Category Theory and Functional Programming
Functors

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2022-2

Preliminaries

Convention

The number assigned to chapters, examples, exercises, figures, pages, sections, and theorems on these slides correspond to the numbers assigned in the textbook [Abramsky and Tzevelekos 2011].

Outline

Introduction

Introduction

Question

What about of morphisms between categories?

Introduction

Question

What about of morphisms between categories?

Answer: Of course, them are functors.

Definition of a Functor

Definition of a Functor

Definition

A **(covariant) functor** $F : \mathcal{C} \to \mathcal{D}$ between categories $\mathcal{C}$ and $\mathcal{D}$ is a mapping of objects to objects and arrows to arrows, that is,†

$$F_0 : \mathrm{Obj}(\mathcal{C}) \to \mathrm{Obj}(\mathcal{D}) \qquad \text{(object-map)}$$

$$F_1 : \mathrm{Ar}(\mathcal{C}) \to \mathrm{Ar}(\mathcal{D}) \qquad \text{(arrow-map)}$$

which for all objects A and arrows f and g, satisfies the **functoriality** conditions

$$F_1\,(g \circ f) = (F_1\,g) \circ (F_1\,f) \qquad \text{(preservation of compositions)}$$

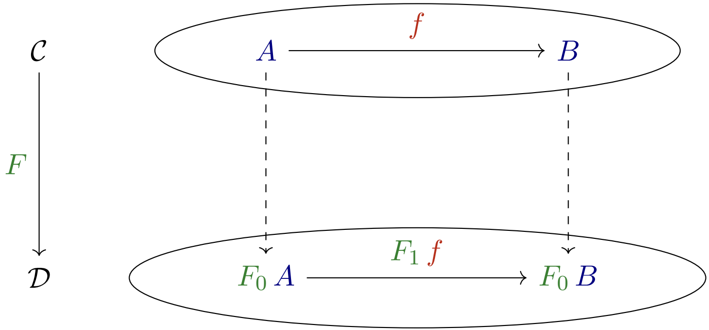
$$F_1\,\mathrm{id}_A = \mathrm{id}_{(F_0\,A)} \qquad \text{(preservation of identities)}$$

† The textbook does not use F_0 and F_1 but F.

Definition of a Functor

Remark

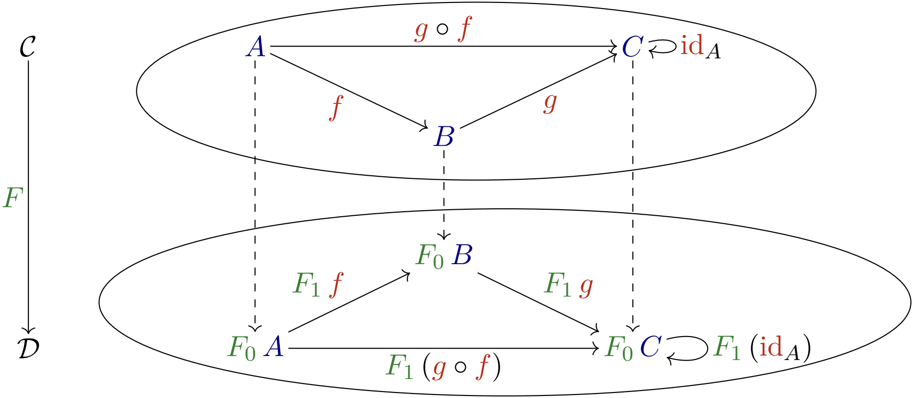
The functor $F : \mathcal{C} \to \mathcal{D}$ maps objects and arrows of $\mathcal{C}$ to objects and arrows of $\mathcal{D}$, respectively.



Definition of a Functor

Remark

The functor $F : \mathcal{C} \to \mathcal{D}$ preserves domains and codomains, identity arrows, and composition. It also maps each commutative diagram in $\mathcal{C}$ into a commutative diagram in $\mathcal{D}$.



Definition of a Functor

Remark

Given a functor $F : \mathcal{C} \to \mathcal{D}$, that is,

$$F_0 : \mathrm{Obj}(\mathcal{C}) \to \mathrm{Obj}(\mathcal{D}),$$
$$F_1 : \mathrm{Ar}(\mathcal{C}) \to \mathrm{Ar}(\mathcal{D}),$$

for all A, B in $\mathrm{Obj}(\mathcal{C})$, there is the map

$$F_{A,B} : \mathcal{C}(A, B) \to \mathcal{D}(F_0\, A, F_0\, B),$$

and for all $f : A \to B$,

$$F_{A,B}\, f : F_0\, A \to F_0\, B.$$

Examples of Functors

Examples of Functors

Example

Let $\mathcal{P}\,S$ be the power set of the set S. The **(covariant) power set functor**

$$P : \mathbf{Set} \to \mathbf{Set}, \qquad \text{is defined by}$$

Examples of Functors

Example

Let $\mathcal{P}\,S$ be the power set of the set S. The **(covariant) power set functor**

$$P : \mathbf{Set} \to \mathbf{Set}, \qquad \text{is defined by}$$

$$P_0 : \mathrm{Obj}(\mathbf{Set}) \to \mathrm{Obj}(\mathbf{Set})$$
$$P_0\,X := \mathcal{P}\,X$$

Examples of Functors

Example

Let $\mathcal{P}\,S$ be the power set of the set S. The **(covariant) power set functor**

$$P : \mathbf{Set} \to \mathbf{Set}, \qquad \text{is defined by}$$

$$P_0 : \mathrm{Obj}(\mathbf{Set}) \to \mathrm{Obj}(\mathbf{Set})$$

$$P_0\,X := \mathcal{P}\,X$$

$$P_1 : \mathrm{Ar}(\mathbf{Set}) \to \mathrm{Ar}(\mathbf{Set})$$

$$P_{X,Y} : \mathbf{Set}(X, Y) \to \mathbf{Set}(P_0\,X, P_0\,Y)$$

$$P_{X,Y}\,f : \mathcal{P}\,X \to \mathcal{P}\,Y$$

$$P_{X,Y}\,f\,S := f(S) = \{\, f(x) \mid x \in S \,\}$$

(continued on next slide)

Examples of Functors

Example (continuation)

Let $X = \{0, 1\}$, $Y = \{\emptyset, X\}$ and $f : X \to Y$ defined by $f(0) = \emptyset$ and $f(1) = X$. Then,

$$P_0 : \mathrm{Obj}(\mathbf{Set}) \to \mathrm{Obj}(\mathbf{Set})$$
$$P_0\, X := \mathcal{P}\, X = \{\emptyset, \{0\}, \{1\}, X\},$$

$$P_{X,Y}\, f : \mathcal{P}\, X \to \mathcal{P}\, Y$$
$$P_{X,Y}\, f\, \emptyset := f(\emptyset) = \emptyset,$$
$$P_{X,Y}\, f\, \{0\} := f(\{0\}) = \{\emptyset\},$$
$$P_{X,Y}\, f\, \{1\} := f(\{1\}) = \{X\},$$
$$P_{X,Y}\, f\, \{0, 1\} := f(\{0, 1\}) = \{\emptyset, X\}.$$

Examples of Functors

Example

Let $(P, \preceq)$ and $(Q, \preceq)$ be two pre-orders seen as categories, denoted $\mathcal{P}$ and $\mathcal{Q}$, respectively. A functor $F : \mathcal{P} \to \mathcal{Q}$ is defined by

Examples of Functors

Example

Let $(P, \preceq)$ and $(Q, \preceq)$ be two pre-orders seen as categories, denoted $\mathcal{P}$ and $\mathcal{Q}$, respectively. A functor $F : \mathcal{P} \to \mathcal{Q}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{P}) \to \mathrm{Obj}(\mathcal{Q})$$

Examples of Functors

Example

Let $(P, \preceq)$ and $(Q, \preceq)$ be two pre-orders seen as categories, denoted $\mathcal{P}$ and $\mathcal{Q}$, respectively. A functor $F : \mathcal{P} \to \mathcal{Q}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{P}) \to \mathrm{Obj}(\mathcal{Q})$$

$$F_1 : \mathrm{Ar}(\mathcal{P}) \to \mathrm{Ar}(\mathcal{Q})$$
$$F_{A,B} : \mathcal{P}(A, B) \to \mathcal{Q}(F_0\, A, F_0\, B)$$
$$F_{A,B}\, f : F_0\, A \to F_0\, B$$

Examples of Functors

Example

Let $(P, \preceq)$ and $(Q, \preceq)$ be two pre-orders seen as categories, denoted $\mathcal{P}$ and $\mathcal{Q}$, respectively. A functor $F : \mathcal{P} \to \mathcal{Q}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{P}) \to \mathrm{Obj}(\mathcal{Q})$$

$$F_1 : \mathrm{Ar}(\mathcal{P}) \to \mathrm{Ar}(\mathcal{Q})$$
$$F_{A,B} : \mathcal{P}(A, B) \to \mathcal{Q}(F_0\, A, F_0\, B)$$
$$F_{A,B}\, f : F_0\, A \to F_0\, B$$

Since $\mathcal{P}(A, B)$ and $\mathcal{Q}(F_0\, A, F_0\, B)$ have at most an arrow, the map $F_{A,B}$ exists iff

$$A \preceq B \quad \text{implies} \quad F_0\, A \preceq F_0\, B.$$

Examples of Functors

Example

Let $(P, \preceq)$ and $(Q, \preceq)$ be two pre-orders seen as categories, denoted $\mathcal{P}$ and $\mathcal{Q}$, respectively. A functor $F : \mathcal{P} \to \mathcal{Q}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{P}) \to \mathrm{Obj}(\mathcal{Q})$$

$$\begin{aligned} F_1 &: \mathrm{Ar}(\mathcal{P}) \to \mathrm{Ar}(\mathcal{Q}) \\ F_{A,B} &: \mathcal{P}(A, B) \to \mathcal{Q}(F_0\,A, F_0\,B) \\ F_{A,B}\,f &: F_0\,A \to F_0\,B \end{aligned}$$

Since $\mathcal{P}(A, B)$ and $\mathcal{Q}(F_0\,A, F_0\,B)$ have at most an arrow, the map $F_{A,B}$ exists iff

$$A \preceq B \quad \text{implies} \quad F_0\,A \preceq F_0\,B.$$

That is, a functor $F : \mathcal{P} \to \mathcal{Q}$ is just a monotone map which sends, if exists, the unique arrow $A \to B$ to the unique arrow $F_0\,A \to F_0\,B$.

Examples of Functors

Example

Let $(P, \preceq)$ and $(Q, \preceq)$ be two pre-orders seen as categories, denoted $\mathcal{P}$ and $\mathcal{Q}$, respectively. A functor $F : \mathcal{P} \to \mathcal{Q}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{P}) \to \mathrm{Obj}(\mathcal{Q})$$

$$F_1 : \mathrm{Ar}(\mathcal{P}) \to \mathrm{Ar}(\mathcal{Q})$$
$$F_{A,B} : \mathcal{P}(A, B) \to \mathcal{Q}(F_0\, A, F_0\, B)$$
$$F_{A,B}\, f : F_0\, A \to F_0\, B$$

Since $\mathcal{P}(A, B)$ and $\mathcal{Q}(F_0\, A, F_0\, B)$ have at most an arrow, the map $F_{A,B}$ exists iff

$$A \preceq B \quad \text{implies} \quad F_0\, A \preceq F_0\, B.$$

That is, a functor $F : \mathcal{P} \to \mathcal{Q}$ is just a monotone map which sends, if exists, the unique arrow $A \to B$ to the unique arrow $F_0\, A \to F_0\, B$.

Example from [Fong, Milewski and Spivak 2020, § 3.2.2].

Examples of Functors

Example

Let $(M, \cdot, \epsilon)$ and $(N, \diamond, \mu)$ be two monoids seen as categories, denoted $\mathcal{M}$ and $\mathcal{N}$, respectively. Let $*$ be the only object in both categories. A functor $F : \mathcal{M} \to \mathcal{N}$ is defined by

Examples of Functors

Example

Let $(M, \cdot, \epsilon)$ and $(N, \diamond, \mu)$ be two monoids seen as categories, denoted $\mathcal{M}$ and $\mathcal{N}$, respectively. Let $*$ be the only object in both categories. A functor $F : \mathcal{M} \to \mathcal{N}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{M}) \to \mathrm{Obj}(\mathcal{N})$$
$$F_0 : \{*\} \to \{*\}$$
$$F_0 * = *$$

Examples of Functors

Example

Let $(M, \cdot, \epsilon)$ and $(N, \diamond, \mu)$ be two monoids seen as categories, denoted $\mathcal{M}$ and $\mathcal{N}$, respectively. Let $*$ be the only object in both categories. A functor $F : \mathcal{M} \to \mathcal{N}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{M}) \to \mathrm{Obj}(\mathcal{N})$$
$$F_0 : \{*\} \to \{*\}$$
$$F_0 * = *$$

$$F_1 : \mathrm{Ar}(\mathcal{M}) \to \mathrm{Ar}(\mathcal{N})$$
$$F_{*,*} : \mathcal{P}(*, *) \to \mathcal{Q}(F_0 *, F_0 *)$$
$$F_{*,*}\, f : * \to *$$

Examples of Functors

Example

Let $(M, \cdot, \epsilon)$ and $(N, \diamond, \mu)$ be two monoids seen as categories, denoted $\mathcal{M}$ and $\mathcal{N}$, respectively. Let $*$ be the only object in both categories. A functor $F : \mathcal{M} \to \mathcal{N}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{M}) \to \mathrm{Obj}(\mathcal{N}) \qquad F_1 : \mathrm{Ar}(\mathcal{M}) \to \mathrm{Ar}(\mathcal{N})$$

$$F_0 : \{*\} \to \{*\} \qquad F_{*,*} : \mathcal{P}(*,*) \to \mathcal{Q}(F_0\,*, F_0\,*)$$

$$F_0\,* = * \qquad F_{*,*}\,f : * \to *$$

The functor F must satisfies:

$$F_{*,*}\,(m_1 \cdot m_2) = (F_{*,*}\,m_1) \diamond (F_{*,*}\,m_2), \qquad \text{for all } m_1, m_2 \text{ in } \mathcal{M},$$

$$F_{*,*}\,\epsilon = \mu.$$

Examples of Functors

Example

Let $(M, \cdot, \epsilon)$ and $(N, \diamond, \mu)$ be two monoids seen as categories, denoted $\mathcal{M}$ and $\mathcal{N}$, respectively. Let $*$ be the only object in both categories. A functor $F : \mathcal{M} \to \mathcal{N}$ is defined by

$$F_0 : \mathrm{Obj}(\mathcal{M}) \to \mathrm{Obj}(\mathcal{N}) \qquad F_1 : \mathrm{Ar}(\mathcal{M}) \to \mathrm{Ar}(\mathcal{N})$$
$$F_0 : \{*\} \to \{*\} \qquad F_{*,*} : \mathcal{P}(*, *) \to \mathcal{Q}(F_0 *, F_0 *)$$
$$F_0 * = * \qquad F_{*,*} f : * \to *$$

The functor F must satisfies:

$$F_{*,*} (m_1 \cdot m_2) = (F_{*,*} m_1) \diamond (F_{*,*} m_2), \qquad \text{for all } m_1, m_2 \text{ in } \mathcal{M},$$
$$F_{*,*} \epsilon = \mu.$$

That is, a functor $F : \mathcal{M} \to \mathcal{N}$ is just a monoid homomorphism.

Examples of Functors

Example

The **identity functor** $\mathrm{Id}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$ in a category $\mathcal{C}$ is the functor that maps each object and each arrow of $\mathcal{C}$ to itself.

Examples of Functors

Example

Let $F : \mathbf{Mon} \to \mathbf{Set}$ be the **forgetful functor** which

(i) sends a monoid to its set of elements and

(ii) sends a homomorphism between monoids to the corresponding function between sets.

Examples of Functors

Example

Let $[S]$ be the set of all finite lists of elements of S. The **list functor**

$$\text{List} : \mathbf{Set} \to \mathbf{Set}, \qquad \text{is defined by}$$

Examples of Functors

Example

Let $[S]$ be the set of all finite lists of elements of S. The **list functor**

$$\mathsf{List} : \mathbf{Set} \to \mathbf{Set}, \qquad \text{is defined by}$$

$$\mathsf{List}_0 : \mathrm{Obj}(\mathbf{Set}) \to \mathrm{Obj}(\mathbf{Set})$$

$$\mathsf{List}_0\, X := [X]$$

Examples of Functors

Example

Let $[S]$ be the set of all finite lists of elements of S. The **list functor**

$$\mathsf{List} : \mathbf{Set} \to \mathbf{Set}, \qquad \text{is defined by}$$

$$\mathsf{List}_0 : \mathrm{Obj}(\mathbf{Set}) \to \mathrm{Obj}(\mathbf{Set})$$

$$\mathsf{List}_0\, X := [X]$$

$$\mathsf{List}_1 : \mathrm{Ar}(\mathbf{Set}) \to \mathrm{Ar}(\mathbf{Set})$$

$$\mathsf{List}_{X,Y} : \mathbf{Set}(X, Y) \to \mathbf{Set}(\mathsf{List}_0\, X, \mathsf{List}_0\, Y)$$

$$\mathsf{List}_{X,Y}\, f : [X] \to [Y]$$

$$\mathsf{List}_{X,Y}\, f\, [x_1, x_2, \ldots, x_n] := [f(x_1), f(x_2), \ldots, f(x_n)]$$

Examples of Functors

Example

The **free monoid functor** MList $: \mathbf{Set} \to \mathbf{Mon}$ maps every set X to the free monoid over X.

Examples of Functors

Example

The **free monoid functor** MList : $\mathbf{Set} \to \mathbf{Mon}$ maps every set X to the free monoid over X.

Let $(-) * (-)$ be the list concatenation function and let ε be the empty list, the functor is defined by

Examples of Functors

Example

The **free monoid functor** MList : $\mathbf{Set} \to \mathbf{Mon}$ maps every set X to the free monoid over X.

Let $(-) * (-)$ be the list concatenation function and let ε be the empty list, the functor is defined by

$$\begin{aligned} \mathsf{MList}_0 &: \mathrm{Obj}(\mathbf{Set}) \to \mathrm{Obj}(\mathbf{Mon}) \\ \mathsf{MList}_0\, X &:= (\mathsf{List}_0\, X, *, \varepsilon) \\ &= ([X], *, \varepsilon) \end{aligned}$$

Examples of Functors

Example

The **free monoid functor** $\mathsf{MList} : \mathbf{Set} \to \mathbf{Mon}$ maps every set X to the free monoid over X.

Let $(-) * (-)$ be the list concatenation function and let ε be the empty list, the functor is defined by

$$\begin{aligned} \mathsf{MList}_0 &: \mathrm{Obj}(\mathbf{Set}) \to \mathrm{Obj}(\mathbf{Mon}) \\ \mathsf{MList}_0\, X &:= (\mathsf{List}_0\, X, *, \varepsilon) \\ &= ([X], *, \varepsilon) \end{aligned}$$

$$\begin{aligned} &\mathsf{MList}_1 : \mathrm{Ar}(\mathbf{Set}) \to \mathrm{Ar}(\mathbf{Mon}) \\ &\mathsf{MList}_{X,Y} : \mathbf{Set}(X, Y) \to \mathbf{Mon}(\mathsf{MList}_0\, X, \mathsf{MList}_0\, Y) \\ &\mathsf{MList}_{X,Y}\, f : ([X], *, \varepsilon) \to ([Y], *, \varepsilon) \\ &\mathsf{MList}_{X,Y}\, f\, [x_1, x_2, \ldots, x_n] := \mathsf{List}_{X,Y}\, f\, [x_1, x_2, \ldots, x_n] \end{aligned}$$

Exercises

Exercise 1

Verify that functors $F : \mathbf{2}_{\rightrightarrows} \to \mathbf{Set}$ correspond to directed graphs (textbook, Exercise 45).

Functors in Haskell

Functors in Haskell

Introduction via `Maybe`

(Whiteboard).

Functors in Haskell

Introduction via Maybe

(Whiteboard).

The typeclass Functor

```
class Functor f where
  fmap :: (a -> b) -> f a -> f b
```

Functors in Haskell

Example

The polymorphic type constructor Maybe is a functor whose instance is defined by

```
instance Functor Maybe where
  fmap _ Nothing  = Nothing
  fmap f (Just a) = Just (f a)
```

Functors in Haskell

Example

The polymorphic type constructor Maybe is a functor whose instance is defined by

```
instance Functor Maybe where
  fmap _ Nothing  = Nothing
  fmap f (Just a) = Just (f a)
```

Exercise 2

Show that the Maybe functor satisfies the functoriality conditions.

Functors in Haskell

Example

ReadInt is a type constructor that turns any type a into a new type that reads a value of Int to create a value of a [Fong, Milewski and Spivak 2020, Example 3.41].

```
data ReadInt a = MkReadInt (Int -> a)
```

Functors in Haskell

Example

ReadInt is a type constructor that turns any type a into a new type that reads a value of Int to create a value of a [Fong, Milewski and Spivak 2020, Example 3.41].

```
data ReadInt a = MkReadInt (Int -> a)
```

ReadInt is a functor via the following instance.

```
instance Functor ReadInt where
  fmap f (MkReadInt g) = MkReadInt (f . g)
```

Functors in Haskell

Example

The (binary) function type `(->) :: a -> b -> (a -> b)` is a functor.

```
instance Functor ((->) a) where
  fmap f g = f . g
```

Note that `fmap :: (b -> c) -> (a -> b) -> (a -> c)`.

Functors in Haskell

Exercise 3

To define an instance of `Functor` for the (binary) product type `(,) :: a -> b -> (a,b)`.

Functors in Haskell

Example

Recall that terminal object (unit type) in Haskell is `()::()`. We can define a constant functor by

```
data CUnit a = MkCU ()

instance Functor CUnit where
  fmap f (MkCU ()) = MkCU ()
```

Functors in Haskell

Exercise 4

Given a constant 'functor' defined by

```
data CBool a = MkCB Bool

instance Functor CBool where
  fmap f (MkCB True)  = MkCB False
  fmap f (MkCB False) = MkCB True
```

Is `CBool` really a functor?

Functors in Haskell

Exercise 5

We define a constant functor by

```
data CInt a = MkCI Int
```

Show that the polymorphic type constructor `CInt` can be given the structure of a functor by saying how it lifts morphisms. That is, provide a Haskell function `mapCInt` of the type `(a -> b) -> (CInt a -> CInt b)` [Fong, Milewski and Spivak 2020, Exercise 3.46].

Functors in Haskell

Exercise 6

For each of the following type constructors, define two versions of `fmap`, one of which has a corresponding functor $\mathbf{Hask} \to \mathbf{Hask}$, and one of which does not [Fong, Milewski and Spivak 2020, Exercise 3.48].

(i) `data WithString a = WithStr (a, String)`

(ii) `data ConstStr a = ConstStr String`

(iii) `data List a = Nil | Cons (a, List a)`

Binary Functors

The Product Category

Definition

Let $\mathcal{C}$ and $\mathcal{D}$ be two categories. The **product category** $\mathcal{C} \times \mathcal{D}$ is defined by:

(i) Objects: (C, D), where C and D are objects in $\mathcal{C}$ and $\mathcal{D}$, respectively.

(ii) Arrows: $(C, D) \xrightarrow{(f,g)} (C', D')$, where $C \xrightarrow{f} C'$ and $D \xrightarrow{g} D'$ are arrows in $\mathcal{C}$ and $\mathcal{D}$, respectively.

(iii) Composition

$$(f', g') \circ (f, g) := (f' \circ f, g' \circ g).$$

(iv) Identities

$$\mathrm{id}_{(C,D)} := (\mathrm{id}_C, \mathrm{id}_D).$$

Definition of a Binary Functor

Definition

Let $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{E}$ be three categories. A **binary functor** (or **bifunctor**) is a functor whose domain is a product category, that is, a binary functor from $\mathcal{C} \times \mathcal{D}$ to $\mathcal{E}$ is a functor

$$F : \mathcal{C} \times \mathcal{D} \to \mathcal{E}.$$

Example of Binary Functors

Example

The projection functors $\mathcal{C} \xleftarrow{\pi_1} \mathcal{C} \times \mathcal{D} \xrightarrow{\pi_2} \mathcal{D}$ are binary functors.

Example of Binary Functors

Example

The projection functors $\mathcal{C} \xleftarrow{\pi_1} \mathcal{C} \times \mathcal{D} \xrightarrow{\pi_2} \mathcal{D}$ are binary functors.

(i) For $\pi_1 : \mathcal{C} \times \mathcal{D} \to \mathcal{C}$ we have:

$$(\pi_1)_0 : \mathrm{Obj}(\mathcal{C} \times \mathcal{D}) \to \mathrm{Obj}(\mathcal{C})$$
$$(\pi_1)_0\,(C, D) := C$$

$$(\pi_1)_1 : \mathrm{Ar}(\mathcal{C} \times \mathcal{D}) \to \mathrm{Ar}(\mathcal{C})$$
$$(\pi_1)_{(C,D),(C',D')} : \mathrm{Mor}_{\mathcal{C}\times\mathcal{D}}((C, D), (C', D')) \to \mathrm{Mor}_{\mathcal{C}}((\pi_1)_0\,(C, D), (\pi_1)_0\,(C', D')),$$

$$(\pi_1)_{(C,D),(C',D')}\,(f, g) : C \to C'$$
$$(\pi_1)_{(C,D),(C',D')}\,(f, g) := f$$

Example of Binary Functors

Example

The projection functors $\mathcal{C} \xleftarrow{\pi_1} \mathcal{C} \times \mathcal{D} \xrightarrow{\pi_2} \mathcal{D}$ are binary functors.

(i) For $\pi_1 : \mathcal{C} \times \mathcal{D} \to \mathcal{C}$ we have:

$$(\pi_1)_0 : \mathrm{Obj}(\mathcal{C} \times \mathcal{D}) \to \mathrm{Obj}(\mathcal{C})$$
$$(\pi_1)_0\,(C, D) := C$$

$$(\pi_1)_1 : \mathrm{Ar}(\mathcal{C} \times \mathcal{D}) \to \mathrm{Ar}(\mathcal{C})$$
$$(\pi_1)_{(C,D),(C',D')} : \mathrm{Mor}_{\mathcal{C}\times\mathcal{D}}((C, D), (C', D')) \to \mathrm{Mor}_{\mathcal{C}}((\pi_1)_0\,(C, D), (\pi_1)_0\,(C', D')),$$

$$(\pi_1)_{(C,D),(C',D')}\,(f, g) : C \to C'$$
$$(\pi_1)_{(C,D),(C',D')}\,(f, g) := f$$

(ii) Similarly for $\pi_2 : \mathcal{C} \times \mathcal{D} \to \mathcal{D}$.

The Product Functor

Definition

Let $\mathcal{C}$ be a category with binaries products, and let $\mathcal{C} \times \mathcal{C}$ be the product category of $\mathcal{C}$ with itself. The **product functor** $\times : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ is a binary functor defined by

The Product Functor

Definition

Let $\mathcal{C}$ be a category with binaries products, and let $\mathcal{C} \times \mathcal{C}$ be the product category of $\mathcal{C}$ with itself. The **product functor** $\times : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ is a binary functor defined by

$$\times_0 : \mathrm{Obj}(\mathcal{C} \times \mathcal{C}) \to \mathrm{Obj}(\mathcal{C})$$
$$\times_0 (A, B) := A \times B \quad \text{(binary product)}$$

$$\times_1 : \mathrm{Ar}(\mathcal{C} \times \mathcal{C}) \to \mathrm{Ar}(\mathcal{C})$$
$$\times_{(A,A'),(B,B')} : \mathrm{Mor}_{\mathcal{C}\times\mathcal{C}}((A, A'), (B, B')) \to \mathrm{Mor}_{\mathcal{C}}(\times_0 (A, A'), \times_0 (B, B'))$$
$$\times_{(A,A'),(B,B')} (f, g) : A \times A' \to B \times B'$$
$$\times_{(A,A'),(B,B')} (f, g) := f \times g \quad \text{(product morphish)}$$

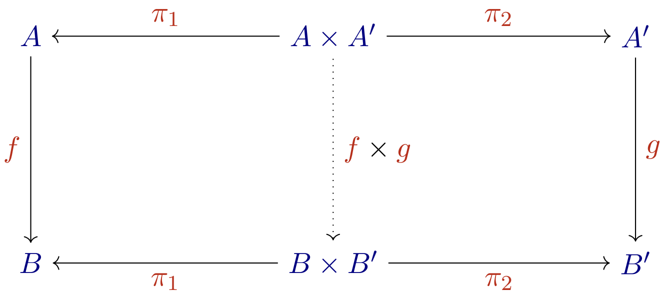
where $f \times g := \langle f \circ \pi_1, g \circ \pi_2 \rangle$.

(continued on next slide)

The Product Functor

Definition (continuation)

That is, both squares in the following diagram commute.



$$\begin{pmatrix} f \circ \pi_1 = \pi_1 \circ (f \times g) \\ g \circ \pi_2 = \pi_2 \circ (f \times g) \end{pmatrix}$$

N-Ary Functors

Remark

Binary functors can be generalised to n-ary functors.

Small, Large and Locally Small Categories

Small and Large Categories

Introduction

Before defining a category of categories, we need to classify the categories in small and large for avoiding that it be an object of itself.

Small and Large Categories

Definition

A category is **small** iff both the collection of its objects and the collection of its arrows are sets. Otherwise, the category is **large** [Awodey 2010].

Small and Large Categories

Example

The finite categories $\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}$, a monoid viewed as a category, and a pre-order viewed as a category are small categories.

Small and Large Categories

Example

The finite categories $\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}$, a monoid viewed as a category, and a pre-order viewed as a category are small categories.

Example

The categories $\mathbf{Set}$, $\mathbf{Pos}$, $\mathbf{Mon}$, $\mathbf{Grp}$ and $\mathbf{Top}$ are large categories.

Locally Small Categories

Definition

A category $\mathcal{C}$ is **locally small** iff for all objects A, B the collection $\mathcal{C}(A, B)$ is a set [Awodey 2010].

Locally Small Categories

Definition

A category $\mathcal{C}$ is **locally small** iff for all objects A, B the collection $\mathcal{C}(A, B)$ is a set [Awodey 2010].

Remark

- Recall from the previous conventions that if the collection $\mathcal{C}(A, B)$ is a set it is called a hom-set and it is denoted $\hom_{\mathcal{C}}(A, B)$.
- Also recall that in the textbook all the collections $\mathcal{C}(A, B)$ are hom-sets.

Locally Small Categories

Example

Any small category is locally small.

Locally Small Categories

Example

Any small category is locally small.

Example

The categories $\mathbf{Set}$, $\mathbf{Pos}$, $\mathbf{Mon}$, $\mathbf{Grp}$ and $\mathbf{Top}$ are locally small categories.

The Category of Small Categories

The Category of Small Categories

Definition

The category $\mathbf{Cat}$ is the category of small categories:

(i) Objects: Small categories

(ii) Arrows: Functors

(continued on next slide)

The Category of Small Categories

Definition (continuation)

(iii) Composition of functors

Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$ two functors, then

$$
\begin{aligned}
G \circ F \quad &: \mathcal{C} \to \mathcal{E} \\
(G \circ F)_0 \quad &: \mathrm{Obj}(\mathcal{C}) \to \mathrm{Obj}(\mathcal{E}) \\
(G \circ F)_0\, A \quad &:= G_0\,(F_0\, A), \\
(G \circ F)_1 \quad &: \mathrm{Ar}(\mathcal{C}) \to \mathrm{Ar}(\mathcal{E}) \\
(G \circ F)_{A,B} \quad &: \mathcal{C}(A, B) \to \mathcal{E}((G \circ F)_0\, A, (G \circ F)_0\, B) \\
(G \circ F)_{A,B}\, f &: G_0\,(F_0\, A) \to G_0\,(F_0\, B) \\
(G \circ F)_{A,B}\, f &:= G_1\,(F_1\, f).
\end{aligned}
$$

(continued on next slide)

The Category of Small Categories

Definition (continuation)

(iii) Composition of functors

Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$, then

$$G \circ F : \mathcal{C} \to \mathcal{E} := \begin{cases} A & \mapsto G\,(F\,A), \\ f & \mapsto G\,(F\,f). \end{cases}$$

The Category of Small Categories

Definition (continuation)

(iii) Composition of functors

Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$, then

$$G \circ F : \mathcal{C} \to \mathcal{E} := \begin{cases} A & \mapsto G\,(F\,A), \\ f & \mapsto G\,(F\,f). \end{cases}$$

(iv) Identity functors

$$\mathrm{id}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C} := \begin{cases} A & \mapsto A, \\ f & \mapsto f. \end{cases}$$

The Category of Small Categories

Remark

The category $\mathbf{Cat}$ is large and therefore it is not object of itself.

Contravariance

Introduction

Description

A covariant functor F preserves the direction of arrows, that is,

$$F_1\,(f : A \to B) : F_0\,A \to F_0\,B.$$

A contravariant functor G reverses the direction of arrows, that is,

$$G_1\,(f : A \to B) : G_0\,B \to G_0\,A.$$

Contravariant Functors

Definition

Let $\mathcal{C}$ and $\mathcal{D}$ be two categories. A **contravariant functor** G from $\mathcal{C}$ to $\mathcal{D}$ is a functor

$$G \; : \mathcal{C}^{\mathsf{op}} \to \mathcal{D} \text{ (or } \mathcal{C} \to \mathcal{D}^{\mathsf{op}}\text{)}$$

$$G_0 : \mathrm{Obj}(\mathcal{C}^{\mathsf{op}}) \to \mathrm{Obj}(\mathcal{D}) \qquad \text{(object-map)}$$

$$G_1 : \mathrm{Ar}(\mathcal{C}^{\mathsf{op}}) \to \mathrm{Ar}(\mathcal{D}) \qquad \text{(arrow-map)}$$

$$G_{A,B} \; : \mathcal{C}^{\mathsf{op}}(A, B) \to \mathcal{D}(G_0\, B, G_0\, A)$$

$$G_{A,B}\, f : G_0\, B \to G_0\, A$$

$$G_1(g \circ f) = (G_1\, f) \circ (G_1\, g) \qquad \text{(preservation of composition)}$$

$$G_1\,(\mathrm{id}_A) \; = \mathrm{id}_{(G_0\, A)} \qquad \text{(preservation of identities)}$$

Contravariant Functors

Example

Let $\mathcal{P}\, S$ be the power set of the set S. The **contravariant power set functor**

$$P^{\mathsf{op}} : \mathbf{Set}^{\mathsf{op}} \to \mathbf{Set}, \qquad \text{is defined by}$$

Contravariant Functors

Example

Let $\mathcal{P}\, S$ be the power set of the set S. The **contravariant power set functor**

$$P^{\mathsf{op}} : \mathbf{Set}^{\mathsf{op}} \to \mathbf{Set}, \qquad \text{is defined by}$$

$$P_0^{\mathsf{op}} : \mathrm{Obj}(\mathbf{Set}^{\mathsf{op}}) \to \mathrm{Obj}(\mathbf{Set})$$
$$P_0^{\mathsf{op}}\, X := \mathcal{P}\, X$$

Contravariant Functors

Example

Let $\mathcal{P}\,S$ be the power set of the set S. The **contravariant power set functor**

$$P^{\mathsf{op}} : \mathbf{Set}^{\mathsf{op}} \to \mathbf{Set}, \qquad \text{is defined by}$$

$$P_0^{\mathsf{op}} : \mathrm{Obj}(\mathbf{Set}^{\mathsf{op}}) \to \mathrm{Obj}(\mathbf{Set})$$
$$P_0^{\mathsf{op}}\,X := \mathcal{P}\,X$$

$$P_1^{\mathsf{op}} : \mathrm{Ar}(\mathbf{Set}^{\mathsf{op}}) \to \mathrm{Ar}(\mathbf{Set})$$
$${P^{\mathsf{op}}}_{X,Y} : \mathbf{Set}^{\mathsf{op}}(X, Y) \to \mathbf{Set}(P_0^{\mathsf{op}}\,Y, P_0^{\mathsf{op}}\,X)$$

$${P^{\mathsf{op}}}_{X,Y}\,f : \mathcal{P}\,Y \to \mathcal{P}\,X$$
$${P^{\mathsf{op}}}_{X,Y}\,f\,T := f^{-1}(T) = \{\, x \in X \mid f(x) \in T \,\}$$

Hom-Functors

Hom-Functors

Definition (first notation)

Let $\mathcal{C}$ be a locally small category and let A be an object of $\mathcal{C}$. The **covariant Set-valued hom-functor** $\mathcal{C}(A,-)$ is defined by

$$\mathcal{C}(A,-) : \mathcal{C} \to \mathbf{Set},$$

$$\mathcal{C}(A,-)_0 : \mathrm{Obj}(\mathcal{C}) \to \mathrm{Obj}(\mathbf{Set})$$
$$\mathcal{C}(A,C)_0 := \mathcal{C}(A,C),$$

$$\mathcal{C}(A,-)_1 : \mathrm{Ar}(\mathcal{C}) \to \mathrm{Ar}(\mathbf{Set})$$
$$\mathcal{C}(A,-)_{C,D} : \mathcal{C}(C,D) \to \mathbf{Set}(\mathcal{C}(A,-)_0\, C, \mathcal{C}(A,-)_0\, D)$$
$$\mathcal{C}(A,f)_{C,D} : \mathcal{C}(A,C) \to \mathcal{C}(A,D)$$
$$\mathcal{C}(A,f)_{C,D}\, g := f \circ g.$$

Hom-Functors

Definition (first notation)

Let $\mathcal{C}$ be a (locally small) category and let B be an object of $\mathcal{C}$. The **contravariant** $\mathbf{Set}$**-valued hom-functor** $\mathcal{C}(-,B)$ is defined by

$$\mathcal{C}(-,B) : \mathcal{C}^{\mathsf{op}} \to \mathbf{Set},$$

$$\mathcal{C}(-,B)_0 : \mathrm{Obj}(\mathcal{C}^{\mathsf{op}}) \to \mathrm{Obj}(\mathbf{Set})$$
$$\mathcal{C}(C,B)_0 := \mathcal{C}(C,B),$$

$$\mathcal{C}(-,B)_1 : \mathrm{Ar}(\mathcal{C}^{\mathsf{op}}) \to \mathrm{Ar}(\mathbf{Set})$$
$$\mathcal{C}(-,B)_{C,D} : \mathcal{C}^{\mathsf{op}}(C,D) \to \mathbf{Set}(\mathcal{C}(-,B)_0\, D, \mathcal{C}(-,B)_0\, C)$$
$$\mathcal{C}(f,B)_{C,D} : \mathcal{C}(D,B) \to \mathcal{C}(C,B)$$
$$\mathcal{C}(f,B)_{C,D}\, g := g \circ f.$$

Hom-Functors

Exercise 7

Let $\mathcal{C}$ be a (locally small) category. Spell out the definition of the $\mathbf{set}$-valued hom-functor $\mathcal{C}(-,-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}$. Verify carefully that it is a functor (textbook, Exercise 47).

Hom-Functors

Notation

Recall that if $\mathcal{C}$ is a locally small category the collection of arrows of an object A to an object B is a set and it is denoted by $\hom_{\mathcal{C}}(A, B)$, that is,

$$\hom_{\mathcal{C}}(A, B) := \left\{ f \in \operatorname{Ar}(\mathcal{C}) \,\middle|\, A \xrightarrow{f} B \right\} =: \mathcal{C}(A, B).$$

Hom-Functors

Definition (second notation)

Let $\mathcal{C}$ be a locally small category and let A be an object of $\mathcal{C}$. The **covariant Set-valued hom-functor** $\hom_{\mathcal{C}}(A, -)$ is defined by

$$\hom_{\mathcal{C}}(A, -) : \mathcal{C} \to \mathbf{Set},$$

$$\hom_{\mathcal{C}}(A, -)_0 : \mathrm{Obj}(\mathcal{C}) \to \mathrm{Obj}(\mathbf{Set})$$
$$\hom_{\mathcal{C}}(A, C)_0 := \hom_{\mathcal{C}}(A, C)$$

$$\hom_{\mathcal{C}}(A, -)_1 : \mathrm{Ar}(\mathcal{C}) \to \mathrm{Ar}(\mathbf{Set})$$
$$\hom_{\mathcal{C}}(A, -)_{C,D} : \hom_{\mathcal{C}}(C, D) \to \mathbf{Set}(\hom_{\mathcal{C}}(A, C), \hom_{\mathcal{C}}(A, D))$$
$$\hom_{\mathcal{C}}(A, f : C \to D) : \hom_{\mathcal{C}}(A, C) \to \hom_{\mathcal{C}}(A, D)$$
$$\hom_{\mathcal{C}}(A, f : C \to D)\, g := f \circ g$$

Hom-Functors

Definition (second notation)

Let $\mathcal{C}$ be a (locally small) category and let B be an object of $\mathcal{C}$. The **contravariant** $\mathbf{Set}$**-valued hom-functor** $\hom_{\mathcal{C}}(-, B)$ is defined by

$$\hom_{\mathcal{C}}(-, B) : \mathcal{C}^{\mathsf{op}} \to \mathbf{Set},$$

$$\hom_{\mathcal{C}}(-, B)_0 : \mathrm{Obj}(\mathcal{C}^{\mathsf{op}}) \to \mathrm{Obj}(\mathbf{Set})$$
$$\hom_{\mathcal{C}}(C, B)_0 := \hom_{\mathcal{C}}(C, B)$$

$$\hom_{\mathcal{C}}(-, B)_1 : \mathrm{Ar}(\mathcal{C}^{\mathsf{op}}) \to \mathrm{Ar}(\mathbf{Set})$$
$$\hom_{\mathcal{C}}(-, B)_{C,D} : \hom_{(\mathcal{C}^{\mathsf{op}})}(C, D) \to \mathbf{Set}(\hom_{\mathcal{C}}(D, B), \hom_{\mathcal{C}}(C, B))$$

$$\hom_{\mathcal{C}}(f : C \to D, B) : \hom_{\mathcal{C}}(D, B) \to \hom_{\mathcal{C}}(C, B)$$
$$\hom_{\mathcal{C}}(f : C \to D, B)\, g := g \circ f$$

Hom-Functors

Exercise 8

Let $\mathcal{C}$ be a (locally small) category. Spell out the definition of the $\mathbf{set}$-valued hom-functor $\hom_{\mathcal{C}}(-,-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}$. Verify carefully that it is a functor (textbook, Exercise 47).

Properties of Functors

Faithful and Full Functors

Definition

Let $\mathcal{C}$ and $\mathcal{D}$ be (locally small) categories and let $F : \mathcal{C} \to \mathcal{D}$ be a functor.

(i) The functor F is **faithful** iff each map $F_{A,B} : \mathcal{C}(A, B) \to \mathcal{D}(F_0\, A, F_0\, B)$ is injective.

(ii) The functor F is **full** iff each map $F_{A,B} : \mathcal{C}(A, B) \to \mathcal{D}(F_0\, A, F_0\, B)$ is surjective.

Faithful and Full Functors

Example

The forgetful functor $F : \mathbf{Mon} \to \mathbf{Set}$ is faithful, but not full (explanation in the whiteboard).

Faithful and Full Functors

Example

The forgetful functor $F : \mathbf{Mon} \to \mathbf{Set}$ is faithful, but not full (explanation in the whiteboard).

Let $(M, \cdot, 1_M)$ and $(N, *, 1_N)$ be two monoids and let $f : M \to N$ be a homomorphism between them.

- Since $F_1\, f = f$, the map F_1 is injective.
- If $g : M \to N$ is any function in $\mathbf{Set}$ such that $g\,(1_M) \neq 1_N$, then g is not a homomorphism between $(M, \cdot, 1_M)$ and $(N, *, 1_N)$. Therefore the map F_1 is not surjective.

Faithful and Full Functors

Exercise 9

Show that the free monoid functor $\mathsf{MList} : \mathbf{Set} \to \mathbf{Mon}$ is faithful, but not full.

Exercise 10 (1.3.5.2)

Let $\mathcal{C}$ be a category with binary products such that, for each pair of objects A, B,

$$\mathcal{C}(A, B) \neq \emptyset. \tag{*}$$

(i) Show that the product functor $\times : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ is faithful.

(ii) Would $- \times -$ still be faithful in the absence of condition (*)?

Preservation and Reflection

Definition

Let P be a property of arrows and let $F : \mathcal{C} \to \mathcal{D}$ be a functor.

(i) The functor F **preserves** the property P iff

$$\text{if } f \text{ satisfies } P \text{ then } F_1\, f \text{ satisfies } P.$$

(ii) The functor F **reflects** the property P iff

$$\text{if } F_1\, f \text{ satisfies } P \text{ then } f \text{ satisfies } P.$$

Preservation and Reflection

Example

Show that all functors preserve isomorphisms.

Preservation and Reflection

Example

Show that all functors preserve isomorphisms.

Example

Show that full and faithful functors reflect isomorphisms.

References

References

- Abramsky, S. and Tzevelekos, N. (2011). Introduction to Categories and Categorical Logic. In: New Structures for Physics. Ed. by Coecke, B. Vol. 813. Lecture Notes in Physics. Springer, pp. 3–94. DOI: 10.1007/978-3-642-12821-9_1 (cit. on p. 2).
- Awodey, S. [2006] (2010). Category Theory. 2nd ed. Vol. 52. Oxford Logic Guides. Oxford University Press (cit. on pp. 63, 66, 67).
- Fong, B., Milewski, B. and Spivak, D. I. (2020). Programming with Categories (DRAFT). URL: http://brendanfong.com/programmingcats.html (cit. on pp. 17–22, 43, 44, 49, 50).