders to graphs.

A monoid as defined above is already a category. Objects are the set of elements $M$ on which the monoid is defined, we require a function $f : M \to M$ which must be associative, and we require an identity mapping $f(id, a) = a$ (in fact, two, because $f(a, id)$ also equals $a$). By definition, all the boxes are checked.

But we can also express a monoid in a manner that doesn't fit the intuitive approach of having a set of elements, some functions between them, and the usual function composition. I'm not clear on the benefits, but is is definitely cooler. Here are the components of another category built from a monoid $(S, id, f)$:

- Objects: following the name *monoid*, this category has exactly one object. Call it §.
- Morphisms: For every monoid element $M$, define a morphism $M : \S \to \S$. Include an identity element $id : \S \to \S$. That is, the same mapping, $f(\S) = \S$, appears several times in the same category, distinguished by different labels.
- Composition: $M_1 \circ M_2 \equiv f(M_1, M_2)$.

A category requires a composition function that is associative and correctly handles the identity morphism, and we get exactly that from the monoid requirements on $f$. More on this below.

Categories are themselves a bundle like the ones from Section 3: a set of objects paired with a set of morphisms between them. So they too could be a node in one of the above diagrams, giving us the category of categories, **Cat**.[2]

Things are already getting self-referential: **PrO** $\to$ **Graph** is a node-and-arrow graph representing an operation involving the set of node-and-arrow graphs. For **Cat**, we can show that **Cat** $\to$ **Graph**, thus drawing a graph for two categories to represent how categories and graphs relate. Mathematicians love this stuff.

## 5 Where commuting gets us

It's already potentially interesting that we can express mappings between different concepts like orderings and graphs, but things don't get interesting until we have multiple paths to the same place.

For example, Spivak [2014] describes multisets as sets of elements $E$ with a set $B$ of symbol names, and a map $\pi : E \to B$. So, for example, we could replace all the words in a book with numbers, producing a sequence $e_1, e_2, \ldots$, then have a dictionary of the words, and $\pi$ tells us that $e_{250}$ maps to *dog*. We may have another set of elements and symbols in another context, maybe a corpus in another language, so we'll need an element mapping $f_1 : E \to E'$ and a symbol mapping $f_2 : B \to B'$. See the flash cards for formalization.

Here is a diagram including both the functions to map a multiset $(E, B, \pi)$ to another, and the $\pi$ functions inside the multisets:

$$\begin{tikzcd}
E \arrow[r, "f_1"] \arrow[d, "\pi"'] & E' \arrow[d, "\pi'"] \\
B \arrow[r, "f_2"'] & B'
\end{tikzcd}$$

The dictionary definition of *commute* is *to exchange*; in math circles, *commutativity* typically refers to reversing the order of things, like $x + y = y + x$. In this case, we can get from the original element set to the primed

[2]Authors refer to *the* category of categories, and I'm unclear whether an alternative cat-of-cats exists or what it would look like, so we may be at the top here—there's no category of categories of categories.

symbol set in two ways: either apply the symbol map $\pi$ first to get to $B$ and then step from the unprimed to primed corpus (calculating $f_2 \circ \pi(e)$), or first step to the primed corpus via $f_1$ and then map from $E$ to $B$ on that side (calculating $\pi \circ f_1(e)$).

When are both routes identical? This is often an interesting question in its own right. For books and word lists in Portuguese and English, I'd guess there's a natural way to map simple nouns: finding that an element in the $P$ corpus is *cão*, then translating that word to the Portuguese-to-English dictionary to find that it means *dog* is likely equivalent to the commuted procedure of first finding the element in the English corpus matching our element of interest, then looking up that that element in the $E$ corpus maps to *dog*. This is going to be easy and natural

Doing the same with *saudade* is going to take some forcing. It seems to me that these exceptions are interesting in their own right: we're going to generate a list of categories that all behave similarly and relate nicely, but what about those things that have to be thrown out to fit the categorical mold? The textbooks that focus on that mold naturally exclude them, but by definition, standardization requires jettisoning the unique.

Indeed, one way to nail down a relationship is to just cull down what is in your sets. The set of books, $B$, can be cataloged either via the Library of Congress classification system (a set of call numbers $L$), or the Dewey decimal system (another set of call numbers $D$). Let us pair together call numbers referring to the same book, in the cross-product space $D \times L$. The full cross product matching every Dewey number with ever LoC number makes no sense, and not all books have a classification in every system, so out of the full cross of all pairs $(d, l)$, we will write down the relatively small list of pairs that are about the same book, herein $D \times_B L$. Finally, we have commutativity:

$$\begin{tikzcd} D \times_B L \arrow[r] \arrow[d] & L \arrow[d] \\ D \arrow[r] & B \end{tikzcd}$$

This is a *pullback*. There is information in the pairing, and maybe even information in the list of books and call numbers that can't be found from $D \times_B L$ and are now lost in the stacks.

## 5.1 Natural transformations

If there is a mapping from one context to another where (apply internal function, then jump) produces the same outcome as (jump, then apply internal function), we call the jump across categories a *natural transformation*. We had a set-level example above where (find English lookup, translate lookup to Portuguese) and (find Portuguese match to element in English corpus, lookup in Portuguese) had the same outcome. We can do the same at the category level.

Given two categories $\mathbf{C}$ and $\mathbf{D}$, we can define $\mathrm{Fun}(\mathbf{C}, \mathbf{D})$ as the category of functors between the two objects, and the set of arrows in that category will be the natural transformations. This motivates defining and identifying natural transformations, so we can bring the functors mapping $\mathbf{C}$ to $\mathbf{D}$ into the categorical club.

Start with two categories and two functors, $F$ and $G$, each independently mapping the first category to the second. Pick two elements from the first category, $a$ and $b$, and a mapping $h : a \to b$. We want mappings $\alpha_a$ and $\alpha_b$ in the second category so that

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

The functors are a natural transformation if, for any two elements of the first category, we have $\alpha$s that make this square commute. Note that all of $F(a), F(h), G(b), \ldots$ are objects or mappings in the second category, and we require the $\alpha$s to be as well. If we have all this, then we have the sort of naturalness we seek: you can start in the $F$-transformed version and walk from $F(a)$ to $F(b)$ and then walk over to the $G$-transformed version and get to $G(b)$; or you can commute the order and $\alpha$-walk over to the $G$-transformed version first and then from $G(a)$ to $G(b)$.

Here is a failed attempt. Consider the category consisting of a single object $A$, one named $B$, and an arrow $A \to B$. We would like to map this category to ordered natural numbers, where the objects are $1, 2, \ldots$ and the relation is $\leq$ by its usual meaning. Here are two functors we can try to naturally transform:

$$\begin{aligned} F(A) &= 2 \quad ; \quad F(B) = 5 \\ G(A) &= 3 \quad ; \quad G(B) = 4 \end{aligned}$$

These are valid functors, because both $2 \leq 5$ and $3 \leq 4$ are arrows that exist in the target category.

But to have a natural transformation from the $F$ functor to the $G$ functor, we need to find those two $\alpha$s to complete the square. We can't just make them up: they have to be part of the set of mappings at the $(\mathbb{N}, \leq)$ category that both functors map to. There is an arrow $2 \to 3$ in that category, so we can do the walk of $F(A) \to G(A) \to G(B)$, or $2 \leq 3 \leq 4$. But we don't have $5 \leq 4$, so the walk from $F(A) \to F(B) \to G(B)$ isn't possible using the arrows the target category gave us.

Given that all our arrows go uphill, we will need transformations where $F(A)$ is downhill from both intermediate points $G(A)$ and $F(B)$, and $G(B)$ is uphill from both intermediate points. For example, given

$$\begin{aligned} F(A) = 2 \quad &; \quad F(B) = 3 \\ G(A) = 4 \quad &; \quad G(B) = 5 \end{aligned}$$

there is a natural transformation from $F$ to $G$ (but not from $G$ to $F$).

If we wanted to start with a category with more than two elements, we would need to show that this relation holds for *any* pair of elements with an arrow between them.

## 5.2 Chaining

Further, all these commutative diagrams chain together. If we had English ($E$), Portuguese ($E'$), and Russian ($E''$) corpora, we could draw:

$$\begin{tikzcd} E \arrow[r, "f_1"] \arrow[d, "\pi"'] & E' \arrow[r, "f_3"] \arrow[d, "\pi'"] & E'' \arrow[d, "\pi''"] \\ B \arrow[r, "f_2"'] & B' \arrow[r, "f_4"'] & B'' \end{tikzcd}$$

It is self-evident that if the left square and the right square commute, than any flow from upper left to lower right will be equivalent. This becomes especially interesting when we have mappings across different contexts: if we can map from multisets to preorders and preorders to graphs, we just got a multiset-to-graph generator for free.

Natural transformations also chain: if there exists an n.t. from functor $F$ to functor $G$, and from functor $G$ to functor $H$, then we have a natural transformation from $F$ to $H$. If we have three categories and natural transformations $F$ and $G$ from first to second, and $H$ and $I$ from second to third, then $H \circ F$ must have a natural transformation to $G \circ I$.

## 5.3 Universal properties

Consider the cross product of two sets, $A \times B$. There is an obvious mapping from $A \times B \to A$ and $A \times B \to B$—just drop the unneeded $B$ or $A$ part.

Here is the shape of this as a diagram, with $\pi_1$ and $\pi_2$ being the *drop one part* functions:

$$
\begin{tikzcd}
 & X \times Y \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
X & & Y
\end{tikzcd}
$$

This is called a span and can also be turned into a little inline diagram as $X \leftarrow (X \times Y) \rightarrow Y$. Of course, there are other things we could put at the top of the fork, some other set $A$ that maps both ways: $X \leftarrow A \rightarrow Y$. But $X \times Y$ is unique in this capacity because every $A$ and its span can be expressed via a morphism to $X \times Y$. Here is the commutative diagram:

$$
\begin{tikzcd}
 & X \times Y \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
X & & Y \\
 & A \arrow[ul, "\forall f"] \arrow[ur, "\forall g"'] \arrow[uu, dashed, "f \times g"] &
\end{tikzcd}
$$

As noted in the diagram, the $A \rightarrow (X \times Y)$ morphism is trivial to generate as the ordered pair $(f(a), g(a))$. For any $f$ and $g$, you could go the short way to map $A$ to $X$ or $Y$, but you are guaranteed that there is also a pair of mappings $A \rightarrow (X \times Y) \rightarrow X$ and $A \rightarrow (X \times Y) \rightarrow Y$ that are the long commute to the same point. Put differently, you can have an arbitrary span composed of any object and mappings iff you can write down a map $A \rightarrow (X \times Y)$. That any span can be expressed by going through $X \times Y$ makes that object a sort of universal spanner. Returning to the problem of combining Dewey and LoC numbers, you could come up with all sorts of other clever combination schemes besides the ordered pair (Dewey, LoC), but you are guaranteed that all of them must reduce to this simpler scheme. This sounds obvious, but people put a lot of time into trying to be clever.

One can generate other universal properties for other common diagrams and operations. As per the names, the intent is that the universal properties do say something fundamental about the relationship diagram, and that natural transformations do describe the natural way to move from one domain to another.

## References

Steve Awodey. *Category theory.* Oxford University Press, Oxford New York, 2010.

F. W. Lawvere. *Conceptual Mathematics: A First Introduction to Categories.* Cambridge University Press, Cambridge, UK New York, 2009.

Emily Riehl. *Category theory in context.* Dover Publications, Inc, Mineola, New York, 2016.

David Spivak. *Category Theory for the Sciences.* The MIT Press, Cambridge, Massachusetts, 2014.