

Categorías y funtores

Mario Román

November 18, 2017

Categorías

Algunas categorías

Funtores

Categorías

Definición (I)

Una categoría \mathcal{C} está formada por

- **objetos** o *puntos*, que llamaremos a, b, c, \dots ,
- y **morfismos** o *flechas* de un objeto a otro, que llamaremos f, g, h, \dots

Que un morfismo f sea de a hacia b se escribe como $f: a \rightarrow b$ o, en ocasiones, $f \in \text{hom}(a, b)$.

Definición (I)

Una **categoría** \mathcal{C} está formada por

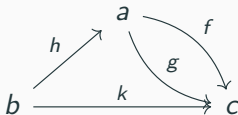
- **objetos** o *puntos*, que llamaremos a, b, c, \dots ,
- y **morfismos** o *flechas* de un objeto a otro, que llamaremos f, g, h, \dots

Que un morfismo f sea de a hacia b se escribe como $f: a \rightarrow b$ o, en ocasiones, $f \in \text{hom}(a, b)$.

Los objetos y las flechas son nociones fundamentales sin definición.

Definición (I)

Hasta aquí, parece un grafo con múltiples aristas.



Hay objetos a, b, c y flechas

- $f: a \rightarrow c$,
- $g: a \rightarrow c$,
- $h: b \rightarrow a$,
- $k: b \rightarrow c$.

Definición (II)

Lo que distingue a las categorías es que las flechas se componen para crear otras flechas. Hay una **operación composición**.

Definición (II)

Lo que distingue a las categorías es que las flechas se componen para crear otras flechas. Hay una **operación composición**.

- La operación **composición**; dadas dos flechas $f: a \rightarrow b$ y $g: b \rightarrow c$, da una flecha $g \circ f: a \rightarrow c$.

$$\begin{array}{ccccc} a & \xrightarrow{f} & b & \xrightarrow{g} & c \\ & \searrow & & \nearrow & \\ & & g \circ f & & \end{array}$$

Definición (II)

Lo que distingue a las categorías es que las flechas se componen para crear otras flechas. Hay una **operación composición**.

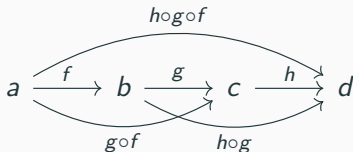
- La operación **composición**; dadas dos flechas $f: a \rightarrow b$ y $g: b \rightarrow c$, da una flecha $g \circ f: a \rightarrow c$.

$$\begin{array}{ccccc} a & \xrightarrow{f} & b & \xrightarrow{g} & c \\ & \searrow & & \nearrow & \\ & & g \circ f & & \end{array}$$

*¡Nótese que no cualesquiera dos flechas pueden componerse!
Una tiene que tener como "dominio" el "codominio" de la otra
para poder ser "componibles".*

Propiedades de la composición

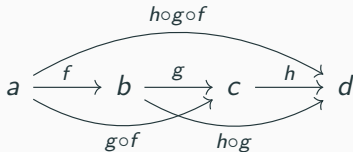
- Es **asociativa**, si tenemos



entonces $h \circ (g \circ f) = (h \circ g) \circ f$ y escribimos simplemente $h \circ g \circ f$.

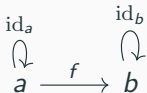
Propiedades de la composición

- Es **asociativa**, si tenemos



entonces $h \circ (g \circ f) = (h \circ g) \circ f$ y escribimos simplemente $h \circ g \circ f$.

- Cada objeto tiene una flecha **identidad**, neutra respecto a la composición



de forma que $f \circ \text{id}_a = f$ y $\text{id}_b \circ f = f$.

Definición formal

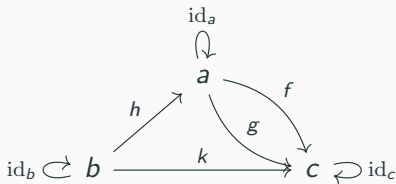
En resumen, una categoría es

1. una colección de **objetos**,
2. con *morfismos* o **flechas** entre ellos,
3. con una **composición** de flechas "componibles",
4. que es **asociativa**,
5. y que tiene una flecha **identidad** en cada objeto.

Algunas categorías

Una categoría cualquiera

Tres **objetos** a, b, c y siete **morfismos**.



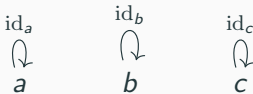
Con la **composición** definida como

- $f \circ h = k$,
- $g \circ h = k$,
- $\text{id}_a \circ h = h$
- $f \circ \text{id}_a = f$
- ...

que se comprueba **asociativa** y con **identidades**.

Una categoría discreta

Una categoría es **discreta** si sólo tiene morfismos identidad. Por ejemplo, la categoría de tres **objetos** con **morfismos**,



donde la **composición** se define de la única forma posible para que sea **asociativa** y con **identidad**: $\text{id}_a \circ \text{id}_a = \text{id}_a$; $\text{id}_b \circ \text{id}_b = \text{id}_b$; $\text{id}_c \circ \text{id}_c = \text{id}_c$.

Una categoría discreta

Una categoría es **discreta** si sólo tiene morfismos identidad. Por ejemplo, la categoría de tres **objetos** con **morfismos**,



donde la **composición** se define de la única forma posible para que sea **asociativa** y con **identidad**: $\text{id}_a \circ \text{id}_a = \text{id}_a$; $\text{id}_b \circ \text{id}_b = \text{id}_b$; $\text{id}_c \circ \text{id}_c = \text{id}_c$.

¡Los conjuntos son lo mismo que las categorías discretas!

Una categoría vacía

La categoría vacía tiene 0 **objetos**, con 0 **morfismos**. La **composición** entre dos morfismos no hay que definirla porque no hay dos morfismos componibles; y como no existe, es claramente **asociativa** y existe una **identidad** para cada uno de los 0 objetos.

Una categoría con un sólo objeto

Una categoría con un sólo **objeto** a e *infinitos* (numerables) **morfismos** $f^0, f^1, f^2, \dots : a \rightarrow a$, todos de a hacia a .



Que se **componen** como $f^n \circ f^m = f^{n+m}$. La composición es **asociativa** y hay una identidad llamada f^0 .

Una categoría con un sólo objeto

Una categoría con un sólo **objeto** a e *infinitos* (numerales)
morfismos $f^0, f^1, f^2, \dots : a \rightarrow a$, todos de a hacia a .



Que se **componen** como $f^n \circ f^m = f^{n+m}$. La composición es **asociativa** y hay una identidad llamada f^0 .

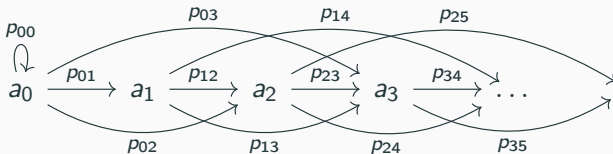
*Esta categoría es el monoide de los naturales con la suma.
¡Todos monoides son categorías de un objeto!*

Una categoría infinita

Una categoría que tiene *infinitos* (numerables) **objetos** a_0, a_1, a_2, \dots
y **morfismos** de la forma $p_{xy}: a_x \rightarrow a_y$ para cualesquiera $x \leq y$.
¡No dibujamos todas las flechas!

Una categoría infinita

Una categoría que tiene *infinitos* (numerables) **objetos** a_0, a_1, a_2, \dots y **morfismos** de la forma $p_{xy} : a_x \rightarrow a_y$ para cualesquiera $x \leq y$.
¡No dibujamos todas las flechas!



La **composición** se define $p_{y,z} \circ p_{x,y} = p_{x,z}$. Se puede ver que es **asociativa** y que tiene una identidad en cada objeto, $p_{x,x}$.

La (enorme) categoría de conjuntos

Una categoría que tiene un **objeto** por cada conjunto que existe; y un morfismo por cada **función** entre conjuntos que existe.

La (enorme) categoría de conjuntos

Una categoría que tiene un **objeto** por cada conjunto que existe; y un morfismo por cada **función** entre conjuntos que existe.

Un dibujo infinito y extraordinariamente complejo iría aquí, pintando todos los conjuntos posibles y todas las funciones posibles.

La (enorme) categoría de conjuntos

Una categoría que tiene un **objeto** por cada conjunto que existe; y un morfismo por cada **función** entre conjuntos que existe.

Un dibujo infinito y extraordinariamente complejo iría aquí, pintando todos los conjuntos posibles y todas las funciones posibles.

La **composición** es la composición usual de funciones, donde $g \circ f$ es la función tal que $(g \circ f)(x) = g(f(x))$. Esta composición es **asociativa** y cada conjunto tiene una función identidad definida $\text{id}(x) = x$.

Parecida a la categoría de conjuntos, pero sólo consideramos los conjuntos que son grupos como **objetos** y sólo consideramos las funciones que son homomorfismos de grupos como **morfismos**.

Parecida a la categoría de conjuntos, pero sólo consideramos los conjuntos que son grupos como **objetos** y sólo consideramos las funciones que son homomorfismos de grupos como **morfismos**.

¡Tenemos que comprobar que la composición de dos homomorfismos de grupos es un homomorfismo de grupos y que la identidad es un homomorfismo de grupos!

Parecida a la categoría de conjuntos, pero sólo consideramos los conjuntos que son espacios topológicos como **objetos** y sólo consideramos las funciones que son continuas como **morfismos**.

Parecida a la categoría de conjuntos, pero sólo consideramos los conjuntos que son espacios topológicos como **objetos** y sólo consideramos las funciones que son continuas como **morfismos**.

¡Tenemos que comprobar que la composición de dos funciones continuas es una función continua y que la identidad es una función continua!

La "categoría" de los tipos en un lenguaje de programación

Los **objetos** son los tipos del lenguaje, como `Int`, `String`, `Bool`, `[Bool]`, `(String, Bool)`, y los **morfismos** son las funciones de un tipo a otro, como

```
isprime :: Int -> Bool
```

```
swap :: (String, Bool) -> (Bool, String)
```

La **composición** está dada como

- $\text{compose } g \ f \ x = g \ (f \ x)$

y es **asociativa** y tiene una **identidad** para cada tipo dada por $\text{id } x = x$.

Funtores

Definición de funtor

Un funtor F de una categoría \mathcal{C} a una categoría \mathcal{D} sería algo así como un homomorfismo de categorías.

Definición de funtor

Un **funtor** F de una categoría \mathcal{C} a una categoría \mathcal{D} sería algo así como un homomorfismo de categorías.

- A cada objeto a de \mathcal{C} , se le asigna un objeto $F(a)$ en \mathcal{D} .

Definición de funtor

Un **funtor** F de una categoría \mathcal{C} a una categoría \mathcal{D} sería algo así como un homomorfismo de categorías.

- A cada objeto a de \mathcal{C} , se le asigna un objeto $F(a)$ en \mathcal{D} .
- A cada morfismo $f: a \rightarrow b$ de \mathcal{C} , se le asigna un morfismo $F(f): F(a) \rightarrow F(b)$ de \mathcal{D} .

Definición de funtor

Un **funtor** F de una categoría \mathcal{C} a una categoría \mathcal{D} sería algo así como un homomorfismo de categorías.

- A cada objeto a de \mathcal{C} , se le asigna un objeto $F(a)$ en \mathcal{D} .
- A cada morfismo $f: a \rightarrow b$ de \mathcal{C} , se le asigna un morfismo $F(f): F(a) \rightarrow F(b)$ de \mathcal{D} .

¡Nótese que la F tiene dos significados!

Definición de funtor II

Además los funtores deben cumplir dos propiedades.

- Respetar composiciones, es decir,

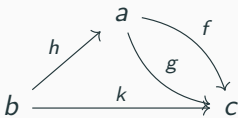
$$F(g \circ f) = F(g) \circ F(f).$$

- Y respetar identidades, es decir,

$$F(\text{id}_a) = \text{id}_{F(a)}.$$

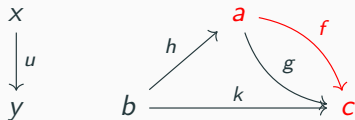
Dibujando un funtor: flechas

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,



Dibujando un funtor: flechas

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

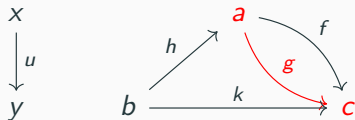


Un **funtor** F estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u): F(x) \rightarrow F(y)$. Por ejemplo:

- $F(x) = a$
- $F(y) = c$
- $F(u) = f$

Dibujando un funtor: flechas

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

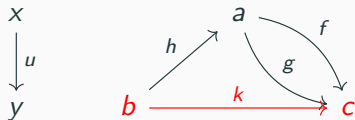


Un **funtor** F estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u): F(x) \rightarrow F(y)$. Por ejemplo:

- $F(x) = a$
- $F(y) = c$
- $F(u) = g$

Dibujando un funtor: flechas

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

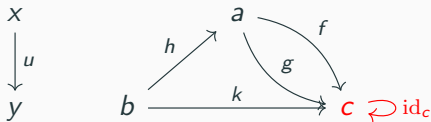


Un **funtor** F estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u): F(x) \rightarrow F(y)$. Por ejemplo:

- $F(x) = b$
- $F(y) = c$
- $F(u) = k$

Dibujando un funtor: flechas

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

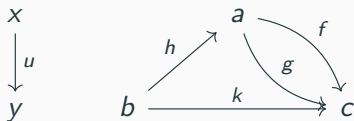


Un **funtor** F estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u): F(x) \rightarrow F(y)$. Por ejemplo:

- $F(x) = c$
- $F(y) = c$
- $F(u) = \text{id}_c$

Dibujando un funtor: flechas

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

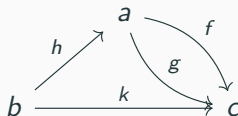
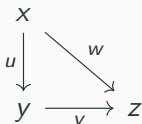


Un **funtor** F estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u): F(x) \rightarrow F(y)$. Por ejemplo:

*Definir un funtor desde la categoría de una flecha es **elegir una flecha**.*

Dibujando un funtor: triángulos

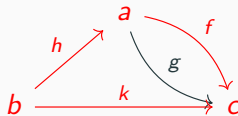
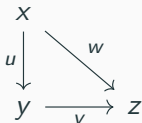
Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,



donde $v \circ u = w$.

Dibujando un funtor: triángulos

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

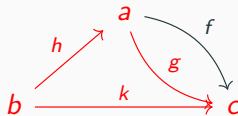
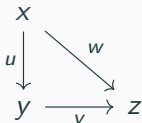


donde $v \circ u = w$. Un funtor F estará definido eligiendo $F(x)$, $F(y)$, $F(z)$ y luego $F(u)$ y $F(v)$ de forma que al componerse den $F(w) = F(v \circ u) = F(v) \circ F(u)$. Por ejemplo:

- $F(u) = h$
- $F(v) = f$
- $F(w) = F(v \circ u) = F(v) \circ F(u) = h \circ f = k$

Dibujando un funtor: triángulos

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

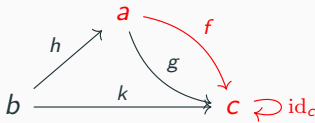
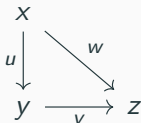


donde $v \circ u = w$. Un **funtor** F estará definido eligiendo $F(x), F(y), F(z)$ y luego $F(u)$ y $F(v)$ de forma que al componerse den $F(w) = F(v \circ u) = F(v) \circ F(u)$. Por ejemplo:

- $F(u) = h$
- $F(v) = g$
- $F(w) = F(v \circ u) = F(v) \circ F(u) = h \circ g = k$

Dibujando un funtor: triángulos

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

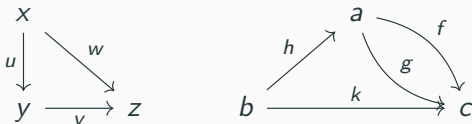


donde $v \circ u = w$. Un funtor F estará definido eligiendo $F(x)$, $F(y)$, $F(z)$ y luego $F(u)$ y $F(v)$ de forma que al componerse den $F(w) = F(v \circ u) = F(v) \circ F(u)$. Por ejemplo:

- $F(u) = f$
- $F(v) = \text{id}_c$
- $F(w) = F(v \circ u) = F(v) \circ F(u) = \text{id}_c \circ f = f$

Dibujando un funtor: triángulos

Sean dos categorías, de las que no dibujamos la identidad. A la izquierda \mathcal{C} y a la derecha \mathcal{D} ,

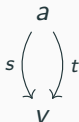


donde $v \circ u = w$. Un **funtor** F estará definido eligiendo $F(x)$, $F(y)$, $F(z)$ y luego $F(u)$ y $F(v)$ de forma que al componerse den $F(w) = F(v \circ u) = F(v) \circ F(u)$. Por ejemplo:

*Definir un funtor desde la categoría de un triángulo es **elegir un triángulo**.*

Dibujando un funtor: grafos

Sean dos categorías, una la llamamos \Rightarrow la dibujamos a la izquierda y la otra es la categoría de conjuntos completa, Set.



Set

Dibujando un funtor: grafos

Sean dos categorías, una la llamamos \Rightarrow la dibujamos a la izquierda y la otra es la categoría de conjuntos completa, Set.

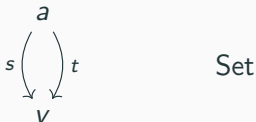


Definir un **funtor** G es elegir:

- un conjunto $G(a)$,
- función $G(s): G(a) \rightarrow G(v)$,
- otro conjunto $G(v)$,
- otra función $G(t): G(a) \rightarrow G(v)$.

Dibujando un funtor: grafos

Sean dos categorías, una la llamamos \Rightarrow la dibujamos a la izquierda y la otra es la categoría de conjuntos completa, Set.



Definir un **funtor** G es elegir:

- un conjunto $G(a)$,
- función $G(s): G(a) \rightarrow G(v)$,
- otro conjunto $G(v)$,
- otra función $G(t): G(a) \rightarrow G(v)$.

¡Esto es equivalente a definir un grafo! Elegimos un conjunto de aristas, uno de vértices, y a cada arista le asignamos un inicio (source) y un final (target). *Los **grafos** son funtores de la categoría \Rightarrow a Set.*

Ejemplo de funtor: palabras

Este es un **endofunctor**, es decir, la categoría de partida y la categoría de llegada serán la misma, ambas serán Set.

Set

Set

Ejemplo de funtor: palabras

Este es un **endofunctor**, es decir, la categoría de partida y la categoría de llegada serán la misma, ambas serán Set.

Set

Set

Para definir ese endofunctor P , asignamos a cada conjunto A otro conjunto $P(A)$, dado por **palabras con letras** en A ; es decir, si $a, b, c \in A$, entonces $abbc, bac, aaac \in P(A)$, por ejemplo.

Ejemplo de funtor: palabras

Este es un **endofunctor**, es decir, la categoría de partida y la categoría de llegada serán la misma, ambas serán Set.

Set

Set

Para definir ese endofunctor P , asignamos a cada conjunto A otro conjunto $P(A)$, dado por **palabras con letras** en A ; es decir, si $a, b, c \in A$, entonces $abbc, bac, aaac \in P(A)$, por ejemplo.

Cada función $f: A \rightarrow B$ puede convertirse en una función $P(f): P(A) \rightarrow P(B)$, que lleva palabras de A en palabras de B aplicándose sobre cada letra. Por ejemplo:

$$P(f)(aacba) = f(a)f(a)f(c)f(b)f(a)$$

Ejemplo de funtor: listas

En programación funcional se trabaja con **endofuntores** de la categoría de tipos.

Types

Types

Ejemplo de functor: listas

En programación funcional se trabaja con **endofuntores** de la categoría de tipos.

Types

Types

Por ejemplo, el **functor lista** $[-]$ lleva cada tipo A al tipo de las listas de elementos de A , llamado $[A]$. Y lleva cada función del tipo $f :: A \rightarrow B$ en una función $[f] :: [A] \rightarrow [B]$ que se suele llamar $\text{map } f$.

$$\text{map } f \ [a, b, c] = [f \ a, f \ b, f \ c].$$