Category Theory

Sandy Maguire

February 18, 2018

Abstract

My notes summarizing Awodey for the purposes of learning Category Theory. It's going to be a great project.

Contents

1 Foundations

1.1 Definition of a Category

A category $\mathscr{C}$ consists of **objects** and **arrows** between them. To be precise, every arrow has a domain and a codomain, both of which are objects in the category. In addition, every object $X \in \mathscr{C}$ has an **identity arrow** $1_X : X \to X$.

If $f : A \to B$, we say $dom(f) = A$ and $cod(f) = B$.

In addition, to be a category, $\mathscr{C}$ must respect the following laws:

1. Composition: If $f : A \to B$ and $g : B \to C$, there exists an arrow $g \circ f : A \to C$.
2. Associative: $f \circ (g \circ h) = (f \circ g) \circ h$.
3. Identity: $1_B \circ f = f = f \circ 1_A$, given $f : A \to B$

Anything that satisfies these laws is a category. It need not correspond to our intuitions that "arrows are functions" or any such silliness.

1.2 Definition of a Functor

A functor $F : \mathscr{C} \to \mathscr{D}$ is a mapping of objects in $\mathscr{C}$ to objects in $\mathscr{D}$, and likewise for arrows. A functor is a homomorphism across domains, codomains and compositions.

That is to say:

1. $F(f : A \to B) = F(f) : F(A) \to F(B)$
2. $F(g \circ f) = F(g) \circ F(f)$
3. $F(1_A) = 1_{F(A)}$

There is an identity functor $1_{\mathscr{C}} : \mathscr{C} \to \mathscr{C}$, and because functors compose, we have a category of categories: $\mathscr{Cat}$.

1.3 Definition of an Isomorphism

An arrow $f : A \to B$ is called an isomorphism if there exists an arrow $g : B \to A$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$.

Theorem 1. *Isomorphisms are unique.*

Proof. For $f : A \to B$ to be an isomorphism, we must have $g : B \to A$. Assume that the isomorphism is not unique, and thus that we also have $g' : B \to A$.

By definition, we have $g \circ f = 1_A$. We can compose on both sides to get $g \circ f \circ g' = 1_A \circ g'$, but recall that g' is an isomorphism, therefore $g \circ 1_B = 1_A \circ g'$. We can emit the identities, and thus $g = g'$. □

Definition 1. *A **group** is a single object category where every arrow is an isomorphism.*

Theorem 2. *Every group G is isomorphic to a group of permutations.*

Proof. Define $f_g(x) = g \times x$ for $g \in G$. Since G is a group, we also have $f_{g^{-1}}(x) = g^{-1} \times x = f_g^{-1}(x)$ which means $(f_g \circ f_g^{-1})(x) = (f_g^{-1} \circ f_g)(x) = x$. Therefore f_g forms a group.

Consider now a function $T : G \to \bar{G}$ where $T(g) = f_g$. T is a group homomorphism because $(f_g \circ f_h)(x) = f_g(f_h(x)) = f_g(h * x) = g * (h * x) = f_{g*h}(x)$.

This is true for any x, therefore $T(g) \circ T(h) = f_g \circ f_h = f_{g*h} = T(g * h)$ □

2 Constructions on Categories

2.1 Product Category

The product of categories $\mathscr{C}$ and $\mathscr{D}$ is $\mathscr{C} \times \mathscr{D}$. It's objects are the cartesian product of objects in $\mathscr{C}$ and $\mathscr{D}$. Arrows are likewise defined in this matter, with composition and units being defined component-wise:

$$1_{(C,D)} = (1_C,\ 1_D)$$
$$(f', g') \circ (f, g) = (f' \circ f,\ g' \circ g)$$

There are also projection functors $\pi_1 : \mathscr{C} \times \mathscr{D} \to \mathscr{C}$ and $\pi_2 : \mathscr{C} \times \mathscr{D} \to \mathscr{C}$ defined in the obvious way.

2.2 Arrow Category

The arrow category $\mathscr{C}^{\to}$ has objects which are arrows in $\mathscr{C}$ and its arrows are commutative squares in $\mathscr{C}$.

For example, given $f,\ f',\ g_1,\ g_2 \in \mathscr{C}$, there is an arrow $g : f \to f' \in \mathscr{C}^{\to}$ such that $g = (g_1,\ g_2)$ if the follow diagram exists in $\mathscr{C}$:

$$\begin{array}{ccc} A & \xrightarrow[g_1]{} & A' \\ \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle f'} \\ B & \xrightarrow[g_2]{} & B' \end{array}$$

Composition acts as you'd expect. Given $h \circ g$, we get the diagram:

$$\begin{array}{ccccc} A & \xrightarrow[g_1]{} & A' & \xrightarrow[h_1]{} & A'' \\ \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle f'} & & \downarrow{\scriptstyle f''} \\ B & \xrightarrow[g_2]{} & B' & \xrightarrow[h_2]{} & B'' \end{array}$$

In other words, there is an arrow $f \to f'$ iff $g_2 \circ f = f' \circ g_1$. But what does this mean?

In a monoid, composition means concatenation, and thus over a free monoid of the alphabet, $f = \text{art}$, $f' = \text{far}$, $g = (\text{t}, \text{f})$ because $\text{f} \circ \text{art} = \text{far} \circ \text{t}$. With the monoid of addition over the natural numbers, the arrow category is complete; all objects have infinite arrows between them because there are infinite ways of adding two numbers to two numbers and having them equate.

more examples plz

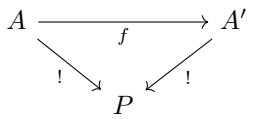
2.3 Slice Category

what are the functors?

The slice category $\mathscr{C}/X$, given $X \in \mathscr{C}$ is a special case of the arrow category when $B = B' = X$.

2.3.1 Principal Ideal

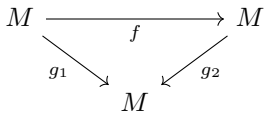
Awodey gives the example of the slice $\mathscr{P}/P$ over a poset category for some $P \in \mathscr{P}$, that this is isomorphic to the principal ideal $\downarrow(P)$. What the heck does this mean? Well, let's draw the diagram:



In a poset category, we can consider an arrow to be $\leq$, therefore, in $\mathscr{P}/P$ there is an arrow $f : A \to A'$ whenever $A \leq A' \leq P$. Therefore, the principal ideal $\downarrow(P)$ is just the subset of the poset that is $\leq P$.

2.3.2 Slice of a Monoid

Out of curiosity, let's look at the slice $\mathscr{M}/M$ over some monoidal category where $M \in \mathscr{M}$.



We can pull equations out of this diagram, namely that $g_1 = g_2 \circ f$. The arrows in $\mathscr{M}/M$ are therefore the elements which can be "decomposed" into the concatenation of two elements. However, because $\mathscr{M}$ is a monoidal category, all elements can be decomposed (by factoring out a unit). Therefore, $\mathscr{M}/M \cong \mathscr{M}$.

2.3.3 Coslice Category

We can define the coslice category $X/\mathscr{C}$, given $X \in \mathscr{C}$ by looking at $(\mathscr{C}/X)^{Op}$. This is obviously a special case of the arrow category when $A = A' = X$.

The coslice of a poset is its $\uparrow(X)$, and the coslice of a monoidal category is still isomorphic to the category itself.

Awodey points out that $1/\mathscr{Set}$ is isomorphic to the category of pointed sets, where each set has a distinguished member called the "point" (recall that Maybe is the free pointed set). Arrows in $1/\mathscr{Set}$ are homomorphisms which preserve the points.

2.4 Free Monoids

Given a set $S = \{s_1, s_2, \ldots, s_n\}$ of "letters", define $S^* = \{\text{words over } S\}$, in other words, every finite sequence of letters in S. The empty word can be denoted as -. S^* forms a monoid under concatenation with - as its unit.

We have an insertion function $i(s) = s : S \hookrightarrow S^*$. The elements of S generate S^*, because every $s \in S^*$ can be made up of some concatenation of $s_{i1}s_{i2}\ldots s_{in}$.

S^* is said to be the free monoid over S because it generates a monoid even when S itself is not one.

But we can state this definition more "categorically" by way of a *universal mapping property*.

The free monoid $M(A) \in \mathscr{Mon}$ over $A \in \mathscr{Set}$ is the object with the property that for any $N \in \mathscr{Mon}$ the following diagram holds:

$$M(A) \overset{!}{\dashrightarrow} N$$

Theorem 3. $M(A) = A^*$

Proof. We can prove the theorem by giving a program in Haskell.

```
foldMap :: Monoid n => (a -> n) -> [a] -> n
foldMap _ []       = mempty
foldMap f (a : as) = f a `mappend` foldMap f as
```

□

Awodey says something about how the monoid of natural numbers under addition is isomorphic to the free monoid over a single element set. This is obviously true, but I'm not entirely sure what his broader argument is.

page 28 – determine what's going on here

2.4.1 Forgetful Functors

For every structured set Z (of category $\mathscr{X}$) there is an underlying set $|Z|$, and likewise for every homomorphism f over the structured set, there is a corresponding homomorphism over sets $|f|$. Therefore we have a functor $U : \mathscr{X} \to \mathscr{Set}$ called the forgetful functor.

2.5 Free Categories

Just like how $A^* \in \mathscr{Mon}$ is the free monoid over $A \in \mathscr{Set}$, $Cat(G) \in \mathscr{Cat}$ is the free category over $G \in \mathscr{Graph}$ (the category of directed graphs.)

A $Cat(G)$ can be generated from G by taking every vertex $V \in G$ and making it an object $V \in Cat(G)$. Every path P made up of a finite sequence of edges $E_1, E_2, \ldots E_n \in G$ (where the target of E_i is the source of E_{i+1}) becomes an arrow in $Cat(G)$. Add the mandatory $1_V : V \to V$ identities, and you have yourself a category where composition of arrows is subsequent traversals of paths in the underlying graph.

There is also *universal mapping property* for free categories, namely that for any category $\mathscr{C} \in \mathscr{Cat}$:

$$Cat(G) \overset{!}{\dashrightarrow} \mathscr{C}$$

prove this

2.6 More Foundations

A category is said to be "small" iff its objects and arrows both form sets. Otherwise it is "large."

This implies that $\mathscr{Set}$ and any other structured set categories are large, and that $\mathscr{Cat}$ is itself only the category of small categories – and thus does not contain itself.

However, a category $\mathscr{C}$ can be said to be "locally small" iff for any objects $X, Y \in \mathscr{C}$, the set of arrows $\{f \in \mathscr{C} \mid f : X \to Y\}$ is a set. $\mathscr{Set}$ is locally small because there is a set of all functions from $X \to Y$, Y^X. Likewise, all other structured sets share this property, since they are necessarily more restrictive than $\mathscr{Set}$.

Awodey asks whether $\mathscr{Cat}$ is locally small. My intuition is yes, because all of its objects are themselves small categories; therefore the collection of functors from one small category to another must form a set.

2.7 Exercises

Exercise 1. *The objects of $\mathscr{Rel}$ are sets, and an arrow $f : A \to B$ is a relation from A to B, that is, $f \subseteq A \times B$. The identity relation $\{\langle a, a\rangle \in A \times A \mid a \in A\}$ is the identity arrow on a set A. Composition in $\mathscr{Rel}$ is to be given by:*

$$g \circ f = \{\langle a, c\rangle \in A \times C \mid \exists b. \langle a, b\rangle \in f \;\&\; \langle b, c\rangle \in g\}$$

Show that $\mathscr{Rel}$ is a category.

Proof. We need to show that $\langle a, a\rangle$ is an identity, and that $f \circ (g \circ h) = (f \circ g) \circ h$.

Given $f : A \to B$, we can show that $1_a = \langle a, a\rangle$ is an identity:

$$\begin{aligned} f \circ 1_a &= \{\langle a, b\rangle \in A \times B \mid \exists x. \langle x, x\rangle \in 1_a \;\&\; \langle x, b\rangle \in f\} \\ &= \{\langle a, b\rangle \in A \times B \mid \langle a, a\rangle \in 1_a \;\&\; \langle a, b\rangle \in f\} \\ &= \{\langle a, b\rangle \in A \times B \mid \langle a, b\rangle \in f\} \\ &= f \end{aligned}$$

The proof that 1_b is a left identity proceeds in exactly the same way. Therefore, the identity relation is in fact an identity on the arrows.

Finally, we need to show that composition is associative. This proceeds immediately from the existential in the definition of composition which asserts an equality between the (set) codomain of f and the domain of g. Because equality is associative, composition in $\mathscr{Rel}$ must too be. □

Exercise 2. *Determine which of the following isomorphisms hold:*

1. $\mathscr{Rel} \cong \mathscr{Rel}^{op}$
2. $\mathscr{Set} \cong \mathscr{Set}^{op}$
3. *For a fixed set X with powerset $P(X)$, as poset categories $P(X) \cong P(x)^{op}$ (the arrows in $P(X)$ are subset inclusions $A \subseteq B$ for $A, B \subseteq X$)*

Proof.

1. $\mathscr{Rel} \cong \mathscr{Rel}^{op}$ because arrows compose due to an existential equality. Since equality is dual to itself, these two categories must be isomorphic.
2. $\mathscr{Set} \ncong \mathscr{Set}^{op}$ because in the dual, initial objects in $\mathscr{Set}$ are mapped to terminal objects in $\mathscr{Set}^{op}$. Since isomorphisms need to preserve initial objects (and other interesting features of a category), these two categories are not isomorphic.
3.

□

Apparently this is in fact isomorphic to its dual, but I don't know why because it seems like the $\mathscr{Set}$ argument should apply here as well.

3 Abstract Structures

3.1 Epis and Monos

A monomorphism (mono) is any morphism $f : A \rightarrowtail B$ such that for any $g, h : C \to A$ the following is true: $fg = fh \implies g = h$. In other words, a mono is always left-cancelable.

$$C \underset{h}{\overset{g}{\rightrightarrows}} A \xrightarrow[f]{} B$$

Dually, an epimorphism (epi) is a morphism $f : A \twoheadrightarrow B$ such that for any $i, j : B \to D$ the following is true: $if = jf \implies i = j$. An epi is always right-cancelable.

$$A \xrightarrow[f]{} B \underset{j}{\overset{i}{\rightrightarrows}} D$$

In $\mathscr{Set}$, monos correspond to injective functions, and epis are surjective functions. Furthermore, in $\mathscr{Set}$ a function which is monic and epic is an iso, but this is **not true** in general.

Theorem 4. *In a poset $\mathscr{P}$, every arrow $p \leq q$ is both monic and epic.*

Proof. In a poset, there is exactly one arrow between any two objects. Therefore if two arrows are equal after composing with a third, they must have been equal to begin with. □

Theorem 5. *Every iso is a mono and an epi.*

Proof. Given an iso $m : A \to B$ with inverse $e : B \to A$:

$$\begin{aligned} mx &= my \\ emx &= emy \\ 1_B x &= 1_B y \\ x &= y \end{aligned}$$

And dually to show epicness. □

3.2 Initial and Terminal Objects

An object $0 \in \mathscr{C}$ is said to be initial if for every other object $X \in \mathscr{C}$ there is a unique arrow $! : 0 \to X$. Dually, a terminal object is one which has a unique arrow coming into it from every object in the category.

Theorem 6. *Initial and terminal objects are unique up to isomorphism.*

Proof. Assume we have two initial objects, X and Y. In order to be initial, they must have arrows $y : X \to Y$ and $x : Y \to X$. Furthermore, we know that there is exactly one arrow from an initial object to any other – including itself. By the category laws, this unique arrow must be 1, which makes the following diagram commute:

$$\begin{array}{ccc} X & \xrightarrow[y]{} & Y \\ & {\scriptstyle 1_X}\searrow \quad \downarrow{\scriptstyle x} \quad \searrow{\scriptstyle 1_Y} & \\ & X \xrightarrow[y]{} Y & \end{array}$$

This argument dualizes to terminal objects.

□

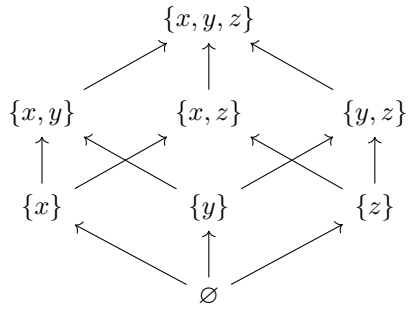
3.3 Generalized Elements

3.3.1 Boolean Algebras

A Boolean algebra is a poset B with initial and terminal objects, products, coproducts, and complements, with the laws:

1. $0 \leq a$ (initial object)
2. $a \leq 1$ (terminal object)
3. $a \leq c, b \leq c \iff a \vee b \leq c$ (coproducts)
4. $c \leq a, c \leq b \iff c \leq a \wedge b$ (products)
5. $a \leq \neg b \iff a \wedge b = 0$ (complement)
6. $\neg\neg a = a$ (complement)

Consider the case of a lattice:



We can define a complement $\neg x$ as $\{a \mid a \in B, a \notin x\}$. This clearly satisfies the $\neg\neg x = x$ law, and some playing around with the $a \leq \neg b \iff a \wedge b = 0$ law seems true, although I don't have a proof for it.

3.3.2 Ultrafilters

A filter F in a boolean algebra $B \in \mathscr{Bool}$ is a subset of B which is closed upwards and under meets (products). Which is to say:

1. $a \in F,\ a \leq b \implies b \in F$ (closed upwards)
2. $a \in F,\ b \leq F \implies a \wedge b \in F$ (closed under meets)

An ultrafilter is a filter F which is maximally big, ie. $F \subset F' \implies F' = B$.

Theorem 7. *F is an ultrafilter iff for all $x \in B$, either $x \in F$ or $\neg x \in F$, but not both.*

Proof. Because a filter is closed upwards, any filter which contains 0 must equal to B itself, and thus not an ultrafilter.

If $x \in F$ and $\neg x \in F$, then $x \wedge \neg x \in F$ by filters being closed under meets. However, $x \wedge \neg x = 0$, therefore, no ultrafilter can contain x and $\neg x$.

What if neither $x \in F$ nor $\neg x \in F$? This is true just if $F = \varnothing$, which is obviously not maximal. □