Teoría de Categorías

Una visión general con aplicaciones a la computación

Juan Marín Noguera

$$
\begin{tikzcd}
S'Sb \arrow[r, "\sigma'_{Sb}"] \arrow[d, "S'\sigma_b"'] \arrow[dr, "\sigma' \circ \sigma"] & T'Sb \arrow[r, "\tau'_{Sb}"] \arrow[d, "T'\sigma_b"] & U'Sb \arrow[d, "U'\sigma_b"] \\
S'Tb \arrow[r, "\sigma'_{Tb}"'] \arrow[d, "S'\tau_b"'] & T'Tb \arrow[r, "\tau'_{Tb}"] \arrow[d, "T'\tau_b"'] \arrow[dr, "\tau' \circ \tau"] & U'Tb \arrow[d, "U'\tau_b"] \\
S'Ub \arrow[r, "\sigma'_{Ub}"'] & T'Ub \arrow[r, "\tau'_{Ub}"'] & U'Ub
\end{tikzcd}
$$

Trabajo de Fin de Grado en Matemáticas

Tutor: Alberto del Valle Robles
Facultad de Matemáticas
Universidad de Murcia

20 de Junio de 2023

Índice general

Índice de figuras

Introduction

Category theory can be roughly defined as the study of mathematical structures. If we look at the structure of the objects of study in several fields of mathematics, we quickly notice that, in some intuitive sense, many of these structures are quite similar. For example, most of the time the objects of study are sets equipped with some additional elements, like binary operations satisfying certain properties, multiplication by elements of a given field of scalars, a collection of subsets that are considered open, etc. In addition, we typically study the class of the functions between those sets that "play well" with the additional elements, generally called homomorphisms, continuous functions, etc., to such an extent that, whenever a function is considered, the first question to ask is, does this function belong to the class of functions of interest? Going deeper, there are constructions on those objects that are, intuitively speaking, "equivalent" to another construction in another kind of objects, even if the structures of the objects involved are nothing alike. Category theory attempts to formalize and thus clarify these intuitions.

Category theory was introduced by mathematicians Samuel Eilenberg and Saunders Mac Lane in a paper published in 1942, where they introduced several of the core concepts of the field as part of their studies in homological algebra.[4, p. 29] Category theory has also been referred to as "abstract nonsense" or "general abstract nonsense", a term coined by mathematician Norman Steenrod to affectionately refer to the kinds of very abstract constructions and methods developed in category theory and related fields of mathematics.[10, preface, p. x]

Nevertheless, the study of abstract nonsense is useful for a number of reasons. First of all, it provides a common vocabulary for the different fields of mathematics, which is useful for learning as having the same name for the "same" concept allows to better develop an intuition of the concept. Category theory is usually not covered by undergraduate curricula, but because the other subjects are studied with a consistent vocabulary, students grasp the idea of concepts that are actually category-theoretic.

Second, it serves as a tool for mathematical thought. Because similar concepts in different fields often have similar properties, it is natural for someone with a good understanding of structures to relate to ask oneself if something that happens in another field of mathematics, or another category, also happens in the one being considered.

Third, because category theory is not about intuitions but about formalisms, it provides mechanisms to transfer mathematical knowledge from one context to another, as well as general, abstract theorems that, when appropriately made concrete, allow proving a great variety of results. One example of this is the Yoneda lemma, which can be used to prove that every group can be viewed as a subgroup of a permutation proof but also that every row operation on matrices can be defined by left multiplication by a square matrix.

Fourth, the methods of reasoning developed in category theory can sometimes be applied to other fields

Finally, when a new field or a new kind of mathematical object is discovered, or invented, it is often possible to use category theory to guide the study of this new kind of object and to "automatically" derive properties for it. An example of this is explored in appendix A. Most of computer science has been developed after category theory, so there are many concepts that have been developed by applying general concepts of category theory to a suitable category. To be fair, a lot of computer science theory is independent from category theory, either because it was developed before category theory was widely known by mathematicians or because it was not developed by mathematicians in the first place, and a lot of it is simply unrelated to categories. Nevertheless, as soon as the usefulness of category theory to write and analyze computer programs was realized, a lot of its concepts, both simple and advanced, have been imported and successfully applied to create programs with less bugs and that are easier to maintain, as well as to design programming languages that support these concepts and formal reasoning tools that use category theory to support the task of verifying correctness of programs in critical systems.

The basic object of study of category theory is a category, which consists of a collection of objects, a collection of morphisms, operations domain (dom) and codomain (cod) that take morphisms to objects, an identity operation (1) that takes objects to morphisms, and a composition operation ($\circ$) that takes pairs of morphisms (g, f) with $\mathrm{dom} g = \mathrm{cod} f$ to morphisms. Furthermore, these operations have to satisfy some properties. First, the domain and codomain of an object's identity are both equal to the object. Second, the domain of a composition is that of the second operand, and its codomain is that of the first operand. Third, the composition is associative. And finally, the composition of a morphism with the identity of its domain, as well as that of the identity of its codomain with the morphism, are both equal to the original morphism. This definition is independent of set theory as it only relies on predicate logic, although we generally interpret it over set theory for convenience.

Categories represent particular mathematical concepts. Here objects might be sets with some structure and morphisms might be functions between those sets with certain properties, so we might have categories for e.g. groups, topological spaces, fields, vector spaces, Banach spaces, or even for all sets. However, objects need not be sets and morphisms need not be functions, so we could have a category where objects are the natural numbers and morphisms from one number to another (that is, morphisms with the first number as the domain and the second one as the codomain) are matrices (over some ring) with so many columns and rows, respectively. Then identities of objects would be the corresponding identity matrices and composition would be the matrix product.

By abstracting away from the way objects and morphisms are represented in a given category, and looking only at the relationships between those, we can define constructions in the same way for several categories, thus formalizing the notion of these constructions being intuitively equivalent. For example, the product of a family a objects is defined as an object along with a family of morphisms, called the projections, going from the product object to each object in the family, such that for any other family of morphisms from any object to the family of objects, there is a morphism from the domain object to the product that, when composed with the projections, yields this family of morphisms. This defines the Cartesian product of sets, the product topological space, the product of groups, of ring modules, and the infimum of a subset of a partially-ordered set.

If we "invert" the morphisms in a category by swapping domain and codomain and inverting the direction of composition, the products of this category are called coproducts of the original one, and they represent the disjoint union of sets, the disjoint union topological space, the direct sum of groups or ring modules, and the supremum of a subset of a partially-ordered set. It turns out that many useful concepts can be derived from others this way.

These definitions can often be described and reasoned about visually with commutative diagrams. If we represent morphisms as arrows between its domain and its codomain, we can draw diagrams that look like graphs where vertices are objects of the category and edges are morphisms, and we say that a diagram commutes if any two paths between two vertices yield the same morphism when all the morphisms across each path are composed together. If the morphisms are functions, we can fix an element of a commutative diagram and "propagate" its value following arrows to get a desired equality, a method known as diagram chasing.

In order to relate concepts from different categories we use functors, which are functions between the objects and morphisms of the two categories that respect the operations. A nice aspect of category theory is that it is reflexive, in the sense that it can be used to analyze itself, so we can define a category whose objects are smaller categories and whose morphisms are functors and we automatically get concepts like products of categories.

We can even define categories where the objects are functors between two given categories, and then the morphisms are natural transformations. Natural transformations are means to relate the images of two functors in a way that plays well with the functors. Whenever some operation in a category is considered "natural", or the "natural" way to do it, natural transformations are usually involved. The go-to example is the standard isomorphism between a finite-dimensional vector space and its double dual. This concept is so important that, according to Mac Lane ([4, p. 18]), " 'category' has been defined in order to define 'functor' and 'functor' has been defined in order to define 'natural transformation' ".

Functors allow us to define the concept of limit, which generalizes that of product as well as other concepts such as that of equalizer, which itself extends the algebraic concept of kernel of a homomorphism. Similarly we have the concept of colimits, which extends concepts like coproducts and quotient sets and spaces.

Limits and colimits are themselves a special case of universal arrows in a category of functors. Roughly speaking, universal arrows formalize the general idea of an entity that satisfies some property and such that any other entity satisfying the property is related to that one entity in a unique way. The most common instance of this rather abstract description is that of a free object in a construct, such as a the free group generated by a set or the free R-module of a given dimension.

Universal arrows are defined in terms of a functor, and when every object in the codomain of the functor admits a universal arrow, we can build an adjunction, a bidirectional "bridge" between two categories that satisfies some coherence properties.

In turn, adjunctions can be used to reason about monads, abstract structures made up of an endofunctor—a functor from a category to itself—and two natural transformations that allow to move, via morphisms, from an object to its image under the functor, and to "coalesce" repeated applications of an endofunctor over an object into only one application. Monads have two main applications. The first one, most relevant to mathematics in general, is that they can be used to describe a lot of common algebraic categories from other categories, typically the category of sets. Since the category of sets is well-understood, monads can be used to reason about these other categories. The second application, more relevant to computer science, is that it allows reasoning about the endofunctor itself. Endofunctors in programming are often used to provide a "context" for computations, and monads can be used to thread computations over that context by leveraging the two natural transformations.

This work is guided by two main goals. The first goal is to understand and explain the main concepts of category theory and relate them to as many other fields of mathematics as reasonably possible, illustrating the unifying character of category theory. The second goal is to show how several concepts of category theory can be applied in practice to computer science, and in particular to programming language design and formal verification. This has been the main motivation for covering monads, as they are incredibly useful to model computations and interactivity in particular, even though the topic of monads is worth studying by itself for the reasons mentioned above. On the other hand, since the concrete explanation of how monads are used in programming is more computer-scientific and not as mathematical in nature, this topic has been included as an appendix.

Capítulo 1

Categorías

Al estudiar distintas ramas de las matemáticas de manera formal es inevitable notar ciertos patrones que se repiten. Todas empiezan con los axiomas que caracterizan los objetos de estudio, como pueden ser los grupos, espacios vectoriales, espacios topológicos, variedades, etc., y pese a que estos se definen de forma muy diferente, en la mayoría de casos se pueden definir naturalmente conceptos como subobjetos, objetos producto u objetos cociente, entre otros. Además, junto a estos objetos se estudian funciones entre objetos que cumplen ciertas propiedades, como las aplicaciones lineales en el álgebra lineal, las aplicaciones continuas en la topología y los homomorfismos en el álgebra abstracta.

Estos patrones no son una coincidencia, pues las propiedades generales que exhiben son las mismas en todos los casos. Por ejemplo, la clase de funciones «destacables» entre objetos es cerrada para la composición, y el producto de subobjetos de dos objetos es un subobjeto del producto de los objetos originales. Esto motiva el estudio de dichos patrones como parte de las matemáticas, con el fin de obtener propiedades aplicables a muchas de las áreas existentes o, incluso, a áreas desarrolladas posteriormente, como veremos que ocurre con la teoría de la computación. Así, en la teoría de categorías, los objetos de estudio son representaciones de los conceptos fundamentales de otras teorías matemáticas, que podemos representar como la clase de los objetos de estudio junto con la clase de funciones destacables entre ellos, llamadas *morfismos*, dando lugar al concepto de *categoría*. Este capítulo introduce las categorías y estudia sus propiedades de manera general, y se basa principalmente en [1, caps. 3, 4, 7 y 10].

Definición 1.1. Una *categoría* es una tupla $\mathcal{C}$ formada por los siguientes elementos:

1. Una clase $\mathrm{Ob}(\mathcal{C})$ de *objetos*.
2. Una clase $\mathrm{Mor}(\mathcal{C})$ de *morfismos*.
3. Dos funciones $\mathrm{dom}, \mathrm{cod} : \mathrm{Mor}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{C})$, llamadas respectivamente *dominio* y *codominio*.

   Para $f \in \mathrm{Mor}(\mathcal{C})$, escribimos $f : a \to b$ si $\mathrm{dom} f = a$ y $\mathrm{cod} f = b$, y llamamos $\hom_{\mathcal{C}}(a, b)$ o simplemente $\hom(a, b)$ a la clase de morfismos $f : a \to b$.
4. Una función $\circ : \bigcup_{a,b,c}(\hom(b, c) \times \hom(a, b)) \to \mathrm{Mor}(\mathcal{C})$ que a cada $f : a \to b$ y $g : b \to c$ le asigna su *composición* $g \circ f : a \to c$, y que debe ser asociativa, es decir, para $f : a \to b$, $g : b \to c$ y $h : c \to d$, $h \circ (g \circ f) = (h \circ g) \circ f$.
5. Una función $1 : \mathrm{Ob}(\mathcal{C}) \to \mathrm{Mor}(\mathcal{C})$ que a cada objeto a le asigna la *identidad* $1_a : a \to a$, que cumple que, para $f : a \to b$, $f = 1_b \circ f = f \circ 1_a$.

Es común en teoría de categorías representar situaciones con diagramas, en los que los vértices representan objetos y las flechas representan morfismos, y una flecha aparece punteada si su existencia se debe a la existencia de las otras flechas del diagrama. No se suelen representar las flechas identidad ni la composición de otras flechas del diagrama.

Un diagrama *conmuta* si, dados dos caminos cualesquiera del diagrama con el mismo objeto de origen y de destino, la composición de los morfismos en cada camino coincide. Así, por ejemplo, los axiomas de la composición e identidad de categorías se pueden expresar como en la figura 1.

Ejemplo 1.2. Ciertos conceptos fundacionales se pueden ver como categorías:

1. La categoría **Set** tiene como objetos todos los conjuntos y como morfismos todas las funciones, cualificadas por su dominio y codominio, con la composición e identidad obvias.

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

(A) Asociatividad

$$\begin{tikzcd} a \arrow[r, "1_a"] \arrow[d, "f"'] \arrow[dr, "f"] & a \arrow[d, "f"] \\ b \arrow[r, "1_b"'] & b \end{tikzcd}$$

(B) Elemento neutro

FIGURA 1. Axiomas de categoría.

2. Un *preorden* es una relación reflexiva y transitiva, que se puede ver intuitivamente como un orden parcial entre clases de equivalencia. Un *conjunto preordenado* es un conjunto con un preorden asociado, y llamamos **Prord** a la categoría de los conjuntos preordenados cuyos morfismos son funciones que conservan el preorden, es decir, funciones $f : (a, \preceq) \to (b, \preccurlyeq)$ tales que para $x, y \in a$, $x \preceq y \implies f(x) \preccurlyeq f(y)$.
3. La categoría **Ord** tiene como objetos los conjuntos parcialmente ordenados y como morfismos las funciones que conservan el orden. **Lat** es similar pero los objetos son retículos y los morfismos deben además preservar supremos e ínfimos de pares de elementos.
4. La categoría **Grph** tiene como objetos los grafos dirigidos, permitiendo ejes reflexivos, y como morfismos las funciones entre los vértices de dos grafos que llevan ejes del primer grafo a ejes del segundo.

El apartado 3 de la lista anterior lleva naturalmente al concepto de subcategoría.

DEFINICIÓN 1.3. Una categoría $\mathcal{B}$ es una *subcategoría* de una $\mathcal{C}$ si $\mathrm{Ob}(\mathcal{B}) \subseteq \mathrm{Ob}(\mathcal{C})$, $\mathrm{Mor}(\mathcal{B}) \subseteq \mathrm{Mor}(\mathcal{C})$ y las funciones dominio, codominio, composición e identidad son restricciones de las correspondientes funciones de $\mathcal{C}$, y es una *subcategoría completa* si además, para $a, b \in \mathrm{Ob}(\mathcal{B})$, $\hom_{\mathcal{B}}(a, b) = \hom_{\mathcal{C}}(a, b)$.

Así, **Lat** es una subcategoría no completa de **Ord**, mientras que **Ord** es una subcategoría completa de **Prord**, y esta a su vez es una subcategoría completa de **Grph** si consideramos las relaciones como grafos dirigidos.

1. Categorías algebraicas

EJEMPLO 1.4. En álgebra, muchas categorías se definen de manera similar:

1. **Smgrp**, la categoría de los semigrupos con las funciones que conservan su operación.
2. **Mon**, la subcategoría no completa de **Smgrp** de los monoides con las funciones que conservan su operación y elemento identidad.
3. **Grp**, la subcategoría completa de **Mon** formada por los grupos y sus homomorfismos.
4. **Ab**, la subcategoría de **Grp** de grupos abelianos.
5. **Rng**, la categoría de anillos unitarios y sus homomorfismos. Incluímos aquí el anillo trivial en el que $1 = 0$.
6. **CRng**, la subcategoría completa de **Rng** de los anillos unitarios conmutativos.

En todos estos casos los objetos son conjuntos con una serie de operaciones, y los morfismos son funciones que respetan esas operaciones en el sentido evidente. La lista de operaciones se puede modelar como sigue.[4, p. 120]

DEFINICIÓN 1.5. Un *conjunto graduado* es un conjunto I de *operadores* junto con una función $a : I \to \mathbb{N}$ llamada *aridad*. Una *acción* de I en un conjunto S es una familia $\{\mu_i : S^{a(i)} \to S\}_{i \in I}$ de operaciones en S asociadas a los operadores de I.

Así, por ejemplo, los operadores en **Ab** serían $+$ de aridad 2, $-$ de aridad 1 y 0 de aridad 0. Queda definir las propiedades de los operadores.

Definición 1.6. Sea (I, a) un conjunto graduado:

1. El conjunto de *operadores derivados* de I es el conjunto graduado Λ formado por las siguientes expresiones:
   a) El *operador identidad*, *id*, de aridad 1.
   b) Todos los operadores de I con su aridad en I.
   c) Para $\omega \in \Lambda$ de aridad n e $i_1, \dots, i_n \in \Lambda$ de aridades $a_1, \dots, a_n$, $\omega(i_1, \dots, i_n)$ de aridad $a_1 + \cdots + a_n$.
   d) Para $\lambda \in \Lambda$ de aridad n y $f : \{1, \dots, n\} \to \{1, \dots, m\}$, λ_f de aridad m.
2. Si μ es una acción sobre I, la *extensión* de μ a Λ es la acción ν sobre Λ definida por las siguientes ecuaciones:
$$\begin{aligned} \nu_i &:= \mu_i, \quad i \in I; \\ \nu_{id}(x) &:= x; \\ \nu_{\lambda_f}(x_1, \dots, x_m) &:= \nu_\lambda(x_{f(1)}, \dots, x_{f(n)}); \\ \nu_{\omega(i_1,\dots,i_n)}(x_{11}, \dots, x_{1a_1}, \dots, x_{n1}, \dots, x_{na_n}) &:= \\ &:= \nu_\omega(\nu_{i_1}(x_{11}, \dots, x_{1a_1}), \dots, \nu_{i_n}(x_{n1}, \dots, x_{na_n})). \end{aligned}$$
3. Una *identidad* sobre I es un par (λ, σ) de operadores Λ de igual aridad.
4. Una acción μ sobre I *satisface* una identidad (λ, σ) sobre I si $\nu_\lambda = \nu_\sigma$, donde ν es la extensión de μ a Λ.

Normalmente las identidades (λ, σ) se representan como igualdades $\lambda = \sigma$, y λ y σ se representan de forma obvia como expresiones que dependen de los operadores base y una serie de parámetros de entrada, de modo que las propiedades de **Ab** son $(x + y) + z = x + (y + z)$, $0 + x = x$, $x + (-x) = 0$ y $x + y = y + x$.

Definición 1.7. Sean Ω un conjunto graduado finito y E un conjunto finito de identidades sobre Ω.

1. Una (Ω, E)-álgebra es un par (S, μ) formado por un conjunto S y una acción μ de Ω sobre S que satisface las identidades en E.
2. Una *variedad algebraica* es una categoría (Ω, E)-**Alg** cuyos objetos son las (Ω, E)-álgebras y cuyos morfismos $(A, \mu) \to (B, \nu)$ son las funciones $f : A \to B$ tales que, para $\omega \in \Omega$ de aridad n y $x_1, \dots, x_n \in A$, $f(\mu_\omega(x_1, \dots, x_n)) = \nu_\omega(f(x_1), \dots, f(x_n))$.

Claramente todas las categorías del ejemplo 1.4 son variedades algebraicas, y de hecho podemos ver **Set** como $(\emptyset, \emptyset)$-**Alg**, pero sin embargo **Field**, la subcategoría completa de **CRng** de los cuerpos (incluyendo el cuerpo trivial), no es una variedad algebraica, ya que requiere una propiedad de la inversa del producto, que no está definida en el 0.

Otra categoría interesante es R-**Mod**, la clase de módulos de un anillo conmutativo R y homomorfismos de R-módulos. $\mathbb{Z}$-**Mod** es esencialmente **Ab** y, para un cuerpo K, K-**Mod** es la categoría de K-espacios vectoriales, por lo que la escribimos como K-**Vec** o simplemente **Vec** si $K = \mathbb{R}$.

2. Categorías abstractas

Hasta ahora, en todas las categorías que hemos visto, los objetos son conjuntos con estructura y los morfismos son funciones entre ellos. Las categorías de esta forma se llaman *constructos*[1], y aunque son muy comunes, también hay muchas categorías relevantes que no son constructos.

Ejemplo 1.8. Para un anillo R, la categoría R-**Mat** tiene como objetos los números naturales, como morfismos de n a m las matrices de tamaño $m \times n$, como composición el producto de matrices y como identidad la matriz identidad del tamaño correspondiente.

[1]Veremos una definición más abstracta de constructo al estudiar los funtores.

Ejemplo 1.9. Algunas estructuras matemáticas se pueden ver como categorías.

1. Una categoría es *discreta* si sólo tiene los morfismos identidad. Un conjunto, o en general una clase, se puede ver como una categoría discreta cuyos objetos son los elementos del conjunto.
2. Una categoría es *fina* si sus *conjuntos hom* (conjuntos de la forma $\hom(a, b)$) tienen como máximo un elemento. Un conjunto preordenado $(X, \preceq)$ se puede ver como una categoría fina cuyos objetos son los elementos del conjunto y tal que, para $x, y \in X$, $\hom(x, y)$ está habitado si y sólo si $x \preceq y$.
3. Un monoide se puede ver como una categoría con un solo objeto, cuyos morfismos son los elementos del monoide y donde la identidad es su elemento neutro y la composición es el producto.

Ejemplo 1.10. Las siguientes categorías se usan principalmente en el estudio de categorías más complicadas:

1. La categoría vacía, $\mathbf{0}$, sin objetos.
2. La categoría discreta unipuntual, $\mathbf{1}$.
3. La categoría discreta de dos objetos, $\mathbf{2}$.
4. La categoría $\downarrow$, con dos objetos y una sola flecha de uno a otro ($\bullet \to \bullet$).
5. La categoría $\downdownarrows$, con dos objetos y dos morfismos de uno a otro ($\bullet \rightrightarrows \bullet$).

3. Categorías topológicas y analíticas

La principal categoría topológica es **Top**, formada por los espacios topológicos y las funciones continuas entre ellos.

Ejemplo 1.11. Algunos constructos usados en topología tienen los mismos objetos pero distintas clases de morfismos, permitiendo estudiar los objetos desde distintas perspectivas. Por ejemplo, las siguientes tres categorías tienen como objetos los espacios métricos:

1. $\mathbf{Met_c}$, con las funciones continuas.
2. $\mathbf{Met_u}$, con las funciones uniformemente continuas.
3. **Met**, con las contracciones, funciones continuas que «acercan» los puntos.

Asimismo tenemos la categoría $\mathbf{Ban_b}$ de los espacios de Banach con las formas lineales acotadas y **Ban** con las contracciones lineales.

Ejemplo 1.12. Sea X un espacio topológico. Recordemos que, para $x, y \in X$, un camino de x a y es una función continua $f : [0, 1] \to X$ con $f(0) = x$ y $f(1) = y$, y que dos caminos f y g de x a y son homotópicamente equivalentes si existe $F : [0, 1] \times [0, 1] \to X$ tal que, para $s, t \in [0, 1]$, $F(t, 0) = f(t)$, $F(t, 1) = g(t)$, $F(0, s) = x$ y $F(1, s) = y$. Entonces el *grupoide fundamental*[4, p. 20] de X es una categoría $\pi(X)$ cuyos objetos son los puntos de X y tal que $\hom(x, y)$ es el conjunto cociente de caminos de x a y por equivalencia homotópica, tomando la concatenación de (clases de) caminos como composición y la clase del camino constante como morfismo identidad.

Para $x \in X$, $\hom(x, x)$ es un grupo con la composición, y si existe un camino $f : x \to y$, $\hom(x, x)$ y $\hom(y, y)$ son isomorfos por el isomorfismo $g \rightsquigarrow fgf^{-1}$, definiendo f^{-1} de la forma obvia, con lo que si X es conexo por arcos, $\hom(x, x)$ es el mismo para todo x salvo isomorfismo, y se llama *grupo fundamental* de X. Esto permite tratar espacios topológicos de manera algebraica.

4. Isomorfismos

Una vez vistas las categorías más importantes de distintas áreas, pasamos a ver algunas propiedades de sus objetos y morfismos. Por ejemplo, en álgebra, se suele definir un isomorfismo como un homomorfismo biyectivo cuya inversa también es un homomorfismo. En topología tenemos también el concepto de homeomorfismo, que viene a ser lo mismo salvo por el cambio de terminología.

Definición 1.13. Un morfismo $f : a \to b$ es un *isomorfismo* si existe otro morfismo $g : b \to a$ tal que $g \circ f = 1_a$ y $f \circ g = 1_b$, en cuyo caso escribimos $f : a \cong b$ y decimos que a y b son *isomorfos*, $a \cong b$, y que g es el *inverso* de f, $g = f^{-1}$.

El inverso de un isomorfismo es único, pues si $f : a \to b$ tiene inversos $g, h : b \to a$, entonces $g = g \circ 1_b = g \circ f \circ h = 1_a \circ h = h$.

Ejemplo 1.14.

1. Son isomorfismos las identidades, el inverso de un isomorfismo y la composición de isomorfismos.
2. En **Smgrp**, **Mon**, **Grp**, **Ab**, **Rng** y R-**Mod**, el concepto de isomorfismo categórico se corresponde con el concepto usual de isomorfismo.
3. En **Top** y $\mathbf{Met_c}$, los isomorfismos son los homeomorfismos.
4. En $\mathbf{Ban_b}$, los isomorfismos son los isomorfismos topológicos, es decir, los isomorfismos vectoriales que son homeomorfismos.
5. En **Met** los isomorfismos son las isometrías, mientras que en **Ban** son los isomorfismos isométricos.
6. En **Set**, los isomorfismos son las funciones biyectivas.
7. Un *grupoide* es una categoría en la que todos los morfismos son isomorfismos. Esto ocurre, por ejemplo, en el grupoide fundamental. Entonces un grupo es un grupoide con un solo objeto.
8. En un conjunto preordenado visto como categoría, dos objetos a y b son isomorfos si y sólo si $a \preceq b \preceq a$, por lo que en particular un conjunto (parcialmente) ordenado visto como categoría no tiene isomorfismos salvo las identidades.
9. En un monoide visto como categoría, los isomorfismos son los elementos invertibles.
10. En R-**Mat**, los isomorfismos son las matrices invertibles.

5. Objetos iniciales y finales

En álgebra, muchas categorías tienen un objeto trivial, como el cuerpo trivial, el espacio vectorial trivial, etc. A la hora de definir la topología trivial, sin embargo, no está claro si deberíamos tomar la topología vacía o la unipuntual, y ninguna de las dos opciones es enteramente satisfactoria. Esto se debe a que, al hablar de un objeto trivial estamos juntando dos propiedades: por un lado, el objeto trivial está contenido en todos los demás, y por otro, siempre es posible pasar de un objeto al objeto trivial por un morfismo.

Definición 1.15. Un objeto a es *inicial* si para cualquier otro objeto x existe un único morfismo $f_x : a \to x$, es *final* si para cualquier otro objeto x existe un único morfismo $g_x : x \to a$, y es *cero* si es inicial y final.

Ejemplo 1.16. Muchas categorías típicas tienen objetos iniciales y finales.

1. En **Mon**, **Grp** y **Ab**, el grupo trivial es un objeto cero; en **CRng** y **Field** lo es el cuerpo trivial, y en R-**Mod** lo es el módulo trivial.
2. En **Set**, el conjunto vacío $\emptyset$ es inicial, mientras que los conjuntos unipuntuales $\{*\}$ son finales. Lo mismo ocurre en **Top** y en **Ord** dotando a $\emptyset$ y $\{*\}$ de la única estructura posible.
3. En un conjunto ordenado visto como categoría, un objeto inicial es un mínimo y un objeto final es un máximo. En particular el único conjunto ordenado con un cero es el unipuntual.
4. En R-**Mat**, el 0 es un cero, pues para cada m hay una única matriz de tamaño $0 \times m$ y una de tamaño $m \times 0$.

Podríamos preguntarnos si categorías como **Set** o **Top** tienen un cero. La respuesta es que no, como se deduce de las siguientes proposiciones.

Proposición 1.17. *Los objetos iniciales son únicos salvo isomorfismo:*

1. *Si a y b son objetos iniciales (de la misma categoría), entonces son isomorfos.*

Demostración. Por hipótesis existen $f : a \to b$ y $g : b \to a$, pero como sólo hay un morfismo $a \to a$, este debe ser 1_a y por tanto $g \circ f = 1_a$, y análogamente $f \circ g = 1_b$, con lo que f es un isomorfismo. □

2. *Si $a \cong b$ y a es inicial, entonces b es inicial.*

Demostración. Sean $f : b \to a$ un isomorfismo y x un objeto cualquiera, existe un único $h : a \to x$ y por tanto $h \circ f$ es un morfismo de b a x, que es único ya que, para $k : b \to x$, $k \circ f^{-1} : a \to x$ y así $k \circ f^{-1} = h$, con lo que $k = h \circ f$. □

La siguiente proposición se prueba de forma análoga.

Proposición 1.18. *Los objetos finales son únicos salvo isomorfismo:*

1. *Si a y b son objetos finales, entonces son isomorfos.*
2. *Si $a \cong b$ y a es final, entonces b es final.*

Así, como $\emptyset$ y $\{*\}$ no son isomorfos en **Set**, **Top** ni **Ord** (ni en ningún otro constructo), estas categorías no tienen objetos cero. En general diremos que un objeto con una cierta propiedad es *único salvo isomorfismo* si los objetos que tienen esa propiedad son precisamente los que son isomorfos a dicho objeto.

Corolario 1.19. *En los grupoides no vacíos, son equivalentes:*

1. *Existe un objeto inicial.*
2. *Existe un objeto final.*
3. *Existe un objeto cero.*
4. *Todos los objetos son cero.*

En particular, en el grupoide fundamental $\pi(X)$ (con X no vacío), esto ocurre si y sólo si X es simplemente conexo.

Demostración. La equivalencia ($1 \iff 2$) se debe a que, por unicidad del isomorfismo inverso, $|\hom(a, b)| = |\hom(b, a)|$ para cualesquiera a y b, y esto prueba la equivalencia con (3). Ahora bien, si a es cero y x es otro objeto, el único morfismo $a \to x$ es un isomorfismo, por lo que x es cero y queda probado ($3 \iff 4$). La última observación es por definición. □

6. Monomorfismos y epimorfismos

En muchas ramas del álgebra llamamos monomorfismos a los morfismos inyectivos y epimorfismos a los morfismos suprayectivos. Esta definición depende de que los morfismos sean funciones, por lo que para categorías generales debemos buscar una definición alternativa.

Para los monomorfismos podemos basarnos en que, en **Set**, los elementos de un conjunto S se identifican con los morfismos $\{*\} \to S$, con lo que la propiedad de que $f(x) = f(y) \implies x = y$ se puede traducir como sigue.

Definición 1.20. Un morfismo $f : a \to b$ es un *monomorfismo* (escrito $f : a \rightarrowtail b$) si es cancelable por la izquierda, es decir, si para cualesquiera $h, k : c \to a$ con $f \circ h = f \circ k$ se tiene $h = k$.

Ejemplo 1.21 (Monomorfismos en constructos).

1. En **Set**, por construcción, todo monomorfismo es inyectivo sin más que tomar $c = \{*\}$ en la definición anterior. Lo mismo ocurre en **Top**, **Met**, **Ord** y otros constructos en que los elementos de un objeto S se identifiquen con los morfismos $\{*\} \to S$.
2. En todos los constructos, los morfismos inyectivos son monomorfismos, pues si $f : X \to Y$ es un morfismo inyectivo y $h, k : S \to X$ con $f \circ h = f \circ k$, entonces para $x \in S$, $f(h(x)) = f(k(x))$ y por tanto $h(x) = k(x)$, de modo que $h = k$.

3. Aunque en la mayoría de constructos relevantes los monomorfismos son precisamente los morfismos inyectivos, esto no es siempre así, como muestra el constructo cuyos objetos son $\{*\}$, $\{\$\}$ y $\{x, y\}$ y cuyos morfismos son las identidades, las funciones con codominio $\{\$\}$ y el morfismo $* \rightsquigarrow x$, en el cual el morfismo $\{x, y\} \to \{\$\}$ es un monomorfismo pero no es inyectivo.

En el ejemplo 1 de la lista anterior usamos que los elementos de un objeto se identifican con los morfismos desde un objeto unipuntual. En otras categorías como R-**Mod** nos gustaría hacer lo mismo, pero el objeto unipuntual no nos sirve.

Definición 1.22. En un constructo, un objeto D es *libre* sobre un conjunto $X \in \text{Ob}(\mathbf{Set})$ respecto a una función $u : X \to D$ si, para todo objeto A del constructo y función $f : X \to A$, existe un único morfismo $\hat{f} : D \to A$ en el constructo tal que $\hat{f} \circ u = f$.

Es fácil ver que el objeto libre sobre un cierto conjunto, si existe, es único salvo isomorfismo, y que un objeto libre sobre un conjunto X también es libre sobre cualquier otro conjunto del mismo tamaño que X. Además, en un constructo en que algún objeto tenga más de un elemento, la función u asociada a un objeto libre es inyectiva, por lo que en general supondremos que es una inclusión. Si X es unipuntual, identificamos los morfismos de D hacia un objeto A con los elementos de A.

Así, por ejemplo, el R-módulo libre sobre X es $R^{(X)}$, la suma directa externa de $|X|$ copias de R, y el conjunto libre sobre X es el propio X.

Proposición 1.23. *Dados un conjunto graduado $\Omega = \{s_1, \dots, s_k\}$ donde cada s_i tiene aridad n_i y un conjunto finito E de identidades en Ω, el objeto libre de (Ω, E)-**Alg** sobre el conjunto X se construye tomando el conjunto de los árboles finitos con raíz y ordenados cuyos nodos se etiquetan con un s_i y tienen n_i hijos, o con un $x \in X$ y tienen 0 hijos, haciendo el conjunto cociente por reescritura según las identidades en E, y definiendo las operaciones por construcción de árboles.*

Ejemplo 1.24 (Objetos libres en variedades algebraicas comunes).

1. El monoide libre sobre X está formado por las cadenas de elementos de X junto a la concatenación, mientras que el semigrupo libre es similar pero excluyendo la cadena vacía.
2. El grupo libre sobre X está formado por cadenas de símbolos de la forma x o $\overline{x}$ para $x \in X$ en las que no aparecen subsecuencias $x\overline{x}$ u $\overline{x}x$, y la operación es la concatenación eliminando sucesivamente las subcadenas de esta forma.
3. El grupo abeliano libre sobre X es $\mathbb{Z}^X$.
4. El anillo conmutativo libre sobre $\{x_1, \dots, x_n\}$ es el anillo de polinomios $\mathbb{Z}[x_1, \dots, x_n]$.

Proposición 1.25. *En los constructos con un objeto libre sobre el conjunto unipuntual, los monomorfismos son precisamente los morfismos inyectivos.*

Demostración. Sean $f : A \to B$ un monomorfismo en un tal constructo $\mathcal{C}$ y $x, y \in A$ con $f(x) = f(y)$. Si F es el objeto libre sobre $\{*\}$, h es el único morfismo $F \to A$ con $h(*) = x$ y $k : F \to A$ es el único con $h(*) = y$, para $c \in F$, $f(h(c))$ y $f(k(c))$ vienen dados por $f(h(*))$ y $f(k(*))$ siguiendo la estructura de c, por lo que $f \circ h = f \circ k$, $h = k$ y, finalmente, $x = y$. $\square$

Proposición 1.26.

1. *La composición de monomorfismos es un monomorfismo.*

   Demostración. Sean $f : A \to B$ y $g : B \to C$ monomorfismos y $h, k : D \to A$ con $g \circ f \circ h = g \circ f \circ k$, entonces $f \circ h = f \circ k$ y por tanto $h = k$. $\square$

2. *Si $g \circ f$ es un monomorfismo, f también lo es.*

   Demostración. Si $g \circ f$ es un monomorfismo y $f \circ h = f \circ k$, entonces $g \circ f \circ h = g \circ f \circ k$ y por tanto $h = k$. $\square$

Respecto a los epimorfismos, en **Set**, si $f : A \to B$ no es suprayectiva, existe $y \in B \backslash \operatorname{Im} f$ y podemos crear dos funciones $g, h : B \to \{0, 1\}$ con $g \circ f = h \circ f$ pero que difieran en y, mientras que si f es suprayectiva, $g \circ f = h \circ f$ implica claramente $g = h$.

DEFINICIÓN 1.27. Un morfismo $f : a \to b$ es un *epimorfismo*, $f : a \twoheadrightarrow b$, si es cancelable por la derecha, es decir, si para $g, h : b \to c$ con $g \circ f = h \circ f$ se tiene $g = h$.

La siguiente proposición se demuestra de forma similar que la correspondiente para epimorfismos.

PROPOSICIÓN 1.28.

1. *La composición de epimorfismos es un epimorfismo.*
2. *Si $g \circ f$ es un epimorfismo, g también lo es.*

EJEMPLO 1.29.

1. En los constructos, las funciones suprayectivas son epimorfismos, por el mismo argumento que en **Set**.
2. En **Set** los epimorfismos son precisamente las funciones suprayectivas, pues para $f : A \to B$ no suprayectiva es fácil encontrar $g, h : B \to \{0, 1\}$ distintas con $g \circ f = h \circ f$. Lo mismo ocurre en **Top** dotando a $\{0, 1\}$ de la topología indiscreta y en **Grph** usando el grafo completo de dos elementos.[2]
3. En R-**Mod**, si $f : M \to N$ es un epimorfismo, sean $p, z : N \to \frac{N}{\operatorname{Im} f}$ la proyección canónica y la función constante en 0, respectivamente, entonces $p \circ f = z \circ f$, luego $p = z$, $\frac{N}{\operatorname{Im} f} \cong 0$ y así $\operatorname{Im} f = N$, con lo que los epimorfismos son suprayectivos.
4. No en todas las variedades algebraicas los epimorfismos son suprayectivos. Por ejemplo, en **Rng**, la inclusión $u : \mathbb{Z} \to \mathbb{Q}$ es un monomorfismo, pues si $f, g : \mathbb{Q} \to R$ cumplen que $f \circ u = g \circ u : \mathbb{Z} \to R$, para $x, y \in \mathbb{Z}$, $f(\frac{x}{y}) = \frac{f(x)}{f(y)} = \frac{(f \circ u)(x)}{(f \circ u)(y)}$, y lo mismo ocurre con g, luego $f = g$.
5. En **Met** y $\mathbf{Met_c}$, los epimorfismos son precisamente las funciones con imagen densa.

   DEMOSTRACIÓN. Si $f : X \to Y$ es un morfismo con imagen densa y $g, h : Y \to Z$ son morfismos con $g \circ f = h \circ f$, para $y \in Y$ existe una sucesión $\{x_n\}_n \subseteq X$ con $y = \lim_n f(x_n)$, y por unicidad del límite y continuidad, $g(y) = \lim_n g(f(x_n)) = \lim_n h(f(x_n)) = h(y)$. Recíprocamente, si $f : X \to Y$ no tiene imagen densa, existen $y_0 \in Y$ y $r > 0$ con $B(y_0, r) \cap \operatorname{Im} f = \emptyset$, por lo que $g, h : Y \rightsquigarrow \mathbb{R}$ dadas por $g(y) := r$ y $h(y) := \min\{d(y_0, y), r\}$ son retracciones continuas distintas con $g \circ f = h \circ f$. $\square$

Los dos últimos ejemplos muestran que, al contrario que en muchos campos del álgebra, no siempre los morfismos que son a la vez monomorfismos y epimorfismos son isomorfismos.

DEFINICIÓN 1.30. Un *bimorfismo* es un morfismo que es a la vez un monomorfismo y un epimorfismo. Una categoría es *equilibrada* si todo bimorfismo es un isomorfismo.

EJEMPLO 1.31.

1. En una categoría fina, todos los morfismos son bimorfismos, pero en general no son isomorfismos.
2. En un monoide visto como categoría, los bimorfismos son los elementos cancelables por ambos lados.
3. Las identidades son bimorfismos, y en particular los isomorfismos son bimorfismos sin más que aplicar las proposiciones 1.26 y 1.28 a la composición de un isomorfismo con su inverso por ambos lados.

[2]En este trabajo, por comodidad, consideramos que el grafo completo tiene también las aristas reflexivas.

7. Secciones y retracciones

Hemos estudiado los morfismos cancelables por uno de los lados, por lo que cabe preguntarse qué ocurre con los morfismos invertibles por uno de los lados. Claramente este es un concepto más fuerte, y de hecho lo es estrictamente.

Definición 1.32. Una *sección* es un morfismo con inverso por la izquierda, y una *retracción* es un morfismo con inverso por la derecha. Dicho de otro modo, si $f : a \to b$ y $g : b \to a$ son tales que $g \circ f = 1_a$, decimos que f es una sección y g es una retracción.

Claramente las secciones son monomorfismos y las retracciones son epimorfismos.

Ejemplo 1.33.

1. En **Set**, las secciones son las funciones inyectivas salvo las que van de un conjunto vacío a uno no vacío, y las retracciones son las funciones suprayectivas. Es fácil ver que esto último es equivalente al axioma de elección.
2. En R-**Mod**, las secciones son los monomorfismos cuya imagen es un sumando directo del codominio, y las retracciones son los epimorfismos cuyo núcleo es un sumando directo del dominio.

   Demostración. Si $f : M \to N$ es un monomorfismo cuya imagen es un sumando directo de N, por ejemplo $N = \operatorname{Im} f \oplus P$, un $g : N \twoheadrightarrow M$ que lleva los elementos de $\operatorname{Im} f$ a su única preimagen por f y los $p \in P$ a 0 es su inverso por la derecha. Si es un epimorfismo cuyo núcleo es un sumando directo de M, por ejemplo $M = \ker f \oplus L$, sean $p : M \to \frac{M}{\ker f}$ la proyección canónica, $h : \frac{M}{\ker f} \cong L$ el isomorfismo «obvio» y $u : L \hookrightarrow M$ la inclusión, entonces $p \circ u \circ h = 1$, pero el teorema del factor nos da un único isomorfismo $\overline{f} : \frac{M}{\ker f} \cong N$ con $\overline{f} \circ p = f$, por lo que $\overline{f} = f \circ u \circ h$ y $f \circ (u \circ h \circ \overline{f}^{-1}) = 1$.

   Para el recíproco, sean $f : M \rightarrowtail N$ y $g : N \twoheadrightarrow M$ con $g \circ f = 1_M$, basta ver que $N = \operatorname{Im} f \oplus \ker g$. En efecto, todo $n \in N$ se descompone como suma de $n_1 := f(g(n)) \in \operatorname{Im} f$ y $n_2 := n - f(g(n)) \in \ker g$, y si $p_1 \in \operatorname{Im} f$ y $p_2 \in \ker g$ cumplen $n = p_1 + p_2$ y $m_1 \in M$ es la preimagen de p_1 por f, entonces $g(n) = g(p_1 + p_2) = g(f(m_1)) + g(p_2) = m_1 + 0$ y $n_1 = f(g(n)) = g(m_1) = p_1$, con lo que $n_2 = p_2$. □

3. En K-**Vec**, como caso especial del apartando anterior, todos los monomorfismos son secciones y todos los epimorfismos son retracciones.

Las secciones y retracciones se conservan por composición del mismo modo que lo hacen los monomorfismos y epimorfismos:

Proposición 1.34.

1. *La composición de secciones es una sección.*

   Demostración. Si f y g son secciones con inversas por la izquierda respectivas $\overline{f}$ y $\overline{g}$, entonces $(\overline{f} \circ \overline{g}) \circ g \circ f = \overline{f} \circ f = 1$. □

2. *Si $g \circ f$ es una sección, f es una sección.*

   Demostración. Si $g \circ f$ es una sección con inversa por la izquierda h, entonces $h \circ g \circ f = 1$, y $h \circ g$ es inversa por la izquierda de f. □

De forma análoga se prueba la siguiente proposición.

Proposición 1.35.

1. *La composición de retracciones es una retracción.*
2. *Si $g \circ f$ es una retracción, g es una retracción.*

Cabe preguntarse si un morfismo que es a la vez sección y retracción es invertible. La respuesta es que sí, y además esta condición se puede relajar.

PROPOSICIÓN 1.36. *Para un morfismo f, las siguientes afirmaciones son equivalentes:*

1. *f es un isomorfismo.*
2. *f es una sección y una retracción.*
3. *f es un monomorfismo y una retracción.*
4. *f es una sección y un epimorfismo.*

DEMOSTRACIÓN. Las implicaciones (1) $\implies$ (2) $\implies$ (3), (4) son obvias.

Para (2) $\implies$ (1), si $f : a \to b$ y $g, h : b \to a$ son tales que $g \circ f = 1_a$ y $f \circ h = 1_b$, entonces $g = g \circ 1_b = g \circ f \circ h = 1_a \circ h = h$, y $g = h$ es la inversa de f.

Para (3) $\implies$ (1), sea g un morfismo con $f \circ g = 1$, entonces $f \circ g \circ f = 1 \circ f = f \circ 1$, y como f es un monomorfismo, cancelando, $g \circ f = 1$ y f es un isomorfismo.

De forma análoga (4) $\implies$ (1). □

8. Dualidad

La mayoría de las propiedades que hemos visto hasta ahora vienen en pares con definiciones muy parecidas, y de hecho, cada vez que demostrábamos algo sobre una de ellas, la misma idea permitía demostrar algo similar sobre la otra. Esto ocurre mucho en teoría de categorías, y se conoce como dualidad.

DEFINICIÓN 1.37. Dada una categoría $\mathcal{C}$, su *categoría dual* es una categoría $\mathcal{C}^{\text{op}}$ con los mismos objetos y morfismos que $\mathcal{C}$ y la misma función identidad pero tal que, para todo morfismo f, $\text{dom}_{\mathcal{C}^{\text{op}}} f = \text{cod}_{\mathcal{C}} f$ y $\text{cod}_{\mathcal{C}^{\text{op}}} f = \text{dom}_{\mathcal{C}} f$, y la composición se define como $f \circ_{\mathcal{C}^{\text{op}}} g := g \circ_{\mathcal{C}} f$.

EJEMPLO 1.38.

1. Si $(X, \preceq)$ es un conjunto preordenado, su dual visto como categoría es $(X, \succeq)$.
2. El dual de una categoría discreta es ella misma.

DEFINICIÓN 1.39. Si P es un predicado aplicable a una o más categorías $\mathcal{C}_1, \dots, \mathcal{C}_n$, su *dual* es $P^{\text{op}}(\mathcal{C}_1, \dots, \mathcal{C}_n) \equiv P(\mathcal{C}_1^{\text{op}}, \dots, \mathcal{C}_n^{\text{op}})$, y decimos que P es *auto-dual* si $P \equiv P^{\text{op}}$.

En esencia, tomar la categoría dual o el predicado dual consiste en invertir el sentido de los morfismos. En general al tratar un concepto categórico es conveniente fijarse en el dual del concepto, ya que suele ser un concepto relevante, y además las propiedades de dicho concepto se derivan directamente de las del concepto original, sin demostración aparte.

9. Producto y coproducto

En muchas categorías es posible tomar el producto de dos o más objetos, como el de dos conjuntos, espacios topológicos, espacios vectoriales, etc. Para definir productos en una categoría arbitraria tenemos que encontrar una propiedad universal del producto, de modo que este quede definido de forma única salvo homomorfismo.

Para ello, si, por ejemplo, A y B son conjuntos, el producto $A \times B$ cumple que, dadas dos funciones $f : X \to A$ y $g : X \to B$, existe una única función $h : X \to (A \times B)$ tal que para todo $x \in X$, $h(x) = (f(x), g(x))$. Dicho de otro modo, si $p : A \times B \to A$ y $q : A \times B \to B$ son las proyecciones sobre las componentes, h es la única función con $f = p \circ h$ y $g = q \circ h$. Esta definición se puede generalizar a categorías arbitrarias y a una cantidad arbitraria de factores. La figura 2 muestra el caso para dos factores.

DEFINICIÓN 1.40. Un objeto b es un *producto* de una familia de objetos $(a_i)_{i\in I}$ si existe una familia de morfismos $(p_i : b \to a_i)_{i\in I}$, llamados *proyecciones*, tales que para cualquier familia de morfismos $(f_i : x \to a_i)_{i\in I}$ existe un único $f : x \to b$ tal que $f_i = p_i \circ f$ para cada i. Llamamos producto a b y a la familia de proyecciones indistintamente.

EJEMPLO 1.41.

1. En **Set**, toda familia pequeña de conjuntos tiene un producto, el producto directo habitual. Lo mismo ocurre en R-**Mod** y en particular en **Vec** y en **Ab**, así como en **Grp** con el producto de grupos y en **Top** con el producto topológico.

$$\begin{tikzcd}
 & x \arrow[dl, "f"'] \arrow[d, dotted, "h"] \arrow[dr, "g"] & \\
a & a \times b \arrow[l, "p"] \arrow[r, "q"'] & b
\end{tikzcd}$$

FIGURA 2. Objeto producto de a y b. Para cualesquiera x, f y g, existe un único h de modo que el diagrama conmuta.

$$\begin{tikzcd}
a \arrow[r, "u"] \arrow[dr, "f"'] & a \oplus b \arrow[d, dotted, "h"] & b \arrow[l, "v"'] \arrow[dl, "g"] \\
 & x &
\end{tikzcd}$$

FIGURA 3. Objeto coproducto de a y b. Para cualesquiera X, f y g, existe un único $f \oplus g$ de modo que el diagrama conmuta.

2. Todo objeto es el producto de sí mismo tomando como proyección el morfismo identidad. De hecho, un objeto a es el producto de la familia unipuntual (b) si y sólo si $a \cong b$.
3. Un objeto es producto de la familia vacía si y sólo si es final.
4. En un conjunto preordenado visto como categoría, el producto de una familia de elementos es su supremo, si existe.

PROPOSICIÓN 1.42. *El producto de una familia de objetos es único salvo isomorfismo.*

DEMOSTRACIÓN. Sean b y c productos de $(a_i)_{i\in I}$ con proyecciones $(f_i : b \to a_i)_{i\in I}$ y $(g_i : c \to a_i)_{i\in I}$, existe $h : b \to c$ tal que cada $f_i = g_i \circ h$ y $k : c \to b$ tal que cada $g_i = f_i \circ k$, pero entonces $g_i = g_i \circ h \circ k$ para cada i, y como para la familia $(g_i)_i$ debe haber un único $f : c \to c$ con cada $g_i = g_i \circ f$, debe ser $h \circ k = f = 1_c$, y análogamente $k \circ h = 1_b$. Es fácil ver que los isomorfismos conservan productos. □

En vista de esto, llamamos $\prod_{i\in I} a_i$ al objeto producto de $(a_i)_{i\in I}$, y $a \times b$ al objeto producto de dos objetos a y b. Esta última notación se puede extender a un número finito de factores $(a_1 \times \cdots \times a_n)$, en vista de la siguiente proposición.

PROPOSICIÓN 1.43. *Si $(p_i : c \to b_i)_{i\in I}$ es un producto y, para cada i, $(q_{ij} : b_i \to a_{ij})_{j\in J_i}$ es un producto, entonces $(q_{ij} \circ p_i : c \to a_{ij})_{i\in I}^{j\in J_i}$ es un producto.*

DEMOSTRACIÓN. Sea $(f_{ij} : x \to a_{ij})_{ij}$ una familia de morfismos, para cada $i \in I$ existe $g_i : x \to b_i$ tal que $f_{ij} = q_{ij} \circ g_i$ para todo j, por lo que existe $h : x \to c$ tal que $g_i = p_i \circ h$ para todo i y este cumple $f_{ij} = q_{ij} \circ p_i \circ h$ para todo i, j. Además, los g_i y h son únicos, por lo que si hubiera otro $k : x \to c$ con $f_{ij} = q_{ij} \circ p_i \circ k$ para todo i, j, necesariamente cada $p_i \circ k = g_i$ y por tanto $k = h$. □

En particular $a \times b \times c = (a \times b) \times c = a \times (b \times c)$, y una categoría tiene todos los productos finitos si y sólo si tiene objeto final y productos de todos los pares de objetos.

DEFINICIÓN 1.44. Dado un objeto a y un conjunto I, llamamos *I-ésima potencia de a* a $a^I := \prod_{i\in I} a$.

El concepto dual del producto es el coproducto, que se muestra gráficamente en la figura 3.

DEFINICIÓN 1.45. Un objeto c es un *coproducto* de una familia de objetos $(a_i)_{i\in I}$ si existe una familia de morfismos $(u_i : a_i \to c)_{i\in I}$, llamados *inyecciones*, tales que para cualquier familia de morfismos $(f_i : a_i \to x)_{i\in I}$ existe un único $f : c \to x$ tal que $f_i = f \circ u_i$ para cada i. Llamamos coproducto tanto a c como a la familia de inyecciones.

Ejemplo 1.46.

1. En **Set**, el coproducto de una familia de conjuntos $(a_i)_{i\in I}$ es la *unión disjunta*, $\biguplus_{i\in I} a_i := \bigcup_{i\in I}(\{i\} \times a_i)$, aunque si los a_i son disjuntos dos a dos se puede tomar como coproducto la unión convencional.
2. En **Top**, el coproducto de una familia de espacios topológicos es la unión disjunta con la topología formada por las uniones de abiertos de cada espacio.
3. En R-**Mod**, el coproducto de una familia pequeña de módulos $\{a_i\}_{i\in I}$ es la suma directa $\bigoplus_{i\in I} a_i$, el subespacio del producto directo $\prod_{i\in I} a_i$ formado por los elementos con casi todas las entradas nulas. En particular, si I es finito el producto coincide con el coproducto.
4. Todo objeto es coproducto de sí mismo.
5. Un objeto es coproducto de la familia vacía si y sólo si es inicial.
6. En un conjunto preordenado visto como categoría, el coproducto de una familia de elementos es su ínfimo, si existe.

Las siguientes propiedades son duales de las correspondientes del producto.

Proposición 1.47. *El coproducto de una familia de objetos es único salvo isomorfismo.*

Esto nos permite llamar $\coprod_{i\in I} a_i$ al objeto coproducto de $(a_i)_{i\in I}$, y $a \oplus b$ al objeto coproducto de dos objetos a y b.

Proposición 1.48. *Si $(u_i : b_i \to c)_{i\in I}$ es un coproducto y, para cada i, $(v_{ij} : a_{ij} \to b_i)_{j\in J_i}$ es un coproducto, entonces $(u_i \circ v_{ij} : a_{ij} \to c)_{i\in I}^{j\in J_i}$ es un coproducto.*

Definición 1.49. Dado un objeto a y un conjunto I, llamamos *I-ésima copotencia de a* a ${}^I a := \coprod_{i\in I} a$.

10. Núcleo y conúcleo

En álgebra es común hablar del núcleo de un morfismo $f : a \to b$ como el conjunto de puntos $x \in a$ con $f(x) = 0$, que es un subobjeto de a. Entonces, si $k \subseteq a$ es el núcleo de f y $u : k \hookrightarrow a$ es la inclusión, $f \circ u = 0$, y si $u' : k' \to a$ es otro morfismo con $f \circ u' = 0$, existe $\tilde{u} : k' \to k$ tal que $u' = u \circ \tilde{u}$. Esta definición nos sirve para teoría de categorías salvo por la presencia del 0.

Si, para cada objeto a, llamamos $i_a : 0 \to a$ a la única flecha desde el objeto cero hasta a y $t_a : a \to 0$ a la única flecha desde a hasta el objeto 0, entonces podríamos escribir «$f \circ u = 0$» como «$f \circ u = i_b \circ t_k$», pero esta caracterización sólo nos sirve para el caso de categorías con cero. En general, podemos notar que $t_k = t_a \circ u$, de modo que u «iguala» f con $i_b \circ t_a$, y aunque no podemos hablar del núcleo de un morfismo, podemos hablar del núcleo de un par de morfismos.

Definición 1.50. Dados dos morfismos $f, g : a \to b$, un morfismo $e : k \to a$ es un *núcleo* de f y g si y sólo si $f \circ e = g \circ e$ y, para todo $e' : k' \to a$ con $f \circ e' = g \circ e'$, existe un único $\tilde{e} : k' \to k$ con $e' = e \circ \tilde{e}$, es decir, tal que la figura 4 conmuta. Si existe un cero 0, un *núcleo* de $f : a \to b$ es un núcleo de f y la composición $a \to 0 \to b$.

Ejemplo 1.51.

1. En **Set**, para $f, g : a \to b$, si $k := \{x \in a \mid f(x) = g(x)\}$, la inclusión $u : k \hookrightarrow a$ es un núcleo de f y g. Lo mismo ocurre en R-**Mod**, **Top**, **Grp**, **Grph** y (Ω, E)-**Alg**, dotando a k de su estructura como submódulo, subespacio topológico, subgrupo, subgrafo o subálgebra.
2. En **CRng**, al contrario que en muchas estructuras algebraicas, el concepto de núcleo de un morfismo en teoría de categorías no coincide con el convencional, pues convencionalmente el núcleo de un anillo no es un subanillo sino un ideal.

$$\begin{tikzcd}
k' \arrow[d, dotted, "\tilde{e}"'] \arrow[dr, "e'"] & & \\
k \arrow[r, "e"'] & a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b
\end{tikzcd}$$

FIGURA 4. Núcleo e de un par de morfismos f y g.

3. En R-**Mat**, dadas dos matrices $A_{n\times m}, B_{n\times m}$, un núcleo de A y B es precisamente una matriz $X_{m\times p}$ cuyas columnas forman una base de soluciones de la ecuación $Ax = Bx$. En efecto, $AX = BX$ y, si $Y_{m\times q}$ cumple que $AY = BY$, cada columna de Y es combinación lineal única de las columnas de X y hay por tanto una única Z con $Y = XZ$. La existencia de Z implica que las columnas de X deben generar el espacio de soluciones, y la unicidad implica que estan deben ser linealmente independientes.

En el último ejemplo hemos tenido que demostrar no sólo que una cierta matriz es el núcleo sino que todos los demás núcleos también tienen la misma forma. A continuación vemos que esto último no es necesario, y basta con encontrar un núcleo.

PROPOSICIÓN 1.52. *El núcleo es esencialmente único, es decir, si $e : k \to a$ es un núcleo de $f, g : a \to b$, el resto de núcleos de f y g son precisamente los morfismos de la forma $e \circ h : k' \to a$ donde $h : k' \to k$ es un isomorfismo.*

DEMOSTRACIÓN. Claramente $e \circ h$ es un núcleo, pues $f \circ e \circ h = g \circ e \circ h$ y, si e' cumple $f \circ e' = g \circ e'$ y $\tilde{e}$ es el único morfismo con $e' = e \circ \tilde{e}$, entonces $h^{-1} \circ \tilde{e}$ es el único con $e' = (e \circ h) \circ (h^{-1} \circ \tilde{e})$. Para el recíproco, sean $e : k \to a$ y $e' : k' \to a$ núcleos de f y g, existen un único $h : k' \to k$ con $e' = e \circ h$ y un único $h' : k \to k'$ con $e = e' \circ h'$, de modo que $e \circ 1_k = e \circ h \circ h'$ y, por unicidad, $1_k = h \circ h'$, pero análogamente $1_{k'} = h' \circ h$, luego h es un isomorfismo. □

Hasta ahora todos los núcleos que hemos visto son monomorfismos, por lo que parece interesante comparar el concepto de núcleo con el de monomorfismo y con el de sección.

PROPOSICIÓN 1.53.

1. *Todo núcleo es un monomorfismo.*

   DEMOSTRACIÓN. Si $e : k \to a$ es el núcleo de $f, g : a \to b$ y $h, h' : c \to k$ cumplen $e \circ h = e \circ h' =: e'$, entonces $f \circ e' = g \circ e'$, luego el h con $e' = e \circ h$ es único y por tanto $h = h'$. □

2. *Toda sección es un núcleo. En concreto, si $f : a \to b$ es una sección y $g : b \to a$ es la correspondiente retracción, entonces f es núcleo de $f \circ g$ y 1_b.*

   DEMOSTRACIÓN. $(f \circ g) \circ f = f \circ (g \circ f) = f = 1_b \circ f$, pero si $e : k \to b$ es tal que $(f \circ g) \circ e = 1_b \circ e$, entonces $g \circ e$ es el morfismo con $f \circ (g \circ e) = e$, y es único porque f es un monomorfismo. □

3. *Los recíprocos no se cumplen.*

   DEMOSTRACIÓN. En **Ab**, el único homomorfismo $\mathbb{Z} \to \mathbb{Z}_2$ tiene como núcleo la inclusión $2\mathbb{Z} \hookrightarrow \mathbb{Z}$, que claramente no es una sección. Y en **CRng**, la inclusión $\mathbb{Z} \hookrightarrow \mathbb{Q}$ es un monomorfismo pero no es un núcleo, ya que de serlo, como también es un epimorfismo, sería un isomorfismo por la siguiente proposición 1.54. □

PROPOSICIÓN 1.54. *Si $e : k \to a$ es núcleo de $f, g : a \to b$, son equivalentes:*

1. $f = g$.
2. 1_a *es núcleo de f y g.*

$$\begin{tikzcd} a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \arrow[r, "c"] \arrow[dr, "c'"'] & q \arrow[d, dotted, "\overline{c}"] \\ & & q' \end{tikzcd}$$

FIGURA 5. Conúcleo q de un par de morfismos f y g.

3. *e es un isomorfismo.*
4. *e es un epimorfismo.*

DEMOSTRACIÓN. Claramente (1) $\Longrightarrow$ (2); (2) $\Longrightarrow$ (3) por (1.52); obviamente (3) $\Longrightarrow$ (4), y (4) $\Longrightarrow$ (1) es por definición de núcleo y de epimorfismo. □

El concepto dual de núcleo es el de conúcleo.

DEFINICIÓN 1.55. Dados dos morfismos $f, g : a \to b$, un morfismo $c : b \to q$ es un *conúcleo* de f y g si y sólo si $c \circ f = c \circ g$ y, para todo $c' : b \to q'$ con $c' \circ f = c' \circ g$, existe un único $\overline{c} : q \to q'$ con $c' = \overline{c} \circ c$, es decir, tal que la figura 5 conmuta. Si existe un cero 0, un *conúcleo* de $f : a \to b$ es un conúcleo de f y la composición $a \to 0 \to b$.

EJEMPLO 1.56.

1. Sean $f, g : a \to b$ en **Set** y sea $\sim$ la menor relación de equivalencia con $f(x) \sim g(x)$ para todo $x \in a$. Entonces la proyección al conjunto cociente $c : b \to \frac{b}{\sim}$ es un conúcleo de f y g.

   DEMOSTRACIÓN. Claramente $c \circ f = c \circ g$ y, si $c' : b \to r$ es tal que $c' \circ f = c' \circ g$, entonces $\overline{c} : \frac{b}{\sim} \to r$ dada por $\overline{c}(\overline{x}) = c'(x)$ es la única función con $\overline{c} \circ c = c'$, y queda ver que está bien definida. Pero si $x \sim y$, bien $x = y$, bien existen una cadena $x = t_0, \dots, t_k = y \in b$ y objetos $s_1, \dots, s_k \in a$ con (t_{i-1}, t_i) igual a $(f(s_i), g(s_i))$ o $(g(s_i), f(s_i))$ para cada i, pero entonces por hipótesis $c'(t_{i-1}) = c'(t_i)$ para cada i y por tanto $c'(x) = c'(y)$. □

2. En **Top**, los conúcleos se calculan como en **Set** asignando a $\frac{b}{\sim}$ la topología cociente, cuyos abiertos son los subconjuntos $S \subseteq \frac{b}{\sim}$ con $c^{-1}(S) \subseteq b$ abierto, pues esta topología hace a c y $\overline{c}$ del apartado anterior continuas (supuesto que c' sea continua). Esto significa que superficies como el toro, la cinta de Möbius o la botella de Klein se definen naturalmente como conúcleos.
3. En **Grph** ocurre lo mismo, estableciendo como ejes en $\frac{b}{\sim}$ las imágenes de ejes en b. En **Prord** los conúcleos se construyen de igual forma pero tomando la clausura transitiva de la relación resultante en $\frac{b}{\sim}$.
4. En (Ω, E)-**Alg** el conúcleo es similar pero tomando como $\sim$ la menor relación de congruencia en b con $f(x) \sim g(x)$ para todo $x \in a$. Una *relación de congruencia* es una relación de equivalencia en que, para cada operación $f : b^n \to b$ en la acción de Ω asociada a b y cada $x_1, \dots, x_n, y_1, \dots, y_n \in b$ se tiene que, si cada $x_i \sim y_i$, entonces $f(x_1, \dots, x_n) \sim f(y_1, \dots, y_n)$. Esta propiedad permite dotar a $\frac{b}{\sim}$ de una estructura algebraica cuyas operaciones se derivan de las de b de la forma evidente.
5. En **CRng**, el conúcleo de $f, g : A \to B$ es la proyección canónica $p : B \to \frac{B}{I}$, donde $I := (\operatorname{Im} f - g)$ es el ideal generado por la imagen de $f - g$.

   DEMOSTRACIÓN. Para $a \in A$, $p(f(a)) - p(g(a)) = p((f - g)(a)) = 0$, pues $(f - g)(a) \in \ker p$, y si $c : B \to C$ es un morfismo que cumple $c \circ f = c \circ g$ y por tanto $c \circ (f - g) = 0$, y los elementos de I son combinaciones lineales de imágenes de $f - g$, se tiene $I \subseteq \ker c$ y, por el teorema del factor, existe un único $\overline{c} : \frac{B}{I} \to C$ con $c = \overline{c} \circ p$. □

Las siguientes proposiciones son las duales de las vistas para el núcleo.

Proposición 1.57. *El conúcleo es esencialmente único, es decir, si* $c : b \to q$ *es un conúcleo de* $f, g : a \to b$*, el resto de conúcleos de* f *y* g *son precisamente los morfismos de la forma* $h \circ c : b \to q'$ *donde* $h : q \to q'$ *es un isomorfismo.*

Proposición 1.58.

1. *Todo conúcleo es un epimorfismo.*
2. *Toda retracción es un conúcleo. En concreto, si* $g : a \to b$ *es una retracción y* $f : b \to a$ *es la correspondiente sección, entonces* g *es conúcleo de* $f \circ g$ *y* 1_a*.*
3. *Los recíprocos no se cumplen.*

Proposición 1.59. *Si* $c : b \to q$ *es conúcleo de* $f, g : a \to b$*, son equivalentes:*

1. $f = g$*.*
2. 1_b *es conúcleo de* f *y* g*.*
3. c *es un isomorfismo.*
4. c *es un monomorfismo.*

11. Subobjetos y objetos cociente

En los ejemplos de núcleos y conúcleos que hemos visto, por lo general los núcleos son subobjetos del dominio de los morfismos involucrados, mientras que los conúcleos son objetos cociente. En teoría de categorías, como las categorías no tienen por qué ser conjuntos, resulta difícil definir estos conceptos, y de hecho no hay una sóla definición para todos los casos, aunque todas las definiciones se basan en una estructura común.

Definición 1.60. Sea M un conjunto de monomorfismos. Un *M-subobjeto* de un objeto a es un par (b, m) formado por un objeto b y un morfismo $m : b \to a$ en M.

Las distintas definiciones dependen, pues, de qué conjunto M usemos. Si estamos en un constructo tiene sentido considerar el conjunto de morfismos que son inclusiones. Para el caso general, sin embargo, existen varios conceptos usados en la práctica, siendo el más amplio el de monomorfismo y el más restringido el de sección. Para categorías algebraicas como R-**Mod**, es útil el concepto de un monomorfismo que es el núcleo de algún morfismo.

Definición 1.61. Un monomorfismo es *regular* si es el núcleo de algún par de morfismos.

Así, si M es el conjunto de los monomorfismos regulares, los M-subobjetos se llamarían *subobjetos regulares*, mientras que si M es el conjunto de todos los monomorfismos, los M-subobjetos son los *subobjetos* a secas. Para estos conceptos, el conjunto de subobjetos es cerrado para isomorfismos, en el sentido siguiente.

Definición 1.62. Dados dos M-subobjetos (a, m) y (b, n) de un objeto c, (a, m) es *más pequeño* que (b, n), $(a, m) \leq (b, n)$, si existe $f : a \to b$ con $m = n \circ f$, y (a, m) y (b, n) son *isomorfos*, $(a, m) \cong (b, n)$, si además f es un isomorfismo.

Cabe destacar la importancia del uso de morfismos para definir subobjetos, pues estos en general aportan mucha más información que los objetos. Por ejemplo, en **Ord**, $\{1, 3, 5\}$ y $\{2, 4, 6\}$ son subconjuntos ordenados de $\mathbb{N}$ y, vistos como objetos, son isomorfos, pero como subobjetos son distintos aun tras componer el monomorfismo inclusión con un isomorfismo a cada lado. En **Set**, aunque no son isomorfos como subobjetos, se pueden igualar componiendo con un isomorfismo por la izquierda, pues la única información que conservaríamos es que se trata de subconjuntos de tamaño 3 de un conjunto numerable. Esto refleja el hecho de que la teoría de categorías estudia la estructura de los objetos, viéndolos «desde fuera», y no estudia su contenido, que es materia de la teoría de conjuntos.

Ejemplo 1.63.

1. En **Set**, los monomorfismos regulares son las funciones inyectivas, por lo que todos los monomorfismos son regulares. En efecto, las funciones inyectivas son aquellas

isomorfas a una inclusión, y toda inclusión $u : E \hookrightarrow A$ es núcleo del par de funciones $A \to \{0, 1\}$ en el que una lleva todos los elementos a 0 y otra lleva a 0 los elementos de E y a 1 el resto. Por tanto los subobjetos regulares se identifican (salvo isomorfismo) con los subconjuntos, y un subobjeto es más pequeño que otro si y sólo si es un subconjunto del otro.

2. En **Top** los monomorfismos regulares son, salvo isomorfismo, las inclusiones de subespacios, usando la prueba del apartado anterior y dotando a $\{0, 1\}$ de la topología indiscreta, por lo que los subobjetos regulares son los subespacios topológicos.
3. En muchas categorías algebraicas como R-**Mod** o **Grp**, todos los monomorfismos son regulares, con lo que los subobjetos regulares son respectivamente los submódulos y los subgrupos. Esto es porque todo monomorfismo $e : K \rightarrowtail M$ es núcleo de la proyección canónica $M \rightsquigarrow \frac{M}{\operatorname{Im} e}$.
4. En R-**Mod**, todos los monomorfismos son regulares. En efecto, un monomorfismo $m : M \to N$ es núcleo de la proyección canónica $p : N \to \frac{N}{\operatorname{Im} m}$.
5. En **CRng** hemos visto (1.53, apartado 3) que no todos los monomorfismos son regulares. Sin embargo, los subobjetos (a secas) son precisamente los subanillos.
6. En una categoría fina, sólo las identidades son regulares.
7. En R-**Mat**, los núcleos son las matrices $m \times p$, $p \le m$, de rango máximo (1.51), con lo que los subobjetos regulares de un número m son los pares formados por un $p \le m$ y una matriz $m \times p$ de rango p.

El concepto dual de subobjeto es el de objeto cociente.

Definición 1.64. Sea E un conjunto de epimorfismos. Un *E-objeto cociente* de un objeto a es un par (b, e) formado por un objeto b y un morfismo $e : a \to b$ en E.

Definición 1.65. Un epimorfismo es *regular* si es el conúcleo de algún par de morfismos.

Si E es el conjunto de los epimorfismos regulares, hablamos de *objetos cociente regulares*, mientras que si es el conjunto de todos los epimorfismos hablamos de *objetos cociente*.

En constructos, tiene sentido considerar los epimorfismos que son proyecciones al conjunto cociente por alguna relación de equivalencia en el dominio, o a algún conjunto irredundante de representantes de dicha relación. En la mayoría de constructos relevantes, si para una cierta relación de equivalencia existe una de estas proyecciones (al cociente o algún conjunto irredundante de representantes, con la estructura apropiada), el resto también existe y son isomorfas en el sentido siguiente.

Definición 1.66. Dados dos E-objetos cociente (b, d) y (c, e) de un objeto a, (b, d) es *más grande* que (c, e), $(b, d) \ge (c, e)$, si existe $f : b \to c$ con $e = f \circ d$, y (b, d) y (c, e) son *isomorfos*, $(b, d) \cong (c, e)$, si además f es un isomorfismo.

Ejemplo 1.67.

1. En **Set**, todos los epimorfismos son regulares, pues si $e : A \twoheadrightarrow B$ es suprayectiva, es el conúcleo de las dos proyecciones $D_e \to A$ con
$$D_e := \{(a, a') \in A \times A \mid e(a) = e(a')\},$$
por un argumento similar al usado en el ejemplo 1.63, apartado 1. Además, claramente los objetos cociente de A son, salvo isomorfismo, los conjuntos cociente con sus proyecciones, y si (B, d) y (C, e) son objetos cociente (regulares) de A, entonces $(B, d) \ge (C, e)$ si y sólo si $D_d \subseteq D_e$.
2. De forma similar se ve que los objetos cociente regulares en **Top** son, salvo isomorfismo, los espacios topológicos cociente con las correspondientes proyecciones. Aquí, sin embargo, no todos los epimorfismos son regulares, pues para ello el codominio debe tener la topología final y no una más gruesa.
3. En R-**Mod**, todos los epimorfismos son regulares. En efecto, si $e : M \to N$ es un morfismo, D_e (apartado 1) es el núcleo del homomorfismo $(m, m') \rightsquigarrow e(m - m')$, por lo que es un R-módulo y e es el conúcleo de las dos proyecciones $D_e \to M$. Además, los objetos cociente son los módulos cociente.

Capítulo 2

Funtores

Buena parte del poder de la teoría de categorías se deriva de su reflexividad: las categorías son estructuras algebraicas y, como tales, es posible estudiarlas usando teoría de categorías. Para ello primero debemos ver cuáles son los morfismos entre categorías, los llamados *funtores*. Este capítulo se basa principalmente en [1, pp. 21–32 y cap. 7] y [4, pp. 13–15].

Definición 2.1. Un *funtor* entre dos categorías $\mathcal{C}$ y $\mathcal{D}$ es un par de funciones $T = (o : \mathrm{Ob}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{D}), m : \mathrm{Mor}(\mathcal{C}) \to \mathrm{Mor}(\mathcal{D}))$ que preserva el dominio, el codominio, las identidades y la composición, es decir, tal que:

1. Para cada morfismo $f : a \to b$ en $\mathcal{C}$, $mf : oa \to ob$ en $\mathcal{D}$.
2. Para $f : a \to b$ y $g : b \to c$ en $\mathcal{C}$, $m(g \circ f) = mg \circ mf$.
3. Para cada objeto c de $\mathcal{C}$, $m(1_a) = 1_{oa}$.

La última condición determina unívocamente la función sobre los objetos a partir de la función sobre los morfismos, por lo que la primera es redundante. Si $\mathcal{C}$ y $\mathcal{D}$ son categorías, escribimos $T : \mathcal{C} \to \mathcal{D}$ para decir que T es un funtor de $\mathcal{C}$ a $\mathcal{D}$, y usamos T para referirnos indistintamente al funtor, a la función sobre los objetos y a la función sobre los morfismos.

Ejemplo 2.2.

1. Toda categoría $\mathcal{C}$ admite un *funtor identidad* $1_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$ que asocia a cada objeto o morfismo el propio objeto o morfismo.
2. Dadas dos categorías $\mathcal{C}$ y $\mathcal{D}$ y $d \in \mathrm{Ob}(\mathcal{D})$, existe un *funtor constante* $C_d : \mathcal{C} \to \mathcal{D}$ que lleva todos los morfismos a 1_d.
3. Un funtor entre dos conjuntos, conjuntos preordenados o monoides vistos como categorías es precisamente un morfismo en **Set**, **Prord** o **Mon**, respectivamente.
4. Si $\mathcal{B}$ es una subcategoría de $\mathcal{C}$, existe un *funtor inclusión* $u : \mathcal{B} \to \mathcal{C}$ que envía cada objeto y morfismo de $\mathcal{B}$ a sí mismo en $\mathcal{C}$.
5. La operación *conjunto potencia* es un funtor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$, que lleva cada función $f : A \to B$ a la función $(\mathcal{P}f)(S) := f[S]$, que asocia a cada subconjunto de A su *imagen* por f.
6. De forma similar podemos definir el funtor *conjunto potencia contravariante*, $\mathcal{Q} : \mathbf{Set} \to \mathbf{Set}^{\mathrm{op}}$, que lleva un conjunto S a su potencia $\mathcal{P}S$ y una función $f : A \to B$ a la función $(\mathcal{Q}f)(T) := f^{-1}[T]$ que asocia a cada subconjunto de B su *preimagen* por f.
7. Para $n \in \mathbb{N}$, existe un funtor $\mathrm{GL}_n : \mathbf{CRng} \to \mathbf{Grp}$ que a cada anillo conmutativo C le asocia el grupo multiplicativo $\mathrm{GL}_n(C)$ de matrices regulares $n \times n$ con entradas en C. Los homomorfismos de anillos se transforman en homomorfismos de grupos que actúan componente a componente.
8. Sea $\mathbf{Top}_*$ el constructo cuyos objetos son pares (X, x) formados por un espacio topológico X y un punto destacado $x \in X$ y cuyos morfismos son funciones continuas que conservan el punto destacado. Entonces podemos definir un funtor grupo de homotopía $\pi : \mathbf{Top}_* \to \mathbf{Grp}$ que a cada espacio topológico X y cada punto $x \in X$ le asocia el grupo de homotopía y a cada morfismo en $\mathbf{Top}_*$ le asocia el correspondiente morfismo de grupos.[4, p. 13]

   Demostración. Primero vemos que la operación sobre morfismos está bien definida. Sea $f : (X, x) \to (Y, y)$ un morfismo en $\mathbf{Top}_*$, y sean $\gamma, \sigma : [0, 1] \to X$

representantes de un mismo elemento de $\pi(X, x)$, entonces existe una homotopía $s : [0,1] \times [0,1] \to X$ de γ a σ, con lo que $f \circ s : [0,1] \times [0,1] \to Y$ es una homotopía de $f(\sigma)$ a $f(\gamma)$ y $(\pi f)(\overline{\gamma}) = (\pi f)(\overline{\sigma})$. Además πf es un morfismo de grupos, pues la composición con f lleva la curva constante en x a la curva constante en y y respeta la concatenación de curvas. Finalmente, dados morfismos $f : (X, x) \to (Y, y)$ y $g : (Y, y) \to (Z, z)$ en $\mathbf{Top}_*$, es fácil ver que $\pi(g \circ f) = \pi g \circ \pi f$ y que $\pi(1_{(X,x)}) = 1_{\pi(X,x)}$. □

9. Los funtores se pueden componer. Dados dos funtores $S : \mathcal{B} \to \mathcal{C}$ y $T : \mathcal{C} \to \mathcal{D}$, el *funtor composición* $T \circ S : \mathcal{B} \to \mathcal{C}$ viene dado sobre los objetos como $a \rightsquigarrow S(Ta)$ y sobre los morfismos como $f \rightsquigarrow S(Tf)$.

1. Categorías de categorías

En vista de los ejemplos anteriores sería razonable considerar una «categoría de las categorías», pero esto plantea ciertos problemas. En primer lugar, esta categoría no se podría contener a sí misma por la paradoja de Russell. De hecho, sólo las categorías que son conjuntos pueden estar dentro de una categoría de categorías, pues las clases propias no pueden estar dentro de otras clases.

Definición 2.3. Una categoría es *pequeña* si es un conjunto, es decir, si tanto su clase de objetos como su clase de morfismos son conjuntos. Llamamos **Cat** a la categoría de las categorías pequeñas y los funtores entre ellas.

Esto no es del todo satisfactorio, pues la mayoría de las categorías no son pequeñas. Es por ello que en teoría de categorías es común usar extensiones de ZFC para lidiar con estos casos. Mac Lane[4, pp. 21–26] propone una extensión basada en universos de Grothendieck.

Definición 2.4. Un *universo* (*de Grothendieck*) es un conjunto $\mathcal{U}$ tal que:

1. Si $x \in u \in \mathcal{U}$ entonces $x \in \mathcal{U}$.
2. Si $u, v \in \mathcal{U}$ entonces $\{u, v\} \in \mathcal{U}$.
3. Si $x \in \mathcal{U}$ entonces $\mathcal{P}x, \bigcup x \in \mathcal{U}$.
4. Si $I \in \mathcal{U}$ y $f : I \to \mathcal{U}$ es una función, entonces $\operatorname{Im} f \in \mathcal{U}$.
5. $\mathbb{N} \in \mathcal{U}$.

La idea es que un universo contendría todos los conjuntos con los que uno trataría normalmente dentro de ZFC, como ilustran las siguientes propiedades fáciles de probar.

Proposición 2.5. *Sea $\mathcal{U}$ un universo:*

1. *Si $v \subseteq u \in \mathcal{U}$ entonces $v \in \mathcal{U}$.*
2. *Si $u, v \in \mathcal{U}$, entonces $(u, v), u \times v \in \mathcal{U}$.*
3. *Si $\{x_i\}_{i \in I}$ es una familia de elementos de $\mathcal{U}$ con $I \in \mathcal{U}$, entonces*
$$\prod_{i\in I} x_i, \bigcup_{i\in I} x_i, \bigcap_{i\in I} x_i \in \mathcal{U}.$$
4. *Si $a, b \in \mathcal{U}$, todas las funciones $f : a \to b$ cumplen $f \in \mathcal{U}$.*
5. *Si $a \in \mathcal{U}$ y $b \subseteq \mathcal{U}$ con $|a| = |b|$ entonces $b \in \mathcal{U}$.*

Grothendieck trabajaba con un axioma adicional que afirmaba que cada conjunto estaba contenido en un universo, desarrollando así una jerarquía de universos. Sin embargo, para nuestros propósitos nos sirve el siguiente axioma más sencillo.

Axioma 2.6 (Mac Lane, 1969[5]). Existe un universo de Grothendieck $\mathcal{U}$.

En adelante fijamos un universo $\mathcal{U}$ y notamos que, cuando hablábamos de conjuntos, ahora hablamos de elementos de $\mathcal{U}$, y cuando hablábamos de clases, ahora hablamos de subconjuntos de $\mathcal{U}$.

Definición 2.7. Un conjunto x es *pequeño* si $x \in \mathcal{U}$, es una *clase* si $x \subseteq \mathcal{U}$, y es *grande* o una *clase propia* si es una clase que no es pequeña.

La mayoría de categorías que hemos definido en el primer capítulo tienen como objetos los conjuntos que cumplen cierta propiedad. Ahora refinamos esta definición: si, por ejemplo, **Vec** era la categoría de los espacios vectoriales y las aplicaciones lineales entre ellos, ahora llamamos $\mathbf{Vec}_X$ a la categoría de los espacios vectoriales *que están en* X y las aplicaciones lineales entre ellos, y definimos $\mathbf{Vec} := \mathbf{Vec}_{\mathcal{U}}$ como la categoría de los espacios vectoriales pequeños. Hacemos lo mismo con todas las categorías definidas de esta forma, de modo que, en particular, $\mathrm{Ob}(\mathbf{Set}) = \mathcal{U}$ y, para un conjunto A, $\mathrm{Ob}(\mathbf{Set}_X) = X$.

Además, eliminamos de la definición de categoría (1.1) la restricción de que las clases de objetos y de morfismos sean clases, lo que nos permite definir la categoría de todas las categorías grandes.

Definición 2.8.

1. Dado un conjunto X, llamamos $\mathbf{Cat}_X$ a la categoría de las categorías en A y los funtores entre ellos.
2. Llamamos $\mathbf{Cat} := \mathbf{Cat}_{\mathcal{U}}$ a la categoría grande de *categorías pequeñas*.
3. Llamamos $\mathbf{CAT} := \mathbf{Cat}_{\mathcal{P}(\mathcal{U})}$ a la categoría de *categorías grandes*.

La eliminación de esta restricción nos permite definir también, por ejemplo, $\mathbf{Cls} := \mathbf{Set}_{\mathcal{P}(\mathcal{U})}$, la categoría de todas las clases.

Una limitación de esta teoría es que, si bien es posible definir la categoría de «todos los conjuntos pequeños» o de «todos los grupos pequeños», etc., no es posible definir la categoría de «todos los conjuntos» o de «todos los grupos», por lo que ha habido bastante debate sobre fundamentos de la teoría de categorías, y en general de todas las matemáticas, no basados en teoría de conjuntos.

2. Equivalencias de categorías

Un funtor en $\mathbf{Cat}_X$ es un isomorfismo si y sólo si es biyectivo sobre los morfismos. Esta idea de isomorfismo es la que podríamos esperar, ya que, por ejemplo, dos conjuntos, conjuntos preordenados o monoides vistos como categorías son isomorfos en **Cat** si y sólo si lo son en **Set**, **Prord** o **Mon**, respectivamente, y en **CAT** $\mathbb{Z}$-**Mod** es isomorfo a **Ab**.

En ocasiones, sin embargo, esta definición de isomorfismo es demasiado estricta. Por ejemplo, si K-$\mathbf{Vec_f}$ es la subcategoría de K-**Vec** de los espacios de dimensión finita, los morfismos de K-$\mathbf{Vec_f}$ se pueden representar mediante matrices, pero en general K-$\mathbf{Vec_f}$ no es isomorfa a K-**Mat** debido a que tiene muchos más objetos, y para obtener el isomorfismo habría que sustituir K-$\mathbf{Vec_f}$ por una subcategoría completa formada por un conjunto irredundante de representantes de los objetos por isomorfismo.

En estos casos es útil la noción de equivalencia. Una equivalencia entre categorías viene a ser un funtor que respeta las clases de isomorfía de los objetos y que «se porta bien» con los morfismos. Para caracterizar el significado de esto último, vemos que un funtor, al estar formado por una función sobre objetos y una sobre morfismos, puede ser inyectivo o suprayectivo sobre los objetos o sobre los morfismos. Serlo sobre los morfismos implica serlo sobre los objetos, y no queremos obligar a que sean biyectivos sobre objetos, por lo que es preferible abordar esta cuestión en función de los conjuntos hom.

Definición 2.9. Sea $T : \mathcal{C} \to \mathcal{D}$ un funtor:

1. T es *fiel* si todas las restricciones $T|_{\hom_{\mathcal{C}}(a,b)} : \hom_{\mathcal{C}}(a,b) \to \hom_{\mathcal{D}}(Ta,Tb)$ son inyectivas.
2. T es *pleno* si todas estas restricciones son suprayectivas.
3. T es una *inmersión* si es inyectivo sobre los morfismos.

Proposición 2.10. *Un funtor es una inmersión si y sólo si es fiel e inyectivo sobre los objetos, y es un isomorfismo si y sólo si es fiel, pleno y biyectivo sobre los objetos.*

Ejemplo 2.11.

1. Los funtores conjunto potencia y conjunto potencia contravariante son inmersiones no plenas.

2. El funtor $u : \mathbf{Met_c} \to \mathbf{Top}$ que lleva los espacios métricos a sus correspondientes espacios topológicos y los morfismos a ellos mismos es fiel y pleno, pero no es una inmersión.
3. Los funtores *espacio discreto* y *espacio indiscreto* $D, N : \mathbf{Set} \to \mathbf{Top}$, que asocian a cada conjunto la topología discreta o indiscreta, respectivamente, y llevan cada función a ella misma, son inmersiones plenas pero no son isomorfismos.
4. $\mathbf{0}$ es un objeto inicial de $\mathbf{Cat}$. El único funtor de $\mathbf{0}$ a una cierta categoría $\mathcal{C}$ es una inmersión, pero en general no es pleno.
5. $\mathbf{1}$ es un objeto final de $\mathbf{Cat}$. El único funtor de una cierta categoría $\mathcal{C}$ a $\mathbf{1}$ es pleno y suprayectivo en objetos, pero en general no es inyectivo en objetos, y es fiel si y sólo si $\mathcal{C}$ es fina.

PROPOSICIÓN 2.12. *La composición de funtores fieles, plenos o inmersiones es, respectivamente, fiel, plena o una inmersión.*

PROPOSICIÓN 2.13. *Dados dos funtores $F : \mathcal{B} \to \mathcal{C}$ y $G : \mathcal{C} \to \mathcal{D}$:*
1. *Si $g \circ f$ es fiel, f también lo es.*
2. *Si $g \circ f$ es pleno, g también lo es.*

Parece razonable entonces requerir que las equivalencias sean fieles y plenas.

DEFINICIÓN 2.14. Un funtor $T : \mathcal{C} \to \mathcal{D}$ es una *equivalencia* si es fiel, pleno y para cada $d \in \mathrm{Ob}(\mathcal{D})$ existe $c \in \mathrm{Ob}(\mathcal{C})$ con $Tc \cong d$, y entonces decimos que $\mathcal{C}$ y $\mathcal{D}$ son *equivalentes*, $\mathcal{C} \simeq \mathcal{D}$.

Estos requisitos son suficientes para satisfacer la noción intuitiva que hemos descrito antes, pues los funtores llevan isomorfismos a isomorfismos y, dados dos objetos isomorfos en $\mathrm{Im}\, T$, como T es biyectiva sobre los conjuntos hom, dos preimágenes cualesquiera de estos dos objetos serán isomorfas en $\mathcal{C}$.

EJEMPLO 2.15.
1. Si K es un cuerpo no trivial, K-$\mathbf{Mat} \simeq K$-$\mathbf{Vec}$. La equivalencia envía el objeto n a K^n y el morfismo $C : m \to n$ a la función $(v \rightsquigarrow Cv) : K^m \to K^n$.
2. La categoría $\mathbf{Top_m}$ de topologías metrizables y funciones continuas es equivalente a $\mathbf{Met_c}$, tomando como equivalencia $\mathbf{Met_c} \to \mathbf{Top_m}$ el funtor que a cada espacio métrico le asocia su correspondiente espacio topológico (y que deja los morfismos como están).
3. Dos conjuntos, conjuntos ordenados o monoides vistos como categorías son equivalentes si y sólo si son isomorfos. Esto no es cierto en general para conjuntos preordenados.

PROPOSICIÓN 2.16.
1. *La composición de equivalencias es una equivalencia.*

   DEMOSTRACIÓN. Claramente el funtor identidad es una equivalencia, por lo que la relación es reflexiva. Para ver que es transitiva, sean $S : \mathcal{B} \to \mathcal{C}$ y $T : \mathcal{C} \to \mathcal{D}$ equivalencias, la composición $T \circ S$ es fiel y plena por ser composición de funtores fieles y plenos (2.12), y para $d \in \mathcal{D}$, existe $c \in \mathcal{C}$ con $Tc \cong d$, pero entonces existe $b \in \mathcal{B}$ con $Sb \cong c$ y así $TSb \cong Tc \cong d$. $\square$

2. *La equivalencia de categorías es una relación de equivalencia en $\mathbf{Cat}_X$.*

   DEMOSTRACIÓN. Claramente el funtor identidad es una equivalencia, por lo que la equivalencia es reflexiva, y el apartado anterior prueba que es transitiva. Para ver que es simétrica, sea $T : \mathcal{C} \to \mathcal{D}$ una equivalencia, usando el axioma de elección, para cada objeto d de $\mathcal{D}$ tomamos un objeto Sd de $\mathcal{C}$ con $T(Sd) \cong d$ y un isomorfismo $h_d : TSd \to d$. Como T es biyectiva en conjuntos hom, para cada morfismo $g : a \to b$ en $\mathcal{D}$ existe un único morfismo $Sg : Sa \to Sb$ para el que la figura 1 conmuta.

$$\begin{tikzcd} TSa \arrow[r, "TSg"] \arrow[d, "h_a"'] & TSb \arrow[d, "h_b"] \\ a \arrow[r, "g"'] & b \end{tikzcd}$$

FIGURA 1. Actuación sobre los morfismos del funtor «inverso» S de una equivalencia.

A partir de esta figura, de la unicidad de Sg y de que T respeta las identidades y la composición, se concluye que S también las respeta, por lo que S es un funtor. S es fiel, pues para $g, g' : a \to b$ en $\mathcal{D}$ con $Sg = Sh$, $g = h_b \circ TSg \circ h_a^{-1} = h_b \circ TSg' \circ h_a^{-1} = g$'. Y S es plena, pues para $f : Sa \to Sb$ en $\mathcal{C}$, $g := h_b \circ Tf \circ h_a^{-1}$ cumple $g \circ h_a = h_b \circ Tf$ y, por la unicidad de la figura 1, $Tf = TSg$ y $f = Sg$.

Finalmente, para $c \in \mathrm{Ob}(\mathcal{C})$, queremos ver que $S(Tc) \cong c$. Pero $h := h_{Tc} : TSTc \cong Tc$ es la imagen por T de un morfismo $\tilde{h} : STc \to c$ y su inverso $k := h^{-1} : Tc \cong TSTc$ es la imagen por T de otro morfismo $\tilde{k} : c \to STc$. Entonces, como $T(\tilde{h} \circ \tilde{k}) = T(\tilde{h}) \circ T(\tilde{k}) = h \circ k = 1$ y T es fiel, $\tilde{h} \circ \tilde{k} = 1$ y $\tilde{h} : c \cong STc$ es un isomorfismo. □

Una primera utilidad de las equivalencias es visualizar la estructura de una categoría abstrayéndonos de los isomorfismos entre sus objetos[1, p. 42].

DEFINICIÓN 2.17. Un *esqueleto* de una categoría $\mathcal{C}$ es una subcategoría completa de $\mathcal{C}$ que tiene exactamente un elemento de cada clase de isomorfía de los objetos de $\mathcal{C}$.

PROPOSICIÓN 2.18.

1. *Toda categoría posee un esqueleto.*
2. *Todo esqueleto de $\mathcal{C}$ es equivalente a $\mathcal{C}$.*
3. *Todos los esqueletos de una misma categoría son isomorfos.*

DEMOSTRACIÓN. Sean $\mathcal{A}$ y $\mathcal{B}$ esqueletos de $\mathcal{C}$, cada objeto de $\mathcal{C}$, y en particular cada objeto a de $\mathcal{A}$, es isomorfo en $\mathcal{C}$ a un único objeto Ta de $\mathcal{B}$ por un isomorfismo $h_a : a \to Ta$, de forma que el funtor $T : \mathcal{A} \to \mathcal{B}$ definido sobre objetos de esta forma y sobre morfismos $f : a_1 \to a_2$ como en la figura 2 es un isomorfismo de categorías.

$$\begin{tikzcd} Ta_1 \arrow[r, "Tf"] & Ta_2 \\ a_1 \arrow[r, "f"'] \arrow[u, "h_{a_1}"] & a_2 \arrow[u, "h_{a_2}"'] \end{tikzcd}$$

FIGURA 2. Isomorfismo entre dos esqueletos de $\mathcal{C}$.

□

EJEMPLO 2.19.

1. El esqueleto de **Set** es la subcategoría completa de los números cardinales.
2. Si K es un cuerpo no trivial, el esqueleto de K-**Mat** es el propio K-**Mat**, y el de K-**Vec** está formado por los K^m para todo cardinal m.

3. El conocido teorema de clasificación de los grupos abelianos finitos da un esqueleto para la subcategoría de **Ab** formada por los grupos abelianos finitos. Lo mismo ocurre con el teorema de clasificación de grupos finitos, mucho más complejo, y la correspondiente subcategoría de **Grp**.

3. Funtores olvidadizos

Un tipo importante de funtor es el de los funtores que «olvidan parte de la estructura» de un objeto. Ya hemos visto alguno de ellos, como la equivalencia $\mathbf{Met_c} \to \mathbf{Top_m}$ que olvida la métrica concreta a utilizar. Otro ejemplo podría ser el funtor $\mathbf{Rng} \to \mathbf{Ab}$ que asocia a cada anillo su grupo abeliano aditivo. No hay una definición formal de este tipo de funtores, por lo que nos limitamos a decir que son fieles.

Definición 2.20. Una *categoría concreta* sobre una categoría $\mathcal{B}$ es un par $(\mathcal{C}, U)$ formado por una categoría $\mathcal{C}$ y un funtor fiel U, llamado *funtor olvidadizo*. Un *constructo* es una categoría concreta sobre **Set**.

Ejemplo 2.21.

1. La definición de constructo en la página 11 coincide con la definición anterior identificando los morfismos en el constructo con su representación en **Set** (debidamente etiquetada con el dominio y codominio) y tomando el funtor olvidadizo evidente.
2. **Ban** puede ser un constructo de dos formas «naturales»: con el funtor olvidadizo «obvio» y con el funtor $O : \mathbf{Ban} \to \mathbf{Set}$ que lleva los objetos X a $OX := \{x \in X \mid \|x\| \leq 1\}$ y los morfismos $f : X \to Y$ a $f|_{OX} : OX \to OY$.
3. La categoría **TopVec** de espacios vectoriales topológicos y las transformaciones lineales continuas se puede ver naturalmente como un constructo o como una categoría concreta sobre **Top** o **Vec**.
4. Las categorías concretas sobre **1** son las categorías finas.

4. Funtores contravariantes

Si $T : \mathcal{C} \to \mathcal{D}$ es un funtor, entonces $T : \mathcal{C}^{\mathrm{op}} \to \mathcal{D}^{\mathrm{op}}$ también es un funtor, y de hecho, podemos tomar el dual de una propiedad sobre categorías y funtores invirtiendo el sentido de los morfismos en las categorías pero no el de los funtores entre dichas categorías.

Sin embargo, es bastante común tener funtores de la forma $S : \mathcal{C}^{\mathrm{op}} \to \mathcal{D}$ o, equivalentemente, $S : \mathcal{C} \to \mathcal{D}^{\mathrm{op}}$. A los funtores $\mathcal{C} \to \mathcal{D}$ los llamamos *funtores covariantes*, y a los funtores $\mathcal{C}^{\mathrm{op}} \to \mathcal{D}$ los llamamos *funtores contravariantes*.

Ejemplo 2.22.

1. Ya hemos visto el funtor covariante $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ y el funtor contravariante $\mathcal{Q} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{Set}$ en los apartados 5 y 6 del ejemplo 2.2.
2. Sea K un cuerpo. $^* : K\text{-}\mathbf{Vec}^{\mathrm{op}} \to K\text{-}\mathbf{Vec}$ es un funtor contravariante que a cada espacio vectorial V le asigna el *espacio dual* V^* de aplicaciones lineales $V \to K$ y a cada morfismo $f : V \to U$ le asigna el morfismo $f^* : U^* \to V^*$ dado por $f^*u := u \circ f$.

5. Funtores hom

Si $\mathcal{C}$ es una categoría con conjuntos hom pequeños, podemos intentar considerar $\hom_{\mathcal{C}} : \mathcal{A} \times \mathcal{B} \to \mathbf{Set}$ como un funtor, donde $\mathrm{Ob}(\mathcal{A}) = \mathrm{Ob}(\mathcal{B}) = \mathrm{Ob}(\mathcal{C})$. Queremos encontrar una forma «natural» de llevar los morfismos del dominio a los del codominio, y para ello una buena idea es fijar uno de los dos componentes y ver cómo podría ser el «funtor parcial» respecto al otro.

Si $\mathcal{B}$ y $\mathcal{C}$ son categorías cualesquiera, su producto $\mathcal{B} \times \mathcal{C}$ (en una categoría de categorías apropiada) es una categoría con $\mathrm{Ob}(\mathcal{B} \times \mathcal{C}) = \mathrm{Ob}(\mathcal{B}) \times \mathrm{Ob}(\mathcal{C})$ y, para $b, b' \in \mathrm{Ob}(\mathcal{B})$ y $c, c' \in \mathrm{Ob}(\mathcal{C})$, $\hom((b, c), (b', c')) = \hom(b, b') \times \hom(c, c')$, con la composición y las identidades definidas por componentes. Entonces podemos definir los funtores parciales como sigue.

$$\begin{tikzcd} a & a \times b \arrow[l] \arrow[r] \arrow[d, dotted, "f \times g"] & b \arrow[d, "g"] \\ a' \arrow[from=u, "f"'] & a' \times b' \arrow[l] \arrow[r] & b' \end{tikzcd}$$

FIGURA 3. Morfismo producto de f y g. Los morfismos sin etiquetar son las proyecciones.

DEFINICIÓN 2.23. Un *bifuntor* en dos categorías $\mathcal{B}$ y $\mathcal{C}$ es un funtor con dominio $\mathcal{B} \times \mathcal{C}$.

DEFINICIÓN 2.24. Sea $T : \mathcal{B} \times \mathcal{C} \to \mathcal{D}$ un bifuntor.

1. Dado un objeto b de $\mathcal{B}$, el *funtor parcial* $T(b, -) : \mathcal{C} \to \mathcal{D}$ viene dado para objetos por $T(b, -)(c) \coloneqq T(b, c)$ y para morfismos por $T(b, -)(g) \coloneqq T(b, g) \coloneqq T(1_b, g)$.
2. Dado un objeto c de $\mathcal{C}$, el *funtor parcial* $T(-, c) : \mathcal{B} \to \mathcal{D}$ viene dado para objetos por $T(-, c)(b) \coloneqq T(b, c)$ y para morfismos por $T(-, c)(f) \coloneqq T(f, c) \coloneqq T(f, 1_c)$.

Para el caso que nos ocupa, sean a, a', b y b' objetos de $\mathcal{C}$. Para $g : b \to b'$, podríamos definir $\hom(a, g) : \hom(a, b) \to \hom(a, b')$ como $\hom(a, g)(f) \coloneqq g \circ f$, lo que nos da un funtor parcial $\hom(a, -) : \mathcal{C} \to \mathbf{Set}$.

Para $f : a \to a'$ intentamos hacer lo mismo y definir $\hom(f, b) : \hom(a, b) \to \hom(a', b)$, pero vemos que aparecen dificultades. Sin embargo, es fácil definir $\hom(f, b) : \hom(a', b) \to \hom(a, b)$ como $\hom(f, b)(g) \coloneqq g \circ f$, lo que nos da un funtor parcial contravariante $\hom(-, b) : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$.

Al primero lo llamamos *funtor hom covariante*, y al segundo, *funtor hom contravariante*. Una vez hecho esto es fácil definir el funtor hom global

$$\begin{aligned} \hom = \hom_{\mathcal{C}} : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} &\to \mathbf{Set} \\ (a, b) &\rightsquigarrow \hom(a, b) \\ (f, g) &\rightsquigarrow (h \rightsquigarrow g \circ h \circ f) \end{aligned}$$

El producto en sí también es un bifuntor, y no solo en **Cat** o en **CAT** sino en toda categoría que tenga productos de pares de objetos. En efecto, sea $\mathcal{C}$ tal categoría, y definimos el funtor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ del siguiente modo. Para objetos a y b, $a \times b$ es un producto cualquiera de a y b, aunque en general elegiremos un producto «canónico». Para morfismos $f : a \to a'$ y $g : b \to b'$, $f \times g : a \times b \to a' \times b'$ es el único morfismo para el que la figura 3 conmuta, donde los morfismos sin etiquetar son las proyecciones.

6. Conservación de propiedades

Puede ser interesante ver qué propiedades de los objetos y morfismos de una categoría son respetadas por funtores y cuáles no, o qué debe cumplir un funtor para que respete esa propiedad.

Para ello, una primera observación es que la idea de «respetar» una propiedad engloba varios conceptos. Por ejemplo, un funtor podría llevar monomorfismos a monomorfismos, pero que sin embargo no todos los monomorfismos del codominio sean imagen de un monomorfismo del dominio, incluso si son imagen de algún otro morfismo. Además, sería concebible un funtor para el que todo monomorfismo en su codominio fuera imagen de un monomorfismo y, sin embargo, hubiera morfismos que no son monomorfismos pero son llevados a uno. Todas estas propiedades de preservación son distintas, pero podemos definirlas de forma abstracta.

DEFINICIÓN 2.25. Sea $P((o_i)_i, (m_j)_j)$ una propiedad relativa a una serie de objetos $(o_i)_i$ y una serie de morfismos $(m_j)_j$ de una cierta categoría, y sea $T : \mathcal{C} \to \mathcal{D}$ un funtor.

1. *T preserva P* si, cuando $P((o_i)_i, (m_j)_j)$ se cumple en $\mathcal{C}$ para ciertos o_i y m_j de $\mathcal{C}$, entonces $P((T(o_i))_i, (T(m_j))_j)$ se cumple en $\mathcal{D}$.

2. *T refleja P* si, cuando $P((T(o_i))_i, (T(m_j))_j)$ se cumple en $\mathcal{D}$ para ciertos o_i y m_j de $\mathcal{C}$, entonces $P((o_i)_i, (m_j)_j)$ se cumple en $\mathcal{C}$.
3. *T levanta P (de forma única)* si, cuando $P((p_i)_i, (n_j)_j)$ se cumple en $\mathcal{D}$ para ciertos p_i y n_j de $\mathcal{D}$, entonces existen (únicos) $(o_i)_i$ y $(m_j)_j$ de $\mathcal{C}$ con cada $T(o_i) = p_i$, cada $T(m_j) = n_j$ y tal que $P((o_i)_i, (m_j)_j)$ se cumple en $\mathcal{C}$.

El concepto de funtor, como el de función, es bastante general y por ello no es de esperar que haya muchas propiedades respetadas por todos los funtores. Sin embargo, algunas son respetadas por todos o por muchos de ellos, como vamos a ver.

Proposición 2.26. *Los funtores preservan isomorfismos, secciones y retracciones.*

Demostración. Sean $T : \mathcal{C} \to \mathcal{D}$ un funtor y f y g morfismos de $\mathcal{C}$ con $g \circ f = 1$, entonces $Tg \circ Tf = T(g \circ f) = T1 = 1$. □

Un funtor entre dos categorías se puede ver como un funtor entre las categorías duales, de modo que si, por ejemplo, todos los funtores fieles, plenos, etc. preservan, reflejan o levantan una cierta propiedad, todos esos funtores también preservan, reflejan o levantan (respectivamente) la propiedad dual.

Proposición 2.27. *Todo funtor fiel y pleno refleja secciones y retracciones.*

Demostración. Si $T : \mathcal{C} \to \mathcal{D}$ es un funtor fiel y pleno y $f : a \to b$ es un morfismo de $\mathcal{C}$ tal que Tf es una sección, sea $g' : Tb \to Ta$ con $g' \circ Tf = 1$, por plenitud existe $g : b \to a$ con $g = Tg'$ y por tanto $Tg \circ Tf = T(g \circ f) = 1$, y por fidelidad $g \circ f = 1$. Las retracciones son la propiedad dual. □

Ambas condiciones de esta proposición son necesarias. Por ejemplo, la inclusión de monoides aditivos $\mathbb{N} \hookrightarrow \mathbb{Z}$ vista como funtor es fiel pero no pleno, y el único morfismo de monoides $\mathbb{N} \twoheadrightarrow \mathbf{1}$ es pleno pero no fiel, y ambos llevan una categoría en que la mayoría de morfismos no son secciones ni retracciones (sólo el 0 lo es) a una en la que todos los son.

Proposición 2.28. *Los funtores hom covariantes reflejan monomorfismos.*

Demostración. Si a es un objeto de $\mathcal{C}$ y $f : b \to c$ un morfismo tal que $\hom(a, f)$ es un monomorfismo, si $f \circ h = f \circ k$, entonces $\hom(a, f)(h) = \hom(a, f)(k)$ y por inyectividad $h = k$. □

Proposición 2.29. *Todo funtor fiel refleja monomorfismos y epimorfismos.*

Demostración. Sean $T : \mathcal{C} \to \mathcal{D}$ un funtor fiel y $f : a \to b$ un morfismo en $\mathcal{C}$ tal que Tf es un monomorfismo. Si $f \circ h = f \circ k$, entonces $Tf \circ Th = Tf \circ Tk$ y $Th = Tk$, y por fidelidad $h = k$. Los epimorfismos son la propiedad dual. □

Capítulo 3

Límites y colímites

Los funtores se pueden usar para modelar diagramas dentro de las matemáticas. Podríamos ver un diagrama «abstracto» como una categoría cuyos objetos y morfismos son los puntos y flechas del diagrama, y una instanciación de ese diagrama como un funtor de dicha categoría a la categoría que nos interesa. Este capítulo estudia una serie de conceptos que nos permitirán razonar sobre propiedades algebraicas en base a diagramas, y se basa principalmente en [1, caps. 11–13] y [4, cap. III.3–4].

1. Diagramas

DEFINICIÓN 3.1. Una categoría es *finita* si lo son su conjunto de objetos y su conjunto de morfismos.

DEFINICIÓN 3.2. Un *diagrama* en una categoría $\mathcal{C}$ es un funtor $D : \mathcal{S} \to \mathcal{C}$, y llamamos *esquema* del diagrama a $\mathcal{S}$. D es *pequeño* o *finito* si lo es $\mathcal{S}$.

EJEMPLO 3.3. Esta definición permite modelar una variedad de situaciones. Por ejemplo:

1. Un diagrama con esquema discreto es una familia de objetos.
2. Un diagrama con esquema $\mathbf{1}$ es un objeto, y uno con esquema $\downarrow$ es un morfismo.
3. Un diagrama con esquema $\downdownarrows$ es un par de morfismos con dominio y codominio común.
4. Un diagrama S cuyo esquema lo forman un punto distinguido d, un conjunto de objetos I y una flecha $f_i : d \to i$ para cada $i \in I$ (además de las identidades) es una *fuente*. Llamamos *dominio* de la fuente a Sd y *codominio* a $(Si)_{i\in I}$, y denotamos la fuente como $(Sf_i : Sd \to Si)_{i\in I}$.
5. De forma dual, un diagrama S cuyo esquema lo forman un punto distinguido c, un conjunto de objetos I y una flecha $g_i : i \to c$ para cada $i \in I$ (además de las identidades) es un *sumidero*. Llamamos *dominio* del sumidero a $(Si)_{i\in I}$ y *codominio* a Sd, y denotamos el sumidero como $(Sg_i : Si \to Sc)_{i\in I}$.

2. Límites

Podemos expresar muchas relaciones entre objetos mediante diagramas. Por ejemplo, un producto de $(a_i)_{i\in I}$ en una categoría $\mathcal{C}$ es una fuente $(p_i : b \to a_i)_{i\in I}$ tal que para cualquier otra fuente $(f_i : x \to a_i)_{i\in I}$ existe un único morfismo $g : x \to b$ tal que $f_i = p_i \circ g$ para todo $i \in I$. Por su parte, si consideramos el esquema $\mathcal{S}$ de la figura 1, el núcleo de dos morfismos f y g de $\mathcal{C}$ es la imagen de $\tilde{e}$ por un diagrama $D : \mathcal{S} \to \mathcal{C}$ tal que $D\tilde{f} = f$, $D\tilde{g} = g$ y, para cualquier otro $D' : \mathcal{S} \to \mathcal{C}$ que cumpla esto, existe un único $\overline{e} : D'k \to Dk$ tal que $D'\tilde{e} = D\tilde{e} \circ \overline{e}$. El hecho de que $f \circ D\tilde{e} = g \circ D\tilde{e}$ se deduce de que el diagrama sólo tiene una flecha $k \to b$.

$$\begin{tikzcd}
 & k \arrow[dl, "\tilde{e}"'] \arrow[dr] & \\
a \arrow[rr, shift left, "\tilde{f}"] \arrow[rr, shift right, "\tilde{g}"'] & & b
\end{tikzcd}$$

FIGURA 1. Esquema del diagrama asociado al núcleo de dos morfismos.

$$\begin{tikzcd}
 & c \arrow[dl, "f_i"'] \arrow[dr, "f_j"] & \\
Di \arrow[rr, "Ds"'] & & Dj
\end{tikzcd}$$

FIGURA 2. Conmutatividad de una fuente $(f_i)_i$ respecto a un diagrama D.

$$\begin{tikzcd}
 & k' \arrow[d, dotted, "\tilde{e}"'] \arrow[ddl, "e'"'] \arrow[ddr] & \\
 & k \arrow[dl, "e"'] \arrow[dr] & \\
a \arrow[rr, shift left, "f"] \arrow[rr, shift right, "g"'] & & b
\end{tikzcd}$$

FIGURA 3. Límite de un diagrama $\downdownarrows \to \mathcal{C}$.

Las descripciones de esta forma son tediosas y muy parecidas unas a otras. Afortunadamente, esta repetición se puede abstraer y usar en razonamientos mediante el concepto de límite.

DEFINICIÓN 3.4. Un *límite* de un diagrama $D : \mathcal{S} \to \mathcal{C}$ es una fuente $(f_i : d \to Di)_{i\in\mathrm{Ob}(\mathcal{S})}$ en $\mathcal{C}$ tal que:

1. Para cada $s : i \to j$ en $\mathcal{S}$, $f_j = Ds \circ f_i$, es decir, el diagrama 2 conmuta.
2. Para cualquier otra fuente $(g_i : x \to Di)_{i\in\mathrm{Ob}(\mathcal{S})}$ con esta propiedad, existe un único $s : x \to d$ con $g_i = f_i \circ s$ para cada $i \in \mathrm{Ob}(\mathcal{S})$.

EJEMPLO 3.5.

1. El producto de una familia $(a_i)_{i\in I}$ en $\mathcal{C}$ es el límite de un diagrama $D : I \to \mathcal{C}$ que a cada objeto del conjunto I visto como categoría discreta le asocia a_i.
2. El núcleo de un par de morfismos $f, g : a \to b$ en $\mathcal{C}$ es el límite de un diagrama $D : \downdownarrows \to \mathcal{C}$ cuyos dos morfismos no identidad van a parar a f y g, o más precisamente es el morfismo de este límite que va a parar a a. Esta situación se muestra en la figura 3.

   Si en vez de dos flechas tenemos un número arbitrario de flechas hablamos de *multi-núcleos*.
3. Una fuente $(f_c : s \to c)_{c\in\mathrm{Ob}(\mathcal{C})}$ es un límite del diagrama identidad $1_\mathcal{C} : \mathcal{C} \to \mathcal{C}$ si y sólo si s es un objeto inicial de $\mathcal{C}$.

   DEMOSTRACIÓN. Si s es inicial, $(f_c)_c$ conmuta respecto al diagrama identidad y, para cualquier otra fuente $(g_c : t \to c)_{c\in\mathrm{Ob}(\mathcal{C})}$ que también conmuta y cada objeto c, $g_c = f_c \circ g_s$. Recíprocamente, si $(f_c)_c$ es un límite del diagrama identidad y $h : s \to c$ un morfismo arbitrario, $h \circ f_s = f_c$ y en particular $f_c \circ f_s = f_c = f_c \circ 1_s$, con lo que tanto f_s como 1_s llevan la fuente $(f_c)_c$ a $(f_c)_c$ y, por la unicidad en la definición de límite, $f_s = 1_s$, de modo que $h = f_c$ y s es inicial. □
4. Si $D : \mathcal{S} \to \mathcal{C}$ es un diagrama y $s \in \mathrm{Ob}(\mathcal{S})$ es inicial, entonces D tiene límite $(Ds \to Di)_{i\in\mathrm{Ob}(\mathcal{S})}$.

PROPOSICIÓN 3.6. *El límite es esencialmente único, es decir, si $(f_i : d \to Di)_i$ es un límite de un diagrama $D : \mathcal{S} \to \mathcal{C}$, el resto de límites son precisamente las fuentes $(f_i \circ h : b \to Di)_i$ donde $h : b \to d$ es un isomorfismo.*

DEMOSTRACIÓN. Es fácil ver que las fuentes de esta forma son límites de D. Para el recíproco, si $(g_i : b \to Di)_i$ es otro límite de D, existe $h : b \to d$ con cada $g_i = f_i \circ h$

y $k : d \to b$ con cada $f_i = g_i \circ k$, pero entonces cada $f_i = f_i \circ h \circ k = f_i \circ 1_d$ y, por la unicidad en la definición de límite, $h \circ k = 1_d$, y análogamente $k \circ h = 1_b$, luego h es un isomorfismo. □

3. Colímites

El concepto dual al de límite es el de colímite.

DEFINICIÓN 3.7. Un sumidero $(g_i : Di \to c)_{i\in\mathrm{Ob}(\mathcal{S})}$ es un *colímite* de un diagrama $D : \mathcal{S} \to \mathcal{C}$ si:

1. Para cada $s : i \to j$ en $\mathcal{S}$, $g_j = g_i \circ Ds$.
2. Para cualquier otro sumidero $(h_i : Di \to x)_{i\in\mathrm{Ob}(\mathcal{S})}$ con esta propiedad, existe un único $s : c \to x$ con $h_i = s \circ g_i$ para cada $i \in \mathrm{Ob}(\mathcal{S})$.

EJEMPLO 3.8.

1. El coproducto de una familia $(a_i)_{i\in I}$ en $\mathcal{C}$ es el colímite de un diagrama $D : I \to \mathcal{C}$ que a cada objeto del conjunto I visto como categoría discreta le asocia a_i.
2. El conúcleo de un par de morfismos $f, g : a \to b$ en $\mathcal{C}$ es el colímite de un diagrama $D : \downdownarrows \to \mathcal{C}$ cuyos dos morfismos no identidad van a parar a f y g, o más precisamente es el morfismo de este colímite que va a parar a b.

   Si en vez de dos flechas tenemos un número arbitrario hablamos de *multiconúcleos*.
3. Un sumidero $(f_c : s \to c)_{c\in\mathrm{Ob}(\mathcal{C})}$ es un colímite del diagrama identidad en $\mathcal{C}$ si y sólo si s es un objeto final de $\mathcal{C}$.
4. Si $D : \mathcal{S} \to \mathcal{C}$ es un diagrama y $t \in \mathrm{Ob}(\mathcal{S})$ es final, entonces D tiene colímite $(Di \to Ds)_{i\in\mathrm{Ob}(\mathcal{S})}$.

Vemos así que un objeto inicial es lo mismo que un colímite del diagrama vacío (con esquema $\mathbf{0}$) y que un límite del diagrama identidad, mientras que un objeto final es lo mismo que un límite del diagrama vacío y que un colímite del diagrama identidad.

PROPOSICIÓN 3.9. *El colímite es esencialmente único, es decir, si $(g_i : Di \to c)_i$ es un colímite de un diagrama $D : \mathcal{S} \to \mathcal{C}$, el resto de colímites son precisamente los sumideros $(h \circ g_i : Di \to b)_i$ donde $h : c \to b$ es un isomorfismo.*

4. Productos fibrados

Los límites y colímites, al ser fuentes y sumideros, respectivamente, se pueden ver como diagramas que pueden tener a su vez un límite y un colímite. Claramente el límite de un límite es el propio límite y el colímite de un colímite es el propio colímite, pues de hecho el límite de una fuente es la propia fuente y el colímite de un sumidero es el propio sumidero. Sin embargo, puede ser interesante estudiar el límite de un sumidero o el colímite de una fuente. Empezamos con el primer caso, y vemos algunas definiciones.

DEFINICIÓN 3.10.

1. Un *producto fibrado múltiple* es un límite de un sumidero.
2. Un *producto fibrado* es un límite de un diagrama con esquema $(\bullet \to \bullet \leftarrow \bullet)$, es decir, de un sumidero de tamaño 2.
3. Dados dos morfismos $f : a \to c$ y $g : b \to c$, llamamos producto fibrado de a y b por c (respecto de f y g), $a \times_c b$, o simplemente producto fibrado de f y g, al producto fibrado del sumidero determinado por f y g.

Al representar gráficamente un producto fibrado, la flecha de dicho producto al codominio del sumidero es superflua, por lo que solemos omitirla. El resultado es un cuadrado como el de la figura 4, llamado *cuadrado cartesiano*. Veamos algunos ejemplos.

$$\begin{tikzcd}
x \arrow[dr, dotted] \arrow[drr, bend left] \arrow[ddr, bend right] & & \\
& a \times_c b \arrow[r, "\overline{f}"'] \arrow[d, "\overline{g}"] & b \arrow[d, "g"] \\
& a \arrow[r, "f"'] & c
\end{tikzcd}$$

FIGURA 4. Producto fibrado de a y b respecto a f y g.

$$\begin{tikzcd}
k \arrow[rr, "p_1 \circ e"] \arrow[dr, "e"'] \arrow[dd, "p_2 \circ e"'] & & a \arrow[dd, "f"] \\
& a \times b \arrow[ur, "p_1"'] \arrow[dl, "p_2"] & \\
b \arrow[rr, "g"'] & & c
\end{tikzcd}$$

FIGURA 5. Construcción canónica de productos fibrados.

EJEMPLO 3.11.

1. En **Set**, el producto fibrado de dos funciones $f : A \to C$ y $g : B \to C$ es $A \times_C B = \{(a, b) \in A \times B \mid f(a) = g(b)\}$ junto con las correspondientes restricciones al dominio de las proyecciones $A \times B \to A$ y $A \times B \to B$.
2. En una categoría fina, el producto fibrado coincide con el producto convencional.
3. En un constructo con un objeto libre 1 sobre el conjunto $\{*\}$, si $e : 1 \to c$ es un elemento de c y $f : b \to c$ es otro morfismo, llamamos *fibra* de f sobre e al producto fibrado de e y f.[10, p. 79] En **Set**, esta fibra coincide con la imagen inversa $f^{-1}(e)$. En **Top**, si $\rho : \mathbb{R} \to \mathbb{S}^1$ es la aplicación $t \rightsquigarrow e^{2\pi i t}$, la fibra de ρ sobre un punto de $\mathbb{S}^1$ es $\mathbb{Z}$ con la inclusión $\mathbb{Z} \hookrightarrow \mathbb{R}$.

El primer ejemplo muestra una relación entre los conceptos de producto fibrado, producto y núcleo, mostrando que en **Set** el primero se puede definir en términos de los otros dos. Esto no es casualidad, sino que de hecho ocurre en todas las categorías en las que dichos límites existen.

PROPOSICIÓN 3.12. *Sean $f : a \to c$ y $g : b \to c$ morfismos, $p_1 : a \times b \twoheadrightarrow a$ y $p_2 : a \times b \twoheadrightarrow b$ proyecciones canónicas y $e : k \rightarrowtail a \times b$ un núcleo de $f \circ p_1$ y $g \circ p_2$, entonces $p_1 \circ e : k \to a$ y $p_2 \circ e : k \to b$ forman un producto fibrado de f y g (figura 5).*

DEMOSTRACIÓN. Obviamente el cuadrado conmuta, y si $r : x \to b$ y $s : x \to a$ son tales que $f \circ s = g \circ r$, existe $t : x \to a \times b$ con $p_1 \circ t = s$ y $p_2 \circ t = r$, y entonces $f \circ p_1 \circ t = f \circ s = g \circ r = g \circ p_2 \circ t$ y existe un único $h : x \to k$ tal que $t = e \circ h$, con lo que $s = (p_1 \circ e) \circ h$ y $r = (p_2 \circ e) \circ h$. $\square$

EJEMPLO 3.13. La anterior proposición permite calcular el producto fibrado en muchas categorías comunes.

1. En **CRng**, **Grp**, R-**Mod** y **Top**, dados dos homomorfismos $f : A \to C$ y $g : B \to C$, el producto fibrado de A y B por C es el subanillo, subgrupo, submódulo o subespacio topológico, respectivamente, de $A \times B$ dado por $A \times_C B = \{(a, b) \in A \times B \mid f(a) = g(b)\}$.
2. En **Ab**, los homomorfismos $\mathbb{Z} \to \mathbb{Z}$ se identifican con los enteros. Entonces el producto fibrado de dos enteros m y n está formado por pares de enteros $(x, y) \in$

$$\begin{tikzcd}
d \arrow[r, "f"] \arrow[d, "g"'] & a \arrow[d, "\overline{g}"] \arrow[ddr, bend left] & \\
b \arrow[r, "\overline{f}"] \arrow[drr, bend right] & a \oplus_c b \arrow[dr, dotted] & \\
 & & x
\end{tikzcd}$$

FIGURA 6. Coproducto fibrado de a y b respecto a f y g.

$\mathbb{Z} \times \mathbb{Z}$ con $xn = ym$. Si $m, n \neq 0$, esto es isomorfo a $\mathbb{Z}$ y los morfismos del producto son enteros a y b tales que $ma = nb$ es el mínimo común múltiplo de a y b.

Si m es 0 pero n no, el producto fibrado es $\mathbb{Z}$, con el morfismo paralelo a m igual a 0 y el otro igual a 1. Si n es 0 pero m no es análogo, y si ambos son 0 el producto fibrado es el producto convencional. Esto encaja con la siguiente proposición.

PROPOSICIÓN 3.14. *El producto fibrado de dos objetos respecto a un objeto terminal es el producto de dichos objetos.*

Los productos fibrados se pueden usar para definir intersecciones. La intersección de dos objetos cualesquiera no tiene una definición general en teoría de categorías, pues en principio, en un constructo, los elementos de un objeto son intercambiables. Sin embargo, sí que se puede definir la intersección entre subobjetos del mismo objeto. En **Set**, el producto fibrado de un par de inclusiones es la intersección de los dominios, lo cual podemos generalizar.

DEFINICIÓN 3.15. La *intersección* de una familia $\{(a_i, m_i)\}_{i \in I}$ de subobjetos de un objeto c es el par (b, n) formado por el dominio b del producto fibrado múltiple de los m_i y el morfismo $n : b \to c$ de dicho producto.

PROPOSICIÓN 3.16. *La intersección de una familia de subobjetos es un subobjeto.*

DEMOSTRACIÓN. Sea (b, n) una intersección de la familia $\{(a_i, m_i)\}_{i \in I}$ de subobjetos de c y sea $(f_i : b \to a_i)_{i \in I \sqcup \{*\}}$ el producto fibrado correspondiente, con $f_* = n$. Para $g, h : d \to b$ con $n \circ g = n \circ h$, $m_i \circ f_i \circ g = m_i \circ f_i \circ h$ para cada $i \in I$ y por tanto $f_i \circ g = f_i \circ h$, y por la unicidad en la definición de límite es $g = h$, con lo que n es un monomorfismo. □

EJEMPLO 3.17.

1. La intersección de una familia vacía de subobjetos es el objeto total.
2. La intersección en **Set**, **Top**, **Rng**, K-**Mod** y (Ω, E)-**Alg** se corresponde con la intersección de conjuntos.
3. En una categoría fina, las intersecciones son los productos.

5. Coproductos fibrados

El concepto dual al producto fibrado es el coproducto fibrado.

DEFINICIÓN 3.18.

1. Un *coproducto fibrado múltiple* es un colímite de una fuente.
2. Un *coproducto fibrado* es un colímite de un diagrama con esquema $(\bullet \leftarrow \bullet \to \bullet)$, es decir, de una fuente de tamaño 2.
3. Dados dos morfismos $f : d \to a$ y $g : d \to b$, llamamos coproducto fibrado de a y b por d (respecto de f y g), $a \oplus_d b$, o simplemente coproducto fibrado de f y g, al coproducto fibrado de la fuente determinada por f y g.

Es común representar el coproducto fibrado con un cuadrado cartesiano como el de la figura 6.

$$\begin{tikzcd}
d \arrow[rr, "g"] \arrow[dd, "f"'] & & b \arrow[dl, "u_2"'] \arrow[dd, "c \circ u_2"] \\
& a \oplus b \arrow[dr, "c"] & \\
a \arrow[ur, "u_1"] \arrow[rr, "c \circ u_1"'] & & q
\end{tikzcd}$$

FIGURA 7. Construcción canónica de coproductos fibrados.

PROPOSICIÓN 3.19. *Sean $f : d \to a$ y $g : d \to b$ morfismos, $u_1 : a \rightarrowtail a \oplus b$ y $u_2 : b \rightarrowtail a \oplus b$ inclusiones canónicas y $c : a \oplus b \twoheadrightarrow q$ un conúcleo de $u_1 \circ f$ y $u_2 \circ g$, entonces $c \circ u_1 : a \to q$ y $c \circ u_2 : b \to q$ forman un coproducto fibrado de f y g (figura 7).*

EJEMPLO 3.20.

1. En **Set**, dadas dos funciones $f : D \to A$ y $g : D \to B$, el coproducto fibrado de A y B por D es el conjunto cociente de $A \sqcup B$ por la menor relación de equivalencia que identifica $f(d)$ con $g(d)$ para cada $d \in D$.
2. En **Rng**, **Grp**, R-**Mod** y **Top**, el coproducto fibrado se obtiene de manera similar.
3. La unión de espacios topológicos por un punto se puede expresar como el coproducto fibrado respecto a $\{*\}$ en **Top** o como el coproducto convencional en $\mathbf{Top}_*$.
4. En una categoría fina, un coproducto fibrado es un coproducto convencional.
5. El concepto dual de la intersección es el de *cointersección* de objetos cociente. Específicamente, en **Set**, si $(\sim_i)_{i \in I}$ son relaciones de equivalencia en un conjunto A, la cointersección de los conjuntos cociente $(\frac{A}{\sim_i})_{i \in I}$ (con las correspondientes proyecciones canónicas) es el conjunto cociente de A por la menor relación de equivalencia que contiene a $\bigcup_{i \in I} \sim_i$. La cointersección de una familia vacía de objetos cociente es el objeto original.

6. Completitud

Hemos visto que los límites y colímites, cuando existen, son únicos salvo isomorfismo, pero todavía no hemos estudiado su existencia. Sabemos que esta depende del caso; por ejemplo, en **Set** todos los productos existen, pero en una categoría discreta no existe el producto de dos o más elementos distintos, y en conjuntos parcialmente ordenados puede haber pares de elementos que tengan producto y otros que no. Se podría pensar entonces que caracterizar la existencia de límites de diagramas arbitrarios es una tarea excesivamente compleja, pero de hecho esta se puede reducir a la existencia de ciertos tipos de límites, como veremos a continuación.

Por brevedad no vemos los casos duales, aunque estos se pueden deducir fácilmente.

DEFINICIÓN 3.21. Sea $\mathcal{C}$ una categoría:

1. $\mathcal{C}$ *tiene límites* (*finitos*) o es (*finitamente*) *completa* si todos los diagramas pequeños (finitos) en $\mathcal{C}$ tienen límite.
2. $\mathcal{C}$ *tiene productos* (*finitos*) si todas las familias pequeñas (finitas) de objetos de $\mathcal{C}$ tienen producto.
3. $\mathcal{C}$ *tiene núcleos* si todo par de morfismos con el mismo dominio y codominio tiene un núcleo.
4. $\mathcal{C}$ *tiene productos fibrados* si todo par de morfismos con el mismo codominio tiene un producto fibrado.
5. $\mathcal{C}$ *tiene intersecciones* (*finitas*) si toda familia pequeña (finita) de subobjetos de un mismo objeto de $\mathcal{C}$ tiene intersección.

El concepto dual de la completitud es la cocompletitud.

Teorema 3.22. *Para una categoría $\mathcal{C}$, son equivalentes:*

1. *$\mathcal{C}$ es completa.*
2. *$\mathcal{C}$ tiene productos e intersecciones finitas.*
3. *$\mathcal{C}$ tiene productos y núcleos.*

Demostración. (1) $\implies$ (2) es obvio.

Veamos (2) $\implies$ (3). Si $\mathcal{C}$ tiene productos e intersecciones finitas, para $f, g : a \to b$, existe $a \times b$. Sean entonces $p_1 : a \times b \to a$ y $p_2 : a \times b \to b$ las proyecciones canónicas y $\hat{f}, \hat{g} : a \rightarrowtail a \times b$ los únicos morfismos con $p_1 \circ \hat{f} = p_2 \circ \hat{f} = 1_a$, $p_2 \circ \hat{f} = f$ y $p_2 \circ \hat{g} = g$, que claramente son monomorfismos, los subobjetos $(a, \hat{f})$ y $(a, \hat{g})$ de $a \times b$ tienen una intersección (k, n). Sean entonces $e_1, e_2 : k \to a$ tales que $n = \hat{f} \circ e_1 = \hat{g} \circ e_2$, y queremos ver que $e_1 = p_1 \circ \hat{f} \circ e_1 = p_1 \circ \hat{g} \circ e_2 = e_2$ es un núcleo de f y g. Pero $f \circ e_1 = p_2 \circ n = g \circ e_2 = g \circ e_1$, y si $e' : k' \to a$ cumple $f \circ e' = g \circ e'$, entonces es fácil ver que $\hat{f} \circ e' = \hat{g} \circ e'$, y por la definición de intersección existe un único $h : k' \to k$ con $e' = e_1 \circ h$.

Queda por probar (3) $\implies$ (1). Si $D : \mathcal{S} \to \mathcal{C}$ es un diagrama pequeño y $\mathcal{C}$ tiene productos y núcleos, existen los productos $(p_j : O \coloneqq \prod_{i \in \mathrm{Ob}(\mathcal{S})} Di \to Dj)_{j \in \mathrm{Ob}(\mathcal{S})}$ y $(\pi_s : C \coloneqq \prod_{t \in \mathrm{Mor}(\mathcal{S})} D(\mathrm{cod} t) \to D(\mathrm{cod} s))_{s \in \mathrm{Mor}(\mathcal{S})}$. Tomando $p_{\mathrm{cod} t}, Dt \circ p_{\mathrm{dom} t} : O \to D(\mathrm{cod} t)$ para cada $t \in \mathrm{Mor}(\mathcal{S})$, el par de morfismos $\hat{c}, \hat{d} : O \to M$ dados por $\pi_t \circ \hat{c} \coloneqq p_{\mathrm{cod} t}$ y $\pi_t \circ \hat{d} \coloneqq Dt \circ p_{\mathrm{dom} t}$ tiene un núcleo $e : k \to O$, y queremos ver que $(p_i \circ e : E \to Di)_{i \in \mathrm{Ob}(\mathcal{S})}$ es límite de D. En primer lugar, para cada $s : i \to j$ en $\mathcal{S}$,

$$p_j \circ e = \pi_s \circ \hat{c} \circ e = \pi_s \circ \hat{d} \circ e = Ds \circ p_i \circ e.$$

En segundo lugar, si $(f_i : x \to Di)_{i \in \mathrm{Ob}(\mathcal{S})}$ es otra fuente tal que $f_j = Ds \circ f_i$ para todo $s : i \to j$ en $\mathcal{S}$, sea $\hat{f} : x \to O$ dada por $p_i \circ \hat{f} \coloneqq f_i$, para $s : i \to j$ en $\mathcal{S}$,

$$\pi_s \circ \hat{c} \circ \hat{f} = p_j \circ \hat{f} = f_j = Ds \circ f_i = Ds \circ p_i \circ \hat{f} = \pi_s \circ \hat{d} \circ \hat{f},$$

y como esto se da para todo s, por definición de producto, $\hat{c} \circ \hat{f} = \hat{d} \circ \hat{f}$, pero por definición de núcleo, existe un único $g : x \to k$ con $\hat{f} = e \circ g$, de modo que $f_i = p_i \circ e \circ g$ para cada i y g es el único morfismo con esta propiedad. □

Para el caso finito tenemos una propiedad similar.

Teorema 3.23. *Para una categoría $\mathcal{C}$, son equivalentes:*

1. *$\mathcal{C}$ es finitamente completa.*
2. *$\mathcal{C}$ tiene productos finitos e intersecciones finitas.*
3. *$\mathcal{C}$ tiene productos finitos y núcleos.*
4. *$\mathcal{C}$ tiene productos fibrados y un objeto terminal.*

Demostración. (1) $\iff$ (2) $\iff$ (3) se prueba como en el teorema anterior, y (1) $\implies$ (4) es obvio usando que un objeto terminal es un límite de un diagrama vacío.

Queda ver (4) $\implies$ (2), pero un producto de una familia vacía es un objeto terminal, los de una unipuntual siempre existen y los de dos objetos son productos fibrados respecto a un objeto terminal, y basta usar la asociatividad del producto. Además, una intersección de una familia vacía es el subobjeto total, la de un sólo subobjeto es el propio subobjeto, la de dos subobjetos es un producto fibrado y es fácil ver que, si $(a_1, m_1), \dots, (a_n, m_n)$ son subobjetos de c con $n > 2$, (b, q) es intersección de $(a_1, m_1), \dots, (a_{n-1}, m_{n-1})$ y (b', q') es intersección de (b, q) y de (a_n, m_n), entonces (b', q') es intersección de $(a_1, m_1), \dots, (a_n, m_n)$. □

Ejemplo 3.24.

1. Las categorías **Set**, R-**Mod**, **Top**, **Ord** y **Grp** tienen productos y núcleos, por lo que son completas.

2. Un conjunto pequeño parcialmente ordenado visto como categoría es completo si y sólo si es un retículo completo, si y sólo si es cocompleto.

   Demostración. Sea $(C, \leq)$ este conjunto. Claramente, si $(C, \leq)$ es un retículo completo, es una categoría completa y cocompleta, pues tiene productos y coproductos (ínfimos y supremos) y, como toda categoría fina, tiene núcleos y conúcleos. Supongamos que $(C, \leq)$ es una categoría completa, y queremos ver que es un retículo completo. Si $S \subseteq C$ es no vacío, S tiene un ínfimo (el producto de los objetos). Ahora bien, sea $X \subseteq C$ el conjunto de cotas superiores de S, X no es vacío porque contiene al máximo de C (el objeto terminal), por lo que tiene un ínfimo p, pero para $s \in S$, $s \leq x$ para todo $x \in X$ y, por definición de ínfimo, $s \leq p$, de modo que p es una cota superior de S y es la menor de ellas, por lo que es un supremo, y así C es un retículo completo. Análogamente, si C es una categoría cocompleta, también es un retículo completo. $\square$

3. La subcategoría completa de **Set** de los conjuntos finitos es finitamente completa y cocompleta, pero no es completa ni cocompleta.

7. Preservación por funtores

En esta sección estudiamos si los funtores respetan los límites y colímites, o más precisamente, qué funtores preservan qué límites y qué colímites, y qué nos dice eso.

Como sabemos, el concepto de respetar una propiedad es ambiguo, y podemos hablar de que la preserva, la refleja, etc. A continuación formalizamos las formas en que un funtor puede respetar un límite y estudiamos sus relaciones. Algunas de estas definiciones son redundantes porque ya se han definido para propiedades, pero las definimos también para límites por claridad.

Por brevedad nos enfocamos en los conceptos relativos a límites, pero los conceptos y las propiedades duales se deducen fácilmente.

Definición 3.25. Un funtor $F : \mathcal{C} \to \mathcal{D}$ *preserva* un límite $(f_i : L \to Di)_i$ de un diagrama $D : \mathcal{S} \to \mathcal{C}$ si $(Ff_i : FL \to FDi)_i$ es un límite de $F \circ D$, y preserva los límites de cierto tipo (los de diagramas con cierto tipo de esquema) si preserva todos los límites de dicho tipo.

Ejemplo 3.26.

1. Los funtores identidad preservan todos los límites.
2. La composición de dos funtores que preserva un tipo de límite también preserva ese tipo de límite.
3. En **Top** y **Grph**, los funtores olvidadizos preservan límites y colímites.

   Demostración. Claramente la conmutatividad de la fuente respecto al diagrama se preserva. Si $(f_i : l \to Di)_i$ es un límite de $D : \mathcal{S} \to \mathbf{Top}$ y $(g_i : x \to FDi)_i$ es una fuente conmutativa al diagrama $F \circ D$, donde $F : \mathbf{Top} \to \mathbf{Set}$ es el funtor olvidadizo, dotando a x de la topología discreta se obtiene una fuente en **Top** que conmuta con D y por tanto una única función continua $h : x \to l$ con cada $g_i = f_i \circ h$. La unicidad de h como función en **Set** se debe a que todas las funciones $x \to l$ son continuas. Para los colímites la prueba es análoga pero usando la topología indiscreta. En **Grph** hacemos lo mismo considerando el grafo completo (incluyendo ejes reflexivos) y el grafo vacío. $\square$

4. Si $\mathcal{C}$ es un constructo con objetos libres para todos los conjuntos pequeños, su funtor olvidadizo conserva límites.

   Demostración. Sea $(f_i : l \to Di)_i$ un límite de $D : \mathcal{S} \to \mathcal{C}$, y sea $(g_i : x \to FDi)_i$ una fuente conmutativa al diagrama $F \circ D$, donde $F : \mathcal{C} \to \mathbf{Set}$ es el funtor olvidadizo. Si $\hat{x} \in \mathrm{Ob}(\mathcal{C})$ es el objeto libre asociado a x por la función $u : x \to F\hat{x}$, entonces para cada i existe $\hat{g}_i : \hat{x} \to Di$ con $g_i = F\hat{g}_i \circ u$, pero por hipótesis existe

$$\begin{tikzcd} a \arrow[r, "1"] \arrow[d, "1"'] & a \arrow[d, "f"] \\ a \arrow[r, "f"'] & b \end{tikzcd}$$

FIGURA 8. Caracterización de un monomorfismo por producto fibrado.

un único $h : \hat{x} \to l$ con cada $\hat{g}_i = f_i \circ h$, con lo que $g_i = Ff_i \circ (Fh \circ u)$. La unicidad de $Fh \circ u$ es clara si u es inyectiva, lo que ocurre si $\mathcal{C}$ tiene algún objeto que, como conjunto, tiene al menos 2 elementos, pero si este no es el caso sólo existe una función $x \to l$ a lo sumo. □

5. En **Grp**, **Rng** y **Vec**, los funtores olvidadizos preservan límites, pero no coproductos ni conúcleos.

   DEMOSTRACIÓN. La preservación de límites es por el apartado anterior. Para los coproductos, el coproducto en estas categorías es la suma directa y en **Set** es la unión disjunta, que en general es estrictamente más pequeña. Para los conúcleos, en **Rng**, si $f, g : \mathbb{Z}_5[X] \to \mathbb{Z}_5[X]$ son respectivamente la identidad y la función $p(X) \rightsquigarrow p(-X)$, el conúcleo de f y g en **Set** es el conjunto cociente resultante de identificar cada polinomio con el resultante de negar sus coeficientes impares, que es infinito, pero en **Rng** es $\frac{\mathbb{Z}_5[X]}{(X)} \cong \mathbb{Z}_5$, que es finito. Del mismo modo, en **Vec**, si $f, g : \mathbb{R} \to \mathbb{R}$ son la identidad y el producto por -1, el conúcleo de f y g es 0 en **Vec** pero es infinito en **Set**. Algo parecido ocurre en **Grp** restringiendo f y g a $\mathbb{Z} \to \mathbb{Z}$. □

6. Los funtores hom preservan límites.

   DEMOSTRACIÓN. Sean $c \in \mathrm{Ob}(\mathcal{C})$,$F \coloneqq \hom(c, -) : \mathcal{C} \to$ **Set**, $(f_i : l \to Di)_i$ un límite de $D : \mathcal{S} \to \mathcal{C}$ y $(g_i : X \to \hom(c, Di))_i$ una fuente en **Set** que conmuta con $F \circ D$. Para cada $x \in X$, $(g_i(x) : c \to Di)_i$ es una fuente en $\mathcal{C}$ que conmuta con D, por lo que existe un único morfismo $\hat{g}(x) : c \to l$ con cada $g_i(x) = f_i \circ \hat{g}(x)$ y así $\hat{g} : X \to \hom(c, l)$ es la única función con $g_i = f_i \circ \hat{g}$. □

7. El funtor potencia $\mathcal{P}$: **Set** $\to$ **Set** no preserva productos, coproductos, núcleos ni conúcleos.

Las propiedades de completitud y cocompletitud se pueden usar a la hora de determinar si un determinado funtor preserva límites, como vemos en las siguientes proposiciones fáciles de probar.

PROPOSICIÓN 3.27. *Si $\mathcal{C}$ es finitamente completa, un funtor $F : \mathcal{C} \to \mathcal{D}$ preserva límites finitos si y sólo si preserva productos finitos y núcleos, si y sólo si preserva productos fibrados y objetos terminales.*

PROPOSICIÓN 3.28. *Si $\mathcal{C}$ es finitamente completa, un funtor $F : \mathcal{C} \to \mathcal{D}$ preserva límites pequeños si y sólo si preserva productos y núcleos.*

PROPOSICIÓN 3.29. *Un funtor que preserva límites finitos preserva también monomorfismos y monomorfismos regulares.*

DEMOSTRACIÓN. Claramente conserva monomorfismos regulares ya que conserva núcleos, y claramente un morfismo f es un monomorfismo si y sólo si la figura 8 muestra un producto fibrado. □

Definición 3.30. Un funtor $F : \mathcal{C} \to \mathcal{D}$ *levanta* un tipo de límites (*de forma única*) si para todo diagrama $D : \mathcal{S} \to \mathcal{C}$ de dicho tipo y todo límite $(g_i)_i$ de $F \circ D$, existe un (único) límite $(f_i)_i$ de D con cada $g_i = Ff_i$. Del mismo modo, F *crea* un tipo de límites si para todo diagrama $D : \mathcal{S} \to \mathcal{C}$ y todo límite $(g_i)_i$ de $F \circ D$ existe una única fuente $(f_i : l \to Di)_{i \in \mathrm{Ob}(\mathcal{S})}$ en $\mathcal{C}$ con cada $g_i = Ff_i$, y además esta fuente es un límite de D.

Claramente todo funtor que crea un tipo de límite lo levanta de forma única, y todo funtor que lo levanta de forma única, lo levanta, pero los recíprocos no son ciertos, como vemos a continuación.

Ejemplo 3.31.

1. Los funtores olvidadizos (Ω, E)-**Alg** $\to$ **Set** y R-**Mod** $\to$ **Set** crean límites.

   Demostración. Sean $F : (\Omega, E)$-**Alg** $\to$ **Set** el funtor olvidadizo, $D : \mathcal{S} \to (\Omega, E)$-**Alg** un diagrama, $(f_i : L \to FDi)_i$ un límite de $F \circ D$ y $(f_i : (L, \mu) \to Di)_i$ una fuente arbitraria preimagen por F de dicho límite, y queremos ver que esta fuente existe, es única y es un límite. Si Ω tiene operadores $(i_1, \dots, i_k)$ con aridades $(n_1, \dots, n_k)$, para cada $p \in \{1, \dots, k\}$ e $i \in \mathrm{Ob}(\mathcal{S})$ definimos $t_{pi} : L^{n_p} \to Di$ como $t_{pi}(x_1, \dots, x_n) \coloneqq \nu_{pi}(f_i(x_1), \dots, f_i(x_n))$, donde ν_{pi} es el p-ésimo operador en Di. Entonces para cada p existe una fuente $(t_{pi} : L^{n_p} \to Di)_i$ y por tanto una única función $\mu_p : L^{n_p} \to L$ con cada $f_i \circ \mu_p = t_{pi} = \nu_p \circ (f_i \times \cdots \times f_i)$. Pero esta es precisamente la condición para que cada f_i sea un homomorfismo $(L, (\mu_1, \dots, \mu_n)) \to Di$, con lo que la fuente existe y es única. Que las operaciones μ_p respetan las igualdades en E se debe a que, al componerlas con cada f_i, las operaciones resultantes en términos de las ν_{pi} las respetan, y a la unicidad en la definición del límite en **Set**.

   Para ver que la fuente es un límite, si $(g_i : (X, \gamma) \to Di)_i$ es otra fuente que conmuta con D, $(Fg_i)_i$ conmuta con $F \circ D$ y por tanto existe una única función $h : X \to L$ con cada $g_i = f_i \circ h$, y queda ver que h es un homomorfismo. Pero para cada p y cada i, usando que los g_i son homomorfismos,

   $$\begin{aligned} f_i \circ h \circ \gamma_p = g_i \circ \gamma_p = \nu_{pi} \circ (g_i \times \cdots \times g_i) = \\ = \nu_{pi} \circ (f_i \times \cdots \times f_i) \circ (h \times \cdots \times h) = f_i \circ \mu_p \circ (h \times \cdots \times h), \end{aligned}$$

   y como $(f_i)_i$ es un límite, $h \circ \gamma_p = \mu_p \circ (h \times \cdots \times h)$, lo que termina la prueba.

   El caso de R-**Mod** se puede reducir al anterior convirtiendo el producto por elementos de R una cantidad potencialmente infinita de operaciones de aridad 1 en el módulo y convirtiendo cada instancia de una propiedad de este producto en una igualdad en la lista de igualdades, y usando que esta prueba no depende de que el número de operaciones e igualdades sea finito. □

2. Los funtores olvidadizos de **Top** y **Grph** a **Set** levantan límites y colímites de forma única, pero no los crean.

   Demostración. Para ver que los crean, en **Top** tomamos respectivamente la topología inicial respecto al límite en **Set** y la final respecto al colímite, y en **Grph** tomamos respectivamente la intersección de las preimágenes de los ejes por las funciones del límite en **Set** y la unión de las imágenes de los ejes por las del colímite. Esto nos da los únicos límites o colímites que son preimagen del correspondiente en **Set** por el funtor, pero en general hay más fuentes o sumideros que también son preimagen, por ejemplo tomando la topología discreta, la indiscreta, el grafo discreto y el grafo completo, respectivamente. □

3. El funtor olvidadizo de $\mathbf{Met_c}$ levanta límites finitos, pero no de forma única.

   Demostración. Sean F el funtor olvidadizo, $D : \mathcal{S} \to \mathbf{Met_c}$ un diagrama finito y $(f_i : L \to FDi)_i$ un límite de $F \circ D$, llamando d_i a la distancia de Di para cada i, $\hat{d}(x, y) \coloneqq \max_{i \in \mathrm{Ob}(\mathcal{S})} d_i(f_i(x), f_i(y))$ es una distancia en L, pues cumple con la simetría y desigualdad triangular y si $\hat{d}(x, y) = 0$, entonces $f_i(x) = f_i(y)$ para todo i y por la definición de límite es $x = y$. Además, si $(g_i : X \to Di)_i$ es una fuente en **Top**, $(g_i : X \to FDi)_i$ lo es en **Set** y existe $h : X \to L$ con cada $g_i = f_i \circ h$, pero para $x \in X$ y $\varepsilon > 0$, para cada i existe $\delta_i > 0$ tal que $g_i(B(x, \delta_i)) \subseteq B(g_i(x), \varepsilon)$, y tomando $\delta' := \text{mín}_i\, \delta_i$, para cada i, $f_i(h(B(x, \delta')) \subseteq B(f_i(h(x)), \varepsilon)$, con lo que $h(B(x, \delta')) \subseteq f_i^{-1}(B(f_i(h(x)), \varepsilon))$ y por tanto $h(B(x, \delta')) \subseteq \bigcap_i f_i^{-1}(B(f_i(h(x))), \varepsilon) = B(h(x), \varepsilon)$, de modo que h es continua. El levantamiento no es único porque multiplicando la métrica por una constante positiva se obtiene otra equivalente. □

TEOREMA 3.32. *Si $F : \mathcal{C} \to \mathcal{D}$ levanta límites y $\mathcal{D}$ es completa, entonces $\mathcal{C}$ es completa y F preserva límites pequeños.*

DEMOSTRACIÓN. Si D es un diagrama pequeño, $F \circ D$ tiene límite y este tiene una preimagen que es un límite de D y que F preserva. Como los límites son únicos salvo isomorfismo, F preserva el resto de límites al preservar isomorfismos. □

DEFINICIÓN 3.33. Un funtor $F : \mathcal{C} \to \mathcal{D}$ *refleja* un tipo de límites si para todo diagrama $D : \mathcal{S} \to \mathcal{C}$ de este tipo, toda fuente $(f_i : c \to Di)_{i \in \text{Ob}(\mathcal{S})}$ tal que $(Ff_i)_i$ es límite de $F \circ D$ es límite de D. F *detecta* un tipo de límites si todo diagrama $D : \mathcal{S} \to \mathcal{C}$ tal que $F \circ D$ tiene un límite tiene un límite.

PROPOSICIÓN 3.34.

1. *Todo funtor que levanta un tipo de límite lo detecta.*
2. *Un funtor crea un tipo de límite si y sólo si lo levanta de forma única y lo refleja.*

EJEMPLO 3.35. No hay una relación de implicación entre los conceptos de levantar límites y reflejarlos, como vemos en estos ejemplos.

1. El funtor olvidadizo **Top** $\to$ **Set** levanta límites de forma única pero no los refleja.
2. Un funtor biyectivo en objetos que lleva una categoría discreta a una no discreta refleja límites pero no los levanta ni los detecta.

Capítulo 4

Transformaciones naturales

Si tenemos un espacio vectorial de dimensión finita, sabemos que su espacio dual es isomorfo a este, por lo que el doble dual también lo es. Sin embargo, mientras el isomorfismo con el dual es algo más ad-hoc, el del doble dual se ve como algo más fundamental, pues se obtiene «de forma natural» un isomorfismo que podríamos llamar «canónico», dado por $v \rightsquigarrow (f \rightsquigarrow f(v))$.

El hecho de considerar una cierta transformación, o una cierta operación, como natural es algo común en distintas áreas de las matemáticas, y si bien el uso del término suele ser informal, en la primera mitad del siglo XX se hicieron esfuerzos por formalizarlo que, de hecho, fueron los que llevaron a la creación de la teoría de categorías. En este capítulo exploramos este concepto, basándonos principalmente en [1, cap. 6] y [4, I.4 y II.4–5].

DEFINICIÓN 4.1. Dados dos funtores $S, T : \mathcal{C} \to \mathcal{D}$, una *transformación natural* $\tau : S \to T$, también escrita como

$$\mathcal{C} \underset{T}{\overset{S}{\underset{\longrightarrow}{\overset{\longrightarrow}{\downarrow \tau}}}} \mathcal{D},$$

es una función $\tau : \mathrm{Ob}(\mathcal{C}) \to \mathrm{Mor}(\mathcal{D})$ que a cada objeto c en $\mathcal{C}$ le asigna un morfismo $\tau_c : Sc \to Tc$ de forma que, para todo morfismo $f : a \to b$ en $\mathcal{C}$, la figura 1 conmuta, esto es, $Tf \circ \tau_a = \tau_b \circ Sf$.

Si pensamos en los funtores S y T como imágenes en $\mathcal{D}$ de los objetos y los morfismos de $\mathcal{C}$, una transformación natural nos da un conjunto de morfismos de la imagen de S a la de T de forma que todos los cuadrados como los de la figura 1 conmutan.

También podemos pensar en la transformación natural como un morfismo «genérico» en $\mathcal{D}$, parametrizado por un objeto de $\mathcal{C}$, y lo que nos dice el diagrama a grandes rasgos es que el morfismo «se comporta igual» para cualquier valor del parámetro.

EJEMPLO 4.2.

1. En $\mathbf{Vec_f}$, un isomorfismo de un espacio V a su dual V^* viene dado por una forma bilineal no degenerada, pero como esta no es única y no hay una forma general de elegir una, el isomorfismo no puede ser natural. Sin embargo, $\phi_V : V \to V^{**}$ dado por $\phi_V(x)(f) \coloneqq f(x)$ es un isomorfismo y se puede definir del mismo modo para todo V, por lo que ϕ es una transformación natural $\phi : 1_{\mathbf{Vec_f}} \to (^{**})$ y, de hecho, es una transformación natural $\phi : 1_{K\text{-}\mathbf{Vec}} \to (^{**})$, donde el funtor $(^{**})$ viene dado por la composición $(^{**}) : \mathbf{Vec} \overset{(^*)}{\to} \mathbf{Vec}^{\mathrm{op}} \overset{(^*)}{\to} \mathbf{Vec}$.
2. Para cada $n \in \mathbb{N}$, el determinante de matrices $n \times n$ es una transformación natural. Más concretamente, si para un anillo conmutativo R tomamos la función

$$\begin{tikzcd} a \arrow[d, "f"] & Sa \arrow[r, "\tau_a"] \arrow[d, "Sf"'] & Ta \arrow[d, "Tf"] \\ b & Sb \arrow[r, "\tau_b"'] & Tb \end{tikzcd}$$

FIGURA 1. Transformación natural.

$$\begin{tikzcd}
\cdot \arrow[r, "h"] \arrow[d, "f"'] & \cdot \arrow[d, "g"] \\
\cdot \arrow[r, "k"'] & \cdot
\end{tikzcd}$$

Figura 2. Morfismos $f \to g$ de la categoría de flechas.

determinante $\det_R : \mathcal{M}_n(R) \to R$, $\det_R$ no es un homomorfismo de anillos porque no conserva la suma, pero sí es un homomorfismo de grupos multiplicativos $\det_R : \mathrm{GL}_n(R) \to R^*$. Si f es un homomorfismo de anillos, podemos definir $\mathrm{GL}_n(f)$ componente a componente y $f^* = f$, y entonces $\det : \mathrm{GL}_n \to (\cdot)^*$ es una transformación natural entre funtores $\mathbf{CRng} \to \mathbf{Grp}$.

1. Categorías de funtores

Las transformaciones naturales se pueden componer de varias formas. Una forma sencilla es componiendo los morfismos imagen de dos transformaciones naturales

$$\mathcal{C} \; \begin{array}{c} \longrightarrow \\ \downarrow\sigma \\ \longrightarrow \\ \downarrow\tau \\ \longrightarrow \end{array} \; \mathcal{D}.$$

Definición 4.3. Dados tres funtores $R, S, T : \mathcal{C} \to \mathcal{D}$ y dos transformaciones naturales $\sigma : R \to S$ y $\tau : S \to T$, llamamos *composición vertical* de σ y τ a la transformación natural $\tau \cdot \sigma : R \to T$ dada por $(\tau \cdot \sigma)_c := \tau_c \circ \sigma_c$.

Claramente esta composición es asociativa y tiene identidad, lo que sugiere la siguiente definición.

Definición 4.4. Dadas dos categorías $\mathcal{C}$ y $\mathcal{D}$, la *categoría de funtores* de $\mathcal{C}$ y $\mathcal{D}$, $\mathcal{D}^{\mathcal{C}}$, es aquella que tiene como objetos los funtores $\mathcal{C} \to \mathcal{D}$, como morfismos las transformaciones naturales entre ellos, como composición la composición vertical y como identidad la *transformación natural identidad*, que para un funtor $T : \mathcal{C} \to \mathcal{D}$ viene dada por $1_T(c) := 1_{Tc}$.

Ejemplo 4.5.

1. Si I es un conjunto, la categoría de funtores $\mathcal{C}^I$ es precisamente la I-ésima potencia de $\mathcal{C}$ en una categoría de categorías apropiada. En concreto, si X es otro conjunto, X^I es el conjunto de familias $(x_i)_{i \in I}$ con entradas en X, o de funciones $I \to X$.
2. Si X es un conjunto, $\{0,1\}^X$ es su conjunto potencia.
3. $\mathcal{C}^{\mathbf{1}}$ es isomorfa a $\mathcal{C}$, mientras que $\mathcal{C}^{\mathbf{0}}$ es la categoría unipuntual.
4. $\mathcal{C}^{\downarrow}$ es la *categoría de flechas* de $\mathcal{C}$, cuyos objetos son los morfismos de $\mathcal{C}$ y cuyos morfismos $f \to g$ son los pares de flechas (h, k) para los que la figura 2 conmuta.
5. Si M es un monoide, $\mathbf{Set}^M$ es la categoría de las acciones de M sobre algún conjunto.
6. Consideremos el *funtor diagonal* $\Delta : \mathcal{C} \to \mathcal{C}^{\mathcal{S}}$. Para cada objeto c, $\Delta c : \mathcal{S} \to \mathcal{C}$ es el funtor constante que lleva cada objeto a c y cada morfismo a 1_c, y para cada morfismo f, $\Delta f : \Delta a \to \Delta b$ es la transformación natural que lleva todos los objetos a f. Si $D : \mathcal{S} \to \mathcal{C}$ es un diagrama, un morfismo $\Delta c \to D$ es una fuente que conmuta con D, y un morfismo $D \to \Delta c$ es un sumidero que conmuta con D.

Proposición 4.6. *Dadas dos categorías $\mathcal{C}$ y $\mathcal{D}$:*

1. *Si $\mathcal{C}$ y $\mathcal{D}$ son pequeñas, también lo es $\mathcal{D}^{\mathcal{C}}$.*

Demostración. Los objetos de $\mathcal{D}^{\mathcal{C}}$ son funciones $\mathrm{Mor}(\mathcal{C}) \to \mathrm{Mor}(\mathcal{D})$, los morfismos son funciones $\mathrm{Ob}(\mathcal{C}) \to \mathrm{Mor}(\mathcal{D})$, y ambos conjuntos de funciones son pequeños. $\square$

$$\begin{tikzcd}
\hom(a,x)\times\hom(b,x) \arrow[r, "\tau_x"] \arrow[d, "\hom(a,f)\times\hom(b,f)"] & \hom(a\oplus b,x) \arrow[d, "\hom(a\oplus b,f)"] \\
\hom(a,y)\times\hom(b,y) \arrow[r, "\tau_y"'] & \hom(a\oplus b,y)
\end{tikzcd}$$

(A) Diagrama asociativo.

$$\begin{tikzcd}
(h,k) \arrow[r, rightsquigarrow] \arrow[d, rightsquigarrow] & h\oplus k \arrow[d, rightsquigarrow] \\
(f\circ h, f\circ k) \arrow[r, rightsquigarrow] & (f\circ h)\oplus(f\circ k) = f\circ(h\oplus k)
\end{tikzcd}$$

(B) Persecución de flechas.

FIGURA 3. Transformación natural en un coproducto.

2. *Si $\mathcal{C}$ es pequeña y $\mathcal{D}$ es una clase, $\mathcal{D}^{\mathcal{C}}$ es una clase.*

   DEMOSTRACIÓN. Si $T : \mathcal{C} \to \mathcal{D}$ es un funtor, como $\mathrm{Mor}(\mathcal{C})$ es pequeño, su imagen por $\mathcal{C}$ también y T es pequeño, por lo que el conjunto de funtores es una clase, y análogamente para el de transformaciones naturales. $\square$

3. *Si $\mathcal{C}$ es pequeña y $\mathcal{D}$ tiene conjuntos hom pequeños, $\mathcal{D}^{\mathcal{C}}$ tiene conjuntos hom pequeños.*

   DEMOSTRACIÓN. Los elementos de $\hom_{\mathcal{D}^{\mathcal{C}}}(S,T)$ son funciones de $\mathrm{Ob}(\mathcal{C})$ a $\bigcup_{c\in\mathrm{Ob}(\mathcal{C})}\hom(Sc,Tc)$, pero estos conjuntos son pequeños y por tanto el conjunto de estas funciones también. $\square$

2. Isomorfismos naturales

Los isomorfismos de las categorías de funtores son particularmente importantes, pues proporcionan una forma general de obtener isomorfismos que, además, conmutan con los morfismos imágenes de los correspondientes funtores, proporcionando una forma directa de pasar de uno al otro.

DEFINICIÓN 4.7. Un *isomorfismo natural* es un isomorfismo en una categoría de funtores, es decir, una transformación natural $\tau : S \to T$ en que todos los τ_c son isomorfismos, y si existe decimos que los funtores S y T son *naturalmente isomorfos*.

EJEMPLO 4.8.

1. La transformación natural $\phi : 1 \to (^{**})$ del ejemplo 4.2 no es un isomorfismo natural, pero sí lo es si restringimos el dominio y codominio de los funtores a K-$\mathbf{Vec_f}$.
2. Si a y b son objetos de una cierta categoría con coproducto $a \oplus b$, existe una biyección $\hom(a,d)\times\hom(b,d) \cong \hom(a\oplus b,d)$ natural respecto a d, como se muestra en la figura 3. Esto se puede generalizar de forma obvia a coproductos de una familia arbitraria de objetos.
3. Del mismo modo, si existe el producto de los objetos a y b, existe una biyección $\hom(c,a)\times\hom(c,b) \cong \hom(c,a\times b)$ natural respecto a c, que también se puede extender a productos arbitrarios.

Ciertas propiedades de funtores se pueden caracterizar mediante isomorfismos naturales.

PROPOSICIÓN 4.9. *Un funtor $T : \mathcal{C} \to \mathcal{D}$ es una equivalencia si y sólo si existe $S : \mathcal{D} \to \mathcal{C}$ tal que $S \circ T = 1_{\mathcal{C}}$ y $T \circ S = 1_{\mathcal{D}}$.*

DEMOSTRACIÓN. Si T es una equivalencia, hemos visto en la prueba de la proposición 2.16 que existen $S : \mathcal{D} \to \mathcal{C}$ y un isomorfismo natural $h : T \circ S \to 1_{\mathcal{D}}$. Además, si c es un objeto de $\mathcal{C}$, $h_{Tc}^{-1} : Tc \to TSTc$ es un isomorfismo, y como T es fiel y pleno, existe un único $\mu_c : c \to STc$ con $T\mu_c = h_{Tc}^{-1}$. La naturalidad de $\mu : 1_{\mathcal{C}} \to S \circ T$ se deriva de la de h y la fidelidad de T.

Recíprocamente, si existen $S : \mathcal{D} \to \mathcal{C}$ e isomorfismos naturales $\mu : 1_{\mathcal{C}} \to S \circ T$ y $\sigma : T \circ S \to 1_{\mathcal{D}}$, para cada objeto d en $\mathcal{D}$, $T(Sd) \cong d$, y queda ver que T es fiel y pleno. Es

$$\begin{tikzcd}
S'Sb \arrow[r, "\sigma_{Sb}"] \arrow[d, "S'\tau_b"'] \arrow[dr, "\sigma \circ \tau"] & T'Sb \arrow[d, "T'\tau_b"] \\
S'Tb \arrow[r, "\sigma_{Tb}"'] & T'Tb
\end{tikzcd}$$

FIGURA 4. Composición horizontal de transformaciones naturales.

fiel porque, dados dos morfismos $f, g : a \to b$ en $\mathcal{C}$,

$$Tf = Tg \implies \mu_b \circ f = STf \circ \mu_a = STg \circ \mu_a = \mu_b \circ g \implies f = g,$$

y es plena porque, dado un morfismo $g : Ta \to Tb$ en $\mathcal{D}$, $f := \mu_b^{-1} \circ Sg \circ \mu_a$ cumple $Tf = g$. $\square$

DEFINICIÓN 4.10. Un funtor $T : \mathcal{C} \to \mathbf{Set}$ es *representable* por un objeto c si es naturalmente isomorfo a $\hom(c, -)$.

EJEMPLO 4.11.

1. Un funtor olvidadizo es representable por un objeto c si y sólo si c es un objeto libre sobre $\{*\}$.
2. El funtor olvidadizo de **Ban** no es representable.

PROPOSICIÓN 4.12. *Dados un funtor $T : \mathcal{C} \to \mathbf{Set}$, un objeto c de $\mathcal{C}$ y $x \in Tc$, existe una única transformación natural $\tau : \hom(c, -) \to T$ con $\tau_c(1_c) = x$.*

DEMOSTRACIÓN. Si τ es tal transformación natural, b es un objeto de $\mathcal{C}$ y $f : c \to b$, necesariamente $\tau_b(f) = \tau_b(f \circ 1_c) = (\tau_b \circ \hom(c, f))(1_c) = (Tf \circ \tau_c)(1_c) = (Tf)(x)$. Esta fórmula define una transformación natural, que es pues la única que cumple la condición. $\square$

3. Composición horizontal

Además de componer transformaciones naturales verticalmente, podemos componer dos transformaciones naturales $\mathcal{B} \underset{\longrightarrow}{\overset{\longrightarrow}{\downarrow \tau}} \mathcal{C} \underset{\longrightarrow}{\overset{\longrightarrow}{\downarrow \sigma}} \mathcal{D}$ de la siguiente manera.

DEFINICIÓN 4.13. Dados cuatro funtores $S, T : \mathcal{B} \to \mathcal{C}$ y $S', T' : \mathcal{C} \to \mathcal{D}$ y dos transformaciones naturales $\tau : S \to T$ y $\sigma : S' \to T'$, llamamos *composición horizontal* de τ y σ a la transformación natural $\sigma \circ \tau : S' \circ S \to T' \circ T$ dada por la figura 4 como $(\sigma \circ \tau)_b := T'\tau_b \circ \sigma_{Sb} = \sigma_{Tb} \circ S'\tau_b$, donde la conmutatividad se debe a la naturalidad de σ.

Es fácil ver que la composición horizontal de transformaciones naturales es otra transformación natural y que esta operación es asociativa. Además, la transformación natural identidad en un funtor identidad es una identidad de esta composición, por lo que las transformaciones naturales son los morfismos de una categoría cuyos objetos son categorías.

PROPOSICIÓN 4.14 (Ley del intercambio). *Dadas las transformaciones naturales*

$$\mathcal{B} \overset{\overset{\longrightarrow}{\downarrow\sigma}}{\underset{\underset{\longrightarrow}{\downarrow\tau}}{\longrightarrow}} \mathcal{C} \overset{\overset{\longrightarrow}{\downarrow\sigma'}}{\underset{\underset{\longrightarrow}{\downarrow\tau'}}{\longrightarrow}} \mathcal{D},$$

se tiene $(\tau' \cdot \sigma') \circ (\tau \cdot \sigma) = (\tau' \circ \tau) \cdot (\sigma' \circ \sigma)$.

DEMOSTRACIÓN. Ver el diagrama en la portada del trabajo. $\square$

Esto muestra un ejemplo de *bicategoría* o *categoría bidimensional*, un par de categorías con el mismo conjunto de morfismos que cumple la ley del intercambio y en que las identidades de una de las dos lo son de la otra. Las categorías implicadas son, por supuesto, una categoría de categorías con sus transformaciones naturales y la unión disjunta de las categorías de funtores entre ellas.

Capítulo 5

Adjunciones

La definición que hemos visto de objeto libre (1.22) está asociada a constructos, pero no hay razón para limitarse a este caso. Los objetos libres pueden existir en categorías cualesquiera que sean el dominio de un cierto funtor, no sólo del funtor olvidadizo de un constructo, y cuando todos los elementos en el codominio del funtor admiten un objeto libre podemos definir un funtor libre asociado a este funtor, que tiene propiedades interesantes. En este capítulo estudiamos las principales propiedades de los objetos libres, funtores libres y otras estructuras relacionadas, basándonos principalmente en [4, II.5, III.1–2 y IV.1–2, 7].

1. Flechas universales

Definición 5.1. Sean $U : \mathcal{C} \to \mathcal{B}$ un funtor y b un objeto de $\mathcal{B}$, una *flecha universal* de b a U es un par (c, u) formado por un objeto c de $\mathcal{C}$, el *objeto libre*, y un morfismo $u : b \to Uc$ en $\mathcal{B}$, tales que para todo morfismo $f : b \to Ux$ en $\mathcal{B}$ existe un único morfismo $\hat{f} : c \to x$ en $\mathcal{C}$ tal que $f = U\hat{f} \circ u$, como se muestra en la figura 1.

Ejemplo 5.2. Las flechas universales suelen representar inmersiones de objetos en un cierto objeto completado o con estructura adicional.

1. En el caso de constructos, esta definición de objeto libre coincide con la vista en el capítulo 1.
2. Si $U : \mathbf{Field} \hookrightarrow \mathbf{Dom}$ es el funtor inclusión de la subcategoría completa de los cuerpos en la categoría de dominios, una flecha universal de un dominio D a U es el conjunto cociente $Q(D)$ junto con la inclusión.
3. Consideremos la categoría **MGrph** de los *multigrafos*, los grafos dirigidos (no necesariamente finitos) que admiten varios ejes entre dos mismos vértices. Si $U : \mathbf{Cat} \to \mathbf{MGrph}$ es el funtor que «olvida» la composición y las identidades, un objeto libre de un multigrafo M a U es una categoría cuyos objetos son los vértices de M y cuyos morfismos entre dos objetos son los caminos entre ellos en M, tomando como composición la concatenación de caminos y como identidad el camino vacío.
4. Si $D : \mathcal{S} \to \mathcal{C}$ es un diagrama, un morfismo $f : D \to \Delta c$ en $\mathcal{C}^{\mathcal{S}}$ es un sumidero con codominio c que conmuta con D, de modo que un colímite de D es una flecha universal de D a $\Delta : \mathcal{C} \to \mathcal{C}^{\mathcal{S}}$.

Nos gustaría ver que las flechas universales son únicas salvo isomorfismo, pero para ello primero tenemos que ver qué significa esto. Una forma de hacerlo es definir la categoría en la que «viven» estas flechas.

Definición 5.3. Si $U : \mathcal{C} \to \mathcal{B}$ es un funtor y b es un objeto de $\mathcal{B}$, la *categoría de objetos U-bajo b*, $(b \downarrow U)$, tiene como objetos los pares (c, f) formados por un objeto c de $\mathcal{C}$

$$\begin{tikzcd} b \arrow[r, "u"] \arrow[dr, "f"'] & Uc \arrow[d, dotted, "U\hat{f}"] & & c \arrow[d, dotted, "\hat{f}"] \\ & Ux & & x \end{tikzcd}$$

Figura 1. Flecha universal de un objeto a un funtor.

$$\begin{tikzcd}
x \arrow[d, dotted, "\hat{f}"'] & Tx \arrow[d, dotted, "T\hat{f}"'] \arrow[dr, "f"] & \\
c & Tc \arrow[r, "v"'] & b
\end{tikzcd}$$

FIGURA 2. Flecha universal de un funtor a un objeto.

y un morfismo $f : b \to Uc$, y como morfismos $h : (c, f) \to (c', f')$ los morfismos $h : c \to c'$ en $\mathcal{C}$ tales que $f' = Uh \circ f$, y como composición e identidad las correspondientes en $\mathcal{C}$.

PROPOSICIÓN 5.4. *La flecha universal de un objeto b de $\mathcal{B}$ a un funtor $U : \mathcal{C} \to \mathcal{B}$, si existe, es única salvo isomorfismo en $(b \downarrow U)$.*

DEMOSTRACIÓN. Es el objeto inicial de $(b \downarrow U)$. □

El concepto dual al de flecha universal de un objeto a un funtor es el de flecha universal de un funtor a un objeto.

DEFINICIÓN 5.5. Sean $T : \mathcal{C} \to \mathcal{B}$ un funtor y b un objeto de $\mathcal{B}$, una *flecha universal* de T a b es un par (c, v) formado por un objeto c de $\mathcal{C}$ y un morfismo $v : Tc \to b$ en $\mathcal{B}$, tales que para todo morfismo $f : Tx \to b$ en $\mathcal{B}$ existe un único morfismo $\hat{f} : x \to c$ tal que $f = v \circ T\hat{f}$, como se muestra en la figura 2.

EJEMPLO 5.6. Si $D : \mathcal{S} \to \mathcal{C}$ es un diagrama, un morfismo $f : \Delta c \to D$ en $\mathcal{C}^{\mathcal{S}}$ es una fuente con dominio c que conmuta con D, con lo que un límite de D es una flecha universal de Δ a D.

DEFINICIÓN 5.7. Si $T : \mathcal{C} \to \mathcal{B}$ es un funtor y b es un objeto de $\mathcal{B}$, la *categoría de objetos T-sobre b*, $(T \downarrow b)$, tiene como objetos los pares (c, f) formados por un objeto c de $\mathcal{C}$ y un morfismo $f : Tc \to b$; como morfismos $h : (c, f) \to (c', f')$ los morfismos $h : c \to c'$ en $\mathcal{C}$ tales que $f = f' \circ Th$.

PROPOSICIÓN 5.8. *La flecha universal de un objeto b de $\mathcal{B}$ a un funtor $T : \mathcal{C} \to \mathcal{B}$, si existe, es única salvo isomorfismo en $(T \downarrow b)$.*

2. Lema de Yoneda

El lema de Yoneda es un resultado clásico sobre transformaciones naturales que apunta a una relación entre estas y las flechas universales. Antes de verlo es conveniente definir algunos funtores útiles.

En esta discusión a veces requeriremos que las categorías tengan conjuntos hom pequeños. Aunque este suele ser el caso, en general es posible obtener el mismo resultado sustituyendo **Set** por una categoría de conjuntos más grande.

DEFINICIÓN 5.9. Si $\mathcal{C}$ es una categoría con conjuntos hom pequeños, definimos el *bifuntor hom* $\hom_{\mathcal{C}} : \mathcal{C}^{\mathrm{op}} \times \mathcal{C}$ sobre objetos (a, b) como $\hom_{\mathcal{C}}(a, b)$, y sobre morfismos $(f, g) : (a, b) \to (a', b')$ como $\hom_{\mathcal{C}}(f, g)(h) := g \circ h \circ f$. Además, dados dos funtores $S : \mathcal{A} \to \mathcal{C}$ y $T : \mathcal{B} \to \mathcal{C}$, definimos el bifuntor $\hom_{\mathcal{C}}(S-, T-) : \mathcal{A}^{\mathrm{op}} \times \mathcal{B} \to \mathcal{C}$ como $\hom_{\mathcal{C}} \circ (S \times T)$.

DEFINICIÓN 5.10. Dadas dos categorías $\mathcal{C}$ y $\mathcal{D}$, definimos el *funtor de evaluación* $E : \mathcal{D}^{\mathcal{C}} \times \mathcal{C} \to \mathcal{D}$ sobre objetos como $E(T, c) := Tc$ y sobre morfismos $(\tau, f) : (T, c) \to (U, c')$ como $E(\tau, f) := \tau f := \tau_{c'} \circ Tf = Uf \circ \tau_c$.

DEFINICIÓN 5.11. Si $\mathcal{C}$ es una categoría con conjuntos hom pequeños, llamamos *funtor de Yoneda* al funtor $Y : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}^{\mathcal{C}}$ que lleva objetos c a funtores $\hom(c, -)$ y morfismos $f : a \to b$ en $\mathcal{C}$ a transformaciones naturales $\hom(f, -) : \hom(b, -) \to \hom(a, -)$ dadas por $\hom(f, -)_c(g) := g \circ f$.

$$\begin{tikzcd}
\hom_{\mathcal{C}}(c,c) \arrow[r, "\tau_c"] \arrow[d, "\hom_{\mathcal{C}}(c,f)"'] & Tc \arrow[d, "Tf"] \\
\hom_{\mathcal{C}}(c,x) \arrow[r, "\tau_x"] & Tx
\end{tikzcd}
\qquad
\begin{tikzcd}
1_c \arrow[r, squiggly] \arrow[d, squiggly] & \tau_c 1_c \arrow[d, squiggly] \\
f \arrow[r, squiggly] & \tau_x f = (Tf)(\tau_c 1_c)
\end{tikzcd}$$

FIGURA 3. Representación del lema de Yoneda.

LEMA 5.12 (Yoneda). *Sea $\mathcal{C}$ una categoría con conjuntos hom pequeños. Para cada funtor $T : \mathcal{C} \to \mathbf{Set}$ y objeto c de $\mathcal{C}$, la función*

$$\gamma_{T,c} : \hom_{\mathbf{Set}^{\mathcal{C}}}(\hom_{\mathcal{C}}(c,-), T) \to Tc$$

dada por $\gamma_{T,c}(\tau) \coloneqq \tau_c 1_c$ es una biyección natural entre el funtor $\hom(Y-,-)$ (con el orden de las entradas cambiado) y el funtor de evaluación $\mathbf{Set}^{\mathcal{C}} \times \mathcal{C} \to \mathbf{Set}$.

DEMOSTRACIÓN. Para la biyección basta observar la figura 3. Para la naturalidad, fijado un morfismo $(\varphi, f) : (S,a) \to (T,b)$ de $\mathbf{Set}^{\mathcal{C}} \times \mathcal{C}$, debemos comprobar que el siguiente diagrama conmuta.

$$\begin{tikzcd}
\hom(\hom(a,-),S) \arrow[r, "\gamma_{S,a}"] \arrow[d, "\hom(\hom(f,-),\varphi)"'] & Sa \arrow[d, "\varphi f"] \\
\hom(\hom(b,-),T) \arrow[r, "\gamma_{T,b}"] & Tb
\end{tikzcd}$$

Sea entonces $\tau : \hom(a,-) \to S$ una transformación natural,

$$\begin{aligned}
(\varphi f)(\gamma_{S,a}(\tau)) = (\varphi f)(\tau_a 1_a) = \varphi_b((Sf)(\tau_a 1_a)) = \varphi_b(\tau_b \hom(a,f)(1_a)) = \varphi_b \tau_b f = \\
= \gamma_{T,b}(\varphi \cdot \tau \cdot \hom(f,-)) = \gamma_{T,b}(\hom(\hom(f,-),\varphi)(\tau)).
\end{aligned}$$

□

Este lema permite demostrar la siguiente relación entre isomorfismos naturales y flechas universales.

PROPOSICIÓN 5.13. *Dado un funtor $U : \mathcal{C} \to \mathcal{B}$ y un objeto b de $\mathcal{B}$, un par $(c, u : b \to Uc)$ es una flecha universal de b a U si y sólo si, para cada objeto x en $\mathcal{C}$, la función $\tau_x : \hom_{\mathcal{C}}(c,x) \to \hom_{\mathcal{B}}(b,Ux)$ dada por $\tau_x(f) \coloneqq Uf \circ u$ es biyectiva, en cuyo caso τ es un isomorfismo natural. Además, dados objetos b de $\mathcal{B}$ y c de $\mathcal{C}$, todo isomorfismo natural*

$$\hom_{\mathcal{C}}(c,-) \cong \hom_{\mathcal{B}}(b,U-)$$

es de esta forma para un único morfismo $u : b \to Uc$ para el que (c,u) es una flecha universal.

DEMOSTRACIÓN. La definición de flecha universal equivale a esta biyección, que es natural ya que, para cada morfismo $g : x \to y$ en $\mathcal{C}$ y $f : c \to x$, $\hom(b,Ug)(\tau_x(f)) = Ug \circ Uf \circ u = U(g \circ f) \circ u = U(\hom(c,g)(f)) \circ u = \tau_y(\hom(c,g)(f))$.

Para la segunda parte, si $\tau : \hom_{\mathcal{C}}(c,-) \to \hom_{\mathcal{B}}(b,U-)$ es un isomorfismo natural, por el lema de Yoneda y la biyectividad de τ_x, todo morfismo $b \to Ux$ se expresa de forma única como $\tau_x f = \hom(b,Uf)(\tau_c 1_c) = Uf \circ \tau_c 1_c$ para cierto $f : c \to x$, lo que significa precisamente que $\tau_c 1_c$ es universal de b a U. □

3. Adjunciones

Sea $U : \mathcal{C} \to \mathcal{B}$ un funtor entre categorías con conjuntos hom pequeños y tal que todo objeto b en $\mathcal{B}$ admite una flecha universal (Fb, u_b) de b a U. Para cada morfismo $f : b \to b'$ en $\mathcal{B}$, siguiendo la figura 1 existe un único morfismo $Ff : Fb \to Fb'$ en $\mathcal{C}$ tal que $UFf \circ u_b = u_{b'} \circ f$, y además claramente $F1_b = 1_{Cb}$ y $Fg \circ Ff = F(g \circ f)$, de modo que $F : \mathcal{B} \to \mathcal{C}$ así construido es un funtor y $u : 1_{\mathcal{B}} \to U \circ F$ es una transformación natural.

El que la imagen de u esté formada por flechas universales permite obtener una especie de transformación inversa. Dados un objeto b en $\mathcal{B}$ y un objeto c en $\mathcal{C}$, para cada morfismo $f : b \to Uc$ existe un único $\hat{f} : Fb \to c$ tal que $f = U\hat{f} \circ u_b$, de modo que $\psi_{b,c} : \hom_{\mathcal{C}}(Fb, c) \to \hom_{\mathcal{B}}(b, Uc)$ dada por $\psi_{b,c}(g) \coloneqq Ug \circ u_b$ es una biyección, y como U es un funtor y u es natural, ψ es un isomorfismo natural entre los funtores $\hom_{\mathcal{C}} \circ (F \times 1), \hom_{\mathcal{B}} \circ (1 \times U) : \mathcal{B}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$. Entonces $u_b \equiv \psi_{b,Fb}(1)$ y, del mismo modo, podemos definir la transformación natural $e : F \circ U \to 1_{\mathcal{C}}$ como $e_c \coloneqq \psi^{-1}_{Uc,c}(1)$, de modo que para cada objeto c, (Uc, e_c) es una flecha universal de F a c.

La relación entre las transformaciones naturales u y e es más estrecha que esto. Para un objeto c en $\mathcal{C}$, $1_{Uc} = \psi(e_c) = Ue_c \circ u_{Uc}$, y para un objeto b en $\mathcal{B}$,

$$\psi(e_{Fb} \circ Fu_b) = Ue_{Fb} \circ UFu_b \circ u_b = Ue_{Fb} \circ u_{UFb} \circ u_b = 1_{UFb} \circ u_b = u_b,$$

y por tanto $1_{Fb} = \psi^{-1}(u_b) = e_{Fb} \circ Fu_b$.

Estas dos identidades se pueden expresar más elegantemente con la notación adecuada. Si $\tau : R \to S$ es una transformación natural entre dos funtores $R, S : \mathcal{B} \to \mathcal{C}$ y $T : \mathcal{C} \to \mathcal{D}$ es otro funtor, podemos definir la transformación natural $T\tau : T \circ R \to T \circ S$ como $(T\tau)_b \coloneqq T(\tau_b)$ para cada objeto b en $\mathcal{B}$. Por otro lado, si $U : \mathcal{A} \to \mathcal{B}$ es otro funtor, podemos definir la transformación natural $\tau U : R \circ U \to G \circ U$ como $(\tau U)_a \coloneqq \tau_{Ua}$ para cada objeto a en $\mathcal{A}$.

Con esto, podemos caracterizar la situación anterior como sigue.

Definición 5.14. Una *adjunción* entre dos categorías $\mathcal{B}$ y $\mathcal{C}$ es una tupla $(F, G, \eta, \varepsilon)$ formada por dos funtores $F : \mathcal{B} \to \mathcal{C}$ y $G : \mathcal{C} \to \mathcal{B}$ y dos transformaciones naturales $\eta : 1 \to GF$ y $\varepsilon : FG \to 1$, llamadas respectivamente *unidad* y *co-unidad*, tales que $G\varepsilon \cdot \eta G = 1_G$ y $\varepsilon F \cdot F\eta = 1_F$.

Cabe preguntarse si todas las adjunciones se pueden construir como en el razonamiento anterior. La respuesta es que sí, como vemos en el siguiente teorema.

Teorema 5.15. *Una adjunción $(F, G, \eta, \varepsilon)$ entre $\mathcal{B}$ y $\mathcal{C}$ viene determinada por cualquiera de las siguientes listas de elementos:*

1. *Funtores F y G y un isomorfismo natural $\psi : \hom(F-, -) \to \hom(-, G-)$. Entonces se definen $\eta_b \coloneqq \psi_{b,Fb}(1)$ y $\varepsilon_c \coloneqq \psi^{-1}_{Uc,c}(1)$.*
2. *El funtor G y, para cada objeto b en $\mathcal{B}$, una flecha universal (c_b, η_b) de b a G. Entonces F se define sobre cada objeto b como c_b y sobre cada morfismo $f : b \to b'$ como el único Ff tal que $GFf \circ \eta_b = \eta_{b'} \circ f$, y el isomorfismo $\psi_{b,c}$ del apartado 1 se define como $g \rightsquigarrow Gg \circ \eta_b$.*
3. *El funtor F y, para cada objeto c en $\mathcal{C}$, una flecha universal (b_c, ε_c) de F a b. Entonces G se define sobre cada objeto c como b_c y sobre cada morfismo $g : c \to c'$ como el único Gg tal que $\varepsilon_c \circ FGg = g \circ \varepsilon_{c'}$, y el isomorfismo $\psi_{b,c}$ se define como el inverso de $f \rightsquigarrow \varepsilon_c \circ Ff$.*

En particular, cada (Fb, η_b) es una flecha universal de b a G y cada (Gc, ε_c) es una flecha universal de F a c.

Demostración. Para (1), si b es un objeto de $\mathcal{B}$ y c uno de $\mathcal{C}$, definimos $\psi : \hom(Fb, c) \to \hom(b, Gc)$ como $\psi(g) \coloneqq Gg \circ \eta_b$ y $\theta : \hom(b, Gc) \to \hom(Fb, c)$ como $\theta(f) \coloneqq \varepsilon_c \circ Ff$. Entonces, como η es natural,

$$\psi(\theta(f)) = G\varepsilon_c \circ GFf \circ \eta_b = G\varepsilon_c \circ \eta_{Gc} \circ f = f,$$

por lo que $\psi \circ \theta = 1$, y análogamente $\theta \circ \psi = 1$, por lo que ψ es un isomorfismo, claramente natural respecto a b y c, y se tiene $\psi(1) = \eta_b$ y $\theta(1) = \varepsilon_c$. Para el recíproco, definiendo η_b y ε_c a partir de ψ como en el enunciado, para todo morfismo $f : Fb \to c$ podemos seguir flechas como sigue.

$$\begin{tikzcd}
\hom(Fb, Fb) \arrow[r, "\psi_{b,Fb}"] \arrow[d, "\hom(Fb,f)"'] & \hom(b, GFb) \arrow[d, "\hom(b,Gf)"] \\
\hom(Fb, c) \arrow[r, "\psi_{b,c}"] & \hom(b, Gc)
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[r, rightsquigarrow] \arrow[d, rightsquigarrow] & \eta_b \arrow[d, rightsquigarrow] \\
f \arrow[r, rightsquigarrow] & \psi_{b,c}(f) = Gf \circ \eta_b
\end{tikzcd}$$

Esto permite probar las dos identidades en la definición de adjunción como en el texto al principio de la sección.

Para (2), (Fb, η_b) es una flecha universal, pues para cada objeto x en $\mathcal{C}$, $\psi_{b,x}(g) = Gg \circ \eta_b$ es una biyección $\hom(Fb, x) \to \hom(b, Ux)$ natural respecto a x (5.13). Recíprocamente, si sólo tenemos G y las flechas universales (c_b, η_b), F definido de esta forma es un funtor que hace a η natural, y $\psi_{b,c}$ es un isomorfismo por (5.13) y es claramente natural.

Finalmente, (3) es dual a (2). □

Ejemplo 5.16. Este teorema permite definir una gran variedad de adjunciones.

1. Si $\mathcal{C}$ es un constructo en el que todos los conjuntos admiten un objeto libre, tenemos una adjunción $(F, U, \eta, \varepsilon)$, donde $U : \mathcal{C} \to \mathbf{Set}$ es el funtor olvidadizo, $F : \mathbf{Set} \to \mathcal{C}$ es el funtor libre, $\eta_X : X \hookrightarrow UFX$ es la inclusión de la base en el conjunto subyacente del objeto y $\varepsilon_c : FUc \twoheadrightarrow c$ es el epimorfismo que aparece al describir un objeto como cociente de un cierto objeto libre, en concreto del que tiene los propios elementos de c como generadores. Por ejemplo, en el caso de R-**Mod**, ε_M lleva sumas formales $\sum_{i=1}^k a_i m_i$, con cada $a_i \in R$ y cada $m_i \in M$, a su evaluación en el módulo M.
2. Entre **Dom** y **Field** hay una adjunción $(Q, U, \eta, \varepsilon)$ formada por la creación de cuerpos de fracciones $Q : \mathbf{Dom} \to \mathbf{Field}$, la inclusión $U : \mathbf{Field} \hookrightarrow \mathbf{Dom}$ de una subcategoría, la inclusión canónica $\eta_D : D \to UQD$ y la identidad $\varepsilon_K : QUK \to K$ (recordemos que el cuerpo de fracciones de un cuerpo es el propio cuerpo).
3. Si $\mathcal{C}$ es una categoría que tiene colímites con diagrama $\mathcal{S}$, entre $\mathcal{C}^{\mathcal{S}}$ y $\mathcal{C}$ hay una adjunción $(\varinjlim, \Delta, \eta, \varepsilon)$, donde $\varinjlim$ lleva cada diagrama a su objeto colímite, $\Delta : \mathcal{C} \to \mathcal{C}^{\mathcal{S}}$ es el funtor diagonal y $\eta_D : D \to \Delta \varinjlim D$ es el sumidero colímite. Respecto a $\varepsilon_c : \varinjlim \Delta c \to c$, el colímite de Δc es ${}^I c$, siendo I el número (cardinal) de componentes conexas de $\mathcal{S}$, y ε_c es el morfismo que «une todas las copias de c».
4. Análogamente, si $\mathcal{C}$ tiene límites con diagrama $\mathcal{S}$, tenemos una adjunción $(\Delta, \varprojlim, \eta, \varepsilon)$ donde $\varprojlim$ lleva cada diagrama a su límite, ε_D es el límite como fuente y η_c tiene un codominio de la forma c^I para cierto conjunto I y actúa como «morfismo diagonal».

4. Adjuntos laterales

Definición 5.17. Dada una adjunción $(F, G, \eta, \varepsilon)$, decimos que F es un *adjunto izquierdo* de G y que G es un *adjunto derecho* de F.

Ejemplo 5.18. El funtor olvidadizo $U : \mathbf{Top} \to \mathbf{Set}$ tiene una adjunción por cada lado:[10, pág. 4.1]

1. Por la izquierda tiene el funtor $D : \mathbf{Set} \to \mathbf{Top}$ que asocia a cada conjunto su topología discreta. Entonces $\eta_X : X \to UDX$ es la identidad en X y $\varepsilon_T : DUT \to T$ es la identidad hacia T desde el mismo conjunto pero con la topología discreta.
2. Por la derecha tiene el funtor $N : \mathbf{Set} \to \mathbf{Top}$ que asocia a cada conjunto su topología indiscreta. Aquí $\varepsilon_X : UNX \to X$ es la identidad en X y $\eta_T : T \to NUT$ es la identidad desde T hacia el mismo conjunto pero con la topología indiscreta.

Proposición 5.19. *El adjunto por la izquierda o por la derecha de un funtor es único salvo isomorfismo natural.*

Demostración. El adjunto por la izquierda de un funtor $G : \mathcal{C} \to \mathcal{B}$ viene dado por una flecha universal (Fb, η_b) de b a G para cada objeto b en $\mathcal{B}$ (5.15), pero estas flechas son únicas salvo isomorfismo. Así, si $(Fb, \eta_b)_b$ y $(F'b, \eta'_b)_b$ son familias de estas flechas, para cada b existe un único isomorfismo $h_b : Fb \to F'b$ tal que $\eta'_b = Gh_b \circ \eta_b$. Para ver que este es natural, para $f : b \to b'$, por la construcción de Ff en 5.15,

$$G(h_{b'} \circ Ff \circ h_b^{-1}) \circ \eta'_b = Gh_{b'} \circ GFf \circ \eta_b = Gh_{b'} \circ GFf \circ \eta_b = Gh_{b'} \circ \eta_{b'} \circ f = \eta'_{b'} \circ f,$$

y por la unicidad en dicha construcción, $F'f = h_{b'} \circ Ff \circ h_b^{-1}$.

El concepto de adjunto por la derecha es el dual. □

5. Transformaciones de adjunciones

Definición 5.20. Una *transformación de adjunciones* de la adjunción $(F, G, \eta, \varepsilon)$ de $\mathcal{B}$ a $\mathcal{C}$ a la adjunción $(F', G', \eta', \varepsilon')$ de $\mathcal{B}'$ a $\mathcal{C}'$ es un par de funtores $B : \mathcal{B} \to \mathcal{B}'$ y $C : \mathcal{C} \to \mathcal{C}'$ tales que $C \circ F = F' \circ B$, $B \circ G = G' \circ C$, $B\eta = \eta' B$ y $\varepsilon' C = C\varepsilon$.

Proposición 5.21. *Dadas dos adjunciones $(F, G, \eta, \varepsilon)$ de $\mathcal{B}$ a $\mathcal{C}$ y $(F', G', \eta', \varepsilon')$ de $\mathcal{B}'$ a $\mathcal{C}'$, y dados dos funtores $B : \mathcal{B} \to \mathcal{B}'$ y $C : \mathcal{C} \to \mathcal{C}'$ con $C \circ F = F' \circ B$ y $B \circ G = G' \circ C$, son equivalentes:*

1. *(B, C) es una transformación de la primera adjunción a la segunda.*
2. *$B\eta = \eta' B$.*
3. *$\varepsilon' C = C\varepsilon$.*
4. *Si $\psi : \hom(F-, -) \to \hom(-, G-)$ es el isomorfismo natural asociado a la primera adjunción por el teorema 5.15 y $\psi' : \hom(F'-, -) \to \hom(-, G'-)$ es el correspondiente a la segunda adjunción, para cada objeto b de $\mathcal{B}$ y c de $\mathcal{C}$, $B|_{hom(b,Gc)} \circ \psi_{b,c} = \psi'_{Bb,Cc} \circ C|_{\hom(Fb,c)}$.*

Demostración. Por definición (1) $\implies$ (2), (3). Dado (2), por la definición de ψ a partir de η, para $f : Fb \to c$,

$$B(\psi_{b,c} f) = B(Gf \circ \eta_b) = BGf \circ B\eta_b = G'Cf \circ \eta'_{Bb} = \psi'_{Bb,Cc}(Cf),$$

lo que nos da (4). Análogamente (3) $\implies$ (4). Para el recíproco, usando las fórmulas de η y η' en función de ψ y ψ' y aplicando (4) a 1_{Fb},

$$B\eta_b = B(\psi(1_{Fb})) = \psi'(C1_{Fb}) = \psi'(1_{CFb}) = \psi'(1_{F'Bb}) = \eta'_{Bb},$$

con lo que (4) $\implies$ (2) y, análogamente, (4) $\implies$ (3), y estas condiciones equivalen a (1). □

Esto nos da una categoría cuyos objetos son adjunciones entre categorías de un cierto conjunto universal y cuyos morfismos son transformaciones de adjunciones, que se componen de la forma evidente.

Capítulo 6

Mónadas

Al igual que cuando el dominio y el codominio de un morfismo coinciden hablamos de un *endomorfismo*, cuando esto le ocurre a un funtor hablamos de un *endofuntor*. Los endofuntores son particularmente relevantes en tanto que se puede estudiar, por ejemplo, la relación de un objeto o morfismo con su imagen, o lo que ocurre al aplicar el endofuntor varias veces.

Por ejemplo, tomemos el endofuntor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$. Dado un conjunto X, entre X y $\mathcal{P}X$ existe un monomorfismo $\eta_X : X \to \mathcal{P}X$ dado por $\eta_X(x) \coloneqq \{x\}$, que es natural. Por otro lado, aunque podemos aplicar este funtor a X varias veces, siempre podemos volver de $\mathcal{P}^n X$ a $\mathcal{P}X$ aplicando sucesivamente la unión, que podemos ver como una función $\mu_X : \mathcal{P}\mathcal{P}X \to \mathcal{P}X$ dada por $\mu_X(\mathcal{A}) \coloneqq \bigcup \mathcal{A}$. Esto también define una transformación natural que, además, «da igual» en qué orden se aplique, en el sentido de que, si $S \in \mathcal{P}^3 X$, aplicar primero $\mu_{\mathcal{P}X}$ a S y luego μ_X al resultado es lo mismo que aplicar μ_X a cada elemento de S y a continuación aplicar μ_X al resultado. Además, intuitivamente μ se puede ver como una inversa por un lado de η, en tanto que $\mu_X(\eta_{\mathcal{P}X}(S)) \equiv S$.

Muchos endofuntores comunes admiten transformaciones naturales con estas propiedades, y cuando esto ocurre hablamos de mónadas. En este capítulo estudiamos las mónadas y sus principales propiedades, basándonos principalmente en [4, pág. VI].

Definición 6.1. Una *mónada* en una categoría $\mathcal{C}$ es una tupla (T, η, μ) formada por un funtor $T : \mathcal{C} \to \mathcal{C}$ y dos transformaciones naturales $\eta : 1_{\mathcal{C}} \to T$ y $\mu : T^2 \to T$ que cumplen las siguientes condiciones, ilustradas en la figura 1.

1. $\mu \cdot T\mu = \mu \cdot \mu T$.
2. $\mu \cdot \eta T = \mu \cdot T\eta = 1$.

Ejemplo 6.2.

1. $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ es una mónada, descrita en la introducción del capítulo.
2. Toda categoría admite una *mónada identidad*, formada por el endofuntor identidad y dos transformaciones naturales identidad.
3. Sea $(^*) : \mathbf{Set} \to \mathbf{Set}$ el endofuntor que asocia a cada objeto X el conjunto subyacente de su monoide libre, dado por $\bigcup_{n \in \mathbb{N}} X^n$, y que lleva cada función $f : X \to Y$ a la función $(x_1, \dots, x_k) \rightsquigarrow (fx_1, \dots, fx_k)$. Este endofuntor es una mónada con las transformaciones naturales $\eta : 1 \to (^*)$ que lleva cada elemento de un conjunto a la lista de un elemento ($\eta_X(x) = (x)$) y $\mu : (^*)^2 \to (^*)$ que concatena una lista de listas en una sola lista.

$$\begin{tikzcd} T^3 \arrow[r, "T\mu"] \arrow[d, "\mu T"'] & T^2 \arrow[d, "\mu"] \\ T^2 \arrow[r, "\mu"'] & T \end{tikzcd}$$

(A) Conmutatividad de la unión

$$\begin{tikzcd} 1 \circ T \arrow[r, "\eta T"] \arrow[dr, "1"'] & T^2 \arrow[d, "\mu"] & T \circ 1 \arrow[l, "T\eta"'] \arrow[dl, "1"] \\ & T & \end{tikzcd}$$

(B) Relación entre la unidad y la unión

Figura 1. Condiciones de coherencia de las mónadas.

4. Esto se puede generalizar a todas las variedades algebraicas. El álgebra libre sobre un conjunto X es un conjunto cociente de árboles formados por operadores y elementos de X (1.23), o de expresiones formales que involucran a dichos operadores y elementos, y las funciones entre conjuntos se pueden llevar a morfismos de álgebras libres que operan elemento a elemento sobre las expresiones. Entonces la unidad η_X llevaría cada elemento de X a (la clase de equivalencia de) la expresión formada sólo por dicho elemento, y la unión μ_X tomaría árboles cuyas hojas son (clases de equivalencia de) otros árboles y sustituiría las hojas por los subárboles que representan. Es fácil ver que estas transformaciones son naturales y forman una mónada.
5. Sean $\mathcal{C}$ una categoría con coproductos finitos y d un objeto de $\mathcal{C}$. Sea $E : \mathcal{C} \to \mathcal{C}$ un endofuntor que a cada objeto c le asocia $c \oplus d$ y a cada morfismo $f : b \to c$ el morfismo «suma» $Ef = f \oplus 1_d : b \oplus d \to c \oplus d$. Si $\eta_c : c \rightarrowtail c \oplus d$ es la inclusión canónica y $\mu_c : c \oplus d \oplus d \twoheadrightarrow c \oplus d$ es la función que «identifica» las dos copias de d en el sentido evidente, entonces (E, η, μ) es una mónada.
6. Sean s un conjunto y $S \coloneqq \hom(s, - \times s)$ un endofuntor en **Set** que lleva cada conjunto X al conjunto de funciones $s \to X \times s$ y cada función f a $\hom(s, f \times 1_s)$. Entonces S es una mónada con las transformaciones naturales $\eta : 1 \to S$ y $\mu : S^2 \to S$ dadas por la formación de pares $\eta_X(x)(y) \coloneqq (x, y)$ y la «doble evaluación» $\mu_X(f)(y) \coloneqq (p_1 f y)(p_2 f y)$, donde p_1 y p_2 son las proyecciones canónicas de $\hom(X, s) \times s$.
7. En un conjunto parcialmente ordenado $(S, \leq)$ visto como categoría, un endofuntor es una función $f : S \to S$ monótona, es decir, tal que $x \leq y \implies fx \leq fy$. Las transformaciones naturales asociadas a f en una mónada están unívocamente determinadas y existen si y sólo si, para todo $x \in S$, $x \leq fx$ y $f^2 x \leq fx$, pues al ser una categoría fina los diagramas correspondientes siempre conmutan. Estas ecuaciones implican que f es idempotente. Así, las mónadas en conjuntos parcialmente ordenados son *operaciones de clausura*, funciones monótonas e idempotentes con $x \leq fx$ para todo x. Este término se usa especialmente cuando el orden es la inclusión de conjuntos.

El ejemplo 4 de la lista anterior se puede generalizar aún más, de las variedades algebraicas a todas las categorías concretas que admitan un funtor libre.

PROPOSICIÓN 6.3. *Si $(F, G, \eta, \varepsilon)$ es una adjunción entre las categorías $\mathcal{B}$ y $\mathcal{C}$, entonces $(G \circ F, \eta, G\varepsilon F)$ es una mónada.*

DEMOSTRACIÓN. Al componer horizontalmente ε consigo mismo (4.13) obtenemos que $\varepsilon \circ \varepsilon = \varepsilon \cdot \varepsilon FG = \varepsilon \cdot FG\varepsilon$, y componiendo con G por la izquierda y con F por la derecha obtenemos $G\varepsilon F \cdot G\varepsilon FGF = G\varepsilon F \cdot GFG\varepsilon F$, que es la primera condición de coherencia. Para la segunda basta componer con F por la derecha en la identidad $G\varepsilon \cdot \eta G = 1$ y con G por la izquierda en $\varepsilon F \cdot F\eta = 1$. □

DEFINICIÓN 6.4. Dada una categoría $\mathcal{C}$, la *categoría de mónadas* de $\mathcal{C}$ es una categoría $\mathrm{Mnd}(\mathcal{C})$ cuyos objetos son las mónadas sobre $\mathcal{C}$ y cuyos morfismos $(S, \eta, \mu) \to (T, \eta', \mu')$ son las transformaciones naturales $\tau : S \to T$ tales que los diagramas en la figura 2 conmutan, con la composición vertical y las identidades evidentes.

El concepto dual al de mónada es el de *comónada*, aunque la utilidad de las comónadas es más limitada.

1. Categorías de Eilenberg-Moore

Hemos visto que toda adjunción genera una mónada, por lo que cabe preguntarse si toda mónada es generada de esta manera por una adjunción. La respuesta es que sí, y de hecho en general cada mónada se puede describir mediante dos adjunciones asociadas a dos categorías distintas, la categoría de Eilenberg-Moore y la categoría de Kleisli. Empezamos viendo la primera, que generaliza el concepto de variedad algebraica.

$$\begin{tikzcd} 1 \arrow[r, "\eta"] \arrow[dr, "\eta'"'] & S \arrow[d, "\tau"] \\ & T \end{tikzcd}$$

(A) Conmutatividad de la unidad

$$\begin{tikzcd} S^2 \arrow[r, "\mu"] \arrow[d, "\tau \circ \tau"'] & S \arrow[d, "\tau"] \\ T^2 \arrow[r, "\mu'"'] & T \end{tikzcd}$$

(B) Conmutatividad de la unión

FIGURA 2. Morfismos de mónadas.

$$\begin{tikzcd} T^2c \arrow[r, "Te"] \arrow[d, "\mu_c"'] & Tc \arrow[d, "e"] \\ Tc \arrow[r, "e"'] & c \end{tikzcd}$$

(A) Propiedad asociativa

$$\begin{tikzcd} c \arrow[r, "\eta_c"] \arrow[dr, "1"'] & Tc \arrow[d, "e"] \\ & c \end{tikzcd}$$

(B) Propiedad unitaria

FIGURA 3. Propiedades de las T-álgebras.

DEFINICIÓN 6.5. Sea $T = (T, \eta, \mu)$ una mónada en una categoría $\mathcal{C}$.

1. Una *T-álgebra* es un morfismo $e : Tc \twoheadrightarrow c$ para cierto objeto c en $\mathcal{C}$ tal que los diagramas en la figura 3 conmutan. Llamamos *mapa de estructura* de la T-álgebra a e y *objeto subyacente* a c.
2. Un morfismo entre dos T-álgebras $e : Tc \to c$ y $e' : Tc' \to c'$ es un morfismo $f : c \to c'$ en $\mathcal{C}$ tal que $f \circ e = e' \circ Tf$.
3. La *categoría de Eilenberg-Moore* asociada a T, escrita $\mathcal{C}^{(T,\eta,\mu)}$ o simplemente $\mathcal{C}^T$, es la que tiene como objetos las T-álgebras y como morfismos los morfismos de T-álgebras, con la composición y las identidades de $\mathcal{C}$.

TEOREMA 6.6. *Sea (T, η, μ) una mónada en $\mathcal{C}$, si $F : \mathcal{C} \to \mathcal{C}^T$ actúa sobre los objetos como μ y sobre los morfismos como T, $G : \mathcal{C}^T \to \mathcal{C}$ «olvida» el mapa de estructura, asociando a cada T-álgebra su objeto subyacente y a cada morfismo el mismo, y $\varepsilon : FG \to 1$ lleva cada T-álgebra a su mapa de estructura, entonces $(F, G, \eta, \varepsilon)$ es una adjunción cuya mónada asociada es (T, η, μ).*

DEMOSTRACIÓN. Para cada objeto c en $\mathcal{C}$, 1_{Tc} es la identidad de la T-álgebra μ_c, y para cada morfismo $f : c \to c'$, $Tf : \mu_c \to \mu_{c'}$ es un morfismo de T-álgebras por la naturalidad de μ, luego F es un funtor. Por otro lado, G claramente es un funtor y ε, que viene dado por $\varepsilon_e \coloneqq e : \mu_c \to e$ para cada $e : Tc \to c$, es una transformación natural. Además, dada una T-álgebra $e : Tc \to c$, $G\varepsilon_e \circ \eta_{Ge} = e \circ \eta_c = 1$, y dado un objeto c de $\mathcal{C}$, $\varepsilon_{Fc} \circ F\eta_c = \mu_c \circ T\eta_c = 1$, luego $(F, G, \eta, \varepsilon)$ es una adjunción. Finalmente, es obvio que $G \circ F = T$ siguiendo su actuación sobre morfismos, y para un objeto c, $G\varepsilon_{Fc} = G\varepsilon_{\eta_c} = \eta_c$ y por tanto $G\varepsilon F = \eta$. □

La adjunción definida en este teorema es final en el sentido siguiente.

TEOREMA 6.7. *Dada una adjunción $(F, G, \eta, \varepsilon)$ de $\mathcal{C}$ a $\mathcal{D}$, si T es la mónada generada por la adjunción y $(F', G', \eta, \varepsilon')$ es la adjunción de $\mathcal{C}$ a $\mathcal{C}^T$ definida en el teorema anterior, existe un único funtor $K : \mathcal{D} \to \mathcal{C}^T$ tal que $(1_{\mathcal{C}}, K)$ es una transformación de adjunciones de $(F, G, \eta, \varepsilon)$ a $(F', G', \eta, \varepsilon')$.*

DEMOSTRACIÓN. Ya tenemos $1\eta = \eta 1$, y queda ver las condiciones $G = G' \circ K$ y $F' = K \circ F$. Sea $(T, \eta, \mu) = (GF, \eta, G\varepsilon F)$ la mónada, para cada objeto d de $\mathcal{D}$, podemos ver $G\varepsilon_d : GFGd \to Gd$ como una T-álgebra en el objeto Gd, pues la propiedad asociativa $G\varepsilon_d \circ GFG\varepsilon_d = \varepsilon_d \circ G\varepsilon_{FGd}$ se deduce de la identidad en la composición horizontal y la unitaria $G\varepsilon_d \circ \eta_{Gd} = 1$ es una identidad de las adjunciones. Podemos entonces definir K sobre objetos como $Kd = G\varepsilon_d$, y sobre morfismos como $Kf = Gf$, pues la naturalidad de ε asegura que Gf es un morfismo de T-álgebras. Para cada objeto c de $\mathcal{C}$, $KFc = G\varepsilon_{Fc} = \mu_c = F'c$, y para cada morfismo f, $KFf = GFf = Tf = F'c$. Del mismo modo, para cada objeto d de $\mathcal{D}$, $G'Kd = G'G\varepsilon_d = Gd$, y para cada morfismo f, $G'Kf = G'Gf = Gf$.

Queda ver que K es único. Si d es un objeto de $\mathcal{D}$, Kd debe ser una T-álgebra con objeto subyacente Gd, pues $G'Kd = Gd$. Por otro lado, si f es un morfismo, $G'Kf = Gf$ implica que $Kf = Gf$. Ahora bien, la caracterización de las transformaciones de adjunciones (5.21) aplicada a $(1_{\mathcal{C}}, K)$ nos da $K\varepsilon = \varepsilon' K$, y entonces, para un objeto d, el mapa de estructura de Kd es $Kd = \varepsilon'_{Kd} = K\varepsilon_d = G\varepsilon_d$. □

La siguiente proposición muestra que, de hecho, estas categorías son una generalización del concepto de variedad algebraica.

Proposición 6.8. *Sean (Ω, E)-**Alg** una variedad algebraica y T la mónada generada por la adjunción entre el funtor libre y el funtor olvidadizo de dicha variedad. Entonces el funtor $K : (\Omega, E)$-**Alg** $\to \mathcal{C}^T$ del teorema anterior es un isomorfismo de categorías.*

Demostración. Sean $s_1, \ldots, s_k$ las operaciones en Ω y $n_1, \ldots, n_k$ sus aridades respectivas, y sea $(F, U, \eta, \varepsilon)$ la adjunción mencionada, de modo que $T = (UF, \eta, U\varepsilon F)$.

Si $(S, (\nu_1, \ldots, \nu_k))$ es una (Ω, E)-álgebra, con cada $\mu_i : S^{n_i} \to S$, los elementos de UFS son las (clases de equivalencia de) expresiones formales construidas a partir de los operadores $s_1, \ldots, s_k$ y los elementos de c, por lo que podemos definir $K(S, (\nu_i)_i)$ como la «función de evaluación» $e : UFS \to S$ que lleva cada elemento de c a sí mismo y cada expresión $s_i(x_1, \ldots, x_{n_i})$ a $\nu_i(e(x_1), \ldots, e(x_{n_i}))$.

En este contexto, la propiedad asociativa de las T-álgebras nos dice que, dada una expresión formal sobre expresiones formales sobre elementos de S, da lo mismo evaluar las expresiones interiores y a continuación la global que considerar las expresiones interiores como partes de la expresión global y evaluar todo a la vez. Por su parte, la propiedad unitaria nos dice que evaluar una expresión formada sólo por un elemento «devuelve» dicho elemento. Claramente ambas propiedades se cumplen, y de hecho juntas implican que cualquier función $e : UFS \to S$ que las cumpla viene dada por su actuación sobre expresiones «de altura 1», es decir, de la forma $s_i(x_1, \ldots, x_{n_i})$ con los $x_j \in S$, lo que nos da una serie de operaciones $\nu_i : S^{n_i} \to S$ que, además, cumplen las igualdades en E al estar e definida sobre clases de equivalencia, de modo que estas igualdades definen una (Ω, E)-álgebra y K es biyectiva sobre objetos.

Para los morfismos $f : (S, (\nu_i)_i) \to (S', (\nu'_i)_i)$, podemos definir $Kf : S \to S'$ como la propia f, que es un morfismo $K(S, (\nu_i)_i) \to K(S', (\nu'_i)_i)$ por la conmutatividad de f respecto a los operadores de Ω. K así definida es fiel y plena, pues la propiedad conmutativa que define los morfismos de T-álgebras es precisamente la que define los morfismos de (Ω, E)-álgebras.

Así, K es un isomorfismo, y claramente esta definición de K coincide con la del teorema anterior. □

2. Categorías de Kleisli

Hemos visto que, dada una mónada en una categoría $\mathcal{C}$, la adjunción de Eilenberg-Moore asociada es un objeto final de la categoría de las adjunciones que definen dicha mónada junto con las transformaciones de mónadas que son la identidad en $\mathcal{C}$. Vamos a ver que esta categoría de adjunciones tiene también un objeto inicial, la categoría de Kleisli, de gran importancia en teoría de la computación.

Definición 6.9. Dada una mónada (T, η, μ) en $\mathcal{C}$, llamamos *categoría de Kleisli* asociada a la mónada, escrita $\mathcal{C}_{(T,\eta,\mu)}$ o simplemente $\mathcal{C}_T$, a la categoría cuyos objetos son los de $\mathcal{C}$ y cuyos morfismos $a \to b$ son los morfismos $a \to Tb$ en $\mathcal{C}$, donde la composición de dos funtores $f : a \to b$ y $g : b \to c$ viene dada por

$$g \hat{\circ} f := \mu_c \circ Tg \circ f$$

y la identidad en un objeto c es μ_c.

Es rutinario comprobar que estas definiciones de composición e identidad definen una categoría.

TEOREMA 6.10. *Sea (T, η, μ) una mónada en $\mathcal{C}$. Si $F : \mathcal{C} \to \mathcal{C}_T$ lleva los objetos a ellos mismos y los morfismos $f : a \to b$ a $\eta_b \circ f$, $G : \mathcal{C}_T \to \mathcal{C}$ actúa sobre los objetos como T y lleva los morfismos $f : a \to b$ a $\mu_b \circ Tf$ y, para cada objeto c, $\varepsilon_c := 1_c \in \hom_{\mathcal{C}_T}(Tc, c)$, entonces $(F, G, \eta, \varepsilon)$ es una adjunción que genera (T, η, μ).*

DEMOSTRACIÓN. Obviamente F preserva identidades, y es fácil ver que también preserva composiciones, por lo que es un funtor. Del mismo modo, es fácil ver que G también preserva identidades y composiciones. Por su parte, para un objeto c, $G\varepsilon_c \circ \eta_{Gc} = \mu_c \circ T1_c \circ \eta_{Gc} = \mu_c \circ \eta_{Tc} = 1$ y $\varepsilon_{Fc} \hat{\circ} F\eta_c = \mu_c \circ T\varepsilon_{Fc} \circ F\eta_c = \mu_c \circ 1_{TFc} \circ \eta_{Tc} \circ \eta_c = \eta_c$, que es la identidad en $\mathcal{C}_T$, por lo que $(F, G, \eta, \varepsilon)$ es una adjunción.

Es fácil ver que $GF = T$ tanto para objetos como para morfismos, y finalmente, para un objeto c, $G\varepsilon_{Fc} = G1_{Fc} = \mu_c \circ 1_{Fc} = \mu_c$, por lo que la adjunción genera (T, η, μ). □

TEOREMA 6.11. *Sean $(F, G, \eta, \varepsilon)$ una adjunción de $\mathcal{C}$ a $\mathcal{D}$, $T = (T, \eta, \mu)$ la mónada en $\mathcal{C}$ definida por la adjunción y $(F', G', \eta, \varepsilon')$ la adjunción de $\mathcal{C}$ a $\mathcal{C}_T$ definida en el teorema anterior, existe un único funtor $L : \mathcal{C}_T \to \mathcal{D}$ tal que $(1, L)$ es una transformación de la segunda adjunción a la primera.*

DEMOSTRACIÓN. Debemos ver que $G' = G \circ L$ y $F = L \circ F'$. Como F' lleva los objetos a sí mismos, por la segunda identidad debe ser $Lc := Fc$ para cada objeto c, y la primera identidad claramente se cumple. Para un morfismo $f : a \to b$ en $\mathcal{C}_T$ ($f : a \to GFb$ en $\mathcal{C}$), definimos $Lf := \varepsilon_{Fb} \circ Ff$, de modo que $LF'f = \varepsilon_{Fb} \circ F\eta_b \circ Ff = Ff$ por las propiedades de la adjunción y, por el otro lado, $GLf = G\varepsilon_{Fb} \circ GFf = \mu_b \circ Tf = G'f$.

Queda ver que L es única. Claramente lo es sobre objetos. Sobre morfismos, la caracterización de las transformaciones de adjunciones (5.21) nos da $L\varepsilon' = \varepsilon L$, por lo que para cada objeto c, $L\varepsilon'_c = \varepsilon_{Lc} = \varepsilon_{Fc}$. Entonces, si $L, L' : \mathcal{C}_T \to \mathcal{D}$ son dos funtores que cumplen la condición, por igualación $L\varepsilon'_c = L'\varepsilon'_c$, pero ε'_c una flecha universal desde un funtor y por tanto es un retracto y un epimorfismo, con lo que $L = L'$. □

Apéndice A

Categorías en programación

Al nivel más básico, un programa es simplemente una serie de instrucciones a ejecutar por un ordenador. Sin embargo, estos programas suelen ser complejos y es común introducir fallos a la hora de crearlos, por lo que son necesarias herramientas para reducir el número de fallos y, en sistemas críticos, para demostrar su ausencia mediante pruebas de corrección.

Para probar la corrección de un programa, primero debemos modelarlo matemáticamente, y para ello existen varias alternativas:

Semántica axiomática: Usa reglas lógicas para describir el efecto de los comandos ejecutados sobre proposiciones lógicas, definiendo relaciones entre las proposiciones que son verdad antes y después de su ejecución. Por ejemplo, uno de los axiomas podría ser «si E es una expresión, v es una variable y P es una propiedad, si se cumple $P(E)$, tras ejecutar $v \leftarrow E$ se cumple $P(v)$».

Semántica denotativa: Los programas se interpretan como descripciones de objetos matemáticos en ciertos dominios, como funciones entre la entrada y la salida.

Semántica operacional: El programa en sí se considera como un objeto matemático, y las pruebas se construyen a partir de axiomas que definen su comportamiento.

Tradicionalmente los programas se han escrito de forma imperativa, como secuencias de comandos que modifican el estado (los valores de las variables) y se comunican con los dispositivos de entrada y salida. Para estos programas es común usar la semántica axiomática, que encaja bien al estar también basada en comandos.

Sin embargo, en general razonar sobre programas imperativos complejos es difícil, pues, por ejemplo, a la hora de razonar no siempre es válido sustituir una variable por la expresión que se le ha asignado, pues el valor de dicha expresión puede cambiar y, además, ejecutar la expresión puede modificar otras partes del estado del programa.

Para facilitar el razonamiento surge la programación funcional. En esta los procedimientos actúan como funciones matemáticas, en el sentido de que no se comunican con el exterior y el valor devuelto depende exclusivamente de los valores de los parámetros de entrada. Además, los valores de las variables no se pueden modificar una vez se ha definido la variable, permitiendo sustituir libremente las variables por su expresión de forma relativamente sencilla.

La semántica axiomática no es apropiada para la programación funcional, pero la semántica denotativa proporciona un modelo muy elegante y sencillo de usar. Este paradigma de programación no se creó sólo para facilitar demostraciones rigurosas sobre los programas, sino también para facilitar el razonamiento intuitivo y la verificación automática y de este modo reducir notablemente la cantidad de fallos que se introducen en el programa en primer lugar.

La programación funcional no sólo tiene ventajas respecto a corrección, sino que también permite reducir notablemente el tamaño de los programas tratando las funciones como valores y operando sobre ellas. Para ello se usa una variedad de técnicas heredadas de la teoría de categorías, como funtores, transformaciones naturales y mónadas.

El lector habrá notado que, si los procedimientos son funciones que no se comunican con el exterior, es imposible crear programas interactivos. Sin embargo, la idea no es definir el programa completo de forma funcional, sino reducir al mínimo las partes no funcionales. Respecto a la formalización, las partes no funcionales se pueden formalizar de otra forma más tradicional o mediante una construcción que involucra categorías de Kleisli, como veremos a continuación.

1. La categoría Type

Por lo general, los tratamientos categóricos de la programación funcional se basan en una categoría **Type** cuyos objetos son tipos de datos y cuyos morfismos son funciones computables. La estructura concreta de **Type** depende del lenguaje de programación utilizado, por lo que primero definiremos un lenguaje de programación funcional sencillo que consideramos representativo.

El lenguaje que definiremos no es especialmente potente o minimalista, aunque permite describir la mayoría de construcciones usadas en lenguajes de programación funcional sin tener que entrar en aspectos de lógica formal que, si bien son interesantes, se salen del ámbito de este trabajo.

El lenguaje que vemos no permite construcciones avanzadas como pueden ser tratar una mónada como un valor; sin embargo, podemos hablar de mónadas de forma extrínseca. Para este tipo de construcciones necesitamos un lenguaje como el descrito en [3], o incluso el descrito en [2], que además puede usarse como fundamento para la lógica formal, aunque estos lenguajes no permiten identificar los tipos con su dominio de valores de forma directa.[8]

Toda expresión e válida en lenguaje tiene un tipo T asociado, y escribimos $e : T$ para expresar esto. Además, todo tipo T tiene un dominio D_T asociado, y evaluar una expresión $e : T$ en un cierto contexto resulta en un valor de D_T. Los tipos que se pueden definir son:

- *Tipos producto.* Si $t_1, \dots, t_k$ son tipos, $(t_1, \dots, t_k)$ es un tipo con dominio $D_{t_1} \times \cdots \times D_{t_k}$, y en particular existe un *tipo unidad* $()$ con un sólo elemento. Los tipos producto existen en la mayoría de lenguaje de programación con el nombre de *estructuras.*
- *Tipos suma.* Si $t_1, \dots, t_k$ son tipos, $[t_1|\dots|t_k]$ es un tipo cuyo dominio es la unión disjunta $D_{t_1} \oplus \cdots \oplus D_{t_k}$, y en particular existe un *tipo vacío* $[]$ con dominio vacío. Los tipos coproducto son poco comunes fuera de lenguajes de programación funcionales, aunque se pueden aproximar mediante uniones en lenguajes de bajo nivel o jerarquías de clases en lenguajes orientados a objetos.
- *Tipos función.* Si a y b son tipos, $a \to b$ es un tipo cuyo dominio son las funciones $D_a \to D_b$ que son *computables*, que son precisamente las que pueden ser expresadas en este lenguaje.
- *Tipos sinónimo.* Una sentencia `type` $T = t$, donde T es un identificador y t es una expresión que representa un tipo, crea un tipo T con el mismo dominio que t, y morfismos $T^* : t \to T$ y $T_* : T \to t$ que actúan sobre los dominios como la identidad. Crucialmente, las sentencias de este tipo pueden hacer referencia unas a otras y t puede hacer referencia a T, lo que permite definir tipos recursivos cuyo dominio es similar al de un tipo suma con infinitas alternativas. Para que una alternativa esté habitada debe tener un caso base; por ejemplo, la sentencia `type` $T = T$ es válida pero crea un tipo vacío.

  Como extensión de esta sintaxis podemos sustituir T por $T_{a_1,\dots,a_k}$, que para tipos cualesquiera $\sigma_1, \dots, \sigma_k$ define $T_{\sigma_1,\dots,\sigma_k}$ como sinónimo de $t[\sigma_1/a_1]\cdots[\sigma_k/a_k]$.

Las expresiones del lenguaje son de la forma siguiente:

- En un contexto en que el identificador x está vinculado a una expresión de tipo T, x es una expresión de tipo T que representa el valor vinculado al identificador en el contexto actual.
- Si $f : S \to T$ y $e : S$, entonces fe es una expresión de tipo T que evalúa f y e y devuelve el resultado de aplicar la función dada por f al valor dado por e.
- Sean x un identificador y e una expresión que puede mencionar a x, si cuando $x : S$ entonces $e : T$, la expresión $\lambda x : S.e$ es de tipo $S \to T$ y representa una función que recibe un valor de tipo x y devuelve el valor de e cuando x está vinculado a dicho valor.
- Dadas expresiones $e_1 : T_1, \dots, e_k : T_k$, $(e_1, \dots, e_k) : (T_1, \dots, T_k)$ representa el valor que en cada posición i contiene el valor resultante de evaluar e_i, mientras que

si $e : (T_1, \dots, T_k)$ e $i \in \{1, \dots, k\}$, la expresión $e_i : T_i$ recupera el valor en la componente i-ésima.

- Dada una expresión $e : T_i$, la expresión $\langle e \rangle_i^{T_1,\dots,T_k} : [T_1|\dots|T_k]$ representa la inclusión del valor de e en la componente i-ésima de un tipo suma, mientras que si $e : [T_1, \dots, T_k]$ y cada $x_i : T_i \to \gamma$, $\texttt{match}\ e\ \{x_1;\ \dots;\ x_k\} : \gamma$ es una expresión que evalúa e y, según la componente del tipo suma en que se encuentre el valor devuelto, ejecuta una de las funciones con dicho valor y devuelve el valor devuelto por dicha función.
- Dadas expresiones $c : [(), ()]$ y $t, e : T$, la expresión $\texttt{if}\ c\ \texttt{then}\ t\ \texttt{else}\ e$ equivale a $(\texttt{match}\ e\ \{\lambda\_ : ().t;\ \lambda\_ : ().e\}()$. Dicho de otro modo, esta función evalúa c y, según el resultado, evalúa t o e y devuelve el resultado.
- Si, dados identificadores $x_1, \dots, x_k$ a los que le asignamos tipos respectivos $T_1, \dots, T_k$, las expresiones $e_1, \dots, e_k, f$ tienen tipos respectivos $T_1, \dots, T_k, S$, entonces

$$\texttt{let}\ x_1 : T_1 = e_1\ \texttt{and}\ \dots\ \texttt{and}\ x_k : T_k = e_k\ \texttt{in}\ f$$

es una expresión de tipo S que evalúa cada e_i y le asigna el valor al correspondiente x_i, en orden, y finalmente evalúa f y devuelve su valor. Todas las evaluaciones se hacen con los x_i vinculados a los correspondientes valores, y es un error si evaluar uno de los e_i requiere evaluar un x_j con $j \geq i$. Pese a ello, es posible hacer referencia a dicho x_j en e_i si es dentro de una función, lo que permite definir funciones recursivas.

En ocasiones omitiremos las declaraciones de tipos, escribiendo por ejemplo $\lambda x.e$ en vez de $\lambda x : T.e$. Además, escribiremos x^* y x_* en vez de T^*x y T_*x cuando quede claro de qué tipo se está hablando.

Un programa está formado por una serie de sentencias que definen tipos sinónimo y sentencias que definen constantes, de la forma $\texttt{const}\ x : T = e$ o, incluso, de la forma $\texttt{const}\ x_{a_1,\dots,a_k} : T = e$, donde los a_i son variables de tipo para definir una constante por cada asignación de los tipos. Es importante destacar que estas constantes se definen de manera uniforme, es decir, la expresión e y el tipo T se definen igual para cada asignación de tipos a_i y el resultado equivale a definir $\texttt{const}\ x_{\sigma_1,\dots,\sigma_k} : T[\sigma_i/a_i] = e[\sigma_i/a_i]$ para cada asignación de tipos $\sigma_1, \dots, \sigma_k$ donde los σ_i no contienen ningún a_j como variable libre.

Las expresiones recursivas son particularmente importantes, pues equivalen a la iteración en programación iterativa. Por ejemplo, podemos obtener el último elemento de una lista como sigue:

$$\texttt{type list}_a = [()|(a, \texttt{list}_a)]$$
$$\texttt{const}\ \bot_a : a = \texttt{let}\ f : () \to a = \lambda x : ().fx\ \texttt{in}\ f()$$
$$\texttt{const last}_a : \texttt{list}_a \to a = \lambda l.\texttt{match}\ l\ \{\bot_{() \to a};\ \lambda c.\texttt{match}\ c_2\ \{\lambda x.c_1;\ \lambda c'.\texttt{last}_a c'\}\}$$

La primera línea define el tipo lista como la lista vacía o el par formado por el primer elemento de la lista y el resto de elementos. La segunda requiere ser analizada con más detalle. Esta define un elemento de cada tipo (¡incluso de los tipos con dominio vacío!) como el resultado de evaluar una expresión que nunca termina. El problema es que la recursividad introduce la posibilidad de crear expresiones que nunca terminan, pero a la vez es necesaria para no limitar el lenguaje.

Para acomodar esto establecemos que el dominio de cada tipo tiene un (y sólo un[1]) elemento $\bot$ que representa cómputos que no terminan. Entonces, una evaluación ef en que e o f es $\bot$ devuelve $\bot$; $(e_1, \dots, e_k)_i$ devuelve $\bot$ si y sólo si algún $e_i = \bot$, $\lambda x.\bot$ no devuelve $\bot$ (pero su aplicación sobre algún valor sí), $\langle \bot \rangle_i = \bot$, y una sentencia $\texttt{match}\ e\ \{x_1;\ \dots;\ x_k\ \}$ devuelve $\bot$ si y sólo si e devuelve $\bot$ o $e = \langle e_0 \rangle_i$ para cierto e_0 y $x_i e_0$ devuelve $\bot$.

[1]En lenguajes como Haskell puede haber varios valores similares a $\bot$; por ejemplo, un elemento de tipo (a, b) puede ser $\bot$, $(x, \bot)$, $(\bot, y)$, etc. Esto se debe a que Haskell sólo evalúa las partes de las expresiones que son necesarias y por tanto puede que no se llegue a evaluar la parte que no termina. En nuestro caso el lenguaje es estricto y esto no ocurre.

Finalmente, la tercera línea define la función buscada. Si la lista es vacía, devuelve el único valor que puede devolver, $\perp$, y de lo contrario comprueba si la lista tiene más de un elemento, devolviendo en tal caso el último elemento del resto de la lista y en otro caso el único elemento que hay. El uso de $\perp$ para indicar un valor de retorno no definido es bastante habitual.

Igual que hemos definido las listas, podemos definir los números naturales como $\texttt{list}_{()}$ identificando cada número n con la lista de longitud n, los booleanos como $[()|()]$, etc. Ya estamos en condiciones de definir **Type**.

Definición A.1. La categoría **Type** (asociada a un programa) es aquella que tiene como objetos los tipos que se pueden definir en el lenguaje usando los tipos sinónimo definidos en el programa, como morfismos $a \to b$ los elementos de $D_{a \to b}$, como composición de dos morfismos $f : a \to b$ y $g : b \to c$ el morfismo dado por la expresión $\lambda x.g(fx)$, y como identidad el morfismo $\lambda x.x$.

$$\begin{aligned} \texttt{const } (\circ)_{a,b,c} : (b \to c) \to (a \to b) \to (a \to c) &= \lambda x.g(fx) \\ \texttt{const } 1_a : a \to a &= \lambda x.x \end{aligned}$$

La definición de $(\circ)$ muestra un patrón común en programación funcional, en el que una función de varios parámetros no se define como de tipo $(a, b) \to c$ sino de tipo $a \to b \to c$, notación que equivale a $a \to (b \to c)$. Esto tiene la ventaja de ser más cómodo, pues si $f : a \to b \to c$, $x : a$ e $y : b$, podemos escribir fxy en vez de $f(x, y)$. Sin embargo, la mayor ventaja viene de poder definir funciones parciales como $fx : b \to c$, que son útiles en algunos casos.

2. Funtores

Podemos definir un funtor en nuestro lenguaje como un sinónimo de tipo de un parámetro F_a junto con una familia de constantes $F_{a,b} : (a \to b) \to (F_a \to F_b)$, suponiendo que cumplan las condiciones necesarias. Esto nos da un endofuntor **Type** $\to$ **Type**.

Ejemplo A.2. Algunos funtores comúnmente usados son los siguientes:

1. Las listas, definidas anteriormente, junto con la evaluación de funtores componente a componente:

   $$\texttt{const map}_{a,b} = \lambda f : a \to b.\lambda l : \texttt{list}_a.\texttt{match } l \ \{\lambda x.\langle()\rangle_1;\ \lambda c.\langle(fc_1, \texttt{map}\, f\, c_2)\rangle_2\}$$

   Esto equivale en cierto modo al funtor libre **Set** $\to$ **Mon**.
2. Si tenemos un objeto de tipo $[A|E]$, donde un elemento de A representa el resultado de un cálculo completado con éxito y uno de E representa que ha habido un error, es común querer realizar alguna operación con el resultado exitoso pero devolver los errores tal cual. Esto se puede hacer con un funtor para cada tipo de error e.

   $$\texttt{type result}_{e,a} = [\, a \,|\, e \,]$$
   $$\texttt{const rmap}_{e,a,b} = \lambda f : a \to b.\lambda r : \texttt{result}_{e,a}.\texttt{match } r_* \ \{\lambda x.\langle fx\rangle_1;\ \lambda y.\langle y\rangle_2\}^*$$

   Cuando no nos interesan los detalles sobre el error y simplemente queremos representar un valor que puede existir o no, podemos tomar $()$ como tipo de error.

   $$\texttt{type option}_a = \texttt{result}_{(),a}; \qquad \texttt{const optmap}_{a,b} = \texttt{rmap}_{(),a,b}$$
   $$\texttt{const some}_a : a \to \texttt{option}_a = \lambda x.\langle x\rangle_1^{**}; \qquad \texttt{const none}_a : \texttt{option}_a = \langle()\rangle_2^{**}$$
3. Por supuesto, podemos definir el funtor identidad.

   $$\texttt{type id}_a = a; \qquad \texttt{const idmap}_{a,b} = \lambda f.\lambda x.(fx_*)^*$$

   Sin embargo, no podemos definir la composición de funtores dentro del lenguaje ya que los funtores no son valores, por lo que debemos razonar sobre ella desde fuera.

4. No todos los funtores que podemos definir en el lenguaje responden a esta forma. Por ejemplo, consideremos la siguiente función:

$$\texttt{const length}_a : \texttt{list}_a \to \mathbb{N} = \lambda l.\,\texttt{match}\ l\ \{\lambda x.0;\ \lambda c.1 + \texttt{length}_a c_2\}$$

Cada $\texttt{length}_a$ devuelve el número de elementos de una lista y es pues un homomorfismo de monoides de un monoide libre a $\mathbb{N}$, por lo que es un funtor entre monoides vistos como categorías. Como identificamos los naturales con $\texttt{list}_{()}$, esta función también se puede definir de la siguiente forma:

$$\texttt{const length}_a = \texttt{map}_{a,()}(\lambda x.())$$

Al permitir llevar funciones de una categoría a otra, los funtores favorecen la reusabilidad del código y evitan tener que escribir cierto código rutinario. Además, las subcategorías así definidas permiten definir funciones de forma más limpia. Por ejemplo, la función $\texttt{last}_a$ vista anteriormente se puede escribir mejor como sigue:

$$\begin{aligned}&\texttt{const last}_a : \texttt{list}_a \to \texttt{option}_a = \\ &\qquad\qquad \lambda l.\,\texttt{match}\ l\ \{\lambda x.\texttt{none}_a;\ \lambda c.\,\texttt{match}\ c_2\ \{\lambda x.\texttt{some}_a c_1;\ \texttt{last}_a\}\}.\end{aligned}$$

La definición es similar, pero en vez de devolver $\perp$ en caso de error, devuelve $\texttt{none}_a$, permitiendo a quien llamó a la función recuperarse del error o propagarlo con `optmap`. Otra observación es que $\texttt{last}_a$ es una transformación natural, es decir, para todo morfismo $f : a \to b$, $\texttt{last}_b \circ \texttt{map}_{a,b} f = \texttt{optmap}_{a,b} f \circ \texttt{last}_a$. Intuitivamente esto tiene sentido, pues las transformaciones naturales son funciones que «actúan igual» sobre todos los tipos y las funciones con un parámetro de tipo cumplen esto. Lo curioso, sin embargo, es que de hecho esto ocurre con todas las funciones de esta forma.

TEOREMA A.3. *Dadas dos expresiones de tipo $S(a)$ y $T(a)$ que dependen de una variable libre a, si S y T son las funciones sobre objetos de dos endofuntores $S, T : \mathbf{Type} \to \mathbf{Type}$, entonces toda familia de funciones $f_a : S(a) \to T(a)$ define una transformación natural $f : S \to T$.*

La demostración de este teorema es complicada y se basa en el teorema de parametricidad de Reynolds[9], que permite demostrar propiedades de valores a partir de sus tipos.

3. Mónadas

El uso de endofuntores permite trasladar funciones para usarlas en subcategorías que nos resulten más convenientes. Sin embargo, también puede ser deseable trabajar en categorías que no sean subcategorías de **Type**, y en ese aspecto las categorías de Kleisli son muy útiles en la práctica, como se muestra en [6]. Esto se debe a que el uso de estas categorías permite tratar funciones de tipo $a \to Tb$ «como si fueran» de tipo $a \to b$, lo que es útil porque Tb suele representar un resultado del tipo b «con alguna pega», como que en vez de un valor de tipo b puede haber un error, o una cantidad variable de elementos de tipo b.

Veamos cómo implementar categorías de Kleisli de forma eficiente. Una secuencia de transformaciones sobre una entrada podría tener el aspecto siguiente:

$$\texttt{let} x_1 = e_1 \texttt{and} x_2 = e_2 \texttt{and} \dots \texttt{in} g x_n.$$

Esta expresión equivale a $(\lambda x_1.(\lambda x_2.\dots((\lambda x_n.gx_n)(e_n)\dots))(e_2))(e_1)$, aunque es más fácil de leer y de escribir, y si cada e_{i+1} solo depende de x_i, esto equivale a $(g \circ (\lambda x_{n-1}.e_n) \circ \dots \circ (\lambda x_1.e_2))(e_1)$, de modo que las secuencias de transformaciones son, en cierto modo, análogas a composiciones de funciones.

Si ahora (T, η, μ) es una mónada en **Type** y $\hat{\circ}$ es la composición en **Type**, una composición de la forma $f_n \hat{\circ} \dots \hat{\circ} f_2 \hat{\circ} f_1$ con cada $f_i : a_{i-1} \to Ta_i$ en **Type** equivale a $\mu_{a_n} \circ Tf_n \circ \dots \circ \mu_{a_2} \circ Tf_2 \circ f_1$. Parece entonces razonable definir una función parametrizada

$\sigma_{a,b} : (a \to Tb) \to (Ta \to Tb)$ como $\sigma_{a,b} f \coloneqq \mu_b \circ Tf$, y entonces $f_n \hat{\circ} \dots \hat{\circ} f_2 \hat{\circ} f_1$ equivale a $\sigma f_n \circ \cdots \circ \sigma f_2 \circ f_1$.

Esta expresión todavía está en orden inverso de lo que seria lo intuitivo para definir secuencias de transformaciones, por lo que en la práctica se define una función $(\Rightarrow)_{a,b} : T_a \to (a \to T_b) \to T_b$ como $x \Rightarrow_{a,b} f \coloneqq \mu_b((Tf)(x))$, y una secuencia de transformaciones en $\mathcal{C}_T$ se puede expresar de forma relativamente conveniente como $e_1 \Rightarrow (\lambda x_1.e_2 \Rightarrow (\lambda x_2.\dots(\lambda x_n.gx_n)))$, o si cada expresión sólo depende del resultado de la anterior, como $e_1 \Rightarrow (\lambda x_1.e_2) \Rightarrow \cdots \Rightarrow g$.

A partir de $(\Rightarrow)$ se pueden recuperar μ y la actuación de T sobre morfismos. En efecto, para un tipo a y un valor $x : T^2a$, $(x \Rightarrow 1_{Ta}) = \mu_{Ta}(1_{T^2a}x) = \mu_{Ta}x$, y para un morfismo $f : a \to b$ y un valor $x : a$, $(x \Rightarrow (\eta_b \circ f)) = \mu_b(T\eta_b((Tf)(x))) = (Tf)(x)$. Es por ello que, en lenguajes de programación funcionales como Haskell, las mónadas se definen como un sinónimo de tipo con un parámetro junto con definiciones para η y $(\Rightarrow)$.[7, p. 6]

Ejemplo A.4. Algunos de los ejemplos en (6.2) son muy comunes en programación funcional.[7, pp. 17–24]

1. En la mónada identidad, $x \Rightarrow f$ es la evaluación $f(x)$.
2. El funtor $\texttt{result}_d$ admite la mónada dada por el ejemplo 5 en dicha lista. Sean d es el tipo de los errores, $x : [\,b\,|\,d\,]$ y $f : b \to [\,c|\,d\,]$, si x es un elemento de b, $x \Rightarrow b$ equivale a fx viendo x como elemento de b, y si x está en d, $x \Rightarrow f$ devuelve x visto como elemento de $[\,c\,|\,d\,]$.

   $$\eta_a = \lambda x.\langle x\rangle_1 \qquad\qquad (\Rightarrow)_{a,b} = \lambda x.\lambda f.\,\texttt{match}\ x\ \{f;\ \lambda e.e\}$$

   La idea es que, si ocurre un error en un paso de la secuencia de operaciones, la ejecución termina inmediatamente devolviendo el error, y cuando una operación termina con éxito, el resultado pasa al siguiente paso.
3. El funtor `option` también admite una mónada, en tanto que es un caso especial de `result`.
4. El funtor `list`, en tanto que se puede considerar como un funtor libre, admite una mónada. Entonces $x \Rightarrow f$ aplica f a cada elemento de la lista x y concatena las listas de resultados. Esta mónada se puede usar para representar secuencias de operaciones en las que cada operación no devuelve necesariamente un sólo resultado, sino que devuelve una cantidad variable, generalmente pequeña, de resultados.
5. La mónada $\mathcal{P}$ se puede usar en un contexto teórico para representar cómputos no deterministas. El funcionamiento de esta mónada es similar al de `list` pero basándose en un conjunto en vez de en una lista.
6. La mónada S definida en el ejemplo 6 de (6.2) permite representar un estado como un elemento de s. La expresión $x \Rightarrow f$ devuelve una función que primero evalúa x sobre su parámetro para obtener un par (x', s') y entonces ejecuta $f(x')(s')$. La idea es que esto actúa como una especie de composición de funciones, de modo que el valor devuelto por una secuencia de operaciones es una función compuesta $s \to (x, s)$, que recibe un estado inicial de tipo s y devuelve un valor resultado x y un estado final, también de tipo s. Las operaciones definidas por η, no usan el estado, pero otras operaciones pueden acceder al estado y devolver de vuelta un estado modificado. Esto permite representar casos en los que tener un estado e ir modificándolo es conveniente.

Hemos mencionado en la introducción de este capítulo que es posible modelar la interactividad mediante categorías de Kleisli. Esto se suele hacer con una mónada especial, generalmente llamada `IO`, cuyos valores representan acciones, secuencias de acciones o programas enteros. Dado un tipo a, el tipo $\texttt{IO}_a$ es el tipo de las acciones que, al ejecutarlas, realizan cálculos, leen datos de entrada, escriben datos de salida, etc., y finalmente devuelven un elemento de tipo a.

Generalmente $\texttt{IO}_a$ se define de forma extrínseca a partir de los valores que se pueden construir, que dependen de las operaciones que se quieran soportar. A nivel más básico, si

`Char` es el tipo de los caracteres, podemos definir las operaciones

$$\texttt{putChar} : \texttt{Char} \to \texttt{IO}_{()}, \qquad\qquad \texttt{getChar} : \texttt{IO}_{\texttt{Char}},$$

que imprimen un caracter en la terminal y leen un caracter del teclado, respectivamente, aunque también se pueden definir operaciones básicas para otros propósitos como dibujar en la pantalla para crear interfaces gráficas, recibir movimientos del ratón, manipular ficheros y directorios, conectarse a Internet, etc. Las operaciones de la mónada `IO` son las siguientes:

1. $\eta_a : a \to \texttt{IO}\,a$ recibe un valor y devuelve la acción que, sin hacer nada, devuelve dicho valor.
2. Si $x : \texttt{IO}\,a$ y $f : a \to \texttt{IO}\,b$, $x \Rightarrow f$ devuelve la acción consistente en ejecutar la acción x, pasar el resultado devuelto a f y ejecutar la acción devuelta por f, devolviendo su valor.
3. Si $x : \texttt{IO}\,(\texttt{IO}\,a)$, $\mu_a x$ es la acción que ejecuta x y, a continuación, ejecuta la acción devuelta por x, devolviendo su valor.
4. Si $f : a \to b$, $\texttt{IO}\,f : \texttt{IO}\,a \to \texttt{IO}\,b$ toma una acción x y devuelve una acción que ejecuta x y devuelve el resultado de aplicar f al valor devuelto por la acción.

Estas operaciones permiten definir cualquier programa interactivo. Por ejemplo, si $\texttt{unit}_a : a \to \texttt{IO}_a$ y $\texttt{bind}_{a,b} : \texttt{IO}_a \to (a \to \texttt{IO}_b) \to \texttt{IO}_b$ son las operaciones η y $(\Rightarrow)$ de `IO`, y extendemos la sintaxis para poder escribir $\texttt{bind}_{a,b}\,x\,f$ como $x \Rightarrow f$, con asociatividad por la izquierda, podemos escribir una cadena de caracteres, representada como una lista de `Char`, con la siguiente función.

$$\begin{aligned}
&\texttt{const forIO}_a : (a \to \texttt{IO}_{()}) \to \texttt{list}_a \to \texttt{IO}_{()} = \\
&\qquad \lambda l.\lambda f.\,\texttt{match}\ l\ \{\texttt{unit};\ \ \lambda c.(\texttt{putChar}\,c_1 \Rightarrow \lambda x.\texttt{forIO}_a\,f\,c_2)\} \\
&\texttt{const putString} : \texttt{list}_{\texttt{Char}} \to \texttt{IO}_{()} = \texttt{forIO}\,\texttt{putChar}
\end{aligned}$$

Del mismo modo podemos querer leer líneas completas de la entrada:

$$\begin{aligned}
&\texttt{const getLine} : \texttt{IO}_{\texttt{list}_{\texttt{Char}}} = \texttt{getChar} \Rightarrow \\
&\qquad (\lambda c.\,\texttt{if}\ \{c \text{ es un salto de línea}\}\ \texttt{then unit}\,\langle()\rangle_1\,\texttt{else getLine} \Rightarrow \lambda s.\langle(c,s)\rangle_2)
\end{aligned}$$

El programa principal sería entonces un elemento de tipo $\texttt{IO}_{()}$, como el siguiente:

$$\begin{aligned}
&\texttt{putString}\,\text{“¿Cómo te llamas? ”} \Rightarrow \lambda_.(\texttt{getLine} \Rightarrow \\
&\qquad\qquad \lambda n.\texttt{putString}\,\text{“¡Hola, ”} \Rightarrow (\lambda_.\texttt{putString}\,n) \Rightarrow (\lambda_.\texttt{putString}\,\text{“!”}))
\end{aligned}$$

En la práctica es común querer combinar la acción de varias mónadas. Para ello se usan *transformadores de mónadas*, que no son otra cosa que transformaciones naturales en Mnd(**Type**) cuyo dominio es el funtor identidad. Por ejemplo, en vez de definir la mónada `list` directamente, se define una transformación natural $\texttt{listT} : 1_{\mathrm{Mnd}(\mathbf{Type})} \to F$ para cierto endofuntor F en Mnd(**Type**) de forma que, si I es la mónada identidad, $FI = \texttt{list}$.[11] Es común aplicar varios de estos transformadores de mónadas a la vez sobre la mónada identidad o sobre la mónada `IO`.

Finalmente, decir que las mónadas y categorías de Kleisli no sólo son útiles para programación funcional, sino que también lo son para razonar sobre programas en paradigmas convencionales y, de hecho, esta es la motivación que se usa en [6], el artículo que introduce las mónadas en la teoría de la computación. La idea es que, por lo general, los aspectos no funcionales de los lenguajes de programación habituales se pueden representar mediante mónadas. Por ejemplo, `IO` permite comunicarse con el exterior, `state` permite manipular un estado global, $\mathcal{P}$ permite representar cómputos no deterministas, `result` puede usarse para modelar las excepciones, etc. Combinar estos efectos mediante transformadores de mónadas permite definir una semántica denotativa para lenguajes de programación imperativos, lo que a su vez permite razonar sobre programas imperativos con un alto nivel de abstracción.

Bibliografía

[1] Jiří Adámek, Horst Herrlich y George Strecker. *Abstract and Concrete Categories: The Joy of Cats*. John Wiley & Sons, Inc., 1990.

[2] Thierry Coquand y Gérard Huet. *The calculus of constructions*. Inf. téc. RR-0530. INRIA, mayo de 1986. DOI: 10.1016/0890-5401(88)90005-3. URL: https://inria.hal.science/inria-00076024/file/RR-0530.pdf.

[3] Luis Damas y Robin Milner. *Principal type-schemes for functional programs*. 1982. DOI: 10.1145/582153.582176. URL: https://web.cs.wpi.edu/~cs4536/c12/milner-damas_principal_types.pdf.

[4] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag New York, Inc., 1971.

[5] Saunders Mac Lane. "One universe as a foundation for category theory". En: *Reports of the Midwest Category Seminar III. Lecture Notes in Mathematics* 106 (1969). DOI: 10.1007/BFb0059147.

[6] Eugenio Moggi. *Notions of computation and monads*. URL: https://person.dibris.unige.it/moggi-eugenio/ftp/ic91.pdf.

[7] Jeff Newbern. *All About Monads: A comprehensive guide to the theory and practice of monadic programming in Haskell, v. 1.1.0*. URL: https://www.cs.rit.edu/~swm/cs561/All_About_Monads.pdf.

[8] John C. Reynolds. "Polymorphism is not set-theoretic". En: *Semantics of Data Types*. Ed. por Gilles Kahn, David B. MacQueen y Gordon Plotkin. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, págs. 145-156. ISBN: 978-3-540-38891-3. DOI: 10.1007/3-540-13346-1_7. URL: https://link.springer.com/chapter/10.1007/3-540-13346-1_7.

[9] John C. Reynolds. *Types, abstraction and parametric polymorphism*. 1983. URL: https://www.cs.cmu.edu/afs/cs/user/jcr/ftp/typesabpara.pdf.

[10] Emily Riehl. *Category Theory in Context*.

[11] Chung-chieh Shan. *Monad transformers*. URL: http://conway.rutgers.edu/~ccshan/wiki/blog/posts/Monad_transformers/ (visitado 20-06-2023).