

Category theory course
Lecture 4
ITI9200, Spring 2020

Fosco Loregian
fouche@yoneda.ninja

February 16, 2020

Contents

1	Functors	1
1.1	Morphisms of categories	1
1.2	Examples of functors	2
1.3	Diagrams or: functors as pictures	4
1.4	General definition of limits and colimits	6
1.5	Examples of limits and colimits	9

1 Functors

1.1 Morphisms of categories

Definition 1 (Functor). *Let \mathcal{C} and \mathcal{D} be two categories; we define a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ as a pair (F_0, F_1) consisting of the following data:*

- F_0 is a function $\mathcal{C}_o \rightarrow \mathcal{D}_o$ sending an object $C \in \mathcal{C}_o$ to an object $FC \in \mathcal{D}_o$;
- F_1 is a family of functions $F_{AB} : \mathcal{C}(A, B) \rightarrow \mathcal{D}(FA, FB)$, one for each pair of objects $A, B \in \mathcal{C}_o$, sending each arrow $f : A \rightarrow B$ into an arrow $Ff : FA \rightarrow FB$, and such that:
 - $F_{AA}(1_A) = 1_{FA}$;
 - $F_{AC}(g \circ_C f) = F_{BC}(g) \circ_D F_{AB}(f)$.

```

-- | All instances of the `Functor` type-class must satisfy two laws.
-- These laws are not checked by the compiler.
--
-- * The law of identity
--   `forall x. (id <$> x) ~ x`
--
-- * The law of composition
--   `forall f g x. (f . g <$> x) ~ (f <$> (g <$> x))`
class Functor k where
-- Pronounced, eff-map.
(<$>) :: (a -> b) -> k a -> k b

```

If $F : \mathcal{C} \rightarrow \mathcal{D}$ is a functor, it corresponds to a data constructor $f :: * \rightarrow *$ (this is the F_0 part, a *function on objects* from the collection of object \mathcal{C}_o of \mathcal{C} to the collection of objects \mathcal{D}_o ¹) and a *function on morphisms*

```

<$> :: (a -> b) -> f a -> f b

```

If you recall how the associativity of \rightarrow works, it is evident that this is a function from the type of maps $a \rightarrow b$ to the type of maps $f a \rightarrow f b$; or rather, this is a function that given a function $u :: a \rightarrow b$, and an “element” of type $f a$, yields an element $x :: f b$.

Remark 1. *There is a category whose objects are (small) categories, and whose morphisms $\mathcal{C} \rightarrow \mathcal{D}$ are functors. The identity functor $1_{\mathcal{C}} : \mathcal{C} \rightarrow \mathcal{C}$ of a category \mathcal{C} , and the composition $G \circ F : \mathcal{A} \xrightarrow{F} \mathcal{B} \xrightarrow{G} \mathcal{C}$ of two functors are defined in the obvious way; all category axioms follow.²*

1.2 Examples of functors

As you can imagine, both Haskell and mathematics are crawling with examples of functors;

Example 1 (Examples of functors).

1. *Let’s rule out a few edge examples: for every category \mathcal{C} , there is a unique functor $F : \emptyset \rightarrow \mathcal{C}$, where \emptyset is the empty category with no objects and no morphisms; for every category \mathcal{C} , there is an obvious*

¹Although in Haskell, $\mathcal{C} = \mathcal{D}$ is always the same nameless category $*$.

²The reader might now wonder if there is a category of Haskell types; unfortunately there is no such thing, but better languages have a better behaviour in this respect (for example [Agda](#) or [Idris](#)).

definition of the identity functor $1_{\mathcal{C}} : \mathcal{C} \rightarrow \mathcal{C}$, whose correspondences on objects and on arrows $1_{\mathcal{C},AB} : \mathcal{C}(A,B) \rightarrow \mathcal{C}(A,B)$ are all identity functions.

2. As we have hinted in the preceding lesson, all correspondences that forget a structure, yielding for example the underlying set UM of a thingum M , are functors $\mathbf{Thng} \rightarrow \mathbf{Set}$ from their structured categories of definition to the category of sets and functions.
3. All correspondences that regard a structure as an example of another are -more or less tautological- examples of functors: a monoid can be seen as a one-object category, and this yields a functor $\mathbf{B} : \mathbf{Mon} \rightarrow \mathbf{Cat}$ because a functor $F : \mathbf{BM} \rightarrow \mathbf{BN}$ between monoids is but a monoid homomorphism $f : M \rightarrow N$; a preset can be seen as a category with at most one arrow between any two objects, and this defines a functor $^{\times}(_) : \mathbf{Pres} \rightarrow \mathbf{Cat}$, because a functor $F : ^{\times}P \rightarrow ^{\times}Q$ between presets is but a monotone function $f : P \rightarrow Q$; a set can be seen as a discrete topological space A^{δ} or as a category having only identity arrows, and these are functors $\mathbf{Set} \rightarrow \mathbf{Top}, \mathbf{Cat}$. (Continue the list at your will.)
4. Let M be a monoid, and Δ be the category having objects nonempty, finite, totally ordered sets and monotone functions as morphisms. The correspondence that sends $[n] \in \Delta$ to the set $M^{[n]}$ of ordered n -tuples $[a_1 | \dots | a_n]$ of elements of M and a monotone function $f : [m] \rightarrow [n]$ to the function $M^f : M^{[n]} \rightarrow M^{[m]}$ defined by

$$M^f[a_1 | \dots | a_n] = [a_{f1} | \dots | a_{fn}]$$

is a functor $\Delta^{op} \rightarrow \mathbf{Set}$. This functor is called the classifying space of the monoid.

5. Let \mathbf{Set}_* be the category of pointed sets: objects are pairs (A, a) where $a \in A$ is a distinguished element, and a morphism $f : (A, a) \rightarrow (B, b)$ is a function $f : A \rightarrow B$ such that $fa = b$. It is easy to prove that this category is isomorphic to the coslice $*/\mathbf{Set}$ of functions $a : * \rightarrow A$.
6. Let $\partial\mathbf{Set}$ be the category so defined: objects are sets, and a morphism $(f, S) : A \rightarrow B$ is a pair $(S \subseteq A, f : S \rightarrow B)$. This is called a partial function from A to B . It is evident how to define the composition of morphisms, and the identity arrow of $A \in \partial\mathbf{Set}$. There is a functor $(_)_{\bullet} : \partial\mathbf{Set} \rightarrow \mathbf{Set}_*$ defined as $A \mapsto (A \sqcup \{\infty\}, \infty)$, where ∞ is an element that does not belong to A , and a function (f, S) goes to $(f, S)_{\bullet} : A \sqcup \{\infty_A\} \rightarrow B \sqcup \{\infty_B\}$, defined sending $S^c \cup \{\infty_A\}$ to $\{\infty_B\}$.

7. Every directed graph \underline{G} , with set of vertices V and set of edges E , defines a quotient set of V by the equivalence relation \approx generated by the subset of those (A, B) for which there is an arrow $A \rightarrow B$; the symmetric and transitive closure of this relation yields a quotient V/R that is usually denoted as the set of connected components $\pi_0(\underline{G})$. This is a functor $\mathbf{Gph} \rightarrow \mathbf{Set}$, and if we now regard a category \mathcal{C} as a mere directed graph we obtain a well-defined set $\pi_0(\mathcal{C})$ of connected components of a category.

1.3 Diagrams or: functors as pictures

Most of category theory is based on the intuition that a diagram of a certain shape in a category \mathcal{C} is a functor $F : \mathcal{J} \rightarrow \mathcal{C}$; the definition of a diagram itself is engineered to blur the distinction between a functor as a morphism of categories, and a functor as a correspondence that “pictures” a small category \mathcal{J} inside a big category \mathcal{C} . After having recalled the definition of diagram, and the definition of commutative diagram, we collect a series of examples to convince the reader that this identification of “diagrams as pictures” is fruitful and suggestive.

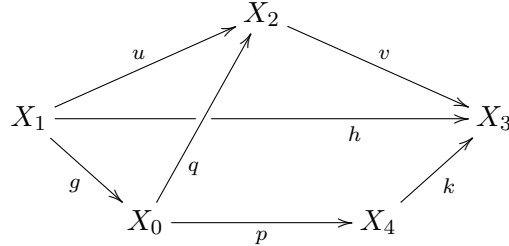
The recommended soundtrack while reading this section is, of course, Emerson, Lake & Palmer’s *Pictures at an Exhibition*.

1.3.1 Diagrams, formally

Arrangements of objects and arrows in a category are called *diagrams*; to some extent, category theory is the art of making diagrams *commute*, i.e. the art of proving that two paths $X \rightarrow A_1 \rightarrow A_2 \rightarrow \cdots \rightarrow A_n \rightarrow Y$ and $X \rightarrow B_1 \rightarrow \cdots \rightarrow B_m \rightarrow Y$ result in the same arrow when they are fully composed.

Remark 2 (Diagrams and their commutation). *Depicting morphisms as arrows allows to draw regions of a given category \mathcal{C} as parts of a (possibly non planar) graph; we call a diagram such a region in \mathcal{C} , the graph whose vertices are objects of \mathcal{C} and whose edges are morphisms of suitable domains*

and codomains. For example, we can consider the diagram



The presence of a composition rule in \mathcal{C} entails that we can meaningfully compose paths $[u_0, \dots, u_n]$ of morphisms of \mathcal{C} . In particular, we can consider diagrams having distinct paths between a fixed source and a fixed ‘sink’ (say, in the diagram above, we can consider two different paths $\mathfrak{P} = [k, p, g]$ and $\mathfrak{Q} = [v, u]$); both paths go from X_1 to X_3 , and we can ask the two compositions $\circ[k, p, g] = k \circ p \circ g$ and $v \circ u$ to be the same arrow $X_1 \rightarrow X_3$; we say that a diagram commutes at $\mathfrak{P}, \mathfrak{Q}$ if this is the case; we say that a diagram commutes (without mention of $\mathfrak{P}, \mathfrak{Q}$) if it commutes for every choice of paths for which this is meaningful.

Searching a formalisation of this intuitive pictorial idea leads to the following:

Definition 2. A diagram is a map of directed graphs (‘digraphs’) $D : J \rightarrow \mathcal{C}$ where J is a digraph and \mathcal{C} is the digraph underlying a category.³ Such a diagram D commutes if for every pair of parallel edges $f, g : i \rightrightarrows j$ in J one has $Df = Dg$.

1.3.2 Diagrams, everywhere

- An object is a diagram: the terminal object in the category of categories is the category

$$[0] = \left\{ \bullet \begin{array}{c} \curvearrowright \\ \curvearrowleft \end{array} 1 \right\}$$

A functor $C : \mathbf{1} \rightarrow \mathcal{C}$ consists of an object of \mathcal{C} .

- An arrow is a diagram: let

$$[1] = \left\{ 0 \longrightarrow 1 \right\}$$

³Every small category has an underlying graph, obtained keeping objects and arrows and forgetting all compositions; there is of course a category of graphs, and regarding a category as a graph is another example of forgetful functor. Of course, making this precise means that the collection of categories and functors form a category on its own right.

be the category with two objects $0, 1$ and a single nonidentity morphism $0 \rightarrow 1$. A functor $f : \mathbf{2} \rightarrow \mathcal{C}$ consists of two objects $X_i = f(i)$ for $i = 0, 1$, and a morphism $X_0 \rightarrow X_1$.

- A chain is a diagram: let $n > 2$ be a positive integer. Let

$$[n] = \left\{ 0 \longrightarrow 1 \longrightarrow \cdots \longrightarrow n \right\}$$

be the category with exactly $n + 1$ objects and exactly one nonidentity arrow $j - 1 \rightarrow j$ for $j = 1, \dots, n$. A functor $c : [n] \rightarrow \mathcal{C}$ consists of a tuple of objects $X_0 \rightarrow X_1 \rightarrow \cdots \rightarrow X_n$, and morphisms $X_{j-1} \rightarrow X_j$ for $j = 1, \dots, n$.

- A span is a diagram: let

$$\mathcal{S} = \left\{ \begin{array}{ccc} & 0 & \\ \swarrow & & \searrow \\ 1 & & 2 \end{array} \right\}$$

be the category with three objects and arrows as depicted. A functor $F : \mathcal{S} \rightarrow \mathcal{C}$ is a diagram of the same shape, labeled in the same way...

So, you get the idea. Define a *cospan* in \mathcal{C} to be a functor F with domain the opposite category \mathcal{S}^{op} .

- A square is a diagram:
- A cube is a diagram:
- A diagram is a diagram is a diagram, ..., is a diagram

1.4 General definition of limits and colimits

Definition 3 (Cone completions of \mathcal{J}). Let \mathcal{J} be a small category; we denote $\mathcal{J}^\triangleright$ the category obtained adding to \mathcal{J} a single terminal object ∞ ; more in detail, $\mathcal{J}^\triangleright$ has objects $\mathcal{J}_o \cup \{\infty\}$, where $\infty \notin \mathcal{J}$, and it is defined by

$$\begin{aligned} \mathcal{J}^\triangleright(J, J') &= \mathcal{J}(J, J') \\ \mathcal{J}^\triangleright(J, \infty) &= \{*\} \end{aligned}$$

and it is empty otherwise. This category is called the right cone of \mathcal{J} .

Dually, we define a category $\mathcal{J}^\triangleleft$, the left cone of \mathcal{J} , as the category obtained adding to \mathcal{J} a single initial object $-\infty$; this means that $\mathcal{J}^\triangleleft(J, J') = \mathcal{J}(J, J')$, $\mathcal{J}^\triangleleft(-\infty, J) = \{*\}$, and it is empty otherwise.

Example 2. Let ω be the ordinal number obtained from the union of all finite ordinals; then, when regarded as a category, ω^\triangleleft is the category $\{-\infty \rightarrow 0 \rightarrow 1 \rightarrow \dots\}$ (and thus is isomorphic to ω in an evident way), whereas ω^\triangleright is $\omega + 1$, in the sense of ordinal sum.

For those willing to embark in an extra exercise: what is G^\triangleright if G is a monoid regarded as a category?

Remark 3. The correspondences $\mathcal{J} \mapsto \mathcal{J}^\triangleright$ and $\mathcal{J} \mapsto \mathcal{J}^\triangleleft$ are functorial. As an easy exercise, define them on morphisms and prove their functoriality. Prove also that there is a natural embedding i_\triangleright of \mathcal{J} into $\mathcal{J}^\triangleright$ and one i_\triangleleft of \mathcal{J} into $\mathcal{J}^\triangleleft$ (that we invariably denote i in the following discussion).

Definition 4 (Cone of a diagram). Let \mathcal{J} be a small category, \mathcal{C} a category, and $D : \mathcal{J} \rightarrow \mathcal{C}$ a functor; all along this section, an idiosyncratic way to refer to D will be as a diagram of shape \mathcal{J} . We call a cone for D any extension of the diagram D to the left cone category of \mathcal{J} defined in , so that the diagram

$$\begin{array}{ccc} \mathcal{J} & \xrightarrow{D} & \mathcal{C} \\ i_\triangleleft \downarrow & \nearrow \bar{D} & \\ \mathcal{J}^\triangleleft & & \end{array}$$

commutes.

Every such extension is thus forced to coincide with D on all objects in $\mathcal{J} \subseteq \mathcal{J}^\triangleleft$; the value of \bar{D} on $-\infty$ is called the base of the cone; dually, the value of an extension of D to $\mathcal{J}^\triangleright$ coincides with D on $\mathcal{J} \subseteq \mathcal{J}^\triangleright$, and $\bar{D}(\infty)$ is called the *tip* of the cone.

There is of course a similar definition of a *cocone* for D : it is an extension of the diagram D to the right cone category of \mathcal{J} so that the diagram

$$\begin{array}{ccc} \mathcal{J} & \xrightarrow{D} & \mathcal{C} \\ i_\triangleright \downarrow & \nearrow \bar{D} & \\ \mathcal{J}^\triangleright & & \end{array}$$

commutes. Exercise sheds a light on and it substantiates the quote from [?] therein: cones for D are exactly cocones for the opposite functor D^{op} .

Remark 4.

- The class of cones for D forms a category $Cn(D)$, whose morphisms are the natural transformations $\alpha : D' \Rightarrow D'' : \mathcal{J} \rightarrow \mathcal{C}$ such that the

right whiskering of α with $i : \mathcal{J} \rightarrow \mathcal{J}^\triangleleft$ coincides with the identity natural transformation of D ; this means that a morphism α of this sort is a natural transformation such that

$$\begin{array}{ccc}
 & \mathcal{J} & \\
 i \swarrow & & \searrow D \\
 \mathcal{J}^\triangleleft & \xrightarrow{D'} & \mathcal{C} \\
 & \Downarrow \alpha & \\
 & \xrightarrow{D''} &
 \end{array}
 = 1_D$$

as a 2-cell $D \Rightarrow D$.

- Dually, the class of cocones for D forms a category $Ccn(D)$, whose morphisms are the natural transformations $\alpha : D' \Rightarrow D'' : \mathcal{J} \rightarrow \mathcal{C}$ such that the right whiskering of α with $i : \mathcal{J} \rightarrow \mathcal{J}^\triangleright$ coincides with the identity natural transformation of D ; this means that a morphism α of this sort is a natural transformation such that

$$\begin{array}{ccc}
 & \mathcal{J} & \\
 i \swarrow & & \searrow D \\
 \mathcal{J}^\triangleright & \xrightarrow{D'} & \mathcal{C} \\
 & \Downarrow \alpha & \\
 & \xrightarrow{D''} &
 \end{array}
 = 1_D$$

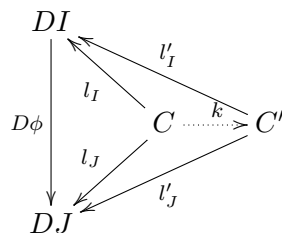
as a 2-cell $D \Rightarrow D$.

Definition 5 (Colimit, limit). *The limit of a diagram $D : \mathcal{J} \rightarrow \mathcal{C}$ is the terminal object denoted ' $\lim_{\mathcal{J}} D$ ' in the category of cones for D ; dually, the colimit of D is the initial object denoted ' $\text{colim}_{\mathcal{J}} D$ ' in the category of cocones for D .*

It is a good idea to unwind definitions and in order to obtain the more classically explained notion of limit and colimit: the key for this unwinding operation is that a co/cone for $D : \mathcal{J} \rightarrow \mathcal{C}$ amounts to a natural family of maps from a constant object (the base) or to a constant object (the tip).

- A *cone* for a diagram $D : \mathcal{J} \rightarrow \mathcal{C}$ is a natural transformation from a constant functor $\Delta_c : \mathcal{J} \rightarrow \mathcal{C}$ to $D(_)$;
- there is a category of cones for $D(_)$, where morphisms between a cone $c \rightarrow D(_)$ and a cone $C' \rightarrow D(_)$ are arrows $k : C \rightarrow C'$ such

that the diagram



is commutative;

- a limit for D is a terminal object in the category of cones for D . This means that given a cone for D , there is a unique arrow k which is a morphism of cones.

Of course, a straightforward dualisation of diagram (??) yields the definition of a cocone, and a colimit for D .

1.5 Examples of limits and colimits

1.5.1 in monoids and Mon

1.5.2 in posets and Pos

1.5.3 in Set, and other algebraic categories

1.5.4 Exercises

Exercises denoted with a star symbol are supposed to be difficult. Don't be put off, and enjoy!

1.