Chapter 1

CATEGORIES

Category Theory studies "objects" and "morphisms" between them. These concepts are both primitive in Category Theory: objects are not collections of "elements," and morphisms do not need to be functions between sets (thus morphisms cannot be applied to "elements" but only composed with other morphisms). Any immediate access to the internal structure of objects is prevented: all properties of objects must be specified by properties of morphisms (existence of particular morphisms, their unicity, validity of some equations among them, and so on). This is quite similar to considering objects as "abstract data types," that is, data specifications that are independent of any particular implementation. The relevance of Category Theory for programming languages comes from the previous consideration: it offers a highly formalized language especially suited for stating abstract properties of structures. Thus, it relates to widely used programming methodologies and provides as well a formal setting for the mathematical investigation of the semantics of programming languages.

1.1 Category: Definition and Examples

As we have mentioned, Category Theory is a theory of functions, and the only basic operation is composition. The concept of **Category** embodies some abstract properties of the composition operator "$\circ$" for functions that "reasonably" must be guaranteed. In particular, if $g: a \to b$ and $h: b \to c$, then there exist $h \circ g: a \to c$; moreover, composition must be associative and an identity must exist for all objects.

This is the formal definition:

1.1.1 Definition *A* ***category*** ***C*** *is*

- *a collection Ob_C of* ***objects****, denoted by a, b . . . A, B . . .*
- *a collection Mor_C of* ***morphisms*** *(****arrows****), denoted by f, g . . . ,*
- *two operations* ***dom****,* ***cod*** *assigning to each arrow f two objects respectively called* ***domain*** *(****source****) and* ***codomain*** *(****target****) of f*
- *an operation* ***id*** *assigning to each object b a morphism id_b (the* ***identity*** *of b) such that $dom(id_b) = cod(id_b) = b$*
- *an operation " $\circ$ " (****composition****) assigning to each pair f, g of arrows with $dom(f) = cod(g)$ an arrow $f \circ g$ such that $dom(f \circ g) = dom(g)$, $cod(f \circ g) = cod(f)$*
- *identity and composition, moreover, must satisfy the following conditions:*

***identity law**: for any arrows f, g such that cod(f) = b = dom(g)*

$id_b \circ f = f$

$g \circ id_b = g$

***associative law:** for any arrows f, g, h such that dom(f) = cod(g) and dom(g) = cod(h)*

$(f \circ g) \circ h = f \circ (g \circ h)$

We write f: a → b to denote a morphism whose source and target are respectively a and b. Given two objects a and b, the collection of all morphisms f such that f: a → b is denoted by **C**[a,b]; the writing f∈**C**[a,b] is thus a third way to express the fact that dom(f) = a, and cod(f) = b. For the moment we shall use one notation or the other indifferently

The following table lists some common categories by specifying their objects and arrows, letting the definition of their operators as an exercise for the reader:

Category	**Objects**	**Morphisms**
Set	sets	functions
Top	topological spaces	continuous functions
Vect	vector spaces	linear transformations
Grp	groups	group homomorphisms
PO	partially ordered sets	monotone functions

The intuition of the notion of "category" suggested by the previous examples is to consider the objects as a collection of "structured" sets and the morphisms as the "associated" or "acceptable" functions with respect to the structure. This is too restrictive, though, since no requirement is made in the definition which may force the morphisms to be "single valued" or to be functions in extenso: a simple example is the category **Rel** with sets as objects and relations as morphisms.

The simplest category has only one object and one arrow (the identity for that object): this category is usually called **1**. Note that, by definition, if **C** is a category, then every object b of **C** has an identity id_b: b→b. The identity is unique, since if id_b' is another identity for b, then for the *identity law*, $id_b' = id_b \circ id_b' = id_b$. A category is called **discrete** if every arrow is the identity of some object: in this case a category is fully determined by the collection of its objects. **1** is a discrete category.

A category is called a **preorder** if for every pair of objects a, b there is *at most* one morphism f: a→b. The reason for the name is that a preorder category is fully determined by a preordering relation among its objects. Indeed, in a preorder **C**, there is only one way that composition may be defined; thus **C** is known when the collection of morphisms $Mor_{\mathbf{C}}$ and the operations dom and cod are known. But every arrow f: a→b may be identified with the pair (a,b), since once the source and target are known there is no choice about what the arrow is to be; thus, all the information about the category **C** is given by the relation $R_{\mathbf{C}} = \{(a,b) / \text{there is an arrow } f\in\mathbf{C}[a,b]\}$, that is, by a preorder relation.(Exercise: prove that the relation $R_{\mathbf{C}}$ is a preorder for every category **C**).

Every discrete category is a preorder. The simplest nondiscrete category which is a preorder is the category **2**, which has two objects, let us call them 0 and 1, and three arrows: the two identities id_0, id_1 and an arrow (0,1): $0\to 1$. In a similar way we can define for each natural number n a preorder category **n**, from the usual ordering on the set {0,1, . . . n-1}. Preorder categories have a common property: they may have plenty of objects, but given two objects, there exists at most one morphism between them.

A dual situation is given by monoids, viewed as categories. A **monoid** is a set having an associative binary operation and an identity element. A category with just one object yields a monoid, where composition of morphisms is the binary operation. Conversely, any monoid $(A, \cdot)$ is a category with just one object. For example, the category with the set of natural numbers as unique object and the recursive functions as morphisms yields the monoid of the recursive functions.

As well as preorders, another example where objects are not necessarily understood as "structured sets" is given by **deductive systems** as categories. In these categories propositions are objects and each morphism $f : a\to b$ corresponds to (a suitable equivalence class of) a proof of $a \vdash b$ (a **entails** b). Observe that a category is obtained easily in the presence of the identical entailment $i_a : a\to a$ and the associative composition of proofs

$$\frac{f : a\to b \qquad g : b\to c}{g \circ f : a\to c}$$

This approach to deduction is very relevant in the categorical understanding of logics of a constructive nature, such as Intuitionistic Logic, where the intended interpretation of proofs is given by (effective) operations. It will be the main paradigm for understanding the relation between types and objects investigated in the second part of this book.

1.2 Diagrams

An important tool in the practice of Category Theory is the use of **diagrams** for representing equations. In a diagram a morphism $f\in\mathbf{C}[a,b]$ is drawn as an arrow from a to b labeled f. A diagram **commutes** if the composition of the morphism along any path between two fixed objects is equal. For example, the *associative* and *identity laws* of the definition of "category" may be nicely visualized by the following commuting diagrams:

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

Diagrams are a typical way, in Category Theory, to describe equational reasoning and turn out to be particularly effective when dealing with several equations at a time. In particular, assertions such as "if $diagram_1$ and . . . $diagram_n$ commute, then $diagram$ commutes" express conditional statements about equalities.

We hope that the reader, while using this book, will acquire some familiarity with diagrams and will learn how to go back and forth from diagrams to equations. Our extended use of equations in this book comes from our desire to stress the "computational" nature of most categorical reasoning.

1.3 Categories out of Categories

A main feature of Category Theory is the facility to define new, more structured categories out of simpler ones. In this section we consider only a few simple constructions; a number of other examples occur throughout the book.

1.3.1 Definition *A category **D** is a **subcategory** of a category **C**, if*

1. *$Ob_{\mathbf{D}} \subseteq Ob_{\mathbf{C}}$;*
2. *for all a, b in $Ob_{\mathbf{D}}$, $\mathbf{D}[a,b] \subseteq \mathbf{C}[a,b]$;*
3. *composition and identities in **D** coincide with those of **C**.*

*A subcategory is **full** if for all a, b in $Ob_{\mathbf{D}}$ $\mathbf{D}[a,b] = \mathbf{C}[a,b]$.*

A full subcategory is fully determined by its collection of objects.

1.3.2 Definition *The **dual** category $\mathbf{C}^{op}$ of a category **C** has the same objects and the same morphisms of **C**, $id^{op}{}_b = id_b$, $dom^{op}(f) = cod(f)$, $cod^{op}(f) = dom(f)$, and $f \circ^{op} g = g \circ f$.*

Note that $\mathbf{C}^{\mathbf{op}}[b,a] = \mathbf{C}[a,b]$ and $(\mathbf{C}^{\mathbf{op}})^{\mathbf{op}} = \mathbf{C}$.

Exercise $\mathbf{Set}^{\mathbf{op}}$ is a subcategory of **Rel**, but not of **Set**. Is it a full subcategory?

Duality is a very powerful technique of Category Theory. If P is a generic proposition expressed in the language of Category Theory, the dual of P (P^{op}) is the statement obtained by replacing the word “dom” by “cod,” “cod” by “dom,” “$g \circ h$” by “$h \circ g$.” If P is true in a category **C**, then P^{op} is true in $\mathbf{C}^{op}$; if P is true in every category, then also P^{op} is, since every category is the dual of its dual.

Duality may be applied to diagrams as well: given a diagram in a category **C**, the dual diagram in $\mathbf{C}^{op}$ is obtained by simply reverting the arrows; of course, a dual diagram commutes if and only if the original one does.

1.3.3 Definition *Given two Categories* **C** *and* **D**, *the* ***product category C×D*** *has for objects the pairs (a,b) where a and b are respectively objects of* **C** *and* **D**, *and for morphisms pairs (f,g): (a,b)→(a',b') where f: a→a' and g: b→b' are respectively morphisms of* **C** *and* **D**. *Finally,* $id_{(a,b)} = (id_a, id_b)$ *and* $(f,g) \circ (f',g') = (f \circ f', g \circ g')$.

1.3.4 Definition *Given a category* **C** *and an object a in* $Ob_{\mathbf{C}}$, *the* ***category C↓a of objects over*** *a is so defined:* $Ob_{\mathbf{C}\downarrow a} = \{f \in Mor_{\mathbf{C}} \mid cod(f) = a\}$; *given two objects f: b→a, g: c→a, a morphism with source f and target g is an arrow* $h \in \mathbf{C}[b, c]$ *such that* $g \circ h = f$. *Identities and composition in* **C**↓*a are inherited from* **C**.

In case **C** is **Set** in the above definition, it is useful to think of an object g: B→A in **Set**↓A as an A-indexed family of disjoint sets, namely, $\{g^{-1}(a)\}_{a \in A}$ (these sets are the inverse images of elements in A under g). Then h: B→B' is a morphism from g: B→A to g': B'→A if and only if it is consistent with the “decomposition” of B and C induced by g and g', i.e., if and only if (iff) $\forall b\ \ b \in g^{-1}(a) \Rightarrow h(b) \in g'^{-1}(a)$.

Since the intended meaning behind the construction of a category **C**↓I is that to consider an object g: A→I as a collection $\{\{i\} \times g^{-1}(i)\}_{i \in I}$, it is usual to call **C**↓I a **slice category** over I (denoted **C**/I). An object g: A→I of the slice category is then called a **generalized object** of **C** at stage I. A **section** of g: A→I is a function s: I→A such that $g \circ s = id_I$; the idea is that s gives, for each index $i \in I$, an element $s(i) \in g^{-1}(i)$.

Exercise Define the dual notion, that is, the category **C**↑a of objects*under* a, whose objects are the arrows with source a.

1.4 Monic, Epic, and Principal Morphisms

A function f between two sets A and B is called “injective” when, for all $a, a' \in A$, if $f(a) = f(a')$ then $a = a'$. In particular, given any two functions g, h : C→A, if for all $c \in C$ $f(g(c)) = f(h(c))$, then for all $c \in C$ $g(c) = h(c)$ or, also, if $f \circ g = f \circ h$ then $g = h$. Thus, every injective function behaves like a left identity (it is left cancellable). The converse is also true: given f: A→B, if for any pair of functions $g, h : C \to A$, $f \circ g = f \circ h$ implies $g = h$, then f is injective. For suppose otherwise: then there are a and a' such that $f(a) = f(a')$ but $a \neq a'$; define then g and h by $g(c) = a$ for all $c \in C$, and $h(c) = a'$ for all $c \in C$; of course $f \circ g = f \circ h$ but $g \neq h$, that is, a contradiction.

We have proved thus that a function f is injective if and only if $f \circ g = f \circ h$ implies $g = h$. In a similar way it is not difficult to prove that f is surjective if and only if $g \circ f = h \circ f$ implies $g = h$. These considerations motivate the following definitions.

1.4.1 Definition. *Let **C** be a category and a, b∈$Ob_{\mathbf{C}}$. Then*

*i. an arrow h∈**C**[a,b] is **epic** (is an **epimorphism**) iff*

$$g \circ h = f \circ h \;\Rightarrow\; g = f\,;$$

*ii. an arrow h∈**C**[a,b] is **monic** (is a **monomorphism**) iff*

$$h \circ g = h \circ f \;\Rightarrow\; g = f\,;$$

*iii. an arrow h∈**C**[a,b] is **iso** (is an **isomorphism**) iff there exists g∈**C**[b,a] such that*

$$g \circ h = id \text{ and } h \circ g = id\,.$$

Two objects a and b are **isomorphic** ($a \cong b$) if there exists an isomorphism h∈**C**[a,b]. Clearly, any isomorphism is monic and epic; the converse, though, does not need to be true (see the example and the exercises below).

A monic (or epic) h∈**C**[a,b] (or h'∈**C**[a,b]) is **split** if there exist g∈**C**[b,a] (or g'∈**C**[b,a]) such that $g \circ h = id$ ($h' \circ g' = id$).

Although the intuition of regarding mono- and epimorphisms as injective and surjective maps is correct for many interesting categories, sometimes it can be misleading. Consider, say, the category **Mon** of monoids and the inclusion inc from ω, the positive integers, into **z**, the relative ones. Clearly mono, inc is also epi, though. As a matter of fact, take g,h∈**Mon**[**z**,a] for some monoid a, and write \g(n) for g(-n). Then $g \circ inc = h \circ inc$ implies $g = h$ for g(-n) = \g(n) =\h(n) = h(-n) (that is, the behavior of the monoids' homomorphism g or h on **z** is entirely determined by their behavior on ω). As a side consequence, we may also conclude that not every arrow that is both monic and epic is an isomorphism: this is clearly in contrast to the set-theoretic intuition.

Exercises

1. Give an epi which is not surjective in **Top**.
2. Find a counterexample for the following assertion: let **C** be a category; if f∈**C**[a,b] and g∈**C**[b,a] are mono, then a is isomorphic to b. (Note that the assertion is true in **Set.**)
3. Prove that a split monic is an iso.

1.4.2 Definition *Let **C** be a category and a, b∈$Ob_{\mathbf{C}}$. Then*

*i. an arrow h∈**C**[a,b] is a **principal** morphism iff*

$\forall f \in \mathbf{C}[a,b]\ \exists g \in \mathbf{C}[a,a]\ \ f = h \circ g\ ;$

ii. a pair of arrows $f \in \mathbf{C}[a,b]$ *and* $g \in \mathbf{C}[b,a]$ *is a* ***retraction pair*** *iff* $g \circ f = id$. *Then,* a *is called a* ***retract*** *of* b $(a<b)$ *via the retraction pair* (f,g).

By diagrams, h is principal iff for all f there is a g such that

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

Principal morphisms have been inspired by recursion theory; the idea they are based on essentially corresponds to a classical notion of reducibility (see the category **EN** in section 2.2 below).

1.4.3 Proposition *Let* $\mathbf{C}$ *be a category and* $a, b \in Ob_{\mathbf{C}}$. *Then*

1. *if* $a<b$ *via* (i,h), *then* h *is epi and principal,* i *is mono;*
2. *if* $h \in \mathbf{C}[a,b]$ *is principal and there exists an epi* $k \in \mathbf{C}[a,b,]$, *then* h *is epi;*
3. *if* $a<b$ *and* $f \in \mathbf{C}[b,a]$ *is principal, then there exists* $g \in \mathbf{C}[a,b]$ *such that* $a<b$ *via* (g,f).

Proof 1. $g \circ h = f \circ h \ \Rightarrow\ g \circ h \circ i = f \circ h \circ i \ \Rightarrow\ g = f$, for $h \circ i = id$.

The proof that h is principal is a simple diagram chase:

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

That is, $\forall f\ \exists g\ f = h \circ g$. Just take $g = i \circ f$; then $h \circ g = h \circ i \circ f = f$.

Finally, $i \circ g = i \circ f \ \Rightarrow\ h \circ i \circ g = h \circ i \circ f \ \Rightarrow\ g = f$.

2. $g \circ h = f \circ h \ \Rightarrow\ g \circ k = g \circ h \circ g' = f \circ h \circ g' = f \circ k$ (for a suitable g') $\Rightarrow\ g = f$.

3. Let $a<b$ via (j,i). Since f is principal, $\exists s \in \mathbf{C}[b,b]\ \ j = f \circ s$. Then, for $g = s \circ i$, one has $f \circ g = j \circ i = id_a$. As a diagram,

$$\begin{tikzcd} b \arrow[r, "i"] \arrow[dr, "g"'] & a \arrow[r, "j"] \arrow[d, "s"] & b \\ & a \arrow[ur, "f"'] & \end{tikzcd}$$

♦

Exercises

1. Characterize retractions in terms of split monos and epis.
2. Show that, given a category **C,** one can define a category $\mathbf{C}^{\mathbf{Ret}}$ whose objects are the same of **C** and whose morphisms are retraction pairs in **C**, that is $F \in \mathbf{C}^{\mathbf{Ret}}[a,b]$ iff $F=(f,g)$ and $a<b$ via (f,g) in **C**.

If $(f:a \to b,\ g:b \to a)$ is a retraction pair, then the function $h = f \circ g: b \to b$ is idempotent, that is, $h \circ h = h$. Indeed, $h \circ h = (f \circ g) \circ (f \circ g) = f \circ (g \circ f) \circ g = f \circ g = h$. This property suggests the following definition:

1.4.4 Definition *Given a category* ***C*** *and an object* $b \in Ob_{\mathbf{C}}$, *the* ***category of idempotents on b*** *(*$\mathbf{Ret}_b$*) is so defined:*

$$Ob_{\mathbf{Ret}_b} = \{ f \in \mathbf{C}[b,b] \ / \ f \circ f = f \}$$
$$Mor_{\mathbf{Ret}_b} = \{ (f, k, g) \ / \ f, g \in Ob_{\mathbf{Ret}_b},\ k \in \mathbf{C}[b,b],\ k = g \circ k \circ f \}$$
$$dom((f, k, g)) = f,\ cod((f, k, g)) = g$$
$$id_f = (f, f, f)$$
$$(f, k, g) \circ (g', k', f) = (g', k \circ k', g)$$

We leave as an exercise for the reader to check the identity and associative laws for the previous category. $\mathbf{Ret}_b$ will be used in several places because of its relevance to this book.

1.5 Subobjects

The concept of *subobject* is the categorical version of the set-theoretical *subset* . The main idea is to regard a subset A of a given object B as a monomorphism $f: D \to B$ (intuitively, a monomorphism f such that "$f(D) = A$"). Of course, many different monic arrows may define the same subset; thus, it is necessary to introduce a reasonable equivalence relation, and define subobjects up to this equivalence.

Let **C** be a category. If $f: b \to a$ and $g: c \to a$ are two monic arrows with common target a, then we say $f \le g$ if and only if there exists $h: b \to c$ such that $g \circ h = f$. Note that in this case, the unique h must be monic too, indeed $h \circ k = h \circ k' \Rightarrow g \circ h \circ k = g \circ h \circ k' \Rightarrow f \circ k = f \circ k' \Rightarrow k = k'$.

Exercise Prove that the preorder $\le$ is the full subcategory of $\mathbf{C}{\downarrow}a$ determined by monomorphisms only.

When $f \leq g$ and $g \leq f$ we write $f \cong g$. Then $\cong$ is an equivalence relation among the monomorphisms with common target a (prove it as an exercise); the equivalence classes of this equivalence relation are called subobjects of a.

1.5.1 Definition *Let a be an object of a category **C**. A **subobject** [f] of a is an equivalence class of a monomorphism $f: b \to a$, with respect to the equivalence relation $\cong$ defined above.*

Very often, we shall make no distinction between equivalence classes and their representatives, and we shall denote a subobject with a single monomorphism.

It should be clear that the categorical approach to "subsets" carries more information than the set-theoretic one. Monomorphisms, like all morphisms, preserve the structural information of the category. For example, in the category **Grp** of groups subobjects are subgroups: (mono)morphisms must take the identity to the identity and preserve the group operation. Similarly consider the category of p.o.sets (partially ordered sets) with a bottom element. A subobject of one such p.o.set must be a structured subset as well, and it must contain an element smaller than all the others.

References: Any book in Category Theory, such as MacLane (1971) Herrlich and Strecker (1973), Arbib and Manes (1975), Barr and Wells (1985), Rydeheard and Burstall (1988). The specific notions and categories introduced (such as retractions) will be used later in more structured settings, with the appropriate references.