Lambda-Calculus & Categories | 2

4 january 2021

Monads

I

Monads

1. every adjunction induces a monad

2. every monad on a category $\mathcal{C}$ induce an adjunction between:

    a. the category $\mathcal{C}$ and the category of Eilenberg-Moore algebras of the monad (or simply T-algebras)

    b. the category $\mathcal{C}$ and the Kleisli category of the monad T.

(can be seen as the category of free T-algebras)
(presented)

(0) definition of a monad.

<u>def</u>. a monad (T, μ, η) on a category $\mathscr{C}$ is a triple consisting of:

dim 1 a functor $T : \mathscr{C} \longrightarrow \mathscr{C}$

dim 2 a pair of natural transformations

$$\mu : T \circ T \Longrightarrow T$$

$$\eta : \mathrm{Id}_{\mathcal{C}} \Longrightarrow T$$

in pasting diagram:

$$\begin{tikzcd}
 & \mathcal{C} \arrow[dr, "T"] \arrow[d, phantom, "\Downarrow\mu"] & \\
\mathcal{C} \arrow[ur, "T"] \arrow[rr, "T"'] & {} & \mathcal{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{C} \arrow[rr, bend left=50, "\mathrm{Id}"] \arrow[rr, bend right=50, "T"'] \arrow[rr, phantom, "\Downarrow\eta"] & & \mathcal{C}
\end{tikzcd}$$

dim 3 satisfying three equations:

a associativity

$$\begin{tikzcd}
 & \mathcal{C} \arrow[r, "T"] \arrow[d, phantom, "\Downarrow\mu"] & \mathcal{C} \arrow[dr, "T"] \arrow[d, phantom, "\Downarrow\mu"] & \\
\mathcal{C} \arrow[ur, "T"] \arrow[urr, "T"'] \arrow[rrr, bend right, "T"'] & {} & {} & \mathcal{C}
\end{tikzcd}
\overset{@}{=}
\begin{tikzcd}
 & \mathcal{C} \arrow[r, "T"] \arrow[drr, "T"'] \arrow[d, phantom, "\Downarrow\mu"] & \mathcal{C} \arrow[dr, "T"] \arrow[d, phantom, "\Downarrow\mu"] & \\
\mathcal{C} \arrow[ur, "T"] \arrow[rrr, bend right, "T"'] & {} & {} & \mathcal{C}
\end{tikzcd}$$

equivalently, the diagram below commutes

$$
\begin{tikzcd}
T \circ T \circ T \arrow[r, Rightarrow, "\mu \circ T"] \arrow[d, Rightarrow, "T \circ \mu"'] & T \circ T \arrow[d, Rightarrow, "\mu"] \\
T \circ T \arrow[r, Rightarrow, "\mu"'] & T
\end{tikzcd}
$$

in the category $\mathrm{Nat}(\mathcal{C}, \mathcal{C})$ of functors from $\mathcal{C}$ to $\mathcal{C}$

equivalently,

for all objects A of the category $\mathcal{C}$

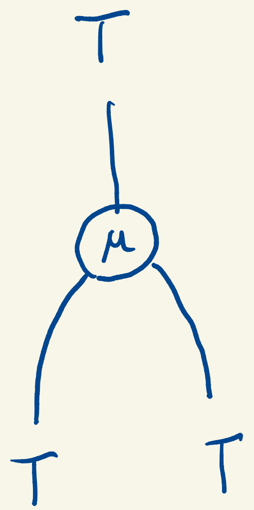
the diagram below commutes:

$$
\begin{tikzcd}
TTTA \arrow[r, "\mu_{TA}"] \arrow[d, "T\mu_A"'] & TTA \arrow[d, "\mu_A"] \\
TTA \arrow[r, "\mu_A"] & TA
\end{tikzcd}
$$

in the category $\mathcal{C}$.

In string diagrams:

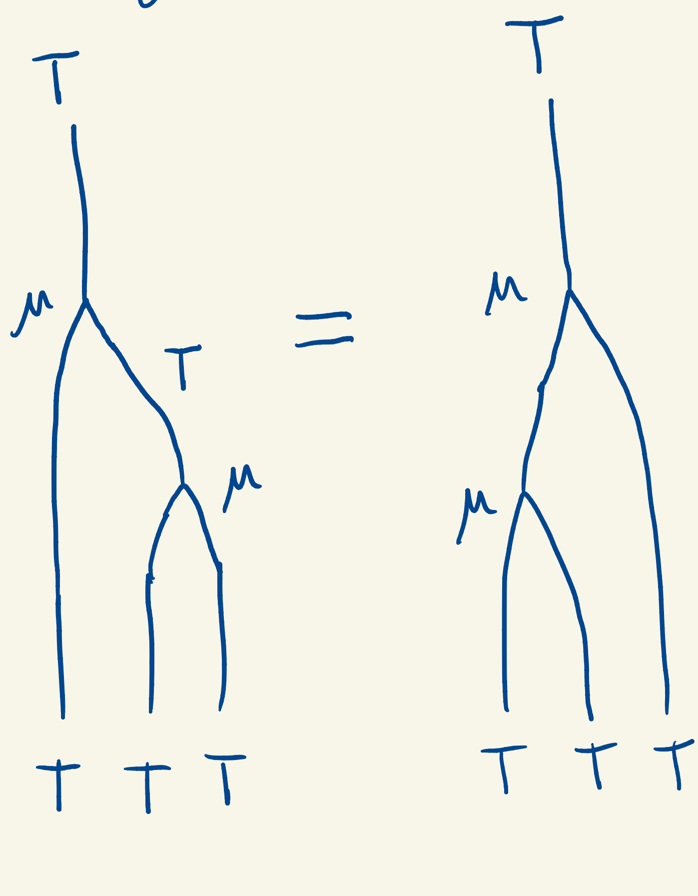
the multiplication is depicted as



$$T \circ T \xRightarrow{\mu} T$$

associativity says that:

$$T \circ T \circ T \xRightarrow{T \circ \mu} T \circ T \xRightarrow{\mu} T$$



$$T \circ T \circ T \xRightarrow{\mu \circ T} T \circ T \xRightarrow{\mu} T$$

(b) two neutrality equations

$$
\begin{tikzcd}
 & \mathcal{C} \arrow[dr, "T"] & \\
\mathcal{C} \arrow[ur, bend left, "Id"] \arrow[ur, bend right, "T"'] \arrow[ur, phantom, "\Downarrow\eta"] \arrow[rr, bend right, "T"'] \arrow[rr, phantom, "\Downarrow\mu"] & & \mathcal{C}
\end{tikzcd}
\overset{(b)}{=}
\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "T"] \arrow[r, bend right, "T"'] \arrow[r, phantom, "\Downarrow Id_T"] & \mathcal{C}
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & \mathcal{C} \arrow[dr, bend left, "Id"] \arrow[dr, bend right, "T"'] \arrow[dr, phantom, "\Downarrow\eta"] & \\
\mathcal{C} \arrow[ur, "T"] \arrow[rr, bend right, "T"'] \arrow[rr, phantom, "\mu\Downarrow"] & & \mathcal{C}
\end{tikzcd}
\overset{(b')}{=}
\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "T"] \arrow[r, bend right, "T"'] \arrow[r, phantom, "\Downarrow Id_T"] & \mathcal{C}
\end{tikzcd}
$$

equivalently, we have the commutative diagrams of natural transformations:

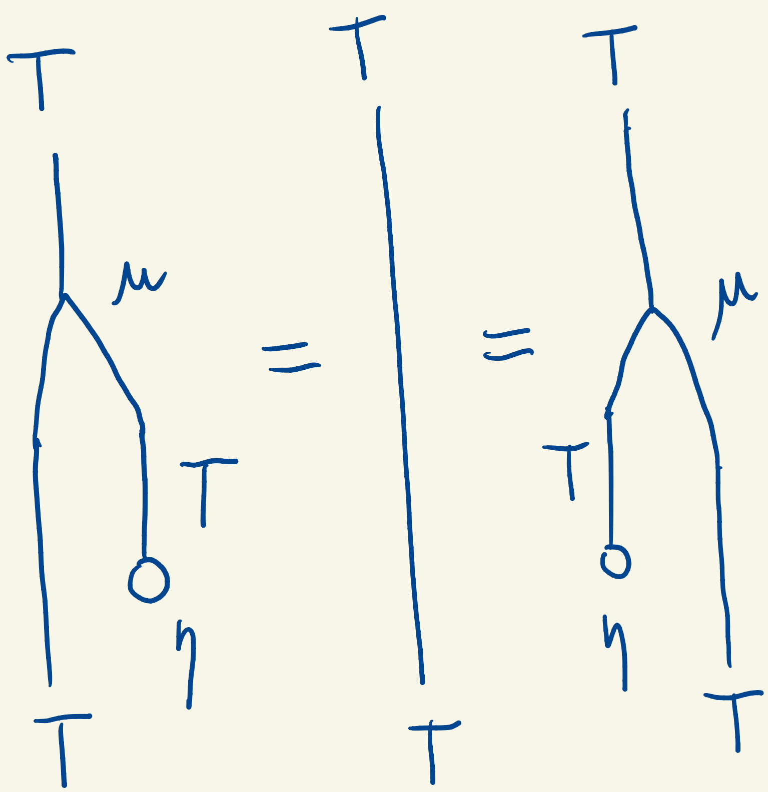
$$
\begin{tikzcd}
 & T \circ T \arrow[dr, Rightarrow, "\mu"] & \\
T \arrow[ur, Rightarrow, "T \circ \eta"] \arrow[rr, Rightarrow, bend right, "Id_T"'] & & T
\end{tikzcd}
\qquad
\begin{tikzcd}
 & T \circ T \arrow[dr, Rightarrow, "\mu"] & \\
T \arrow[ur, Rightarrow, "\eta \circ T"] \arrow[rr, Rightarrow, bend right, "Id_T"'] & & T
\end{tikzcd}
$$

equivalently, for all objects A of the category $\mathcal{C}$, we have commutative diagrams

$$\begin{tikzcd} & TTA \arrow[dr, "\mu_A"] & \\ TA \arrow[ur, "T\eta_A"] \arrow[rr, "\mathrm{id}_{TA}"'] & & TA \end{tikzcd} \qquad \begin{tikzcd} & TTA \arrow[dr, "\mu_A"] & \\ TA \arrow[ur, "\eta_{TA}"] \arrow[rr, "\mathrm{id}_{TA}"'] & & TA \end{tikzcd}$$

equivalently, we have the equation

$$\begin{array}{c} T \\ \Uparrow \mu \\ T\circ T \\ \Uparrow T\circ\eta \\ T \end{array}$$



$$\begin{array}{c} T \\ \Uparrow \mu \\ T\circ T \\ \Uparrow \eta\circ T \\ T \end{array}$$

Example.

(1) the free monoid $\boxed{\text{monad}}$

$$T : \mathbf{Set} \longrightarrow \mathbf{Set}$$

which transports every set A to the set $TA = A^*$ of finite words on the alphabet A,

$$TA = A^* = \coprod_{n \in \mathbb{N}} A^n$$

where $A^n = \underbrace{A \times \ldots \times A}_{n \text{ times}}$

is the set of words of length n.

on the alphabet A

the functor

$$T : Set \longrightarrow Set$$

transports every function

$$f : A \longrightarrow B$$

to the function

$$\begin{array}{ccc} Tf : TA & \longrightarrow & TB \\ \parallel & & \parallel \\ A^* & & B^* \end{array}$$

defined as follows:

$$Tf : [a_1 \ldots a_k] \longmapsto [fa_1 \ldots fa_k]$$

$$\underset{\text{empty word}}{[\,]} \longmapsto [\,]$$

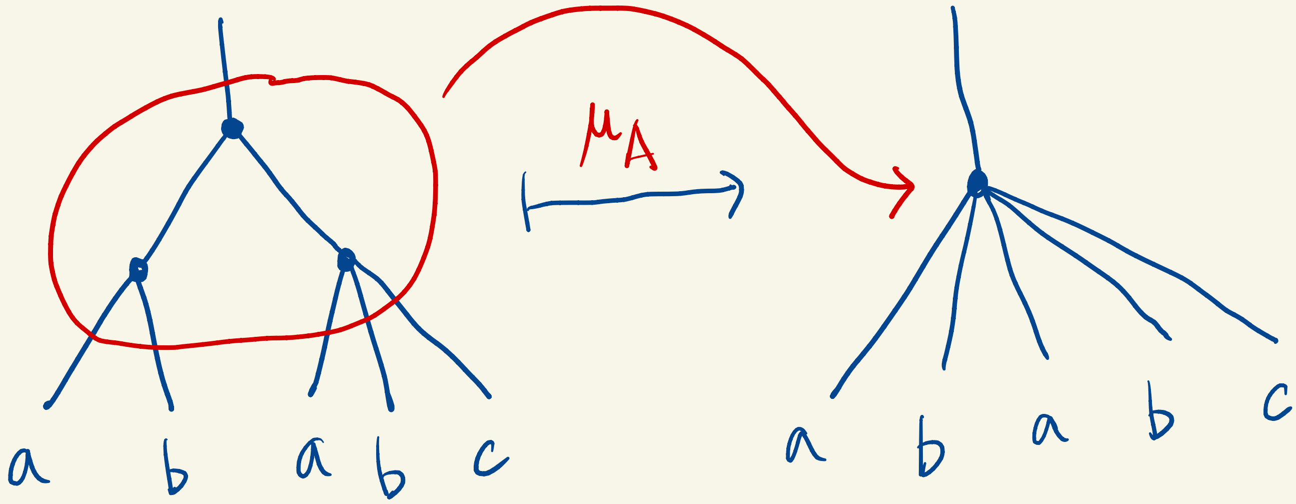
Multiplication is defined as the $\boxed{\text{natural}}$ family of functions

$$\mu_A : \underset{\substack{\| \\ TTA}}{A^{**}} \longrightarrow \underset{\substack{\| \\ TA}}{A^{*}}$$

which "flattens" or "removes brackets" in a word of words.

$$A = \{a, b, c\}$$

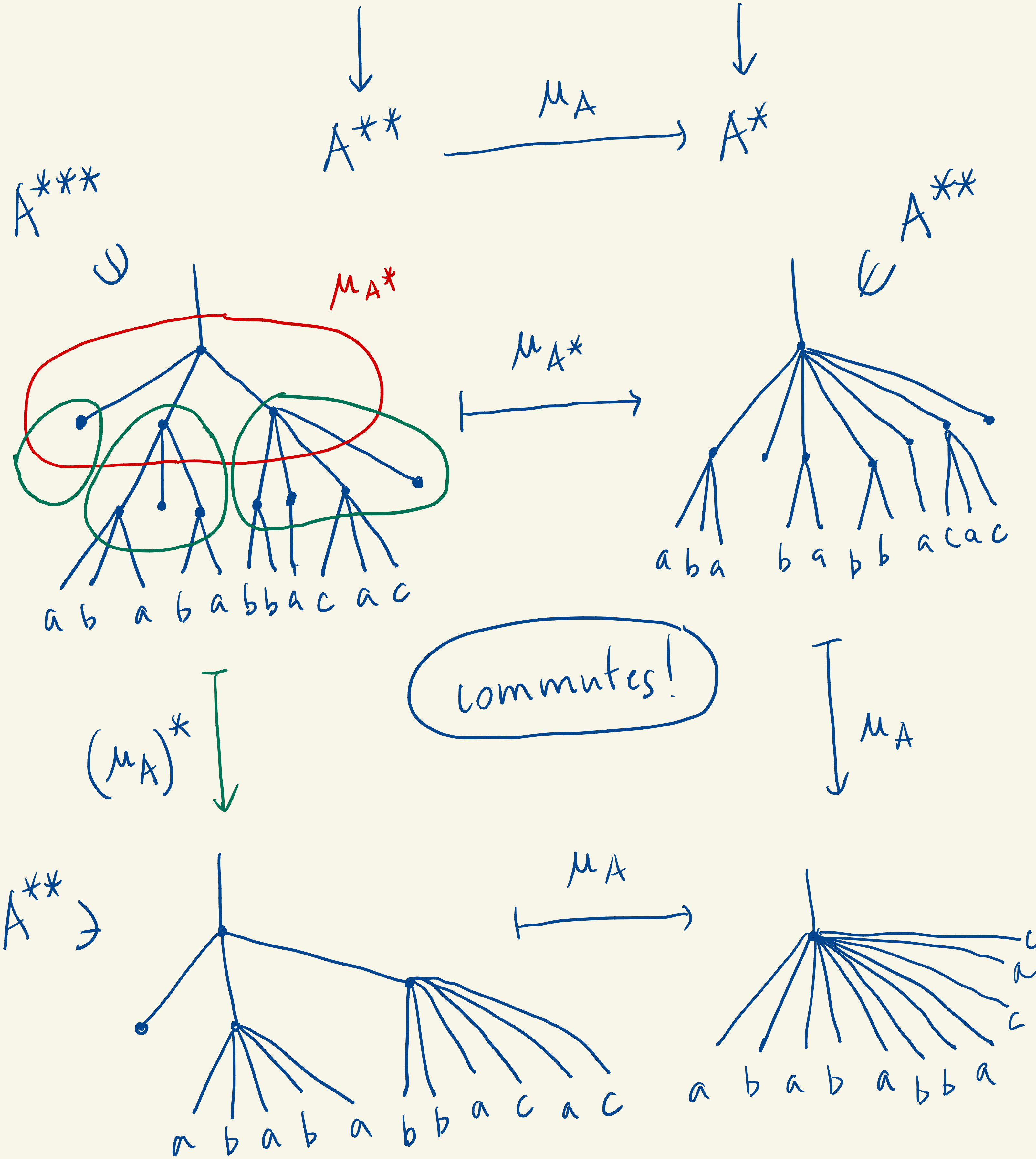
$$\underset{\substack{\cap \\ A^{**} \\ TTA}}{[[a,b],[a,b,c]]} \longmapsto \underset{\substack{\cap \\ A^{*} \\ TA}}{[a,b,a,b,c]}$$



the multiplication μ
at instance A

associativity means that
given a word of words of words
ie an element of $TTTA = A^{***}$
the two different ways to flatten it
induce equal functions.

$$\begin{tikzcd} A^{***} \arrow[r, "\mu_{A^*}"] \arrow[d, "(\mu_A)^*"'] & A^{**} \arrow[d, "\mu_A"] \\ {} & {} \end{tikzcd}$$


$A^{**} \xrightarrow{\mu_A} A^*$
$A^{***} \ni$
μ_{A^*}
$\ni A^{**}$
a b a b a b b a c a c
a b a b a b b a c a c
commutes!
$(\mu_A)^*$
μ_A
$A^{**} \ni$
μ_A
a b a b a b b a c a c
a b a b a b b a c a c

the unit η is defined as

the $\boxed{\text{natural}}$ family of functions

$$\begin{array}{rccl} \eta_A : & A & \longrightarrow & A^* \\ & a & \longmapsto & [a] \end{array}$$

the word with one letter a

let us check the two neutrality equations:

$$\begin{tikzcd} & A^{**} \arrow[dr, "\mu_A"] & \\ A^* \arrow[ur, "\eta_A^*"] \arrow[rr, "id_{A^*}"'] & & A^* \end{tikzcd}$$

commutes

$$\begin{array}{rccl} \eta_A^* : & A^* & \longrightarrow & A^{**} \\ & [a_1 \ldots a_k] & \longmapsto & [[a_1] \ldots [a_k]] \end{array}$$

$$\begin{tikzcd} & A^{**} \arrow[dr, "\mu_A"] & \\ A^* \arrow[ur, "\eta_{A^*}"] \arrow[rr, "\mathrm{id}_{A^*}"'] & & A^* \end{tikzcd}$$

commutes

$$\eta_{A^*} : A^* \longrightarrow A^{**}$$

$$[a_1 \ldots a_k] \longmapsto \underbrace{[[a_1 \ldots a_k]]}_{\substack{\text{a word} \\ \text{with a unique letter} \\ \text{which is the word} \\ [a_1 - a_k] \text{ itself.}}}$$

The fact that μ and η are natural transformations means that the diagrams

$$\begin{tikzcd} TTA \arrow[r, "TTf"] \arrow[d, "\mu_A"'] & TTB \arrow[d, "\mu_B"] \\ TA \arrow[r, "Tf"'] & TB \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\eta_A"'] & B \arrow[d, "\eta_B"] \\ TA \arrow[r, "Tf"] & TB \end{tikzcd}$$

commute in Set for all functions $f : A \longrightarrow B$.

<u>Rem</u>. in this case, the function f can be seen as a "relabelling" Functions.

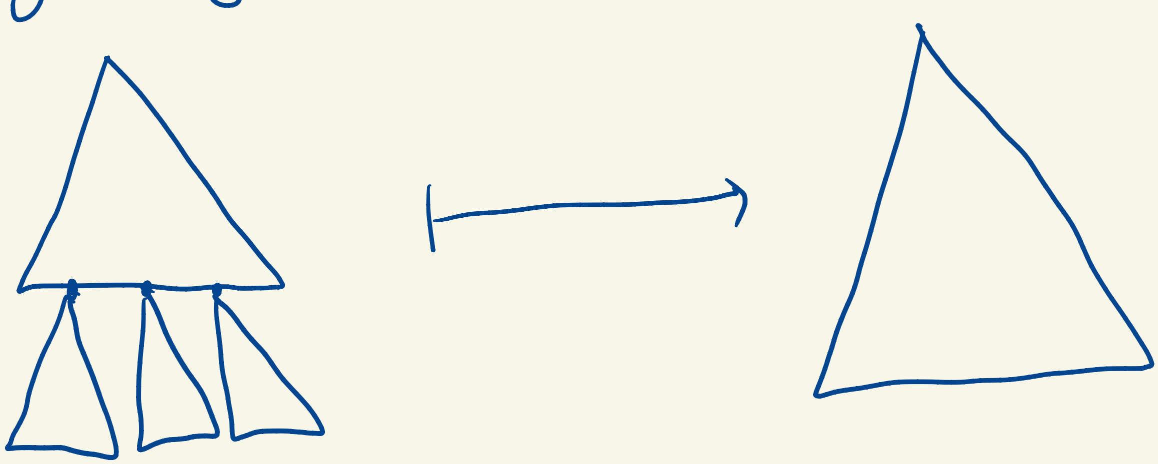
Example 2 the free magma monad

$$\mathrm{Mag} : \mathrm{Set} \longrightarrow \mathrm{Set}$$

Mag transports every set A to the set of binary trees with leaves labelled by the elements of A.

Multiplication is "substitution" of trees inside a tree

$$\mathrm{Mag}(\mathrm{Mag}\,A) \xrightarrow{\mu_A} \mathrm{Mag}\,A$$



The unit is just associating a trivial tree with one leaf to any element of A

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & \mathrm{Mag}\,A \\ a & \longmapsto & \begin{array}{c} | \\ a \end{array} \end{array}$$

equational theory

More generally any algebraic theory on a given signature of operations defines a monad on Set.

(called finitary monad)

1 every adjunction induces a monad

suppose given an adjunction

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathcal{B} \arrow[l, bend left, "R"] \end{tikzcd}$$

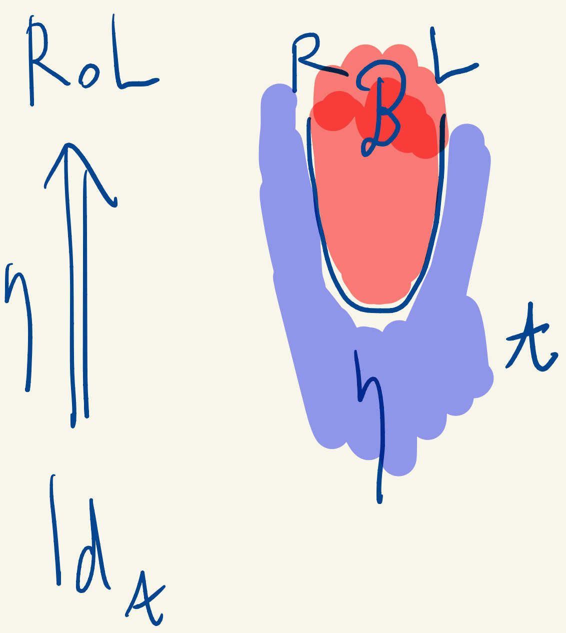
presented by its unit

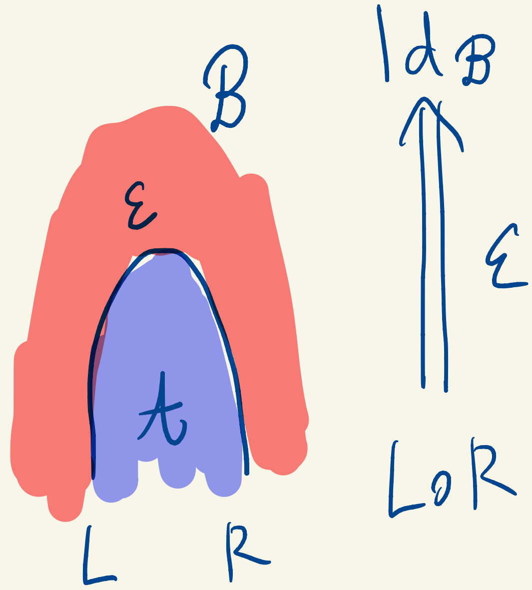
$$\eta : Id_{\mathcal{A}} \Longrightarrow R \circ L$$

and its counit

$$\varepsilon : L \circ R \Longrightarrow Id_{\mathcal{B}}$$

in string diagrams :





<u>Claim</u>: the composite

$$T = R\circ L$$

equipped with the multiplication

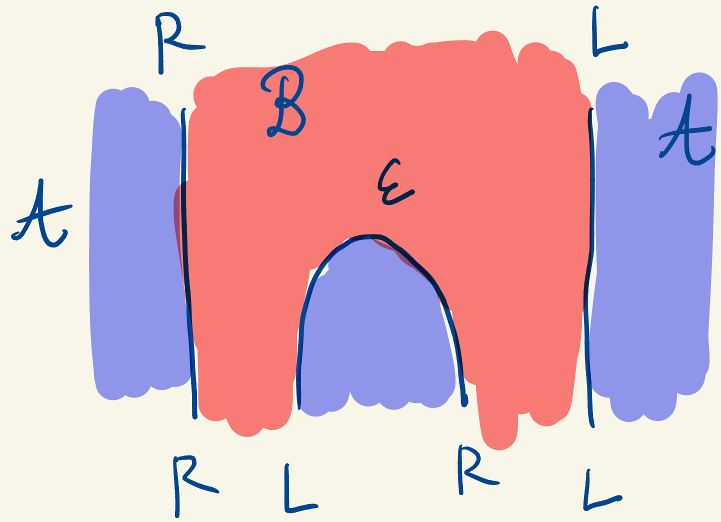
$$T\circ T = R\circ L\circ R\circ L \xRightarrow[R\circ\varepsilon\circ L]{\mu} R\circ L = T$$

and the unit

$$\eta : \mathrm{Id}_{\mathcal{A}} \Longrightarrow R\circ L = T$$

In string diagram:

$$\begin{tikzcd} R\circ L \\ R\circ L\circ R\circ L \arrow[u, Rightarrow, "\mu", "R\circ\varepsilon\circ L"'] \end{tikzcd}$$



We need to check

<u>associativity</u>

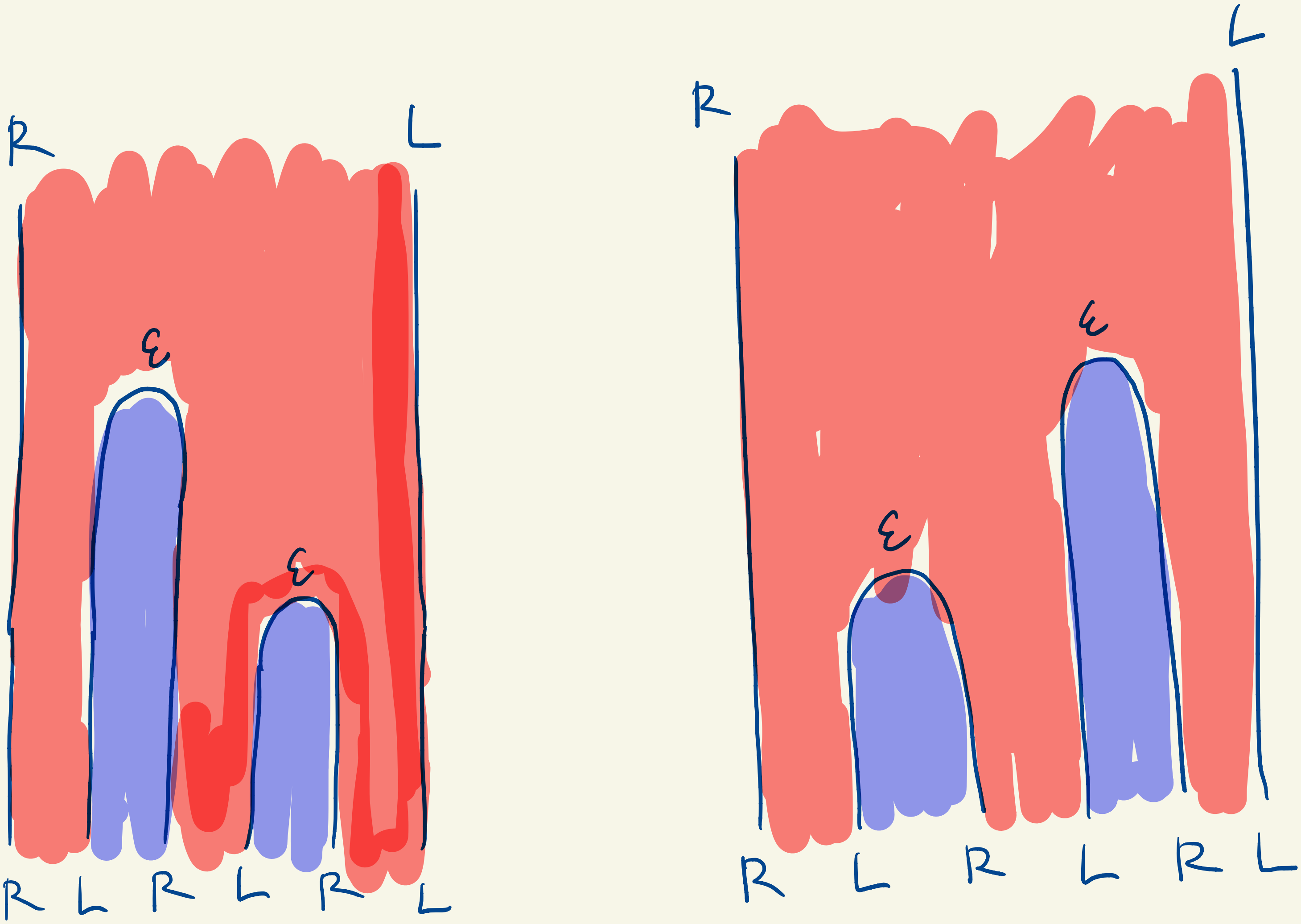
<u>neutrality</u>

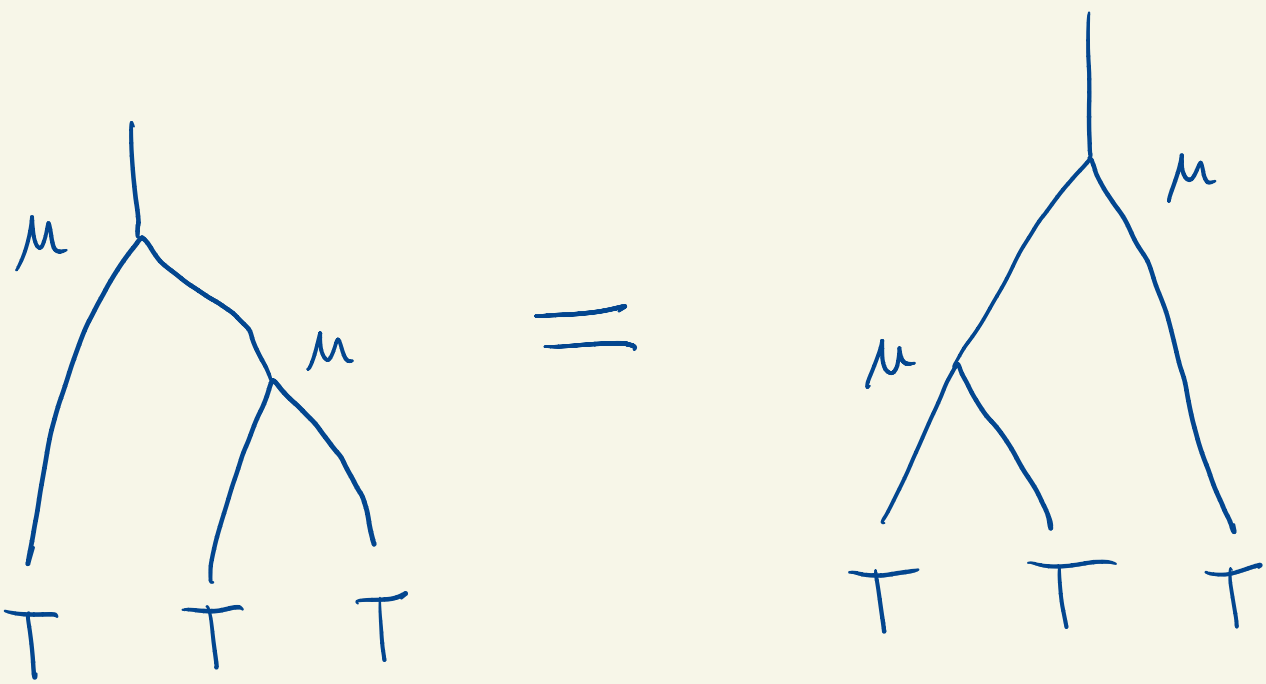
in order to be sure that

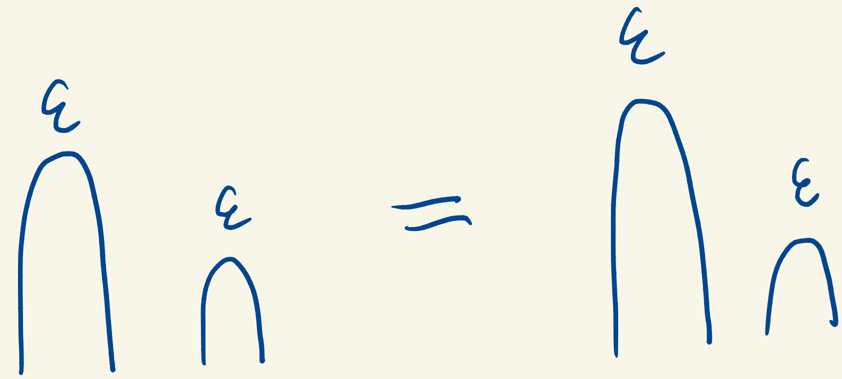
$$(T,\ \mu = R\circ\varepsilon\circ L,\ \eta)$$

defines a monad.

Associativity:







because of naturality

and in fact from the

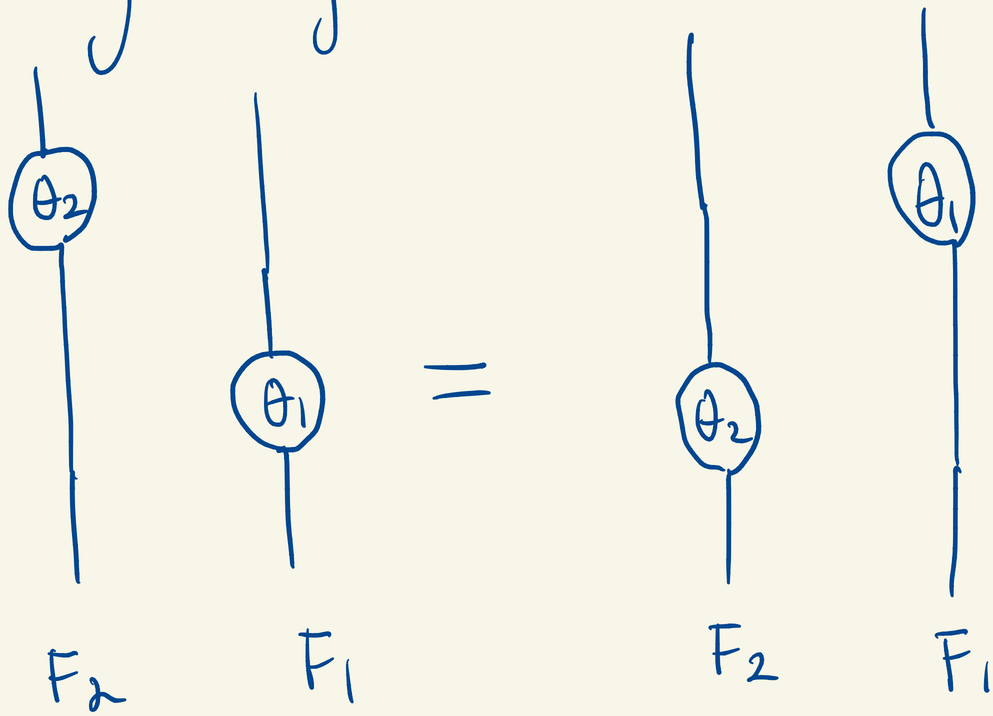
2-categorical structure

of categories, functors

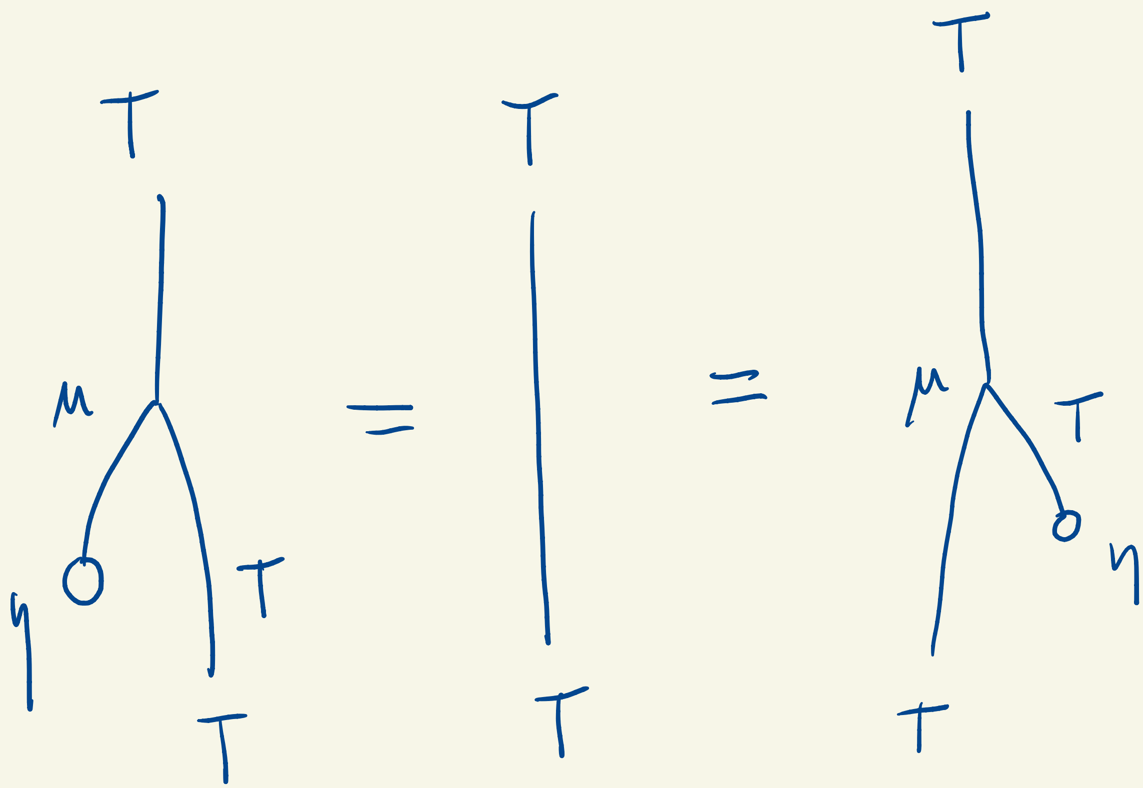
and natural transformations

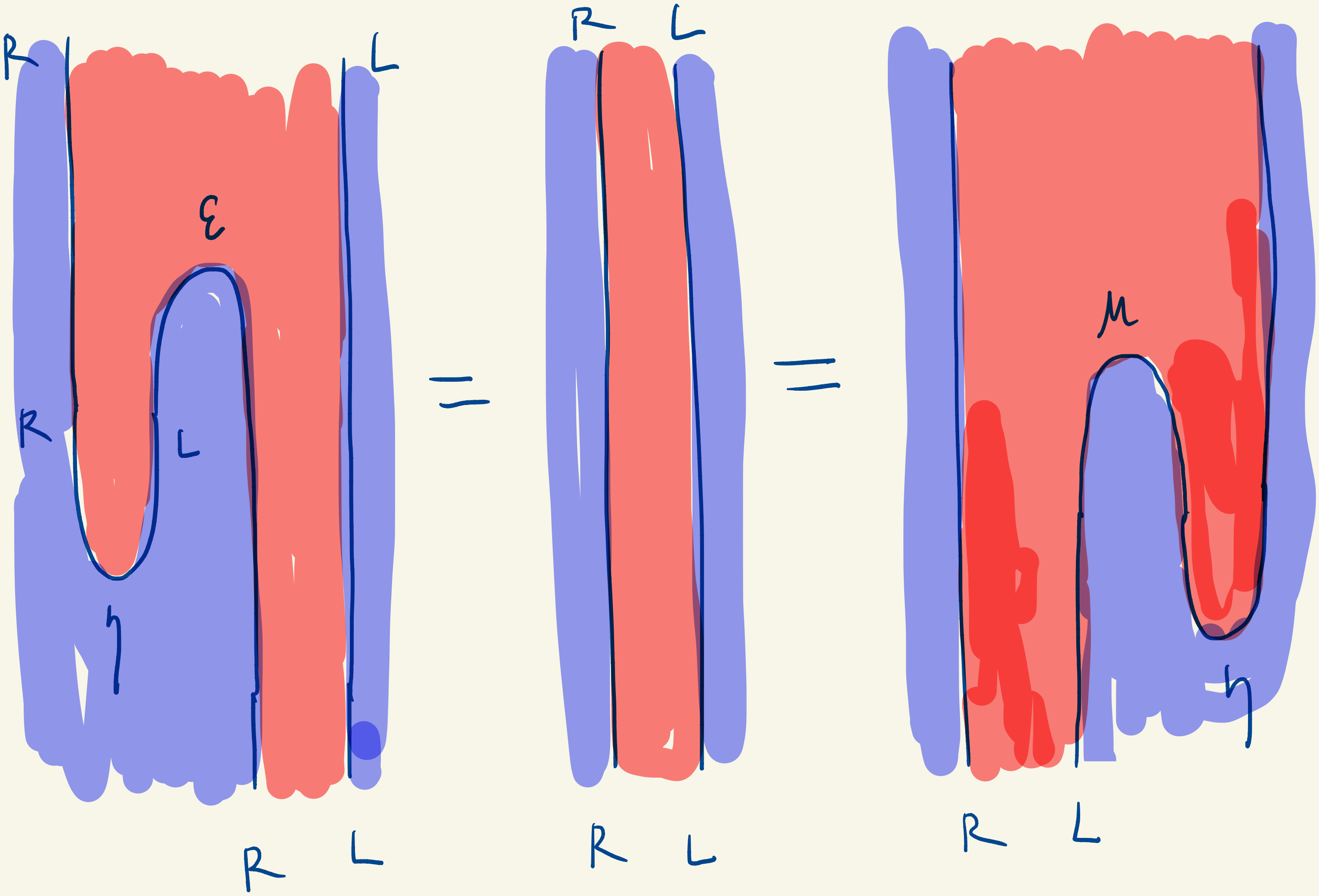
$$\begin{tikzcd}
\mathcal{A} \arrow[r, bend left=60, "F_1"] \arrow[r, bend right=60, "G_1"'] \arrow[r, phantom, "\Downarrow\theta_1"] & \mathcal{B} \arrow[r, bend left=60, "F_2"] \arrow[r, bend right=60, "G_2"'] \arrow[r, phantom, "\Downarrow\theta_2"] & \mathcal{C}
\end{tikzcd}$$

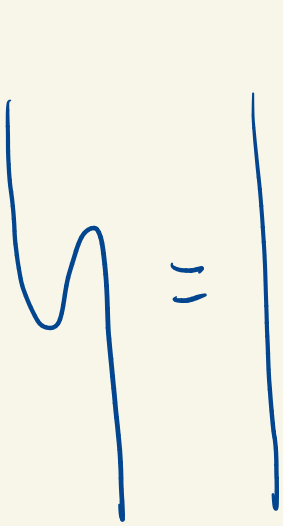
the order in which we apply θ_1 and θ_2 does not matter in string diagrams:



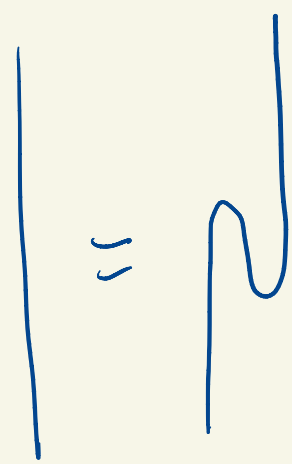
<u>neutrality</u>:







zig zag
equations
of
an adjunction



this establishes the equation!

<u>Example</u>: the adjunction

$$\begin{tikzcd} \mathrm{Set} \arrow[rr, bend left, "L = \mathrm{Free}"] \arrow[rr, phantom, "\perp"] & & \mathrm{Mon} \arrow[ll, bend left, "R = U"] \end{tikzcd}$$

the category of monoids

L the free monoid functor

$$A \longmapsto \underbrace{(A^*, \text{concat}, \text{empty word})}_{\text{the free monoid generated by } A}$$

A a set

R the forgetful functor

$$M \longmapsto UM$$

M monoid

UM the underlying set
the support of M.

the associated monad T on Set is defined as

$$T = R \circ L : \underset{\text{Set}}{A} \longmapsto \underset{\text{set}}{A^*} : \text{Set} \longrightarrow \text{Set}$$

the multiplication:

$$\mu : R \circ L \circ R \circ L \xRightarrow[R \circ \varepsilon \circ L]{} R \circ L$$

$$\mu_A : RLRLA \xrightarrow{R\varepsilon_{LA}} RLA$$

$$\varepsilon_M : M^* \longrightarrow M$$

M a monoid $\quad [m_1 \ldots m_k] \longmapsto m_1 \bullet_M \ldots \bullet_M m_k$

$$M = LA = A^*$$

$$\varepsilon_{A^*} : A^{**} \xrightarrow{\text{flattening}} A^*$$

$$[w_1 \_ w_k] \longmapsto w_1 \cdot \_ \cdot w_k$$

where $\cdot$ is concatenation

hence

$$\mu_A : A^{**} \longrightarrow A^*$$

is defined in just the same way as we described earlier.

Similarly

$$\begin{aligned} \eta_A : \quad A &\longrightarrow A^* \\ a &\longmapsto [a] \end{aligned}$$

So we deduce from this that the monad $T = R \circ L$ associated to the adjunction between Set and Mon coincides with the "free monoid" monad described earlier in the lesson.

Example 2 similarly

the adjunction

$$\begin{tikzcd} \mathrm{Set} \arrow[r, bend left, "\text{free magma}"] \arrow[r, phantom, "\perp"] & \mathrm{Magma} \arrow[l, bend left, "\text{forgetful functor}"] \end{tikzcd}$$

induces the "free magma" monad on Set described earlier.

More generally, any algebraic theory induces an adjunction

$$\begin{tikzcd} \mathrm{Set} \arrow[r, bend left, "\text{free}"] \arrow[r, phantom, "\perp"] & \mathrm{Alg} \arrow[l, bend left, "\text{forgetful}"] \end{tikzcd}$$

where Alg is the category of algebras of the theory (models) with morphisms defined as homomorphisms (= functions preserving the operations of the algebraic theory)

(2) Eilenberg-Moore algebras of a monad T

T-algebras

the idea :

show that every monad T on a category $\mathcal{C}$ induces a category of T-algebras Alg_T and T-homomorphisms defining an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left=50, "Free"] \arrow[r, phantom, "\perp"] & Alg_T \arrow[l, bend left=50, "Forget"] \end{tikzcd}$$

in such a way that (for instance) in the case of the monad

"free monoid"

$$\boxed{\begin{aligned} T : \mathbf{Set} &\longrightarrow \mathbf{Set} \\ A &\longmapsto A^* \end{aligned}}$$

(μ, η)

unit

multiplication

one <u>recovers</u> the category $\boxed{\mathbf{Mon}}$ of monoids and homomorphisms as the category $\mathbf{Alg}_T$ of T-algebras.

There is a little bit of magic here!

Meditative exercise for next week:

Find a definition of monoid ⊛ just using the free monoid monad (T, μ, η) on Set.

⊛ and of monoid homomorphism.