

(Co)Monads

Alyson Mei

January 4, 2026

Contents

1	Monads	2
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
2	Comonads (not started)	9

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:

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

Example 1.2.

- Categories with finite products $(\mathcal{C}, \times, 1)$ are monoidal:
 - In $\mathcal{S}et$, monoid objects are ordinary monoids;
 - In $\text{Vect}_{\mathbf{k}}$, 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:

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

1.2 Kleisli category

Definition 1.4 (Kleisli category). Let (T, η, μ) 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:

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

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

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

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

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 :

$$\begin{array}{ccc} \eta : \text{id}_{\mathcal{C}} \rightarrow GF, \varepsilon : FG \rightarrow \text{id}_{\mathcal{D}}, & & \\ F \xrightarrow{F\eta} FGF \quad GFG \xleftarrow{\eta G} G & \downarrow \varepsilon F \quad \downarrow G\varepsilon & \\ \text{id} \searrow \quad \downarrow \quad \swarrow \text{id} & F \quad G & \end{array} .$$

We define the underlying functor M of the monad as

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

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

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

The ε here essentially collapses the forget-free structure from GF .

The unit and associativity diagrams:

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

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

1.4 Examples

1.4.1 List monad

Definition and monad structure.

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

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

From a categorical perspective, the functor List can be presented as follows:

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

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:

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

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

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

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

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

- 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 $\mathcal{S}et$. The corresponding diagrams are straightforward in this case.

Second, it provides a monad structure on List:

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

The corresponding diagrams:

$$\begin{array}{ccc}
\text{List } A & \xrightarrow{\eta_{\text{List } A}} & \text{List}(\text{List } A) \\
& \searrow \text{id} & \downarrow \mu_A \\
& & \text{List } A
\end{array}
\quad
\begin{array}{ccc}
l & \xrightarrow{\quad} & [l] \\
\swarrow & & \downarrow \\
l = [l]_0 & &
\end{array} , \\[10pt]
\begin{array}{ccc}
\text{List}(\text{List } A) & \xleftarrow{\text{List } \eta_A} & \text{List } A \\
\downarrow \mu_A & \nearrow \text{id} & \\
\text{List } A & &
\end{array}
\quad
\begin{array}{ccc}
[[l]_0], \dots, [l]_{\text{len}(l)-1}] & \xleftarrow{\quad} & l \\
\downarrow & \swarrow & \\
l = \sum_{i=0}^{\text{len}(l)-1} [l]_i & &
\end{array} , \\[10pt]
\begin{array}{ccc}
\text{List}^3 A & \xrightarrow{\text{List } \mu_A} & \text{List}^2 A \\
\downarrow \mu_{\text{List } A} & & \downarrow \mu_A \\
\text{List}^2 A & \xrightarrow[\mu_A]{\quad} & \text{List } A
\end{array}
\quad
\begin{array}{ccc}
l & \xrightarrow{\quad} & [\sum_{j=0}^{\text{len}(l_0)-1} (l_0)_j, \dots, \sum_{j=0}^{\text{len}(l_{\text{len}(l)-1}-1)} (l_{\text{len}(l)-1})_j] \\
\downarrow & & \downarrow \\
\sum_{i=0}^{\text{len}(l)-1} l_i & \xrightarrow{\quad} & \sum_{i,j} (l_i)_j
\end{array} .$$

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:

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

1.4.2 Exception monad

Definition and monad structure.

Consider a type E of exceptions (error values) and a functor $\text{Exception} : \mathcal{S}et \rightarrow \mathcal{S}et$, 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 ∇ is the codiagonal map.

The diagrams are obvious on $\mathcal{S}et$:

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

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

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

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:

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

2 Comonads (not started)