# Categorías y funtores

Mario Román

December 2, 2017

# Outline

— title: Categorías y funtores date: 2017-11-02 —

# Categorías

# Definición (I)

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

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

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

# Definición (I)

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

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

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

*Los objetos y las flechas son nociones fundamentales sin definición.*

Hasta aquí, parece un grafo con múltiples aristas.

$$
\begin{tikzcd}
& a \arrow[dr, bend left, "f"] \arrow[dr, bend right, "g"] & \\
b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}
$$

Hay objetos $a, b, c$ y flechas

- $f \colon a \to c$,
- $g \colon a \to c$,
- $h \colon b \to a$,
- $k \colon b \to c$.

# 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 $f\colon a \to b$ y $g\colon b \to c$, da una flecha $g \circ f\colon a \to c$.

$$\begin{tikzcd} a \arrow[r, "f"] \arrow[rr, bend right, "g \circ f"'] & b \arrow[r, "g"] & c \end{tikzcd}$$

# 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 $f\colon a \to b$ y $g\colon b \to c$, da una flecha $g \circ f\colon a \to c$.

$$
\begin{tikzcd}
a \arrow[r, "f"] \arrow[rr, bend right, "g\circ f"'] & b \arrow[r, "g"] & c
\end{tikzcd}
$$

*¡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

$$\begin{tikzcd} a \arrow[r, "f"] \arrow[rrr, bend left, "h\circ g\circ f"] \arrow[rr, bend right, "g\circ f"'] & b \arrow[r, "g"] \arrow[rr, bend right, "h\circ g"'] & c \arrow[r, "h"] & d \end{tikzcd}$$

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

- Es asociativa, si tenemos

$$\begin{tikzcd} a \arrow[r, "f"] \arrow[rrr, bend left, "h\circ g\circ f"] \arrow[rr, bend right, "g\circ f"'] & b \arrow[r, "g"] \arrow[rr, bend right, "h\circ g"'] & c \arrow[r, "h"] & d \end{tikzcd}$$

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

$$\begin{tikzcd} a \arrow[loop above, "\mathrm{id}_a"] \arrow[r, "f"] & b \arrow[loop above, "\mathrm{id}_b"] \end{tikzcd}$$

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

# 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.

$$
\begin{tikzcd}
 & a \arrow[loop above, "\mathrm{id}_a"] \arrow[dr, bend left, "f"] \arrow[dr, bend right, "g"'] & \\
b \arrow[loop left, "\mathrm{id}_b"] \arrow[ur, "h"] \arrow[rr, "k"] & & c \arrow[loop right, "\mathrm{id}_c"]
\end{tikzcd}
$$

Con la composición definida como

- $f \circ h = k$,
- $g \circ h = k$,
- $\mathrm{id}_a \circ h = h$
- $f \circ \mathrm{id}_a = f$
- $\dots$

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,

$$
\begin{tikzcd}
a \arrow["\mathrm{id}_a", loop above] & b \arrow["\mathrm{id}_b", loop above] & c \arrow["\mathrm{id}_c", loop above]
\end{tikzcd}
$$

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

# Una categoría discreta

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

$$
\begin{tikzcd}
a \arrow[loop above, "\mathrm{id}_a"] & b \arrow[loop above, "\mathrm{id}_b"] & c \arrow[loop above, "\mathrm{id}_c"]
\end{tikzcd}
$$

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

*¡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.

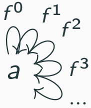
# Una categoría con un sólo objeto

Una categoría con un sólo objeto $a$ e *infinitos* (numerables) morfismos $f^0, f^1, f^2, \dots : a \to a$, 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* (numerables) morfismos $f^0, f^1, f^2, \dots : a \to a$, 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 $a_0, a_1, a_2, \ldots$
y morfismos de la forma $p_{xy} \colon a_x \to a_y$ 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 $p_{xy} \colon a_x \to a_y$ para cualesquiera $x \leq y$.
*¡No dibujamos todas las flechas!*

$$
\begin{tikzcd}
a_0 \arrow[loop above, "p_{00}"] \arrow[r, "p_{01}"] \arrow[rr, bend right, "p_{02}"'] \arrow[rrr, bend left, "p_{03}"] & a_1 \arrow[r, "p_{12}"] \arrow[rr, bend right, "p_{13}"'] \arrow[rrr, bend left, "p_{14}"] & a_2 \arrow[r, "p_{23}"] \arrow[rr, bend right, "p_{24}"'] \arrow[rrr, bend left, "p_{25}"] & a_3 \arrow[r, "p_{34}"] \arrow[rr, bend right, "p_{35}"'] & \dots & {}
\end{tikzcd}
$$

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 $\mathrm{id}(x) = x$.

# Grupos

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.

# Grupos

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!

# Espacios topológicos

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

- `compose g f x = g (f x)`

y es asociativa y tiene una identidad para cada tipo dada por `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 $f\colon a \to b$ de $\mathcal{C}$, se le asigna un morfismo $F(f)\colon F(a) \to F(b)$ 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 $f\colon a \to b$ de $\mathcal{C}$, se le asigna un morfismo $F(f)\colon F(a) \to F(b)$ 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,

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

# Dibujando un funtor: flechas

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

$$
\begin{tikzcd}
x \arrow[d, "u"] & & & a \arrow[dr, bend right, "g"'] \arrow[dr, bend left, "f"] & \\
y & & b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}
$$

# Dibujando un funtor: flechas

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

$$\begin{tikzcd} x \arrow[d, "u"] \\ y \end{tikzcd} \qquad \begin{tikzcd} & {\color{red}a} \arrow[dr, bend left, "g"'] \arrow[dr, bend left=50, red, "f"] & \\ b \arrow[ur, "h"] \arrow[rr, "k"] & & {\color{red}c} \end{tikzcd}$$

Un funtor $F$ estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u)\colon F(x) \to F(y)$. 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 $\mathcal{D}$,

$$
\begin{tikzcd}
x \arrow[d, "u"] \\
y
\end{tikzcd}
\qquad
\begin{tikzcd}
 & a \arrow[dr, bend left, "f"] \arrow[dr, "g"'] & \\
b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}
$$

Un funtor $F$ estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u)\colon F(x) \to F(y)$. 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 $\mathcal{D}$,

$$\begin{tikzcd}
x \arrow[d, "u"] & & a \arrow[dr, bend right, "g"'] \arrow[dr, bend left, "f"] & \\
y & b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}$$

Un funtor $F$ estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u)\colon F(x) \to F(y)$. 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 $\mathcal{D}$,

$$
\begin{tikzcd}
x \arrow[d, "u"] & & & a \arrow[drr, bend left, "f"] \arrow[drr, bend right, "g"'] & & \\
y & & b \arrow[ur, "h"] \arrow[rrr, "k"] & & & c \arrow[loop right, "\mathrm{id}_c"]
\end{tikzcd}
$$

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

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

# Dibujando un funtor: flechas

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

$$
\begin{tikzcd}
x \arrow[d, "u"] \\
y
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & a \arrow[dr, bend right, "g"'] \arrow[dr, bend left, "f"] & \\
b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}
$$

Un funtor $F$ estaría definido eligiendo un $F(x)$, un $F(y)$ y un $F(u)\colon F(x) \to F(y)$. 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 $\mathcal{D}$,

$$
\begin{tikzcd}
x \arrow[d, "u"'] \arrow[dr, "w"] & \\
y \arrow[r, "v"'] & z
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & a \arrow[dr, bend right, "g"'] \arrow[dr, bend left, "f"] & \\
b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}
$$

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 $\mathcal{D}$,

$$\begin{tikzcd} x \arrow[d, "u"'] \arrow[dr, "w"] & \\ y \arrow[r, "v"'] & z \end{tikzcd} \qquad \begin{tikzcd} & a \arrow[dr, bend left, "f"] \arrow[dr, bend right, "g"'] & \\ b \arrow[ur, "h"] \arrow[rr, "k"] & & c \end{tikzcd}$$

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 $\mathcal{D}$,

$$
\begin{tikzcd}
x \arrow[d, "u"'] \arrow[dr, "w"] & \\
y \arrow[r, "v"'] & z
\end{tikzcd}
\qquad
\begin{tikzcd}
& a \arrow[dr, bend right, "g"'] \arrow[dr, bend left, "f"] & \\
b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}
$$

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 $\mathcal{D}$,

$$\begin{tikzcd}
x \arrow[d, "u"'] \arrow[dr, "w"] & \\
y \arrow[r, "v"'] & z
\end{tikzcd}
\qquad
\begin{tikzcd}
& a \arrow[dr, bend left, "f"] \arrow[dr, bend right, "g"'] & \\
b \arrow[ur, "h"] \arrow[rr, "k"] & & c \arrow[loop right, "\mathrm{id}_c"]
\end{tikzcd}$$

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:

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

# Dibujando un funtor: triángulos

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

$$
\begin{tikzcd}
x \arrow[d, "u"'] \arrow[dr, "w"] & \\
y \arrow[r, "v"'] & z
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & a \arrow[dr, bend right, "g"] \arrow[dr, bend left, "f"] & \\
b \arrow[ur, "h"] \arrow[rr, "k"] & & c
\end{tikzcd}
$$

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 $\rightrightarrows$ la dibujamos a la izquierda y la otra es la categoría de conjuntos completa, Set.

$$
\begin{tikzcd}
a \arrow[d, bend right, "s"'] \arrow[d, bend left, "t"] \\
v
\end{tikzcd}
\qquad\qquad \mathrm{Set}
$$

# Dibujando un funtor: grafos

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

$$
\begin{tikzcd}
a \arrow[d, bend right, "s"'] \arrow[d, bend left, "t"] \\
v
\end{tikzcd}
\qquad\qquad \mathrm{Set}
$$

Definir un funtor $G$ es elegir:

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

# Dibujando un funtor: grafos

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

$$
\begin{tikzcd}
a \arrow[d, bend right, "s"'] \arrow[d, bend left, "t"] \\
v
\end{tikzcd}
\qquad\qquad \mathsf{Set}
$$

Definir un funtor $G$ es elegir:

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

¡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 $\rightrightarrows$ a Set.

# Ejemplo de funtor: palabras

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

$$\mathrm{Set} \qquad \mathrm{Set}$$

# Ejemplo de funtor: palabras

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

$$\text{Set} \qquad \text{Set}$$

Para definir ese endofuntor $P$, 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 endofuntor, es decir, la categoría de partida y la categoría de llegada serán la misma, ambas serán Set.

$$\mathrm{Set} \qquad \mathrm{Set}$$

Para definir ese endofuntor $P$, 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 $f\colon A \to B$ puede convertirse en una función $P(f)\colon P(A) \to P(B)$, 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.

$$\text{Types} \qquad \text{Types}$$

# Ejemplo de funtor: listas

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

$$\text{Types} \qquad \text{Types}$$

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

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