Category theory course
Lecture 4
ITI9200, Spring 2020

Fosco Loregian
fouche@yoneda.ninja

February 11, 2020

Contents

0.1 Functors: morphisms of categories

Si prende un foglio di esercizi di data61 che ti costringono a implementare istanze di `Functor` per aggeggi di uso comune (le liste, gli alberi, `Maybe`, `Either` etcetera) e fargli vedere prima come si scrive in hs, e poi come si scrive in CT, per far loro capire che sono la stessa cosa

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

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

```haskell
-- | 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 $F : \mathcal{C} \to \mathcal{D}$ is a functor, it corresponds to a data constructor `f :: * -> *` (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*

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

If you recall how the associativity of `->` works, it is evident that this is a function from the type of maps `a -> b` to the type of maps `f a -> f b`; or rather, this is a function that given a function `u :: a -> b`, and an "element" of type `f a`, yields an element `x :: f b`.

0.2 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 $F : \varnothing \to \mathcal{C}$, where $\varnothing$ is the empty category with no objects an no morphisms; for every category $\mathcal{C}$, there is an obvious definition of the* identity functor $1_\mathcal{C} : \mathcal{C} \to \mathcal{C}$, *whose correspondences on objects and on arrows $1_{\mathcal{C},AB} : \mathcal{C}(A, B) \to \mathcal{C}(A, B)$ 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 $U\mathbb{M}$ of a thingum $\mathbb{M}$, are functors* $\mathsf{Thng} \to \mathsf{Set}$ *from their structured categories of definition to the category of sets and functions.*

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

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* $\boldsymbol{B} : \mathsf{Mon} \to \mathsf{Cat}$ *because a functor* $F : \boldsymbol{B}M \to \boldsymbol{B}N$ *between monoids is but a monoid homomorphism* $f : M \to N$*; a preset can be seen as a category with at most one arrow between any two objects, and this defines a functor* ${}^{\chi}(\_) : \mathsf{Pres} \to \mathsf{Cat}$*, because a functor* $F : {}^{\chi}P \to {}^{\chi}Q$ *between presets is but a monotone function* $f : P \to Q$*; a set can be seen as a discrete topological space* A^{δ} *or as a category having only identity arrows, and these are functors* $\mathsf{Set} \to \mathsf{Top}, \mathsf{Cat}$*. (Continue the list at your will.)*

4. *Let* M *be a monoid, and* $\boldsymbol{\Delta}$ *be the category having objects nonempty, finite, totally ordered sets and monotone functions as morphisms. The correspondence that sends* $[n] \in \boldsymbol{\Delta}$ *to the set* $M^{[n]}$ *of ordered* n*-tuples* $[a_1| \dots |a_n]$ *of elements of* M *and a monotone function* $f : [m] \to [n]$ *to the function* $M^f : M^{[n]} \to M^{[m]}$ *defined by*

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

is a functor $\boldsymbol{\Delta}^{op} \to \mathsf{Set}$*. This functor is called the* classifying space *of the monoid.*

5. *Let* Set_* *be the category of* pointed sets*: objects are pairs* (A, a) *where* $a \in A$ *is a distinguished element, and a morphism* $f : (A, a) \to (B, b)$ *is a function* $f : A \to B$ *such that* $fa = b$*. It is easy to prove that this category is isomorphic to the coslice* $*/\mathsf{Set}$ *of functions* $a : * \to A$*.*

6. *Let* $\partial\mathsf{Set}$ *be the category so defined: objects are sets, and a morphism* $(f, S) : A \to B$ *is a pair* $(S \subseteq A, f : S \to B)$*. 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* $A \in \partial\mathsf{Set}$*. There is a functor* $(\_)_{\bullet} : \partial\mathsf{Set} \to \mathsf{Set}_*$ *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* $(f, S)_{\bullet} : A \sqcup \{\infty_A\} \to B \sqcup \{\infty_B\}$*, 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* $A \to B$*; 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* $\mathsf{Gph} \to \mathsf{Set}$*, 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.*

0.3 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 $F : \mathcal{J} \to \mathcal{C}$; 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*.

0.3.1 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 $X \to A_1 \to A_2 \to \cdots \to A_n \to Y$ and $X \to B_1 \to \cdots \to B_m \to Y$ result in the same arrow when they are fully composed.

Remark 1 (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*

$$\begin{tikzcd}
 & & X_2 \arrow[drr, "v"] & & \\
X_1 \arrow[urr, "u"] \arrow[rrrr, "h"'] \arrow[dr, "g"'] & & & & X_3 \\
 & X_0 \arrow[uur, "q"] \arrow[rr, "p"'] & & X_4 \arrow[ur, "k"'] &
\end{tikzcd}$$

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 $X_1 \to X_3$; 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')* $D : J \to \mathcal{C}$ *where* J *is a digraph and* $\mathcal{C}$ *is the digraph underlying a category.*[2] *Such a diagram* D commutes *if for every pair of parallel edges* $f, g : i \rightrightarrows j$ *in* J *one has* $Df = Dg$*.*

0.3.2 Diagrams, everywhere

- An object is a diagram
- A morphism is a diagram
- A chain is a diagram
- A span is a diagram
- A cospan is a diagram
- A square is a diagram
- A cube is a diagram
- A diagram is a diagram is a diagram,. . . , is a diagram

0.4 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*

$$\mathcal{J}^{\triangleright}(J, J') = \mathcal{J}(J, J')$$
$$\mathcal{J}^{\triangleright}(J, \infty) = \{*\}$$

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* 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.*

[2] Every 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.

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

$$\begin{tikzcd} \mathcal{J} \arrow[r, "D"] \arrow[d, "i_{\triangleleft}"'] & \mathcal{C} \\ \mathcal{J}^{\triangleleft} \arrow[ur, dotted, "\bar{D}"'] & \end{tikzcd}$$

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

$$\begin{tikzcd} \mathcal{J} \arrow[r, "D"] \arrow[d, "i_{\triangleright}"'] & \mathcal{C} \\ \mathcal{J}^{\triangleright} \arrow[ur, dotted, "\bar{D}"'] & \end{tikzcd}$$

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 3.

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

$$\begin{tikzcd} & \mathcal{J} \arrow[dl, "i"'] \arrow[dr, "D"] & \\ \mathcal{J}^{\lhd} \arrow[rr, bend left, "D'"] \arrow[rr, bend right, "D''"'] \arrow[rr, phantom, "\Downarrow\alpha"] & & \mathcal{C} \end{tikzcd} = 1_D$$

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 $\alpha : D' \Rightarrow D'' : \mathcal{J} \to \mathcal{C}$ such that the right whiskering of α with $i : \mathcal{J} \to \mathcal{J}^{\rhd}$ coincides with the identity natural transformation of D; this means that a morphism α of this sort is a natural transformation such that*

$$\begin{tikzcd} & \mathcal{J} \arrow[dl, "i"'] \arrow[dr, "D"] & \\ \mathcal{J}^{\rhd} \arrow[rr, bend left, "D'"] \arrow[rr, bend right, "D''"'] \arrow[rr, phantom, "\Downarrow\alpha"] & & \mathcal{C} \end{tikzcd} = 1_D$$

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

Definition 5 (Colimit, limit)**.** *The* limit *of a diagram $D : \mathcal{J} \to \mathcal{C}$ is the terminal object denoted '$\mathrm{lim}_{\mathcal{J}} D$' in the category of cones for D; dually, the* colimit *of D is the initial object denoted '$\mathrm{colim}_{\mathcal{J}} D$' 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 $D : \mathcal{J} \to \mathcal{C}$ 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 $D : \mathcal{J} \to \mathcal{C}$ is a natural transformation from a constant functor $\Delta_c : \mathcal{J} \to \mathcal{C}$ to $D(\_)$;
- there is a category of cones for $D(\_)$, where morphisms between a cone $c \to D(\_)$ and a cone $C' \to D(\_)$ are arrows $k : C \to C'$ such

that the diagram

$$\begin{tikzcd}
DI \arrow[dd, "D\phi"'] & & \\
& C \arrow[ul, "l_I"'] \arrow[dl, "l_J"] \arrow[r, dotted, "k"] & C' \arrow[ull, "l'_I"'] \arrow[dll, "l'_J"] \\
DJ & &
\end{tikzcd}$$

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.

0.5 Examples of limits and colimits

0.5.1 in monoids and Mon

0.5.2 in posets and Pos

0.5.3 in Set, and other algebraic categories