Contents

0.1 (Review) Lecture -3: Basic category theory

Several people said they wanted to learn basic category theory. We already needed to review some category theory for this seminar, so we'll take the opportunity to review enough theory to hopefully help everyone.

The short-term plan is to review some category theory, Grothendieck sites, and sheaves. Then we will start actually talking about condensed things. I only reluctantly call this "review," since we are *not* assuming everyone is familiar with all the material moving forward. The goal is only to catch folks up to a working knowledge.

Probably everything we say here is said better somewhere in Emily Riehl's *Category Theory in Context*. That book also has many examples. You should read it.

Definition 0.1.1. A *category* C consists of the following data.

(1) A collection $Ob(\mathsf{C})$ we call *objects*.

(2) A collection $Mor(\mathsf{C})$ we call *morphisms*.

(3) For each morphism f, a *source* and *target* object. (We write $f : X \to Y$ to express that f is a morphism with source X and target Y.)

(4) For each object X, a distinguished morphism $\mathrm{id}_X : X \to X$ we call the *identity morphism*.

(5) For each pair of morphisms $f : X \to Y, g : Y \to Z$ such that $target(f) = source(g)$, a distinguished morphism $gf : X \to Z$ we call the *composite morphism*.

And this data must satisfy the following properties.

- Given a morphism $f : X \to Y$, we have $f = \mathrm{id}_Y f = f \mathrm{id}_G$.
- Given morphisms f, g, h, we have $(fg)h = f(gh)$ (when the source/targets match appropriately).

In practice, we think of categories as "like a collection of objects and maps between them, with all the structure that should accompany the word *maps*—identity self-maps, composites, associativity."

0.1.1 Examples of categories

(See Riehl for more)

Example 0.1.2. The category of sets Set has sets as objects and functions as morphisms. The rest of the structure is "obvious": source/target is domain/range, the composition of morphisms is defined as the composition of functions, and identity morphisms are the identity functions.

Example 0.1.3. The category of spaces Top has topological spaces as objects and continuous maps as morphisms. Again, the rest of the structure is "obvious."

Example 0.1.4. Define the category Top_* to have *based spaces*[1] as objects and *based maps*[2] as morphisms.

Example 0.1.5. Define the category Grp to have groups as objects and homomorphisms as morphisms.

Example 0.1.6. Define the category Ab to have abelian groups as objects and homomorphisms as morphisms.

Example 0.1.7. Denote by k a field (e.g., $k = \mathbb{R}$). Define the category $Vect_k$ to have k-vector fields as objects and k-linear maps as morphisms.

Example 0.1.8 (Morphisms do not have to be functions!)**.** Define a category $\mathsf{Naturals}$ to have

[1] A based space is pair (X, x) where X is a space and $x \in X$.

[2] A based map $f : (X, x) \to (Y, y)$ is a continuous map $f : X \to Y$ such that $f(x) = y$.

- As objects, the natural numbers $Ob(\mathsf{Naturals}) := \{0, 1, 2, \dots\}$; and
- A morphism $a \to b$ for each pair of numbers (a, b) such that $a \leq b$.

Thus, given objects $a, b \in \{0, 1, 2, \dots\}$, there is at most one arrow $a \to b$, and it exists iff $a \leq b$.

Example 0.1.9 (Morphisms do not have to be functions!)**.** Define a category $\mathsf{Skel}(\mathsf{FinSet}_{\mathsf{iso}})$ to have

- Natural numbers n as objects; and
- A morphism $n \to n$ for each element of the symmetric group Σ_n, and no morphisms $m \to n$ when $m \neq n$.

Example 0.1.10 (Example from Peter's talk)**.** A *poset* is a set S with a relation $\leq$ that is reflexive, transitive, and antisymmetric. A *morphism of posets* $f : (S, \leq) \to (S', \leq')$ is a function $f : S \to S'$ that respects the partial orderings, i.e. $x \leq y \iff f(x) \leq' f(y)$. We denote by $Poset$ the category of posets and morphisms of posets.

0.1.2 Isomorphisms

All sorts of objects—groups, rings, sets, spaces—have a notion of "sameness." In an arbitrary category, we formalize this notion as *isomorphisms*.

Definition 0.1.11. Let C be a category. Suppose that $f : c \to c'$ is a morphism and that there exists a morphism $g : c' \to c$ such that $fg = \mathrm{id}_{c'}$ and $gf = \mathrm{id}_c$. Then we say f and g are *isomorphisms* and we say that c, c' are *isomorphic*.

Example 0.1.12. Isomorphisms in Set are bijections. Isomorphisms in Top are homeomorphisms. Isomorphisms in Top_* are based homeomorphisms. Isomorphisms in Grp and Ab are group isomorphisms.

Exercise: Figure out what the isomorphisms in $Poset$ are.

0.1.3 Functors

Definition 0.1.13. Let C, D be categories. A *functor from* C *to* D (which we write $F : \mathsf{C} \to \mathsf{D}$) is "a map of objects and morphisms that preserves categorical structure, i.e. sources, targets, composites, and identities." Formally, it consists of the following data.

(1) An object $FX \in \mathsf{D}$ for each object $X \in \mathsf{C}$.

(2) For each morphism $f : X \to Y$ in C, a morphism $Ff : FX \to FY$ in D.

And this data must satisfy the following properties.

- For any pair of composable morphisms f, g in C, we have $F(gf) = F(g)F(f)$.
- For any identity morphism id_X in C, we have $F(\mathrm{id}_X) = \mathrm{id}_{FX}$.

Very often, we say "(one type of object) are the same thing as (another type of objects)." Categories give us a great, concrete way to talk about "types of objects." Functors give us a way to "modify and compare" objects of different types. Can functors tell us when (one type of object) are "the same thing as" (another type)? Yes, and this is a very useful notion.

For the following definition, we will denote by $Mor(X, Y)$ the set of morphisms $X \to Y$ between objects $X, Y \in \mathsf{C}$.

Definition 0.1.14. Let C, D be categories. An *equivalence of categories* is a functor $F : \mathsf{C} \to \mathsf{D}$ that is

1. *Full*: for every pair of objects $X, Y \in \mathsf{C}$, the mapping $f \mapsto F(f)$ defines a surjection $Mor(X, Y) \to Mor(FX, FY)$;

2. *Faithful*: for every pair of objects $X, Y \in \mathsf{C}$, the mapping $f \mapsto F(f)$ defines an injection $Mor(X,Y) \to Mor(FX, FY)$; and
3. *Essentially surjective*: for every object $d \in \mathsf{D}$, there exists some $c \in \mathsf{C}$ such that $Fc \cong d$.

Remark 0.1.15. The analogy is, “full and faithful is like injectivity" and “essentially surjective is like surjectivity." If you have both, you have an isomorphism. Note that a full and faithful functor need not actually be surjective *on objects*.

Remark 0.1.16. There are other common, equivalent definitions of an equivalence of categories.

0.1.4 Examples of functors

(see also Riehl, p. 13)

Example 0.1.17. Let C denote one of the categories $\mathsf{Top}, Grp, \mathsf{Ab}$, or $Vect_k$. We can define a functor $U : \mathsf{C} \to \mathsf{Set}$ by mapping objects to their underlying sets and morphisms to their underlying set-maps. We call U the *forgetful functor*. (We generally refer to any functor that “tosses out" structure, e.g. a topology on a set, as a *forgetful functor*.)

Example 0.1.18. If C is any category, then we can form its *opposite category* C^{op} to have the same objects but with “flipped arrows," i.e. swapped source/targets of C’s morphisms. There is a functor $\mathsf{C} \to \mathsf{C}^{\mathrm{op}}$ that takes objects to themselves and morphisms to their “flip."

Exercise: prove that $(\mathsf{C}^{\mathrm{op}})op$ is equivalent to C.

Example 0.1.19. For $V \in Vect_k$, recall that its *dual* is defined as the vector space $V^* := \{\text{linear maps } V \to k\}$. Given a linear map $f : V \to W$, there is induced a map $f^* : W^* \to V^*$ that sends $v : W \to k$ to $v \circ f : V \to k$. The mapping $V \mapsto V^*, f \mapsto f^*$ defines the *dualization functor* $(-)^* : Vect_k \to Vect_k^{\mathrm{op}}$.

Remark 0.1.20. You have heard probably heard that we have an isomorphism $V \cong V^{**}$ that is “canonical" or “natural" or “very nice," but that we do not have such an isomorphism $V \cong V^*$. (Although the two are isomorphic.) This can be expressed very concretely as a statement about the functor $(-)^*$ and its self-composite $(-)^{**}$. We do not yet have the language for this (natural transformations); the non-categorical reason is that an isomorphism $V \cong V^*$ requires a choice of basis, but there is an isomorphism $V \cong V^{**}$ that does not need any choice.

Many—and historically, the motivating—examples of functors come from algebraic topology.

Example 0.1.21. Let (X, x) be a based space (i.e., X is a space and $x \in X$). We define the *fundamental group* $\pi_1(X, x)$ as the set of based continuous maps $\ell : [0,1] \to X$ such that $\ell(0) = \ell(1) = x$, modulo homotopy equivalence. The group structure is loop concatenation: given $\ell, \ell' : [0,1] \to X$, define $\ell'\ell : [0,1] \to X$ to do one loop over $[0.5]$ then the other over $[0.5, 1]$. Given a based map $f : (X,x) \to (Y,y)$, there is induced a map $f_* : \pi_1 X \to \pi_1 Y$ given by $\ell \mapsto f \circ \ell$. This defines a functor

$$\pi_1(-) : \mathsf{Top}_* \to Grp.$$

Example 0.1.22. For each n, singular homology defines a functor $H_n(-) : \mathsf{Top} \to \mathsf{Ab}$. Similarly, singular cohomology defines a functor $H^n(-) : \mathsf{Top}^{\mathrm{op}} \to \mathsf{Ab}$.

0.1.5 Natural transformations

We often want to compare *functors*. This will help us explain why e.g. “an arbitrary vector space is not *naturally* isomorphic to its dual" but “an arbitrary vector space *is* naturally isomorphic to its double-dual."

There are more serious examples where we *really* care about comparisons between functors. For example, the *Hurewicz homomorphism* from algebraic topology is a comparison $h_X : \pi_n(X) \to H_n(X)$ for every space X. But in fact, more can be said—for any continuous based map $f : X \to Y$, the Hurewicz homomorphism satisfies $h_Y \circ \pi_n(f) = H_n(f) \circ h_X$. (Here, $\pi_n(f)$ and $H_n(f)$ are the maps induced by f on pi_n and H_n.) This is a seriously useful fact that is not "formally guaranteed" to be true. One might phrase this as, "the Hurewicz homomorphism compares objects $\pi_n(X) \to H_n(X)$ in a way that respects how maps induce homomorphisms via the functors $\pi_n(-), H_n(-)$."

Natural transformations give a simple way to express this.

Definition 0.1.23. Let $F, G : \mathsf{C} \to \mathsf{D}$ be two functors. A *natural transformation from F to G*, which we denote as $\alpha : F \implies G$, is the data of

- For each object $c \in \mathsf{C}$, a morphism $\alpha_c : F(c) \to G(c)$ in D

such that for every morphism $f : c \to c'$ in C, one has $G(f) \circ \alpha_c = \alpha'_c \circ F(f)$. In other words, the following diagram commutes.

$$\begin{tikzcd}
F(c) \arrow[r, "F(f)"] \arrow[d, "\alpha_c"] & F(c') \arrow[d, "\alpha'_c"] \\
G(c) \arrow[r, "G(f)"] & G(c')
\end{tikzcd}$$