# Category theory notes

December 26, 2018

Right now the document only touches on and motivates what seem to be the key ideas pushed by the basic category theory textbooks I've been reading. There is still a long to-do list I am trying to get through: duality, more about natural transformations and how they chain together, adjoints, probably more examples.

## 1 Intro

The question I've gotten from everybody I talk to about this: why study category theory at all? Remember I'm a novice and this is forward looking, but here are my impressions halfway through a few textbooks:

- I'm hoping it'll be a shortcut to learning more math. If I get this, then algebraic topology should be metaphorical cake.
- By *category*, we mean *context*, and we're looking for patterns across contexts: a lot of mathematical pursuits do the same thing over and over. Can we formalize exactly how whatever two things are actually the same thing? 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.
- Also, there is a trend in programming languages toward languages that use mappings from a certain category, the monad, back to itself (aka an endofunctor).
- 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 textbook intros seem to motivate it via verbs and motion, as if you are touring from category to category, but maybe it's the opposite.

Maybe math is not 'plug 4 into the square functions and we do some things and return 16' but 'the square of 4 is 16 and always has been and always will be.' 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.<br>
And the line where the incoming swell from the sunset Pacific<br>
First leans and staggers to break will tell all you need to know<br>
About submarine geography, and your father's death rattle<br>
Provides all biographical data required for the *Who's Who* of the<br>
dead.

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

Everything seems an echo of something else.

. . . I watched the sheep huddling. Their eyes<br>
Stared into nothingness. In that mist-diffused light their eyes<br>
Were stupid and round like the eyes of fat fish in muddy water,<br>
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 I've answered that 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 to its own set, so we'd draw it as a loop:

$$A \circlearrowleft$$

But we can bundle further. In the last diagram we had a single function. But consider the collections of all mappings of the form $f : A \to B$. The first diagram with three elements in $A$ and three in $B$ could be rewired nine different ways. The full set of these mappings (aka homomorphisms) is $\mathrm{Hom}_{\mathbf{set}}(\mathrm{A}, \mathrm{B})$. But our collection doesn't have to be the whole set; it's our choice. Sets use capitals and their elements lower case ($s \in S$), so by analogy, given lower-case functions $f$, let a collection of those be $F$; we would draw these bundles as

$$A \xrightarrow{F} B$$

I think of $F$ 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

$$\mathbb{R} \circlearrowleft$$

This is 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 the set of all functions on the natural numbers, $g : \mathbb{N} \to \mathbb{N}$, notated $\mathbf{N}$.

$$\mathbf{N} \longrightarrow \mathbf{R}$$

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 $\mathbb{R} \to \mathbb{R}$. necessarily going to make it.

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 its own set 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, and 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 $f(id) = id$.
- *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 (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 characteristics

See the flash cards for the formalization, but less formally, 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 (herien, 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.

It also has a composition function, so given $F : A \to B$ and $G : B \to C$, we have to include $G \circ F : A \to C$. I haven't yet seen this composition function be anything but the obvious $g(f(a)) = c$, so this is not difficult, but in case you get creative, the textbooks also require that $id_b \circ f(a) = f(a)$ and $g \circ id(b) = g(b)$.

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.

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:

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

### 3.3 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.

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**.[1]

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

## 4 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

[1] 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.

$e_1, e_2, \ldots$, 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 for mapping a multiset $(E, B, \pi)$ to another:

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

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 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, while doing the same with *saudade* is going to take some forcing.

A slight digression to further clarify commutative diagrams: one way to nail down the 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 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:

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

There is information in the pairing, and there is 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.

These examples were at the set level, but one could do the same at the category level. I'd used the word *natural* to describe the procedure to map across languages, then find in dictionary; or identically find in dictionary then map across languages. 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*.

Further, all these commutative diagrams chain together:

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

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.

## 4.1 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, maybe 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, this 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.

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.

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.