

# (Co)Monads

Alyson Mei

January 4, 2026

## Contents

<b>1</b>	<b>Monads</b>	<b>2</b>
1.1	Monoid object and Monad . . . . .	2
1.2	Kleisli category . . . . .	3
1.3	Monads and adjunctions (unfinished) . . . . .	4
1.4	Examples . . . . .	5
1.4.1	List monad . . . . .	5
1.4.2	Exception monad . . . . .	7
<b>2</b>	<b>Comonads (not started)</b>	<b>9</b>

### Conventions:

- Standard polymorphic notation is used freely when no ambiguity arises;
- All diagrams are assumed to commute unless stated otherwise.

# 1 Monads

## 1.1 Monoid object and Monad

**Definition 1.1** (Monoid object). Let  $(\mathcal{C}, \otimes, I)$  be a monoidal category. A monoid object in  $\mathcal{C}$  consists of:

- an object  $M \in \text{obj}(\mathcal{C})$ ,
- a multiplication morphism

$$\mu : M \otimes M \rightarrow M,$$

- a unit morphism

$$\eta : I \rightarrow M,$$

satisfying the unit and associativity diagrams:

$$\begin{array}{ccc} M & \xrightarrow{M \otimes \eta} & M \otimes M & \xleftarrow{\eta \otimes M} & M \\ & \searrow \text{id} & \downarrow \mu & \swarrow \text{id} & \\ & & M & & \end{array} \qquad \begin{array}{ccc} M^{\otimes 3} & \xrightarrow{\mu \otimes M} & M^{\otimes 2} \\ M \otimes \mu \downarrow & & \downarrow \mu \\ M^{\otimes 2} & \xrightarrow{\mu} & M \end{array} .$$

**Example 1.2.**

- Categories with finite products  $(\mathcal{C}, \times, 1)$  are monoidal:
  - In  $\mathbf{Set}$ , monoid objects are ordinary monoids;
  - In  $\mathbf{Vect}_{\mathbf{k}}$ , monoid objects are associative unital algebras over a field  $\mathbf{k}$ ;
- The category of endofunctors  $([\mathcal{C}, \mathcal{C}], \circ, \text{id})$  is monoidal; monoid objects in this category are called monads.

**Definition 1.3** (Monad). Monad is a triple

$$(M : \mathcal{C} \rightarrow \mathcal{C}, \eta : \text{id}_{\mathcal{C}} \rightarrow M, \mu : M^2 \rightarrow M),$$

satisfying the unit and associativity diagrams:

$$\begin{array}{ccc} M & \xrightarrow{\eta \circ M} & M^2 & \xleftarrow{M \circ \eta} & M \\ & \searrow \text{id}_M & \downarrow \mu & \swarrow \text{id}_M & \\ & & M & & \end{array} \qquad \begin{array}{ccc} M^3 & \xrightarrow{M \circ \mu} & M^2 \\ \mu \circ M \downarrow & & \downarrow \mu \\ M^2 & \xrightarrow{\mu} & M \end{array} .$$

## 1.2 Kleisli category

**Definition 1.4** (Kleisli category). Let  $(T, \eta, \mu)$  be a monad on  $\mathcal{C}$ . The Kleisli category  $\text{Kl}(T)$  of the monad  $T$  is defined as follows:

- Objects of  $\text{Kl}(T)$  are the same as in  $\mathcal{C}$ ,
- For  $A, B \in \text{obj}(\mathcal{C})$ , a morphism  $f^\bullet : A \rightarrow B$  in  $\text{Kl}(T)$  is a morphism  $f : A \rightarrow TB$  in  $\mathcal{C}$ ,
- Composition:

$$\begin{array}{ccccc}
 A & B & TB & \xleftarrow{f} & A \\
 f \downarrow & g \downarrow & Tg \downarrow & & \downarrow \mu_C \circ Tg \circ f \\
 TB & TC & T^2C & \xrightarrow{\mu_C} & TC
 \end{array}
 \quad
 \begin{array}{ccc}
 A & & \\
 \downarrow f^\bullet \circ g^\bullet \in \text{Kl}(T) & , & \\
 C & & 
 \end{array}$$

- Identity morphism  $\text{id}_A^\bullet$  in  $\text{Kl}(T)$  is a morphism  $\eta_A : A \rightarrow TA$  in  $\mathcal{C}$  :

$$\begin{array}{ccccc}
 A & \xrightarrow{\eta_A} & TA & & TA & \xleftarrow{\eta_A} & A & & A \\
 f \downarrow & & \downarrow Tf & & Tf \downarrow & & \downarrow \mu_B \circ \eta_{TB} \circ f = f & & \downarrow f \circ \text{id}_A^\bullet \\
 TB & \xrightarrow{\eta_{TB}} & T^2B & & T^2B & \xrightarrow{\mu_B} & TB & & B
 \end{array}$$
  

$$\begin{array}{ccccc}
 & & TA & \xleftarrow{g} & C & & A \\
 & & T\eta_A \downarrow & & \downarrow \mu_A \circ T\eta_A \circ g = g & & \downarrow \text{id}_A^\bullet \circ g \\
 & & T^2A & \xrightarrow{\mu_A} & TA & & B
 \end{array}$$

### 1.3 Monads and adjunctions (unfinished)

**Theorem 1.5** (Monads  $\approx$  adjunctions). Every adjunction gives rise to a monad, and every monad arises (up to equivalence) from an adjunction.

**From adjunction to monad.**

Let  $F : \mathcal{C} \rightarrow \mathcal{D}$ ,  $G : \mathcal{D} \rightarrow \mathcal{C}$ , and  $F$  be the left adjoint to  $G$ :

$$\eta : \text{id}_{\mathcal{C}} \rightarrow GF, \quad \varepsilon : FG \rightarrow \text{id}_{\mathcal{D}},$$

$$\begin{array}{ccc} F & \xrightarrow{F\eta} & FGF \\ & \searrow \text{id} & \downarrow \varepsilon F \\ & & F \end{array} \quad \begin{array}{ccc} GFG & \xleftarrow{\eta G} & G \\ G\varepsilon \downarrow & & \swarrow \text{id} \\ & & G \end{array} .$$

We define the underlying functor  $M$  of the monad as

$$M := GF : \mathcal{C} \rightarrow \mathcal{C}.$$

Its unit is exactly  $\eta$ . We can define the multiplication  $\mu : M^2 \rightarrow M$  using the counit:

$$\mu := G\varepsilon F.$$

The  $\varepsilon$  here essentially collapses the forget-free structure from  $GF$ .

The unit and associativity diagrams:

$$\begin{array}{ccc} GF & \xrightarrow{\eta GF} & GF GF \\ (G\varepsilon \circ \eta G)F = \text{id} \searrow & & \downarrow \mu = G\varepsilon F \\ & & GF \end{array} \quad \begin{array}{ccc} GFGF & \xleftarrow{GF\eta} & GF \\ \mu = G\varepsilon F \downarrow & & \swarrow G(\varepsilon F \circ F\eta) = \text{id} \\ & & GF \end{array} ,$$

$$\begin{array}{ccc} GFGFGF & \xrightarrow{G\varepsilon FGF} & GFGF \\ GFG\varepsilon F \downarrow & & \downarrow G\varepsilon F \\ GFGF & \xrightarrow{G\varepsilon F} & GF \end{array} .$$

## 1.4 Examples

### 1.4.1 List monad

#### Definition and monad structure.

Consider the functor  $\text{List} : \mathcal{Set} \rightarrow \mathcal{Set}$ , which sends a set to the set of all finite lists of its elements:

$$\begin{array}{ccc} A & \text{List } A & [a_1, a_2, \dots, a_n] \\ \downarrow f & \downarrow \text{List } f & \downarrow \text{List } f \\ B & \text{List } B & [f(a_1), f(a_2), \dots, f(a_n)] \end{array} .$$

From a categorical perspective, the functor  $\text{List}$  can be presented as follows:

$$\begin{array}{ccc} \mathcal{Set} & \xrightarrow{\Delta} & \mathcal{Set}^{\mathbb{N}} \\ \text{List} \downarrow & & \downarrow \prod_{n \in \mathbb{N}} \times^n \\ \mathcal{Set} & \xleftarrow{\prod_{n \in \mathbb{N}}} & \mathcal{Set}^{\mathbb{N}} \end{array} \quad \begin{array}{ccc} A & \xrightarrow{\quad} & A^{\mathbb{N}} \\ \text{List}(A) \downarrow & & \downarrow \\ \coprod_{n \in \mathbb{N}} A^n & \xleftarrow{\quad} & (A^n)_{n \in \mathbb{N}} \end{array} .$$

This presentation gives rise to the API:

- The collection of constructors

$$[\ ]_{n,A} := A^n \xrightarrow{i_n} \text{List } A := \lambda a_1 \dots a_n. [a_1, \dots, a_n],$$

in particular:

- The single-element list constructor

$$\eta_A : A \rightarrow \text{List } A := \lambda a. [a],$$

- The empty list constructor

$$[\ ]_0 : 1 \rightarrow \text{List } A := \lambda a. [];$$

- Length of a list via the power of  $A$  in the coproduct:

$$\text{len}_A : \text{List } A \rightarrow \mathbb{N};$$

- List concatenation via the natural isomorphism  $A^n \times A^m \cong A^{n+m}$ :

$$\begin{array}{ccc} \text{id} \times \text{id} & \xrightarrow{i \times i} & \text{List} \times \text{List} \\ \simeq \downarrow & & \downarrow + \\ \text{id} & \xrightarrow{i} & \text{List} \end{array} \quad \begin{array}{ccc} A^m \times A^n & \xrightarrow{i_m \times i_n} & \text{List } A \times \text{List } A \\ \cong \downarrow & & \downarrow +_A \\ A^{m+n} & \xrightarrow{i_{m+n}} & \text{List } A \end{array} ,$$

$$+_A : \text{List } A \times \text{List } A \rightarrow \text{List } A := \lambda l m. [l_0, \dots, l_{\text{len}(l)-1}, m_0, \dots, m_{\text{len}(m)-1}],$$

$$+_A : (\text{List } A)^n \rightarrow \text{List } A := \lambda l_0 \dots l_n. l_0 + \dots + l_n;$$

– List destructors:

$$\mu_A : \text{List}(\text{List } A) \rightarrow \text{List}(A) := \lambda l. \sum_{i=0}^{\text{len}(l)-1} l_i,$$

$$\mu_A : \text{List}^n A \rightarrow \text{List } A := \mu_A^{\circ n}.$$

First of all, this API provides an internal monoid structure on lists: for a fixed set  $A$ , we have a monoid object

$$(\text{List } A, [], +_A)$$

in  $\mathcal{Set}$ . The corresponding diagrams are straightforward in this case.

Second, it provides a monad structure on  $\text{List}$ :

$$(\text{List}, \eta : \text{id} \rightarrow \text{List}, \mu : \text{List}^2 \rightarrow \text{List}).$$

The corresponding diagrams:

$$\begin{array}{ccc} \text{List } A & \xrightarrow{\eta_{\text{List } A}} & \text{List}(\text{List } A) \\ & \searrow \text{id} & \downarrow \mu_A \\ & & \text{List } A \end{array} \quad , \quad \begin{array}{ccc} l & \xrightarrow{\quad} & [l] \\ & \searrow & \downarrow \\ & & l = [l]_0 \end{array}$$

$$\begin{array}{ccc} \text{List}(\text{List } A) & \xleftarrow{\text{List } \eta_A} & \text{List } A \\ \downarrow \mu_A & \swarrow \text{id} & \\ \text{List } A & & \end{array} \quad , \quad \begin{array}{ccc} [[l_0], \dots, [l_{\text{len}(l)-1}]] & \xleftarrow{\quad} & l \\ \downarrow & \swarrow & \\ l = \sum_{i=0}^{\text{len}(l)-1} [l_i] & & \end{array}$$

$$\begin{array}{ccc} \text{List}^3 A & \xrightarrow{\text{List } \mu_A} & \text{List}^2 A \\ \downarrow \mu_{\text{List } A} & & \downarrow \mu_A \\ \text{List}^2 A & \xrightarrow{\mu_A} & \text{List } A \end{array} \quad , \quad \begin{array}{ccc} l & \xrightarrow{\quad} & [\sum_{j=0}^{\text{len}(l_0)-1} (l_0)_j, \dots, \sum_{j=0}^{\text{len}(l_{\text{len}(l)-1}-1)} (l_{\text{len}(l)-1})_j] \\ \downarrow & & \downarrow \\ \sum_{i=0}^{\text{len}(l)-1} l_i & \xrightarrow{\quad} & \sum_{i,j} (l_i)_j \end{array} .$$

**Kleisli category**  $\text{Kl}(\text{List})$ .

One of the possible interpretations of  $f : A \rightarrow \text{List } B$  is a non-deterministic calculation:  $a$  is mapped by  $f$  to the list of all possible results. The composition corresponds to getting all possible results from sequential non-deterministic calculations:

$$\begin{array}{ccc} A & \xrightarrow{f^\bullet} & B \\ & \searrow g^\bullet \circ f^\bullet & \downarrow g^\bullet \\ & & C \end{array} \quad , \quad \begin{array}{ccc} [b_0, \dots, b_n] & \xleftarrow{\quad f \quad} & a \\ Tg = [g_0(b_0), \dots, g_n(b_n)] \downarrow & & \downarrow \\ [[b_0^0, b_1^0, \dots, b_{m_0}^0], \dots, [b_0^n, b_1^n, \dots, b_{m_n}^n]] & \xrightarrow{\mu_B = \sum_i g_i(b_i)} & [b_0^0, b_1^0, \dots, b_{m_0}^0, \dots, b_0^n, b_1^n, \dots, b_{m_n}^n] \end{array} .$$

### 1.4.2 Exception monad

#### Definition and monad structure.

Consider a type  $E$  of exceptions (error values) and a functor  $\text{Exception} : \mathcal{Set} \rightarrow \mathcal{Set}$ , which sends a type to its coproduct with the exception type:

$$\begin{array}{ccc} A & & A + E \\ f \downarrow & & \downarrow f + \text{id} \\ B & & B + E \end{array}$$

The monad structure

$$(\text{Exception}, \eta : \text{id} \rightarrow \text{Exception}, \mu : \text{Exception}^2 \rightarrow \text{Exception}).$$

is defined as follows:

- Unit represents successful calculation:

$$\eta_A := i_1 : A \rightarrow A + E,$$

- Multiplication represents the error propagation:

$$\mu_A : A + E + E \rightarrow A + E = \text{id}_A + \nabla_{E,E} = \nabla_{A+E,E},$$

where  $\nabla$  is the codiagonal map.

The diagrams are obvious on  $\mathcal{Set}$ :

$$\begin{array}{ccc} A + E & \xrightarrow{\eta_{A+E} = i_1, A+E} & (A + E) + E \\ & \searrow \text{id} & \downarrow \nabla_{A+E,E} \\ & & A + E \end{array} \qquad \begin{array}{ccc} a_A \parallel \text{error}_E & \longmapsto & (a \parallel \text{error})_{A+E} \\ & \searrow & \downarrow \\ & & a_A \parallel \text{error}_E \end{array},$$

$$\begin{array}{ccc} (A + E) + E & \xleftarrow{\eta_{A+E} = i_1, A+E} & A + E \\ \nabla_{A+E,E} \downarrow & \swarrow \text{id} & \\ A + E & & \end{array} \qquad \begin{array}{ccc} a_{A+E} \parallel \text{error}_E & \longleftarrow & a_A \parallel \text{error}_E \\ \downarrow & \swarrow & \\ a_A \parallel \text{error}_E & & \end{array},$$

$$\begin{array}{ccc} ((A + E) + E) + E & \xrightarrow{\nabla_{A+E,E} + \text{id}_E} & (A + E) + E \\ \nabla_{(A+E)+E,E} \downarrow & & \downarrow \nabla_{A+E,E} \\ (A + E) + E & \xrightarrow{\nabla_{A+E,E}} & A + E \end{array} \qquad \begin{array}{ccc} a_A \parallel \text{error}_{E^{(1)}+E^{(2)}+E^{(3)}} & \longmapsto & a_A \parallel \text{error}_{E^{(2)}+E^{(3)}} \\ \downarrow & & \downarrow \\ a_A \parallel \text{error}_{E^{(1)}+E^{(2)}} & \longmapsto & a_A \parallel \text{error}_E \end{array}.$$

Here  $a_A \parallel b_B$  informally denotes the element of  $A + B$ , subscripts annotate the domains.

**Kleisli category**  $\text{Kl}(\text{Exception})$ .

The Kleisli composition for the Exception monad tracks the source of the possible error:

$$\begin{array}{ccc}
 A & \xrightarrow{f^\bullet} & B \\
 \searrow g^\bullet \circ f^\bullet & & \downarrow g^\bullet \\
 & & C
 \end{array}
 \qquad
 \begin{array}{ccc}
 f(a)_B \parallel \text{error}_E & \xleftarrow{f} & a_A \\
 \downarrow g + \text{id}_E & & \downarrow \\
 (gf(a)_C \parallel \text{error})_{C+E} \parallel \text{error}_E & \xrightarrow{\mu_C} & gf(a)_C \parallel \text{error}_E
 \end{array}
 .$$



## 2 Comonads (not started)