

Understanding Natural Transformations

Brendan Good

2018-04-11 Wed 19:44

Natural transformations act as the gatekeeper to the rest of category theory. Once you understand them, adjoint functors and even the (in)famous monads unveil themselves. Even the "correct" notion of two categories being equivalent (unsurprisingly called an equivalence of categories) relies on natural transformations.

Despite this notion being critical, I've only really heard two explanations of them; both are one sentence long and both are unenlightening:

- Natural transformations are "coordinate free transformations" (what does that even mean)
- A natural transformation is a "homomorphism of functors" (this is a bit too high-level to make sense to me "out of the box")

As is often the case for me, the best approach was to stare at the diagram and figure out the intuition for myself.

In this post, I'll talk about my intuition for natural transformations and, roughly, how I came to it. Although the understanding I ultimately arrived at is pretty much identical to the "homomorphism of functors" viewpoint, this post unwinds that analogy a bit more.

What is a homomorphism, really?

At its core, a homomorphism is just a map which preserves the structure of its argument. For example, a group homomorphism is a map which sends groups to groups, a "category homomorphism" (functor) sends categories to categories, etc. More generally, for an arbitrary category C , if there exists a morphism $f : X \rightarrow Y$, where $X, Y \in \text{ob}(C)$ ¹ then we generally think of Y

¹Here I take $\text{ob}(C)$ to mean the objects of the category C .

as having the same structure as X within C . Even though that last point sounds like a needless abstraction, that insight will be useful later.

With this in mind, what is the structure of a functor? It's a strange question with a straightforward answer: you apply it. You give me an object $X \in ob(C)$, I'll give you an object $F(X) \in ob(D)$. Similarly, you give me a morphism $f : X \rightarrow Y$ where $X, Y \in ob(C)$, I'll give you a morphism from $F(f) : F(X) \rightarrow F(Y)$. In short, we will show that a natural transformation is the correct notion of one functor having the same structure as another.

Natural transformations

Definition

A natural transformation between two functors $F, G : C \rightarrow D$ associates to each object $X \in ob(C)$ a morphism $\eta_X : F(X) \rightarrow G(X)$ such that the following diagram commutes (which is a 90 degree rotation from the one listed on Wikipedia).

$$\begin{array}{ccc} F(X) & \xrightarrow{\eta_X} & G(X) \\ F(f) \downarrow & & \downarrow G(f) \\ F(Y) & \xrightarrow{\eta_Y} & G(Y) \end{array}$$

Where $f \in mor(X, Y)$ ²

The intuition

How does this construction have those properties I talked about earlier? If I have two functors F and G , I can apply it them to an object X and get two objects from D , the top arrow says that we require a morphism from $\eta_X : F(X) \rightarrow G(X)$, which is to say that $F(X)$ and $G(X)$ have the same structure (whatever that means in our particular category).

That is only half the story, however. We also want to make sure that morphisms $F(f)$ and $G(f)$ have the same structure as well; but first, we have to define what that even means.

²Here I take $mor(X, Y)$ to mean the morphisms between the objects X and Y in the category C .

Structure of morphisms

Consider two morphisms $f : X \rightarrow Y$ and $g : Z \rightarrow W$ where $X, Y, Z, W \in \text{ob}(C)$. What is a reasonable criterion to say that g has the same structure as f ? A morphism consists broadly of two things, a domain and a codomain. What does it mean to preserve the structure of the domain and codomain?

Recall that in a category C , object Y "has the same structure" as X if there exists a morphism $f : X \rightarrow Y$. With that in mind, we might say that a reasonable definition would be that two morphisms have the same structure such that the domains and the codomains have the same structure (since the domain and codomain are just objects). In other words, we want a diagram that looks like this (not necessarily commutative):

$$\begin{array}{ccc} X & \xrightarrow{j} & Z \\ f \downarrow & & \downarrow g \\ Y & \xrightarrow{k} & W \end{array}$$

Diagram A

I want to re-emphasize that, at this point, we do **not** require that this diagram be commutative (that is to say $g \circ j$ does not necessarily equal $k \circ f$).

This looks fine at first glance, since we have that the domain and the codomain have the same structure, but consider the following (silly) example:

$$\begin{array}{ccc} \mathbb{Z} & \xrightarrow{x \mapsto x} & \mathbb{Z} \\ x \mapsto x \pmod{2} \downarrow & & \downarrow x \mapsto x \pmod{3} \\ \mathbb{Z}/2\mathbb{Z} & \xrightarrow{0} & \mathbb{Z}/3\mathbb{Z} \end{array}$$

If we compose the top arrow with the right arrow, we get the morphism $x \mapsto x \pmod{3}$. However, when we compose the left arrow with the bottom arrow, we get the morphism $x \mapsto 0$. Because these resulting morphisms are different, it is possible to recover whether we mapped over via the domain or the codomain depending on what the resulting morphism is. This indicates that the domain and codomain somehow have a different structure in this case.

So what we really want is to be able map from either the domain or the codomain and achieve the same morphism. Therefore, we say that g has the same structure as f if diagram A (above) is commutative.

Intuition wrap-up

Therefore, a natural transformation is just a way of saying that one functor maps objects and morphisms in a way that has the same structure as another functor.

An odd example that helped it click for me

At a Haskell meetup, the organizer (Gershom Bazerman) said "When people ask me for references on category theory, I say 'you always learn category theory from the second book you end up reading'"; I suspect that there may be an element of that here. Nonetheless, even though the example I'm about to provide is not a natural transformation, it helped me further understand "functions that preserve the structure of other functions".

Chain complexes

A chain complex C_\bullet is a collection of R – modules³ C_i along with module homomorphisms $d_i : C_i \rightarrow C_{i-1}$ such that $d_{i-1} \circ d_i = 0$

$$\dots \xrightarrow{d_{n+2}} C_{n+1} \xrightarrow{d_{n+1}} C_n \xrightarrow{d_n} C_{n-1} \xrightarrow{d_{n-1}} \dots$$

In particular, the kernel of d_{i-1} is a submodule of the image of d_i (since $d_{i-1} \circ d_i = 0$), one of the things that we're interested in when we study chain complexes is the kernel of d_{i-1} (called cycles) and the image of d_i (called boundaries). In particular, we care about how much bigger the kernel is than the image; in other words, we want to know the extent to which this diagram is not exact.

From now on, I'll simply refer to the morphisms d_i as simply d .

Morphisms of chain complexes

A morphism of chain complexes from $C_\bullet \rightarrow D_\bullet$ is a collection of morphisms $u_i : C_i \rightarrow D_i$ such that the following diagram commutes:

$$\begin{array}{ccccccc} \dots & \xrightarrow{d} & C_{n+1} & \xrightarrow{d} & C_n & \xrightarrow{d} & C_{n-1} & \xrightarrow{d} & \dots \\ & & \downarrow u_{n+1} & & \downarrow u_n & & \downarrow u_{n-1} & & \\ \dots & \xrightarrow{d} & D_{n+1} & \xrightarrow{d} & D_n & \xrightarrow{d} & D_{n-1} & \xrightarrow{d} & \dots \end{array}$$

³If you aren't familiar with modules, replace all instances with "R-module" with "vector space" or "abelian group".

It can be proven via diagram chasing that u sends cycles to cycles and boundaries to boundaries; which is to say, chain complex morphisms preserve precisely the structure that we're interested in.

Even though natural transformations may be intimidating at first, it is simply a way to say that two functors have the same structure; I hope this explanation has been helpful!