Definition (Monad). A monad in a given 2-category is comprised of the following:

- 0-cell $C$
- 1-cell $m : C \to C$ (generally referred to as the monad)
- 2-cells $\eta : id_C \Rightarrow m : C \Rightarrow C$ (called the unit) and $\mu : m ; m \Rightarrow m : C \to C$ (called the join)
- such that the following identity and associativity laws hold:

\[
\begin{align*}
\eta \circ m &= \eta \\
(m \circ m) \circ m &= m \circ (m \circ m)
\end{align*}
\]

Remark. In terms of string diagrams, the identity and associative laws are formulated as follows:

Remark. Given a 2-category, one can construct a multicategory whose objects are the 1-cells of the multicategory and whose morphisms are 2-cells from the composition of the inputs to the output. A monad is an internal monoid of that multicategory.

Theorem. For any monad $\langle C, m, \eta, \mu \rangle$ and $n : \mathbb{N}$, all 2-cells from $m^n$ to $m$ built from $\eta$, $\mu$, and identities are equal.
Example. The functor \( 0 \) serves as a basis for a monad on \( \text{Set} \) in the 2-category \( \text{Set} \). The unit is \( \text{some} \), and the join \( \mu_A : \text{O}(0(A)) \to \text{O}(A) \) maps \( \text{some} (\text{some}(a)) \) to \( \text{some}(a) \) and maps \( \text{some}(\text{none}) \) and \( \text{none} \) to \( \text{none} \).

Example. \((\text{Set}, \text{L}, \text{singleton}, \text{flatten})\) is a monad in \( \text{Cat} \). Similarly, \((\text{Set}, \text{P}, \lambda x. \{x\}, \cup)\) is also a monad in \( \text{Set} \).

There is also a monad for the functor \( \text{L} : \text{Set} \to \text{Set} \) that maps sets \( A \) to the set of finite multisets/bags of \( A \), i.e. finite collections of \( A \) elements in which duplicates matter but order does not. And there is a monad for the functor \( \text{F} : \text{Set} \to \text{Set} \) that maps \( A \) to the set of finite subsets of \( A \).

Example. Given a set \( C \), the functor \( C \to : \text{Set} \to \text{Set} \) is a monad. The unit is the natural transformation mapping \( a \in A \) to \( (\lambda c \in C. a) \in C \to A \). The join is the natural transformation mapping \( f \in C \to (C \to A) \) to \( (\lambda e \in C. f(e)(e)) \in C \to A \).

Example. Given a set \( S \), the functor \( S \to S \times \cdots : \text{Set} \to \text{Set} \) is a monad. The unit is the natural transformation mapping \( a \in A \) to \( (\lambda s \in S. (s, a)) \in S \to S \times A \). The join is the natural transformation mapping \( f \in S \to (S \times (S \to S \times A)) \) to \( (\lambda s. \pi_2(f(s))((\pi_1(f(s)))) \in S \to S \times A \).

Example. Given a monoid \( \langle M, e, * \rangle\), the functor \( M \times : \text{Set} \to \text{Set} \) is a monad. The unit is the natural transformation mapping \( a \in A \) to \( (e, a) \in M \times A \). The join is the natural transformation mapping \( \langle m, \langle m', a \rangle \rangle \in M \times (M \times A) \) to \( \langle m * m', a \rangle \in M \times A \).

Example. Given a graph \( \langle V, E, s, t \rangle \) one can define the set of paths as alternating lists of vertices and edges \( (v_0, e_0, v_1, e_1, \ldots, v_n) \) with the property that \( s(e_i) = v_i \) and \( t(e_i) = v_{i+1} \) for all indices \( i \). The source of such a path is \( v_0 \) and the target is \( v_n \). Thus we have a graph \( \langle V, \text{Path}(E), s_{\text{Path}}, t_{\text{Path}} \rangle \). This Path construction extends to a monad. For both the unit and join, the function on vertices on simply the identity. As for edges, the unit maps an edge \( e \) to the path \( (s(e), e, t(e)) \), and the join essentially flattens paths of paths.

**Definition** (Monad Morphism). An (oplax) monad morphism from \( \langle C_1, m_1, \eta_1, \mu_1 \rangle \) to \( \langle C_2, m_2, \eta_2, \mu_2 \rangle \) is a 1-cell \( f : C_1 \to C_2 \) and a 2-cell \( \alpha : m_1 ; f \Rightarrow f ; m_2 \) such that

\[
\begin{align*}
\text{C}_2 & \xrightarrow{m_2} \text{C}_2 \\
\text{C}_1 & \xrightarrow{m_1} \text{C}_1
\end{align*}
\]

\[\alpha\]

\[
\begin{align*}
\text{C}_1 & \xrightarrow{\eta_1} \text{C}_1 \\
\text{C}_2 & \xrightarrow{\eta_2} \text{C}_2
\end{align*}
\]

\[f = f\]

\[
\begin{align*}
\text{C}_1 & \xrightarrow{\mu_1} \text{C}_1 \\
\text{C}_2 & \xrightarrow{\mu_2} \text{C}_2
\end{align*}
\]

\[f = f\]

\[
\begin{align*}
\text{C}_1 & \xrightarrow{f} \text{C}_2 \\
\text{C}_1 & \xrightarrow{f} \text{C}_2
\end{align*}
\]

\[f\] and \[f\]

Example. The obvious natural transformations from \( \text{L} \) to \( \text{F} \) to \( \text{P} \) are all monad morphisms where the 1-cell \( f \) is the identity functor of \( \text{Set} \).

Example. The functor \( \pi_E : \text{Graph} \to \text{Set} \) has the property that \( \text{Path} ; \pi_E \) equals \( \pi_E ; \text{L} \). This 1-cell \( \pi_E \) in fact forms a monad morphism from \( \text{Path} \) to \( \text{L} \) where the 2-cell is the identity 2-cell.

**Definition** (Comonad). A comonad in a 2-category is a 0-cell \( C \), a 1-cell \( c : C \to C \) (typically called the comonad), and 2-cells \( \varepsilon : m \Rightarrow id_C \) (often called the counit) and \( \delta : m \Rightarrow m ; m \) (often called the cojoin or comultiplication) satisfying equalities dual to the identity and associativity laws of monads.

Example. Given any set \( C \), the functor \( C \times : \text{Set} \to \text{Set} \) is a comonad on \( \text{Set} \) in \( \text{Cat} \). The counit is the natural transformation mapping \( \langle c, a \rangle \in C \times A \) to \( a \in A \). The cojoin is the natural transformation mapping \( \langle c, \langle c, a \rangle \rangle \in C \times (C \times A) \) to \( \langle c, a \rangle \in C \times C \times A \).

Example. Given a monoid \( \langle M, e, * \rangle \), the functor \( M \to : \text{Set} \to \text{Set} \) is a comonad. The counit is the natural transformation mapping \( f \in M \to A \) to \( f(e) \in A \). The cojoin is the natural transformation mapping \( f \in M \to A \) to \( \lambda m \in M. \lambda m' \in M. s(m * m') \in M \to (M \to A) \). When the monoid is \( \langle N, 0, + \rangle \), this is known as the stream comonad.

**Definition** (Comonad Morphism). Just as a comonad in a 2-category \( C \) coincides with a monad in \( C^{co} \), a comonad morphism in \( C \) coincides with a monad morphism in \( C^{co} \).