# An Invitation to Game Comonads

## Review of Category Theory

Tomáš Jakl

Luca Reggio

Department of Computer Science and Technology, University of Cambridge, UK
*Email address*: `tomas.jakl@cl.cam.ac.uk`
*URL*: `https://tomas.jakl.one`

Department of Computer Science, University College London, UK
*Email address*: `l.reggio@ucl.ac.uk`
*URL*: `https://lucareggio.github.io`

Draft version compiled on 27th July 2022.

# Contents

**Note to the reader** . . . ii

**Chapter 1 Category theory** . . . 1

1.1 Categories . . . 1
1.2 Reasoning with arrows . . . 5
1.3 Functors . . . 8
1.4 Natural transformations . . . 11
1.5 Monads and comonads . . . 14
1.6 Kleisli and Eilenberg–Moore categories for a comonad . . . 18
1.7 Adjunctions . . . 22

# Note to the reader

The purpose of these notes is to recall the basic notions of category theory that will be used throughout the course on *Game comonads* at ESSLLI 2022. More details about the course can be found here:

`https://tomas.jakl.one/teaching/2022-su-game-comonads`

The main definitions and properties will be recalled in the lectures, however we warmly suggest reading these notes before the beginning of the course. Especially for the students who are not familiar with basic category theory, it can be useful to attempt to solve (some of) the exercises spread out through the text.

More advanced exercises are marked with the symbol 🚀, whereas the symbol ⌬ denotes a paragraph or result whose content is not needed for the course and can be safely skipped (sometimes, these can assume familiarity with specific notions, such as *elementary classes*).

CHAPTER 1

# Category theory

## 1.1. Categories

**Definition 1.1.** A *category* consists of:

- A class $\mathrm{Ob}(\mathcal{C})$ of *objects*, typically denoted by $A, B, C$.
- A class $\mathrm{Mor}(\mathcal{C})$ of *morphisms* (also called *arrows*), typically denoted by $f, g, h$.
- Two functions
$$\mathrm{dom}, \mathrm{cod}\colon \mathrm{Mor}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{C})$$
assigning to a morphism its *domain* and *codomain*, respectively. We write $f\colon A \to B$ to indicate that $f$ is a morphism with domain $A$ and codomain $B$. For each pair of objects $A, B$, we define the associated *hom-set* to be the collection
$$\mathcal{C}(A, B) \coloneqq \{f \in \mathrm{Mor}(\mathcal{C}) \mid f\colon A \to B\}.$$
- For any triple of objects $A, B, C$, a *composition map*
$$\mathcal{C}(A, B) \times \mathcal{C}(B, C) \to \mathcal{C}(A, C), \quad (f, g) \mapsto g \circ f.$$
- For each object $A$, an *identity* morphism $\mathrm{id}_A\colon A \to A$.

The data above must satisfy the following equations for all arrows $f, g, h$ and all objects $A, B$, whenever the compositions are well-defined:
$$h \circ (g \circ f) = (h \circ g) \circ f \qquad \text{(Associativity law)}$$
$$f \circ \mathrm{id}_A = f = \mathrm{id}_B \circ f \qquad \text{(Identity laws)}$$

**Remark 1.2.** Properly speaking, the hom-sets $\mathcal{C}(A, B)$ need not be sets and may be *proper classes*. Cf. Definition 1.25. ▲

Here is our first example of category:

- **Set**: the objects are sets and the morphisms are the functions between them.

We can always produce a new category from an old one by selecting some objects and taking all morphisms between them. For instance, we can define the category

- $\mathbf{Set}_{fin}$: the objects are *finite* sets and the morphisms are the functions between them.

In general, new categories can be obtained from old ones by restricting either their arrows or objects, or both.

**Definition 1.3.** Suppose $\mathcal{C}$ is a category and consider collections $\mathrm{Ob}(\mathcal{D}) \subseteq \mathrm{Ob}(\mathcal{C})$ and, for all $A, B \in \mathrm{Ob}(\mathcal{D})$, $\mathcal{D}(A,B) \subseteq \mathcal{C}(A,B)$. Then $\mathcal{D}$ is a *subcategory* of $\mathcal{C}$ if

$$\mathrm{id}_A \in \mathcal{D}(A,A)$$

for all $A \in \mathrm{Ob}(\mathcal{D})$ and, for all $f \in \mathcal{D}(A,B)$ and $g \in \mathcal{D}(B,C)$,

$$g \circ f \in \mathcal{D}(A,C).$$

If, moreover, $\mathcal{D}(A,B) = \mathcal{C}(A,B)$ for all $A, B \in \mathrm{Ob}(\mathcal{D})$, then $\mathcal{D}$ is a *full subcategory* of $\mathcal{C}$.

For example, $\mathbf{Set}_{fin}$ is a full subcategory of $\mathbf{Set}$.

**Exercise 1.4.** Show that a subcategory $\mathcal{D}$ of $\mathcal{C}$ is itself a category, with respect to the obvious composition maps and identity morphisms. ▲

In order to provide further examples of categories, we shall recall some basic mathematical notions.

**$\sigma$-Structures.** Recall that a *relational signature* $\sigma$ is a set of relation symbols $\{R_i \mid i \in I\}$ such that each $R_i$ is assigned a positive integer $\mathtt{ar}(R_i)$, called the *arity* of $R_i$. A *$\sigma$-structure* is given by a set $A$ (the *universe* of the structure) together with an interpretation of the relation symbols in $\sigma$. That is, for each $R_i \in \sigma$ of arity $n$ we have a set of $n$-tuples

$$R_i^A \subseteq A^n.$$

A *homomorphisms of $\sigma$-structures* (or *$\sigma$-homomorphism*, for short) from a $\sigma$-structure $A$ to a $\sigma$-structure $B$ is a function $f\colon A \to B$ between their universes that preserves the interpretations of the relations, i.e. for each $R_i \in \sigma$ of arity $n$ and all $(a_1, \dots, a_n) \in A^n$,

$$(a_1, \dots, a_n) \in R_i^A \implies (f(a_1), \dots, f(a_n)) \in R_i^B.$$

For any relational signature $\sigma$, we have a category

- $\mathbf{Str}(\sigma)$: the objects are $\sigma$-structures and the morphisms are the $\sigma$-homomorphisms.

Restricting to the *finite* $\sigma$-structures, i.e. those $\sigma$-structures whose universe is a finite set, we obtain the category

- $\mathbf{Str}_{fin}(\sigma)$: the objects are finite $\sigma$-structures and the morphisms are the $\sigma$-homomorphisms.

**Relations.** Let $X, Y$ be any two sets. A *relation* from $X$ to $Y$, written $X \nrightarrow Y$, is a subset $R$ of the Cartesian product $X \times Y$. If $X = Y$, we simply refer to $R$ as a (binary) relation on $X$. Given a pair $(x, y) \in X \times Y$, we sometimes write $xRy$ instead of $(x, y) \in R$. The *identity relation* on a set $X$ is the *diagonal relation*

$$\Delta_X \coloneqq \{(x, x) \in X \times X \mid x \in X\},$$

and the composition of two relations $R\colon X \nrightarrow Y$ and $S\colon Y \nrightarrow Z$ is the relation $R;S\colon X \nrightarrow Z$ defined by

$$R;S \coloneqq \{(x, z) \in X \times Z \mid \exists y \in Y.\ xRy \text{ and } ySz\}.$$

These data define a category

- **Rel**: the objects are sets and the morphisms are relations.

**Set** is a subcategory of **Rel**, but not a full subcategory.

**Partial orders.** A *partial order* on a set $X$ is a binary relation on $X$ satisfying the following conditions for all $x, y, z \in X$:

$$x \leq x \qquad \text{(Reflexivity)}$$
$$x \leq y \ \wedge\ y \leq z \implies x \leq z \qquad \text{(Transitivity)}$$
$$x \leq y \ \wedge\ y \leq x \implies x = y. \qquad \text{(Antisymmetry)}$$

A *partially ordered set* (or *poset*, for short) is a pair $(X, \leq)$ where $X$ is a set and $\leq$ is a partial order on $X$. A *monotone* (or *order-preserving*) map from a poset $(X, \leq_X)$ to a poset $(Y, \leq_Y)$ is a function $f\colon X \to Y$ such that $x_1 \leq_X x_2$ implies $f(x_1) \leq_Y f(x_2)$ for all $x_1, x_2 \in X$.

We obtain a category

- **Pos**: the objects are posets and the morphisms are monotone maps.

More generally, a relation that is reflexive and transitive—but not necessarily antisymmetric—is a called a *preorder*. In the same spirit as above, we can define a *preordered set* as a pair $(X, \leq)$ consisting of a set $X$ and a preorder $\leq$ on it. This yields a category

- **PreOrd**: the objects are preordered sets and the morphisms are monotone maps.

**Pos** is a full subcategory of **PreOrd**.

**Forests and trees.** Let $(X, \leq)$ be a poset. A subset $C \subseteq X$ is said to be a *linear order* (or a *chain*) if, for all $x, y \in C$, either $x \leq y$ or $y \leq x$. For any element $x \in X$, let us write

$$\downarrow x \coloneqq \{y \in X \mid y \leq x\}$$

for the *downset* of $x$.

A *forest* is a poset $(X, \leq)$ such that, for all $x \in X$, the downset $\downarrow x$ is a finite linear order. The *roots* of a forest are the minimal elements, i.e. those elements that are not strictly above any other element. The *covering* *relation* $\prec$ associated with the partial order $\leq$ is defined by $x \prec y$ if and only if $x < y$ (that is, $x \leq y$ and $x \neq y$) and there is no $z$ such that $x < z < y$.

A *forest morphism* from a forest $(X, \leq_X)$ to a forest $(Y, \leq_Y)$ is a function $f\colon X \to Y$ that preserves roots and the covering relation. That is, for all $x_1, x_2 \in X$, if $x_1$ is a root of $(X, \leq_X)$ then $f(x_1)$ is a root of $(Y, \leq_Y)$, and if $x_1 \prec x_2$ then $f(x_1) \prec f(x_2)$. This defines a category

- **Forests**: the objects are forests and the arrows are forest morphisms.

A forest with at most one root is a called a *tree*. Considering only those objects of **Forests** that are (non-empty) trees, we obtain a new category

- **Trees**: the objects are *non-empty* trees and the arrows are forest morphisms.

The category of trees is a full subcategory of the category of forests.

**Monoids and groups.** A *monoid* is a triple $(M, \cdot, 1)$ where $M$ is a set,

$$\_ \cdot \_ : M \times M \to M$$

is a binary operation on $M$ (the *multiplication* of $M$), and 1 is an element of $M$ (the *identity element*), satisfying the equational axioms:

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \qquad \text{(Associativity law)}$$
$$x \cdot 1 = x = 1 \cdot x \qquad \text{(Identity laws)}$$

A *monoid homomorphism* from $(M, \cdot_M, 1_M)$ to $(N, \cdot_N, 1_N)$ is a function $f\colon M \to N$ that preserves the multiplication and the identity element, i.e. for all $m_1, m_2 \in M$

$$f(m_1 \cdot_M m_2) = f(m_1) \cdot_N f(m_2) \quad \text{and} \quad f(1_M) = 1_N.$$

We thus get a category

- **Mon**: the objects are monoids and the arrows are monoid homomorphisms.

Moreover, a *group* is a monoid $(G, \cdot, 1)$ in which every element has an inverse. That is, for all $g \in G$ there is an element $g^{-1} \in G$ such that

$$g \cdot g^{-1} = 1 = g^{-1} \cdot g.$$

A *group homomorphism* from a group $(G, \cdot_G, 1_G)$ to a group $(H, \cdot_H, 1_H)$ is a function $f\colon G \to H$ that preserves the multiplication and the identity element (i.e., is a monoid homomorphisms), as well as inverses: for all $g \in G$, $f(g)^{-1} = f(g^{-1})$. This yields a category

- **Grp**: the objects are groups and the arrows are group homomorphisms.

**Exercise 1.5.** Prove that **Grp** is a full subcategory of **Mon**. ▲

**Elementary classes.** Let $T$ be a first-order theory in a signature $\tau$ (possibly containing both relation and function symbols). A *$\tau$-homomorphism* between $\tau$-structures is a function that preserves all relation and function symbols in $\tau$. These data define a category

- $\mathbf{Mod}(T)$: the objects are models of $T$ and the arrows are the $\tau$-homomorphisms between them.

Note that the categories **Pos**, **PreOrd** and **Mon** are of the form $\mathbf{Mod}(T)$ for an appropriate signature $\tau$ and theory $T$. The same holds for **Grp**, because every monoid homomorphism between two groups is a group homomorphism (see Exercise 1.5). On the other hand, **Rel**, **Forests** and **Trees** are not of the form $\mathbf{Mod}(T)$ since the arrows in these categories are not all "structure-preserving functions" between the appropriate objects.

Finally, every poset can be regarded as a category; this simple observation leads to a considerable source of examples for many categorical notions:

**Example 1.6.** Any partially ordered set $(P, \leq)$ can be seen as a category in the following way: the objects are the elements of $P$ and, for all $x, y \in P$, the hom-set $P(x, y)$ is given by

$$P(x,y) \coloneqq \begin{cases} \{\star\} & \text{if } x \leq y \\ \emptyset & \text{otherwise.} \end{cases}$$

In particular, between any two objects there is at most one arrow.[1] The reflexivity law $x \leq x$ yields the identity morphisms, and the transitivity law $x \leq y \leq z \Rightarrow x \leq z$ gives compositions of morphisms.

Note that we did not use the fact that posets satisfy the antisymmetry law, hence this construction works more generally for preordered sets. ▲

## 1.2. Reasoning with arrows

In ordinary mathematics, based on set-theoretic foundations, we are used to element-wise reasoning. In order to show that two functions $f, g \colon X \to Y$ are distinct, we seek to find an element $x \in X$ such that $f(x) \neq g(x)$. This hinges on the observation that two functions (with the same domain and codomain) are equal if, and only if, they coincide at each element of their domain. Similarly for continuous maps between topological spaces, group homomorphisms, linear maps between vector spaces, and so forth.

This simple principle, which is an intrinsic part of the "logic of sets" (and is a consequence of the *axiom of extensionality* in set theory), does not hold in all categories. In fact, the notion of *element* is not even available in

[1]This implies that *any* diagram in this category commutes!

an arbitrary category. For this reason, it is important to learn to reason in terms of arrows, rather than elements.[2]

A first useful observation is that, when comparing arrows, equations can be rephrased in terms of commutative diagrams. Consider for instance arrows $f\colon A \to B$, $g\colon B \to D$, $h\colon A \to C$ and $i\colon C \to D$. Then the equation

$$g \circ f = i \circ h$$

holds if, and only if, the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\ C \arrow[r, "i"] & D \end{tikzcd}$$

This sort of rephrasing is at the base of *diagrammatic reasoning*, which is pervasive in category theory.

Many mathematical notions that are usually defined in terms of elements admit purely arrow-theoretic reformulations which can be used to generalise these concepts to arbitrary categories. Take, for instance, the notion of *bijection*: a function $f\colon X \to Y$ between sets is bijective if

$$\forall y \in Y \ \exists ! x \in X.\ f(x) = y.$$

We can avoid any reference to the elements of $X$ and $Y$ by noting that $f$ is a bijection precisely when there exists a function $g\colon Y \to X$ such that

$$g \circ f = \mathrm{id}_X \quad \text{and} \quad f \circ g = \mathrm{id}_Y.$$

The latter property only mentions arrows and makes sense in any category.

**Definition 1.7.** An arrow $f\colon A \to B$ in a category $\mathcal{C}$ is an *isomorphism* if there exists an arrow $g\colon B \to A$ such that

$$g \circ f = \mathrm{id}_A \quad \text{and} \quad f \circ g = \mathrm{id}_B.$$

If there exists an isomorphism $A \to B$, we say that $A$ is *isomorphic* to $B$ and write $A \cong B$.

**Exercise 1.8.** Show that the isomorphisms in the category **Mon** of monoids are precisely the bijective monoid homomorphisms. ▲

**Exercise 1.9.** Prove that the isomorphisms in the category $\mathbf{Str}(\sigma)$, for any relational signature $\sigma$, are precisely the *$\sigma$-isomorphisms*, i.e. the bijective functions $f\colon A \to B$ such that, for each relation symbol $R \in \sigma$ of arity $n$,

$$(a_1, \ldots, a_n) \in R^A \iff (f(a_1), \ldots, f(a_n)) \in R^B.$$

[2] Having said that, *(generalised) elements* can be defined in a large class of categories, and the question of whether a category has *enough elements* is of interest in several contexts, such as topos theory or categorical logic. For example, Gödel's Completeness Theorem is equivalent to the statement that a certain category has enough elements.

Give an example of a bijective $\sigma$-homomorphism that is not an isomorphism. ▲

**Exercise 1.10.** Describe the isomorphisms in the category **Pos** of posets. ▲

Note that every isomorphism $f$ in a category satisfies the conditions

$$f \circ g = f \circ h \implies g = h$$

and

$$g \circ f = h \circ f \implies g = h$$

whenever the compositions are well defined. These cancellation laws define two important classes of morphisms in any category:

**Definition 1.11.** Let $f \colon A \to B$ be an arrow in a category $\mathcal{C}$. We say that

- $f$ is *monic* (or a *monomorphism*) if, for all arrows $g, h \colon C \to A$,
$$f \circ g = f \circ h \implies g = h.$$
- $f$ is *epic* (or an *epimorphism*) if, for all arrows $g, h \colon B \to C$,
$$g \circ f = h \circ f \implies g = h.$$

**Exercise 1.12.** Prove that, in **Set**, the monomorphisms and epimorphisms coincide, respectively, with the injective and surjective functions. Conclude that, in **Set**, a morphism that is both monic and epic is an isomorphism. ▲

**Exercise 1.13.** Give an example of a category admitting an arrow that is both epic and monic, but not an isomorphism. ▲

**Exercise 1.14.** Show that a morphism in **Mon** is monic if, and only if, it is an injective monoid homomorphism. Is every epimorphism in **Mon** surjective? ▲

**Exercise 1.15.** Prove that, in the categories **Pos**, **Forests** and **Str**$(\sigma)$, the monomorphisms and epimorphisms are those morphisms whose underlying function is, respectively, injective and surjective. ▲

By definition, a morphism in a category has a domain and a codomain; this is akin to the case of directed graphs, where edges have a source and a target. Given an arbitrary category $\mathcal{C}$, we can construct a new category $\mathcal{C}^{\mathrm{op}}$ by *reversing* the direction of arrows in $\mathcal{C}$. That is, an arrow $A \to B$ in $\mathcal{C}^{\mathrm{op}}$ is defined to be an arrow $B \to A$ in $\mathcal{C}$. Formally:

**Definition 1.16.** The *opposite category* $\mathcal{C}^{\mathrm{op}}$ of a category $\mathcal{C}$ is defined by $\mathrm{Ob}(\mathcal{C}^{\mathrm{op}}) := \mathrm{Ob}(\mathcal{C})$ and, for all objects $A, B$ of $\mathcal{C}^{\mathrm{op}}$, $\mathcal{C}^{\mathrm{op}}(A, B) := \mathcal{C}(B, A)$. Identities in $\mathcal{C}^{\mathrm{op}}$ are the same as in $\mathcal{C}$, and the composition $g \circ f$ in $\mathcal{C}^{\mathrm{op}}$ is defined as $f \circ g$ in $\mathcal{C}$.

For example, let $(P, \leq)$ be a poset regarded as a category as explained in Example 1.6. Its opposite category can be identified with the poset $(P, \leq^{\mathrm{op}})$ obtained by turning the order of $P$ upside down. That is, for all $x, y \in P$,

$$x \leq^{\mathrm{op}} y \iff y \leq x.$$

The passage from a category to its opposite is a purely formal operation but is at the heart of deep connections between e.g. algebra and geometry (mathematics), syntax and semantics (logic), and observables and states (physics). This is the subject of *duality theory*.

For now, it suffices to mention the following fact, sometimes referred to as *principle of duality*: An arrow-theoretic statement $\varphi$ holds in a category $\mathcal{C}$ precisely when the dual statement (obtained from $\varphi$ by reversing the direction of arrows) holds in $\mathcal{C}^{\mathrm{op}}$.

**Exercise 1.17.** Prove that an arrow in a category $\mathcal{C}$ is monic (respectively, epic) if, and only if, it is epic (respectively, monic) in $\mathcal{C}^{\mathrm{op}}$. ▲

## 1.3. Functors

At the heart of category theory is the idea that morphisms between objects are as important as the object themselves. So, having defined the notion of category, it is natural to ask what is a "morphism of categories".

**Definition 1.18.** A *functor* $F\colon \mathcal{C} \to \mathcal{D}$ from a category $\mathcal{C}$ to a category $\mathcal{D}$ consists of:

- A map $\mathrm{Ob}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{D})$ that assigns an object $FA$ of $\mathcal{D}$ to every object $A$ of $\mathcal{C}$.
- For all $A, B \in \mathrm{Ob}(\mathcal{C})$, a map $\mathcal{C}(A, B) \to \mathcal{D}(FA, FB)$ that assigns an arrow $Ff\colon FA \to FB$ in $\mathcal{D}$ to every arrow $f\colon A \to B$ in $\mathcal{C}$, preserving compositions and identities:
$$F(g \circ f) = Fg \circ Ff \quad \text{and} \quad F(\mathrm{id}_A) = \mathrm{id}_{FA}.$$

Trivial examples of functors are obtained by considering a subcategory $\mathcal{C}$ of a category $\mathcal{D}$. Then there is a functor $\mathcal{C} \to \mathcal{D}$ that acts as the inclusion on both the objects and arrows.

What is a functor from a poset $P$ to a poset $Q$ (regarded as categories)? The object map is simply a function $F\colon P \to Q$. On arrows, whenever $x, y \in P$ satisfy $x \leq y$, we must assign an arrow $Ff\colon Fx \to Fy$ to the unique arrow $x \to y$. But if there exists an arrow $Fx \to Fy$, it is unique. So, this amounts to saying that $F$ is monotone. Note that the requirement that compositions and identities be preserved is trivially satisfied, as any diagram in a poset commutes. Therefore, functors between posets are precisely monotone maps.

Further examples of functors are presented in the following exercises.

**Exercise 1.19.** For any set $X$, denote its power-set by $\mathcal{P}X$. Furthermore, given a function $f\colon X \to Y$ between sets, let $\mathcal{P}f\colon \mathcal{P}X \to \mathcal{P}Y$ be the *direct image map* that sends a subset $S \subseteq X$ to $f[S] \coloneqq \{f(x) \in Y \mid x \in S\}$.

Show that these assignments yield a functor $\mathcal{P}\colon \mathbf{Set} \to \mathbf{Set}$, known as the *(covariant) power-set functor.* ▲

**Exercise 1.20.** Given a set $X$, let $X^*$ denote the monoid of all finite *lists* (i.e., sequences) of elements of $X$. The monoid operation of $X^*$ is given by concatenation of lists, and the identity element is the empty list. If $f\colon X \to Y$ is a function between sets, then $f^*\colon X^* \to Y^*$ sends a list $[x_1, \dots, x_n]$ to $[f(x_1), \dots, f(x_n)]$. Show that this construction determines a functor $\mathbf{Set} \to \mathbf{Mon}$. ▲

**Exercise 1.21.** For any set $X$, let $FX$ be the set of all *non-empty* finite lists of elements of $X$. We equip $FX$ with a partial order defined as follows: for all $s, t \in FX$, $s \le t$ if and only if $s$ is a prefix of $t$. Any function $f\colon X \to Y$ induces a map $FX \to FY$ that sends a (non-empty) sequence $[x_1, \dots, x_n]$ to $[f(x_1), \dots, f(x_n)]$. Prove that these data define a functor $\mathbf{Set} \to \mathbf{Forests}$. ▲

**Exercise 1.22.** Given a forest $X$, let $X_+$ be the tree obtained by adding a least element to $X$. Any forest morphism $X \to Y$ can be extended to a forest morphism $X_+ \to Y_+$ by sending the root of $X_+$ to the root of $Y_+$. Check that this gives a functor $\mathbf{Forests} \to \mathbf{Trees}$. ▲

Many examples of functors arise from "forgetting" part of the structure; these are generally (and informally) referred as *forgetful functors*. For example, given a group $(G, \cdot_G, 1_G)$, we can forget its algebraic structure and only retain the information about its underlying set $G$. Similarly, if $f$ is a group homomorphism from $(G, \cdot_G, 1_G)$ to $(H, \cdot_H, 1_H)$, we can simply regard $f$ as a function $G \to H$. These assignments determine a forgetful functor

$$\mathbf{Grp} \to \mathbf{Set}.$$

Likewise, we can define a functor $\mathbf{Grp} \to \mathbf{Mon}$ by only retaining the monoid structure of groups and group homomorphisms.

In the same way that morphisms in a category can be composed (whenever the codomain of one matches the domain of the other), functors between categories can be composed. If $F\colon \mathcal{B} \to \mathcal{C}$ and $G\colon \mathcal{C} \to \mathcal{D}$ are functors, then there is a *composite functor*

$$GF\colon \mathcal{B} \to \mathcal{D}$$

that sends an object $A$ of $\mathcal{B}$ to the object $GFA$ of $\mathcal{D}$, and an arrow $f\colon A \to B$ in $\mathcal{B}$ to the arrow $GFf\colon GFA \to GFB$ in $\mathcal{D}$.

**Notation 1.23.** Given a functor $F\colon \mathcal{C} \to \mathcal{C}$, we sometimes denote the composite $FF\colon \mathcal{C} \to \mathcal{C}$ by $F^2$. Similarly, $F^3$ stands for $FFF$, and so forth.

Moreover, any category $\mathcal{C}$ admits an *identity functor*

$$\mathrm{id}_{\mathcal{C}}\colon \mathcal{C} \to \mathcal{C}$$

that acts as the identity on both objects and morphisms. This suggests that categories, together with functors between them, form themselves a category. This is indeed the case, but a precise definition requires extra care so as to avoid *Russell's paradox* ("the set of all sets is not a set").

**Exercise 1.24.** Define a chain of functors

$$\mathbf{Forests} \to \mathbf{Pos} \to \mathbf{PreOrd} \to \mathbf{Set}$$

whose composition is the obvious forgetful functor **Forests** $\to$ **Set**. ▲

An important role is played by *set-valued functors*, i.e. functors $\mathcal{C} \to \mathbf{Set}$. If we regard the category of sets and functions as the "universe" where ordinary (classical) mathematics is carried out, set-valued functors correspond to interpretations (e.g. of theories) in this universe. This is the perspective adopted in *categorical logic*, where models of a theory are defined as set-valued functors satisfying appropriate properties. It is therefore pertinent to ask if, for any category $\mathcal{C}$, there are any functors $\mathcal{C} \to \mathbf{Set}$. The answer is *yes*—whenever $\mathcal{C}$ satisfies a mild set-theoretic "smallness condition".

**Definition 1.25.** Let $\mathcal{C}$ be a category and suppose that, for all $A, B \in \mathrm{Ob}(\mathcal{C})$, the collection $\mathcal{C}(A, B)$ is a set (as opposed to a proper class). Then $\mathcal{C}$ is said to be *locally small.*

Any object $A$ of a locally small category $\mathcal{C}$ determines a *hom-set functor*

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

that sends an object $B$ of $\mathcal{C}$ to the set $\mathcal{C}(A, B)$, and an arrow $f \colon B \to C$ in $\mathcal{C}$ to the function

$$\mathcal{C}(A, f) \colon \mathcal{C}(A, B) \to \mathcal{C}(A, C), \quad g \mapsto f \circ g.$$

**Properties of functors.** In the same way that we discussed properties of arrows (e.g., being monic, epic, or an isomorphism), it is useful to consider properties that a functor may, or may not, satisfy.

**Definition 1.26.** A functor $F \colon \mathcal{C} \to \mathcal{D}$ is said to be *faithful* (respectively, *full*) if, for all objects $A, B$ of $\mathcal{C}$, the map

$$\mathcal{C}(A, B) \to \mathcal{D}(FA, FB), \quad f \mapsto Ff$$

is injective (respectively, surjective).

Note that, if $\mathcal{C}$ is a subcategory of $\mathcal{D}$, then the inclusion functor $\mathcal{C} \to \mathcal{D}$ is always faithful, and is full precisely when $\mathcal{C}$ is a full subcategory of $\mathcal{D}$ (see Definition 1.3).

Typically, forgetful functors are faithful but not full.

**Exercise 1.27.** Prove that the forgetful functors **Mon** $\to$ **Set** and **Pos** $\to$ **Set** are faithful but not full. ▲

**Exercise 1.28.** Consider the functors **Set** $\to$ **Mon** and **Set** $\to$ **Forests** introduced, respectively, in Exercise 1.20 and Exercise 1.21. Are they faithful? Are they full? ▲

**Exercise 1.29.** Show that every functor preserves isomorphisms, but need not preserve monomorphisms nor epimorphisms. ▲

Note that the composition of faithful functors is again faithful, and the composition of full functors is full (check this!).

**Definition 1.30.** A functor $F\colon \mathcal{C} \to \mathcal{D}$ is an *isomorphism* if there exists a functor $G\colon \mathcal{D} \to \mathcal{C}$ such that

$$GF = \mathrm{id}_{\mathcal{C}} \quad \text{and} \quad FG = \mathrm{id}_{\mathcal{D}}.$$

If it exists, the functor $G$ in the previous definition is unique and referred to as the *inverse* of $F$.

**Exercise 1.31.** Prove that the functor **Forests** $\to$ **Trees** defined in Exercise 1.22 is an isomorphism. Describe its inverse. ▲

If a functor is an isomorphism, then it is full and faithful (why?). However, the notion of isomorphism is typically too strong. This leads to the weaker concept of *equivalence* which will be introduced in the next section (see Definition 1.40).

## 1.4. Natural transformations

We have seen that categories consist of objects and morphisms between them, and moreover there is an appropriate notion of morphism between categories—namely, functors. One could go further and consider morphisms of morphisms of categories, morphisms between the latter, etc. This is the framework of *higher category theory*; in the present notes we shall only take one more step and discuss "morphisms between functors".

**Definition 1.32.** A *natural tranformation* $\alpha\colon F \to G$ between functors $F, G\colon \mathcal{C} \to \mathcal{D}$ is a collection

$$\{\alpha_A\colon FA \to GA \mid A \in \mathrm{Ob}(\mathcal{C})\}$$

of arrows in $\mathcal{D}$ indexed by objects of $\mathcal{C}$ satisfying the following *naturality condition*: For all arrows $f\colon A \to B$ in $\mathcal{C}$, the following square commutes.

$$\begin{tikzcd} FA \arrow[r, "Ff"] \arrow[d, "\alpha_A"'] & FB \arrow[d, "\alpha_B"] \\ GA \arrow[r, "Gf"] & GB \end{tikzcd}$$

The morphism $\alpha_A$ is called the *component of $\alpha$ at $A$*.

**Example 1.33.** Consider the (covariant) power-set functor $\mathcal{P}\colon \mathbf{Set} \to \mathbf{Set}$ defined in Exercise 1.19, along with the identity functor $\mathrm{id}_{\mathbf{Set}}\colon \mathbf{Set} \to \mathbf{Set}$. We show that there is a natural transformation

$$\eta\colon \mathrm{id}_{\mathbf{Set}} \to \mathcal{P}$$

whose component at a set $X$ is the function

$$\eta_X\colon X \to \mathcal{P}X, \quad x \mapsto \{x\}.$$

To this end, we must verify that the following square commutes for all functions $f\colon X \to Y$ between sets:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\eta_X"'] & Y \arrow[d, "\eta_Y"] \\ \mathcal{P}X \arrow[r, "\mathcal{P}f"] & \mathcal{P}Y \end{tikzcd}$$

In turn, this follows by observing that, for all $x \in X$,

$$(\eta_Y \circ f)(x) = \eta_Y(f(x)) = \{f(x)\} = \mathcal{P}f(\{x\}) = (\mathcal{P}f \circ \eta_X)(x). \quad \blacktriangle$$

**Exercise 1.34.** As in the previous example, consider the (covariant) power-set functor $\mathcal{P}\colon \mathbf{Set} \to \mathbf{Set}$. Show that there exists a natural transformation

$$\mu\colon \mathcal{P}\mathcal{P} \to \mathcal{P}$$

whose component at a set $X$ is the function

$$\mu_X\colon \mathcal{P}\mathcal{P}X \to \mathcal{P}X, \quad S \mapsto \bigcup S. \quad \blacktriangle$$

**Exercise 1.35.** Let $F\colon \mathbf{Set} \to \mathbf{Mon}$ be the functor defined in Exercise 1.20, and let $U\colon \mathbf{Mon} \to \mathbf{Set}$ be the forgetful functor. Recall that, for all sets $X$, $FX$ is the monoid of finite lists of elements of $X$, hence $UFX$ is the *set* of finite lists of elements of $X$. Denoting the composite functor by $T \coloneqq UF\colon \mathbf{Set} \to \mathbf{Set}$, check that there is a natural transformation

$$\eta\colon \mathrm{id}_{\mathbf{Set}} \to T$$

whose component at a set $X$ is the function

$$\eta_X\colon X \to TX, \quad x \mapsto [x].$$

Moreover, show that there is a natural transformation

$$\mu\colon TT \to T$$

whose component at $X$ is the *flatten map*

$$\mu_X\colon TTX \to TX,$$
$$[[x_{1,1}, \dots, x_{1,n_1}], \dots, [x_{k,1}, \dots, x_{k,n_k}]] \mapsto [x_{1,1}, \dots, x_{1,n_1}, \dots, x_{k,1}, \dots, x_{k,n_k}].$$

Note the similarity between these natural transformations and those defined for the power-set functor $\mathcal{P}$ in Example 1.33 and Exercise 1.34. These are two instances of the same concept, namely that of *monad*. See Section 1.5. ▲

**Exercise 1.36.** Let $F\colon \mathbf{Set} \to \mathbf{Forests}$ be the functor defined in Exercise 1.21, and consider the composite functor $G \coloneqq UF\colon \mathbf{Set} \to \mathbf{Set}$ where $U\colon \mathbf{Forests} \to \mathbf{Set}$ is the forgetful functor. For every set $X$, $GX$ is the set of all non-empty finite lists of elements of $X$. Verify that there is a natural transformation

$$\varepsilon\colon G \to \mathrm{id}_{\mathbf{Set}}$$

whose component at $X$ is the map sending a list to its last element:

$$\varepsilon_X \colon GX \to X, \quad [x_1, \dots, x_n] \to x_n.$$

(Note that $\varepsilon_X$ is well defined because $FX$ consists of *non-empty* lists!)

Furthermore, show that there is a natural transformation

$$\delta \colon F \to GG$$

whose component at $X$ sends a list to the list of its (non-empty) prefixes:

$$\delta_X \colon GX \to GGX, \quad [x_1, \dots, x_n] \mapsto [[x_1], [x_1, x_2], \dots, [x_1, \dots, x_n]]. \quad ▲$$

**Functor categories.** To give a precise meaning to the assertion that natural transformations are morphisms of functors, we introduce the notion of *functor category.* For any two categories $\mathcal{C}, \mathcal{D}$, there is a category

$$[\mathcal{C}, \mathcal{D}]$$

whose objects are functors $F \colon \mathcal{C} \to \mathcal{D}$ and whose arrows are natural transformations. Given a functor $F \colon \mathcal{C} \to \mathcal{D}$, its identity is the natural transformation $\alpha \colon F \to F$ such that, for all objects $A$ of $\mathcal{C}$, $\alpha_A = \mathrm{id}_{FA}$. Natural transformations can be composed in the obvious way: if $\alpha \colon F \to G$ and $\beta \colon G \to H$ are natural transformations, then their composite is the natural transformation

$$\beta\alpha \colon F \to H, \quad (\beta \circ \alpha)_A \coloneqq \beta_A \circ \alpha_A.$$

**Exercise 1.37.** Verify the statements in the previous paragraph and check carefully that $[\mathcal{C}, \mathcal{D}]$ is indeed a category. ▲

What are the isomorphisms in a functor category?

**Definition 1.38.** Let $\alpha \colon F \to G$ be a natural transformation between functors $F, G \colon \mathcal{C} \to \mathcal{D}$. If all components of $\alpha$ are isomorphisms in $\mathcal{D}$, then $\alpha$ is called a *natural isomorphism.*

**Exercise 1.39.** Prove that the isomorphisms in a functor category $[\mathcal{C}, \mathcal{D}]$ are precisely the natural isomorphisms. ▲

We can use the notion of natural isomorphism to weaken the concept of isomorphism between categories. This is akin to the passage from homeomorphism to homotopy equivalence in topology.

**Definition 1.40.** A functor $F \colon \mathcal{C} \to \mathcal{D}$ is an *equivalence* if there exists a functor $G \colon \mathcal{D} \to \mathcal{C}$ and natural isomorphisms

$$\alpha \colon GF \to \mathrm{id}_{\mathcal{C}} \quad \text{and} \quad \beta \colon FG \to \mathrm{id}_{\mathcal{D}}.$$

If there exists an equivalence $\mathcal{C} \to \mathcal{D}$, we shall say that $\mathcal{C}$ is *equivalent* to $\mathcal{D}$ and write $\mathcal{C} \simeq \mathcal{D}$.

When $\mathcal{D} = \mathbf{Set}$, the objects of $[\mathcal{C}, \mathbf{Set}]$ are set-valued functors defined on $\mathcal{C}$. Likewise, one can consider the functor category $[\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$ of set-valued functors defined on the opposite category $\mathcal{C}^{\mathrm{op}}$. These play a central role in category theory and are called *presheaves on* $\mathcal{C}$. The category of presheaves on $\mathcal{C}$ is in general much larger than the original category $\mathcal{C}$, yet it admits a copy of $\mathcal{C}$ as a full subcategory. This is known as the *Yoneda embedding*, which we recall below.

**Exercise 1.41.** Let $\mathbf{2}$ denote a category with precisely two objects, their identities, and two distinct parallel morphisms. This category can be depicted as follows (where we omit the identity arrows for convenience):

$$\begin{tikzcd}\bullet \arrow[r, shift left] \arrow[r, shift right] & \bullet\end{tikzcd}$$

Show that the functor category $[\mathbf{2}, \mathbf{Set}]$ is isomorphic to the category of simple, directed (multi)graphs and graph homomorphisms. ▲

**Exercise 1.42.** Consider the ordered set $(\mathbb{N}, \leq)$ of natural numbers as a category (see Example 1.6). Prove that the category $[\mathbb{N}^{\mathrm{op}}, \mathbf{Set}]$ of presheaves on $\mathbb{N}$ is equivalent to **Forests**. ▲

Recall the notion of hom-set functor defined on Page 10. Every object of a locally small category $\mathcal{C}$ induces a presheaf

$$\mathcal{C}^{\mathrm{op}}(A, -) = \mathcal{C}(-, A) \colon \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}.$$

Further, any arrow $f \colon A \to B$ in $\mathcal{C}$ induces a natural transformation

$$\mathcal{C}(-, A) \to \mathcal{C}(-, B)$$

whose component at an object $C$ of $\mathcal{C}$ (equivalently, of $\mathcal{C}^{\mathrm{op}}$) is the function

$$\mathcal{C}(C, A) \to \mathcal{C}(C, B), \quad g \mapsto f \circ g.$$

(Check that this is indeed a natural transformation!) This determines a functor $\mathcal{C} \to [\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$.

Theorem 1.43 (Yoneda embedding). *Let $\mathcal{C}$ be a locally small category. The functor*

$$\mathcal{C} \to [\mathcal{C}^{\mathrm{op}}, \mathbf{Set}], \quad A \mapsto \mathcal{C}(-, A) \colon \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$$

*is full and faithful.*

## 1.5. Monads and comonads

Together with the notions of *functor* and *natural transformation* (and that of *adjunction*, to be discussed in Section 1.7), the concept of *monad* is one of the pillars of basic category theory. Its importance is in large part due to the fact that it shows how a number of constructions throughout mathematics are instances of the same abstract notion. Monads are pervasive in algebra, but appear also in topology, probability theory, functional programming, and other areas. The dual notion, that of *comonad*, is equally important but perhaps less familiar to many researchers. We shall present both notions, but focus in particular on comonads as they play a key role in relation with finite model theory.

Monads and comonads are functors of type $\mathcal{C} \to \mathcal{C}$, i.e. from a category to itself, satisfying appropriate properties. The intuitions are very different though: whereas monads encode ways to *combine* distinct parts or elements, comonads correspond to *decompositions* (or *unravellings*) of an object.

To make this idea more precise, let us look at the case of monads first. Consider a monoid $(M, \cdot, 1)$: the monoid operation $\cdot$ tells us how to combine any two elements of $M$, and 1 is the identity element for this operation. In a sense, monads are generalised monoids.[3] As such, they come equipped with a multiplication and an identity satisfying the usual monoid laws.

**Definition 1.44.** A *monad* on a category $\mathcal{C}$ is a tuple $(T, \mu, \eta)$ where

- $T\colon \mathcal{C} \to \mathcal{C}$ is a functor,
- $\mu\colon T^2 \to T$ is a natural transformation, called *multiplication*,
- $\eta\colon \mathrm{id}_{\mathcal{C}} \to T$ is a natural transformation, called *unit*,

such that the following diagrams commute for all objects $A$ of $\mathcal{C}$:

$$\begin{tikzcd}
T^3A \arrow[r, "\mu_{TA}"] \arrow[d, "T\mu_A"'] & T^2A \arrow[d, "\mu_A"] & TA \arrow[r, "\eta_{TA}"] \arrow[d, "T\eta_A"'] \arrow[dr, "\mathrm{id}_{TA}"] & T^2A \arrow[d, "\mu_A"] \\
T^2A \arrow[r, "\mu_A"] & TA & T^2A \arrow[r, "\mu_A"] & TA
\end{tikzcd}$$

In the previous definition, $T^2$ denotes the composite functor $TT$, and similarly for $T^3$; see Notation 1.23.

**Example 1.45.** Recall from Exercise 1.35 that there is a functor $T\colon \mathbf{Set} \to \mathbf{Set}$ that assigns to a set $X$ the set of all finite lists of elements of $X$, and it comes equipped with natural transformations

$$\mu\colon T^2 \to T \quad \text{and} \quad \eta\colon \mathrm{id}_{\mathbf{Set}} \to T.$$

The components of $\mu$ are the flatten maps that transform a list of lists into a list, e.g. $[[x, y], [z], [y]]$ is sent to $[x, y, z, y]$, and the components of $\eta$ send an element $x$ to the one-element list $[x]$.

The tuple $(T, \mu, \eta)$ is a monad on **Set**, called the *free monoid monad*. This amounts to saying that the diagrams in Definition 1.49 commute for all objects of **Set**. We give a "proof by example"; the formal proof follows the same ideas (with some extra bookkeeping) and is left to the reader. Suppose we have a set $X = \{x, y, z\}$. For the left-hand diagram in Definition 1.49, we must consider an element of $T^3X$, i.e. a list of lists of lists of elements of $X$. For instance,

$$[[[x], [x, y]], [[x, y, z]]].$$

Chasing this element around the diagram, we get:

$$\begin{tikzcd}
{[[[x], [x, y]], [[x, y, z]]]} \arrow[r, mapsto, "\mu_{TX}"] \arrow[d, mapsto, "T\mu_X"'] & {[[x], [x, y], [x, y, z]]} \arrow[d, mapsto, "\mu_X"] \\
{[[x, x, y], [x, y, z]]} \arrow[r, mapsto, "\mu_X"] & {[x, x, y, x, y, z]}
\end{tikzcd}$$

[3]This is a small lie: the truth is that a monad *is* (a special case of) a monoid, namely a monoid object in the functor category $[\mathcal{C}, \mathcal{C}]$.

The right-hand diagram in Definition 1.49 is easier to check: given an element of $TX$, say $[z, y, y, x]$, we have:

$$\begin{tikzcd}
{[z,y,y,x]} \arrow[r, mapsto, "\eta_{TX}"] \arrow[d, mapsto, "T\eta_X"'] \arrow[dr, mapsto, "\mathrm{id}_{TX}"] & {[[z,y,y,x]]} \arrow[d, mapsto, "\mu_X"] \\
{[[z],[y],[y],[x]]} \arrow[r, mapsto, "\mu_X"] & {[z,y,y,x]}
\end{tikzcd}$$

▲

**Exercise 1.46.** Fill in the details of the proof in Example 1.45. ▲

It is instructive to look at what is a monad on a poset (regarded as a category according to Example 1.6). Let $(P, \leq)$ be a poset, and let $(T, \mu, \eta)$ be a monad on $P$. Then $T\colon P \to P$ is a monotone map (cf. the discussion on Page 8), and for each $x \in P$ the component of $\mu$ at $x$ yields an arrow $\mu_x\colon T^2x \to Tx$. That is,

$$T^2x \leq Tx.$$

Similarly, the component of $\eta$ at $x$ yields an arrow $\eta_x\colon x \to Tx$ and so

$$x \leq Tx.$$

By monotonicity, applying $T$ to both sides of the latter inequation we obtain $Tx \leq T^2x$ and therefore

$$T^2x = Tx.$$

In other words, $T$ is a *closure operator* on $P$.

**Definition 1.47.** A *closure operator* on a poset $P$ is a monotone map $t\colon P \to P$ that is

(1) *increasing*, i.e. for all $x \in P$, $x \leq tx$, and
(2) *idempotent*, i.e. for all $x \in P$, $t^2x = tx$.

**Exercise 1.48.** Show that monads on posets are precisely the closure operators. That is, for any closure operator $t\colon P \to P$ there exist unique natural transformations $\mu$ and $\eta$ such that $(t, \mu, \eta)$ is a monad on $P$. ▲

Therefore, a monad on a poset is akin to the modality $\Diamond$ in modal logic, and dually the modal operator $\Box$ is an instance of a comonad.

A comonad on a category $\mathcal{C}$ can be succinctly defined as a monad on the opposite category $\mathcal{C}^{\mathrm{op}}$. More explicitly:

**Definition 1.49.** A *comonad* on a category $\mathcal{C}$ is a tuple $(G, \delta, \varepsilon)$ where

- $G\colon \mathcal{C} \to \mathcal{C}$ is a functor,
- $\delta\colon G \to G^2$ is a natural transformation, called *comultiplication*,
- $\varepsilon\colon G \to \mathrm{id}_{\mathcal{C}}$ is a natural transformation, called *counit*,

such that the following diagrams commute for all objects $A$ of $\mathcal{C}$:

$$\begin{tikzcd}
GA \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] & G^2A \arrow[d, "\delta_{GA}"] \\
G^2A \arrow[r, "G\delta_A"] & G^3A
\end{tikzcd}
\qquad
\begin{tikzcd}
GA \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] \arrow[dr, "\mathrm{id}_{GA}"] & G^2A \arrow[d, "\varepsilon_{GA}"] \\
G^2A \arrow[r, "G\varepsilon_A"] & GA
\end{tikzcd}$$

When defining comonads (and similarly, monads), there are a number of things to be verified: one should give a functor, two natural transformations, and check that the appropriate diagrams commute. We now recall an equivalent description of comonads that allows us to reduce these verifications and is very useful in concrete cases; a similar description for monads is of course available using the principle of duality.

**Definition 1.50.** A *comonad in Kleisli–Manes form* on a category $\mathcal{C}$ is given by:

- an object map $G\colon \mathrm{Ob}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{C})$,
- a morphism $\varepsilon_A\colon GA \to A$ for every $A \in \mathrm{Ob}(\mathcal{A})$,
- a *coextension operation* associating with any morphism $f\colon GA \to B$ a morphism $f^*\colon GA \to GB$.

These must satisfy the following equations for all morphisms $f\colon GA \to B$ and $g\colon GB \to C$:

$$\varepsilon_A^* = \mathrm{id}_{GA}, \quad \varepsilon_B \circ f^* = f, \quad (g \circ f^*)^* = g^* \circ f^*. \tag{1.1}$$

Given a comonad in Kleisli–Manes form, we can extend the object map $G\colon \mathrm{Ob}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{C})$ to a functor $\mathcal{C} \to \mathcal{C}$ by setting

$$Gf \coloneqq (f \circ \varepsilon_A)^*$$

for every morphism $f\colon A \to B$. Furthermore, the arrows $\delta_A \coloneqq \mathrm{id}_{GA}^*$ are the components of a natural transformation $\delta\colon G \to G^2$, the arrows $\varepsilon_A$ are the components of a natural transformation $\varepsilon\colon G \to \mathrm{id}_{\mathcal{C}}$, and $(G, \delta, \varepsilon)$ is a comonad on $\mathcal{C}$.

**Exercise 1.51.** Prove that, conversely, every comonad on $\mathcal{C}$ induces a comonad in Kleisli–Manes form, and the two assignments are inverse to each other. (Hint: define the coextension of $f\colon GA \to B$ as $f^* \coloneqq Gf \circ \delta_A$.) ▲

**Example 1.52.** We define a comonad on **Set** using the Kleisli–Manes form. The object map

$$G\colon \mathrm{Ob}(\mathbf{Set}) \to \mathrm{Ob}(\mathbf{Set})$$

sends a set $X$ to the set of all non-empty finite lists of elements of $X$. For each set $X$,

$$\varepsilon_X\colon GX \to X, \quad [x_1, \dots, x_n] \mapsto x_n$$

is the function sending a list to its last element. Finally, the coextension operation sends a function $f\colon GX \to Y$ to the function

$$f^*\colon GX \to GY, \quad [x_1, \dots, x_n] \mapsto [f([x_1]), f([x_1, x_2]), \dots, f([x_1, \dots, x_n])].$$

It remains to show that the equations

$$\varepsilon_X^* = \mathrm{id}_{GX}, \quad \varepsilon_Y \circ f^* = f, \quad (g \circ f^*)^* = g^* \circ f^*$$

are satisfied for all functions $f\colon GX \to Y$ and $g\colon GY \to Z$. So, we fix an arbitrary element $[x_1, \dots, x_n] \in GX$ and compute:

$$\begin{aligned}\varepsilon_X^*([x_1, \dots, x_n]) &= [\varepsilon_X([x_1]), \varepsilon_X([x_1, x_2]), \dots, \varepsilon_X([x_1, \dots, x_n])] \\ &= [x_1, \dots, x_n],\end{aligned}$$

$$\begin{aligned}(\varepsilon_Y \circ f^*)([x_1, \dots, x_n]) &= \varepsilon_Y([f([x_1]), \dots, f([x_1, \dots, x_n])]) \\ &= f([x_1, \dots, x_n]),\end{aligned}$$

$$\begin{aligned}(g \circ f^*)^*([x_1, \dots, x_n]) &= [(g \circ f^*)([x_1]), \dots, (g \circ f^*)([x_1, \dots, x_n])] \\ &= [g([f(x_1)]), \dots, g([f([x_1]), \dots, f([x_1, \dots, x_n])])] \\ &= g^*([f([x_1]), \dots, f([x_1, \dots, x_n])]) \\ &= (g^* \circ f^*)([x_1, \dots, x_n]).\end{aligned}$$

The associated comonad $(G, \delta, \varepsilon)$ is the one described in Exercise 1.36 (check this!). ▲

**Exercise 1.53.** Prove that comonads on a poset $P$ are precisely the *interior operators* on $P$, i.e. the monotone maps $g\colon P \to P$ that are

(1) *decreasing*, i.e. $gx \leq x$ for all $x \in P$, and
(2) *idempotent*, i.e. $g^2x = gx$ for all $x \in P$.

Either give a direct proof or use Exercise 1.48 combined with the principle of duality. ▲

## 1.6. Kleisli and Eilenberg–Moore categories for a comonad

Monads and comonads induce, respectively, categories of *algebras* and categories of *coalgebras*. For example, algebras for monads over **Set** essentially correspond to varieties of algebras in the sense of universal algebra.[4] In this section, we shall focus exclusively on coalgebras for comonads, as these are relevant in connection with finite model theory.

Given a comonad, there are two categories of coalgebras that are worth looking at: the *Kleisli category* is the "minimal" one and consists only of the *co-free* coalgebras, whereas the *Eilenberg–Moore category* is the "maximal" one and consists of all coalgebras. The former is easier to describe, but some constructions require working in the latter category. Let us start by introducing the Kleisli category of a comonad:

**Definition 1.54.** Let $G$ be a comonad (in Kleisli–Manes form) on a category $\mathcal{C}$. The *Kleisli category* of $G$, denoted by $\mathbf{K}(G)$, is defined as follows:

- $\mathrm{Ob}(\mathbf{K}(G)) = \mathrm{Ob}(\mathcal{C})$.

[4]To make the statement precise, one should restrict to those monads on **Set** that are *finitary*, i.e. that preserve so-called *directed colimits*.

- For all $A, B \in \mathrm{Ob}(\mathcal{C})$, $\mathbf{K}(G)(A, B) = \mathcal{C}(GA, B)$.

For any two arrows $f \in \mathbf{K}(G)(A, B)$ and $g \in \mathbf{K}(G)(B, C)$, their composite is defined as the following composition in $\mathcal{C}$:

$$GA \xrightarrow{f^*} GB \xrightarrow{g} C.$$

The identity $\mathrm{id}_A \in \mathbf{K}(G)(A, A)$ is the arrow $\varepsilon_A \colon GA \to A$ in $\mathcal{C}$.

**Exercise 1.55.** Verify that $\mathbf{K}(G)$ is a category. That is, the composition operation is associative and the identity arrows are identities for the composition operation. ▲

To get a better intuition of the Kleisli category, it is useful to compare it to the Eilenberg–Moore category.

**Definition 1.56.** Le $G$ be a comonad on a category $\mathcal{C}$. An *Eilenberg–Moore coalgebra* for $G$ is a pair $(A, \alpha)$ such that $A \in \mathrm{Ob}(\mathcal{C})$, $\alpha \in \mathcal{C}(A, GA)$, and the following diagrams commute.

$$\begin{tikzcd} A \arrow[r, "\alpha"] \arrow[dr, "\mathrm{id}_A"'] & GA \arrow[d, "\varepsilon_A"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "\alpha"] \arrow[d, "\alpha"'] & GA \arrow[d, "\delta_A"] \\ GA \arrow[r, "G\alpha"] & G^2A \end{tikzcd}$$

The arrow $\alpha$ is called the *structure map* of the coalgebra. A *morphism of Eilenberg–Moore coalgebras* $(A, \alpha) \to (B, \beta)$ is an arrow $f \in \mathcal{C}(A, B)$ compatible with the structures maps, i.e. making the following square commute.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] & B \arrow[d, "\beta"] \\ GA \arrow[r, "Gf"] & GB \end{tikzcd}$$

**Definition 1.57.** Let $G$ be a comonad on a category $\mathcal{C}$. The *Eilenberg–Moore category* of $G$, denoted by $\mathbf{EM}(G)$, consists of the Eilenberg–Moore coalgebras for $G$ and their morphisms. Compositions and identities are the obvious ones.

For any comonad $G$, there is a functor

$$\nabla \colon \mathbf{K}(G) \to \mathbf{EM}(G).$$

At the level of objects, $\nabla$ sends $A \in \mathrm{Ob}(\mathbf{K}(G)) = \mathrm{Ob}(\mathcal{C})$ to $(GA, \delta_A)$. With regards to morphisms, $\nabla$ assigns to an arrow $f \in \mathcal{C}(GA, B)$ the arrow $f^*$.

**Proposition 1.58.** *Let $G$ be a comonad on a category $\mathcal{C}$. Then*

$$\nabla \colon \mathbf{K}(G) \to \mathbf{EM}(G)$$

*is a full and faithful functor.*

PROOF. We first check that $\nabla$ is well defined. Let $A$ be an arbitrary object of $\mathbf{K}(G)$. To show that $(GA, \delta_A)$ is an Eilenberg–Moore coalgebra, we must prove that the following diagrams commute.

$$\begin{tikzcd} GA \arrow[r, "\delta_A"] \arrow[dr, "\mathrm{id}_{GA}"'] & G^2A \arrow[d, "\varepsilon_{GA}"] \\ & GA \end{tikzcd} \qquad \begin{tikzcd} GA \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] & G^2A \arrow[d, "\delta_{GA}"] \\ G^2A \arrow[r, "G\delta_A"] & G^3A \end{tikzcd}$$

In turn, this follows at once from the fact that $G$ is a comonad (see Definition 1.49).

Now, fix an arbitrary arrow $f$ in $\mathbf{K}(G)$, i.e. $f \in \mathcal{C}(GA, B)$. To see that $f^* \colon GA \to GB$ is a morphism of Eilenberg–Moore coalgebras, we must check that the following square commutes.

$$\begin{tikzcd} GA \arrow[r, "f^*"] \arrow[d, "\delta_A"'] & GB \arrow[d, "\delta_B"] \\ G^2A \arrow[r, "Gf^*"] & G^2B \end{tikzcd}$$

To this end, recall that

$$f^* = Gf \circ \delta_A \tag{1.2}$$

for all arrows $f \colon GA \to B$ (see Exercise 1.51). In particular, $\mathrm{id}^*_{GA} = \delta_A$. Thus,

$$\begin{aligned} Gf^* \circ \delta_A &= f^{**} && \text{Eq. (1.2)} \\ &= (\mathrm{id}_{GB} \circ f^*)^* \\ &= \mathrm{id}^*_{GB} \circ f^* && 3^{\text{rd}} \text{ equation in Eq. (1.1)} \\ &= \delta_B \circ f^*. \end{aligned}$$

The fact that $\nabla$ preserves compositions and identities is an immediate consequence of the third and first equations, respectively, for a comonad in Kleisli–Manes form (check the details!).

It remains to show that $\nabla$ is full and faithful. Faithfulness is clear: just observe that, for all $f, g \in \mathcal{C}(GA, B)$, $f^* = g^*$ implies

$$f = \varepsilon_B \circ f^* = \varepsilon_B \circ g^* = g$$

by virtue of the second equation for a comonad in Kleisli–Manes form. To establish fullness of $\nabla$, let $g \colon (GA, \delta_A) \to (GB, \delta_B)$ be a morphism of Eilenberg–Moore coalgebras. We have

$$\begin{aligned} (\varepsilon_B \circ g)^* &= G(\varepsilon_B \circ g) \circ \delta_A && \text{Eq. (1.2)} \\ &= G\varepsilon_B \circ Gg \circ \delta_A && G \text{ is a functor} \\ &= G\varepsilon_B \circ \delta_B \circ g && g \text{ coalgebra morphism} \\ &= \varepsilon^*_B \circ g, && \text{Eq. (1.2)} \end{aligned}$$

which coincides with $g$ by the first equation for comonads in Kleisli–Manes form. Note that $\varepsilon_B \circ g \in \mathcal{C}(GA, B) = \mathbf{K}(G)(A, B)$, hence $\nabla$ is full. □

Let us look at an example. Consider the comonad $G$ on **Set** defined in Example 1.52. Recall that, for any set $X$, $GX$ is the set of non-empty finite lists of elements of $X$, and a function $f\colon X \to Y$ is sent to the function

$$Gf\colon GX \to GY, \quad [x_1, \ldots, x_n] \mapsto [fx_1, \ldots, fx_n].$$

We claim that there is an isomorphism of categories

$$\mathbf{EM}(G) \cong \mathbf{Forests}.$$

Suppose $(X, \alpha)$ is an Eilenberg–Moore coalgebra for $G$. The set $GX$ carries a natural forest order, namely the prefix order. The commutativity of the first diagram in Definition 1.56 tells us that $\varepsilon_X \circ \alpha = \mathrm{id}_X$ and so the structure map $\alpha\colon X \to GX$ is injective. Hence the forest order on $GX$ induces a partial order on $X$ given by

$$x \leq y \iff \alpha(x) \leq \alpha(y)$$

for all $x, y \in X$. To show that this is a forest order on $X$, it suffices to show that the image of $\alpha$ is a *downwards closed* subset of $GX$ (why?). That is, any element of $GX$ that is below some element in the image of $\alpha$ is also in the image of $\alpha$.

Fix an arbitrary $x \in X$ and suppose that $\alpha(x)$ is of the form $[x_1, \ldots, x_n]$ (incidentally, note that $x_n = x$ because $\varepsilon_X \circ \alpha = \mathrm{id}_X$). The commutativity of the second diagram in Definition 1.56 implies that

$$[\alpha(x_1), \ldots, \alpha(x_n)] = [[x_1], \ldots, [x_1, \ldots, x_n]].$$

But any element that is below $\alpha(x)$ in the prefix order of $GX$ is of the form $[x_1, \ldots, x_j]$ for some $j \in \{1, \ldots, n\}$, and in view of the previous equation all these elements are in the image of $\alpha$.

Therefore, any structure map $\alpha\colon X \to GX$ defines a forest order on $X$. Further, the coalgebra morphisms preserve these forest orders. To see this, suppose that $f\colon (X, \alpha) \to (Y, \beta)$ is a morphism of coalgebras, i.e. the following square commutes.

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\alpha"'] & Y \arrow[d, "\beta"] \\ GX \arrow[r, "Gf"] & GY \end{tikzcd}$$

An element $x \in X$ is a root precisely when $\alpha(x) = [x]$ (why?), and similarly for elements of $Y$. Thus, $f$ preserves roots because $\alpha(x) = [x]$ entails

$$\beta(fx) = [fx].$$

To see that $f$ preserves the covering relation, suppose that $x, x' \in X$ satisfy $x \prec x'$ and $\alpha(x') = [x_1, \ldots, x_n]$. Observe that $x \prec x'$ if and only if $fx \prec fx'$ (why?), and so $\alpha(x) = [x_1, \ldots, x_{n-1}]$. The commutativity of the previous square yields

$$\beta(fx) = [fx_1, \ldots, fx_{n-1}] \quad \text{and} \quad \beta(fx') = [fx_1, \ldots, fx_n]$$

which shows that $\beta(fx) \prec \beta(fx')$.

It is straightforward to check that compositions and identities are preserved, hence this construction of a forest order from a structure map gives a functor

$$\mathbf{EM}(G) \to \mathbf{Forests}.$$

Conversely, given a forest order $(X, \leq)$, let $\alpha\colon X \to GX$ be the function that sends $x \in X$ to the list $[x_1, \ldots, x_n]$ of (non-strict) predecessors of $x$. That is, $x_1$ is a root and

$$x_1 \prec \cdots \prec x_n = x.$$

The first diagram in Definition 1.56 commutes simply because the last element of the list $\alpha(x)$ is $x$, and the second diagram commutes by definition of $\alpha$ (check this!). In other words, $(X, \alpha)$ is an Eilenberg–Moore coalgebra. Furthermore, any forest morphism $f\colon (X, \leq) \to (Y, \leq)$ preserves the corresponding structure maps because, for all $x \in X$, if the predecessors of $x$ are $x_1 \prec \cdots \prec x$ then the predecessors of $fx$ are $fx_1 \prec \cdots \prec fx$ (spell out the details of this argument!).

Again, it is clear that compositions and identities are preserved, so we obtain a functor

$$\mathbf{Forests} \to \mathbf{EM}(G).$$

**Exercise 1.59.** Prove that the functors $\mathbf{EM}(G) \leftrightarrows \mathbf{Forests}$ defined above are inverse to each other. ▲

What is the Kleisli category $\mathbf{K}(G)$ of the comonad $G$? By Proposition 1.58, combined with the previous discussion, it can be identified with the full subcategory of **Forests** defined by the forests of the form $GX$ (endowed with the prefix order).

## 1.7. Adjunctions

**Note.** *This section will appear in due course. Familiarity with adjunctions is not necessary for most of the material in the course, and this notion will be recalled in the lectures when needed.*

Adjunctions are a fundamental notion in category theory and provide a vast generalisation of free constructions such as free groups, free modules, etc. This is related to the fact that every adjunction induces a monad—and every monad arises from an adjunction (not a unique one, though).

Importantly, adjunctions have a symmetric nature. In fact, every adjunction induces not only a monad but also a comonad, and every comonad arises from some adjunction.