Introduction to String Diagrams

Alek Sherstynk

For Category Theory, Math 9000 something

In category theory, we have categories, functors between categories, and natural transformations between functors.

Functors combine via composition in a particular way, depending on their domains and codomains.

Natural transformations may compose with each other in 2 different ways, again, depending on their domains and codomains.

Certain rules govern how these compositions interact, such as the "interchange law":

$$(\alpha * \beta) \circ (\gamma * \theta) = (\alpha \circ \gamma) * (\beta \circ \theta)$$

where these compositions are defined.

Purpose:

String diagrams provide a natural "diagramatic algebra" whose algebra is exactly what we want: the compositions of functors and natural transformations.

They let us visualize horizontal and vertical composition in a context where the interchange law becomes obvious. Lets begin

One way of visualizing a system of functors and natural transformations is the so called <u>Globular</u> diagram.

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{D} \arrow[r, bend left, "H"] \arrow[r, bend right, "J"'] \arrow[r, phantom, "\Downarrow\beta"] & \mathcal{E} \end{tikzcd}$$

This conveys the information that there are categories $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{E}$, functors F, G, H, J, natural transformations α and β and

$$F:\mathcal{C}\to\mathcal{D},\ G:\mathcal{C}\to\mathcal{D},\ H:\mathcal{D}\to\mathcal{E},\ J:H\to J,\ \alpha:F\to G,\ \beta:H\to J.$$

You can see why the globular diagram conveys info more clearly than listing every piece of data in a row.

The globular diagram also makes obvious that: $\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "HF"] \arrow[r, bend right, "JG"'] \arrow[r, phantom, "\Downarrow\beta*\alpha"] & \mathcal{E} \end{tikzcd}$

In words, we can define the natural transformation $\beta*\alpha$.

The globular diagram for vertical composition is:

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left=60, "F"] \arrow[r, "G" description] \arrow[r, bend right=60, "H"'] \arrow[r, phantom, shift left=3ex, "\Downarrow\alpha"] \arrow[r, phantom, shift right=3ex, "\Downarrow\beta"] & \mathcal{D} \end{tikzcd} = \begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "H"'] \arrow[r, phantom, "\Downarrow\beta\circ\alpha"] & \mathcal{D} \end{tikzcd}$$

Globular diagrams lead naturally to string diagrams:

Notice, a globular diagram $\begin{tikzcd} \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow"] & \bullet \end{tikzcd}$ represents

- categories (essentially) by points, functors as "oriented 1-cells (directed edges)" between points, and natural transformations as "oriented 2-cells (directed regions) between 1-cells.

(Incidentally, pharased in this language, category theory may be generalized to ∞-category theory, but this isnt necessary for us now.)

The key with string diagrams is to take the dual.

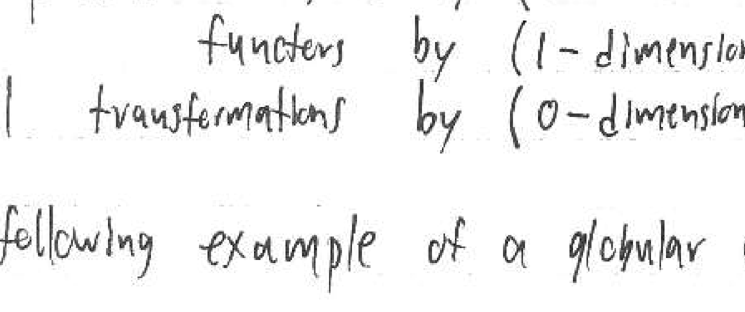
- We represent categories by (2-dimensional) regions, functors by (1-dimensional) edges, and natural transformations by (0-dimensional) points.

Use the following example of a globular diagram and its dual:

$$\begin{tikzcd} \mathcal{C} \arrow[rr, bend left, "H"] \arrow[dr, "F"'] \arrow[rr, phantom, "\Uparrow\alpha"] & & \mathcal{D} \\ & \mathcal{E} \arrow[ur, "G"'] & \end{tikzcd}$$

Here, $\alpha : GF \to H$

dual to

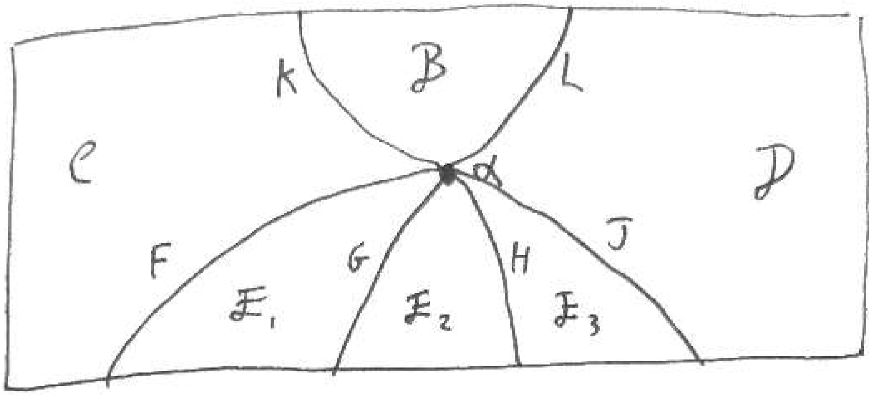


The strings G and F enter the node α, and become H, if we read from bottom to top.

To read the string diag, notice, the strings labled G and F come from the bottom side, and H enters the top.

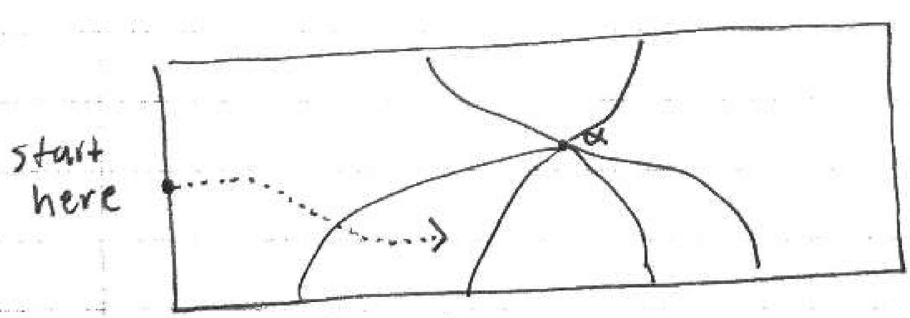
We read string diagrams from bottom to top, and ~~right~~ left to right (although we are about to change this convention superficially).

As a more complicated example, take



The strings coming from the bottom side are F, G, H, J in that order, left to right, so they represent the composite functor JHGF. The top represents LK.

To picture the situation better, imagine a person standing on the very left region. To walk rightward without turning back, we must cross certain strings.

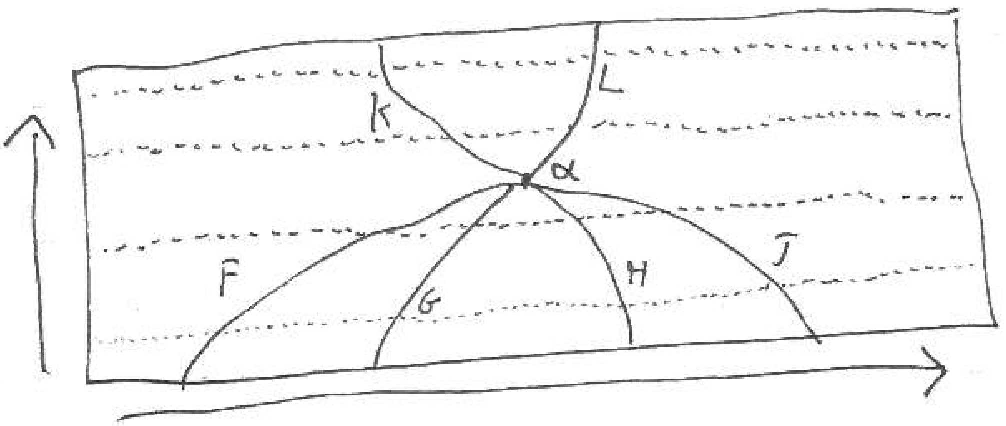


For example, to cross from the region labled $\mathcal{C}$ to the one labled $\mathcal{E}_1$, we must cross the string labled F. This means F is a functor from the category $\mathcal{C}$ to the category $\mathcal{E}_1$.

In general, this is how we read string diagrams. Strings separate regions, and crossing a string is thought of as applying the functor that the string represents.

5 Notice, there are 2 main paths from the left to the right: a path under the node labled α, and a path over the α.

Reading bottom up, we would say that this means $\alpha: JHGF \to LK$. Lets analyse again:



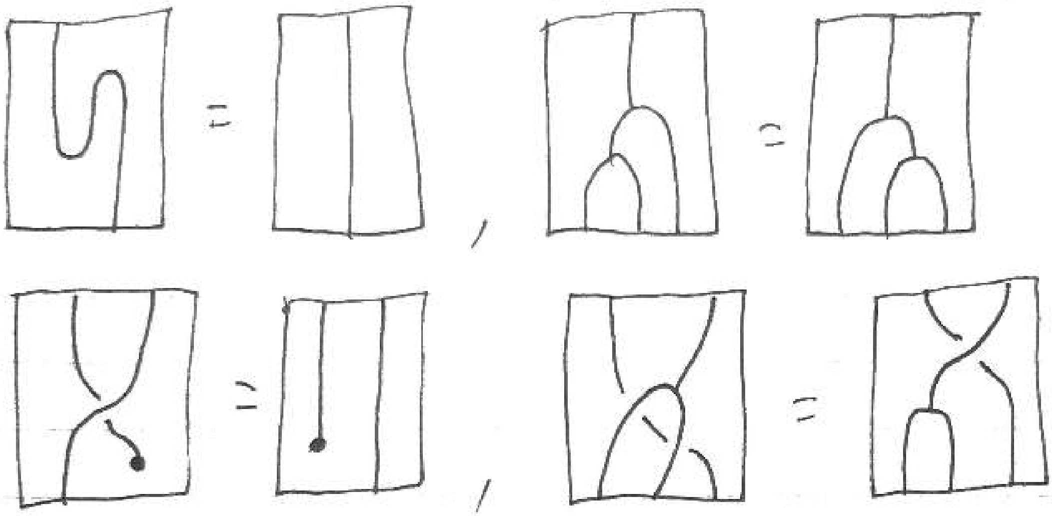
If we take horizontal cross sections, we notice that we start by hitting F, G, H, and J, and above the node α, the horizontal cross sections change and become K and L.

This is how we read the effect of α: it turned paths that intersect F, G, H and J into paths that intersect (cross) K and L, which corresponds to α being a natural transformation from JHGF to LK.

For this interpretation to make sense, strings must "rise from the bottom side into the top" without turning around, and the "changes" to the strings along the way from bottom to top correspond to natural transformations.

Why is this useful? What can we do with string diagrams?

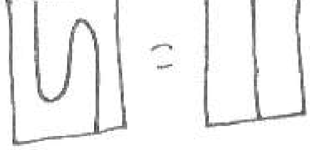
Now that we have a basic sense of orientation, I will explain how complicated axioms for categorical structures can be depicted as simply as:



and other visually spectacular facts. Dispite the fact that all lables have been dropped, the unique shapes of these string diagrams allow us to infer all of the correct details in each of the above situations (most, <u>but not all</u>, of which are beyond the scope of what we currently covered).

Another advantage is that, even with no context, these equalities seem to make a bit of sense, on a puely visual, topological level. Indeed, they are axioms that various categorical constructions satisfy, and without string diagrams, those axioms are overall far less "obvious" or intuitive.

I will explain the first equality, which we actually covered!

The equality , and the dual ,

actually refer to the triangle laws for an adjunction:

Let $\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$ be an adjoint pair, given by

the unit $\eta : 1_{\mathcal{C}} \to GF$ and counit $\varepsilon : FG \to 1_{\mathcal{D}}$

satisfying

$$\begin{tikzcd} F \arrow[r, "F\eta"] \arrow[dr, "1_F"'] & FGF \arrow[d, "\varepsilon F"] \\ & F \end{tikzcd}$$

commutes in the functor category $[\mathcal{C}, \mathcal{D}]$

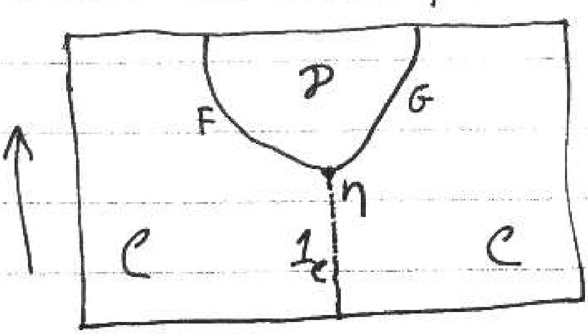
and

$$\begin{tikzcd} G \arrow[r, "\eta G"] \arrow[dr, "1_G"'] & GFG \arrow[d, "G\varepsilon"] \\ & G \end{tikzcd}$$

commutes in the functor category $[\mathcal{D}, \mathcal{C}]$,

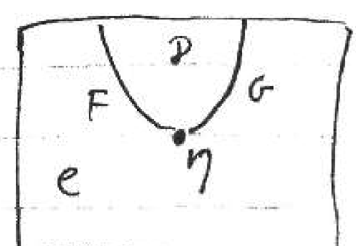
How would we visualize η as a string diagram? :

Like this



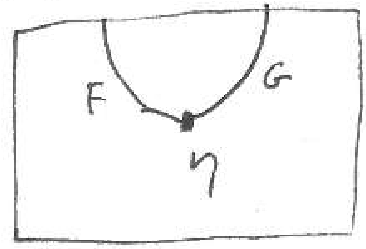
. We will make a preliminary simplification and leave of the identity functor $1_{\mathcal{C}}$ (which we already drew a bit dashed).

The revised diagram is this :

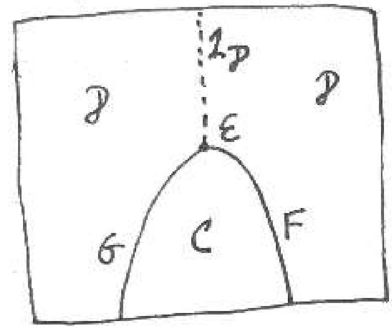


We will soon make another simplification and cease to lable the regions by the categories they represent, since this can be inferred from the lables for the strings, so long as we keep track of our domains and codomains.

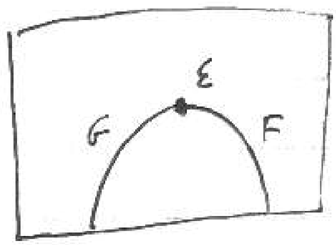
The revised diagram is ⇢



Dually, we have:



which, simplified, becomes this ⇢



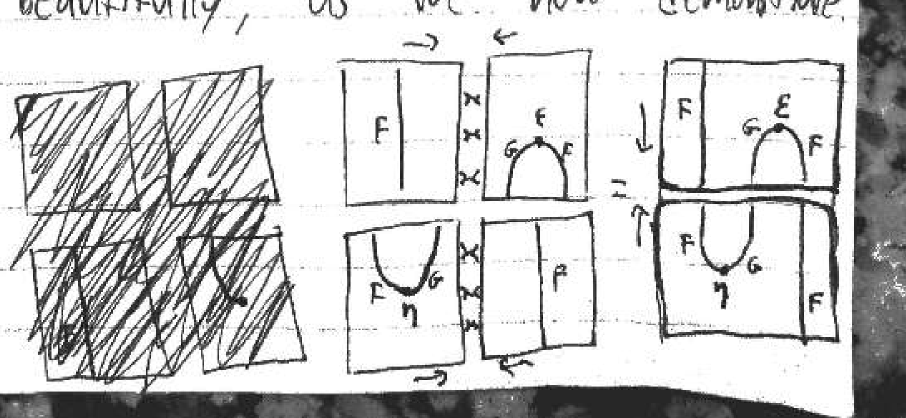
We run into a slight problem now. How do we visualize $F\eta$? What about $\varepsilon F \circ F\eta$?

We need to understand how string diagrams treat horizontal and vertical composition now.
(and we finally have a non-abstract reason to. We want to use string diagrams to understand adjunctions).

Fortunately, string diagrams respect horizontal and vertical composition beautifully, as we now demonstrate.

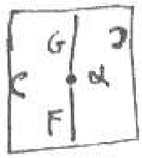
to give a sneak peak:

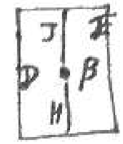
(wrong variance) ⟶



9 Horizontal composition:

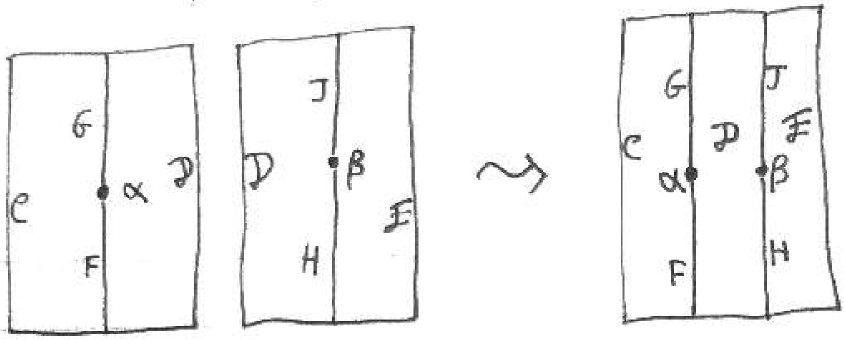
if
$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "G"] \arrow[r, bend right, "F"'] \arrow[r, phantom, "\Uparrow\alpha"] & \mathcal{D} \arrow[r, bend left, "J"] \arrow[r, bend right, "H"'] \arrow[r, phantom, "\Uparrow\beta"] & \mathcal{E} \end{tikzcd}$$
then the string diagram

for α is 

, and for β, 

.

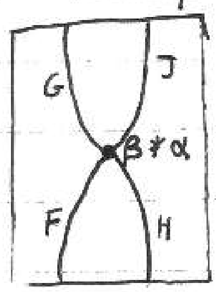
To form $\beta * \alpha$, we put the diagrams next to each other, as

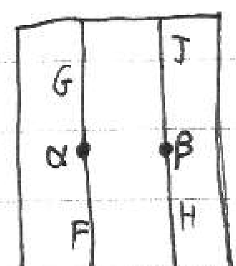


.

Notice how the regions align at $\mathcal{D}$.

So, when we refer to $\beta * \alpha : HF \to JG$, it's this diagram.

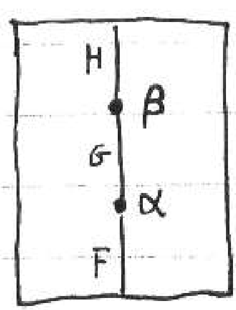
In other words 

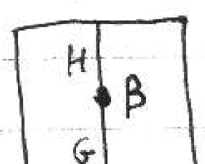
is defined as 

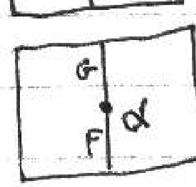
when we "unpack" $\beta * \alpha$.

Vertical Composition:

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left=60, "H"] \arrow[r, "G"] \arrow[r, bend right=60, "F"'] & \mathcal{D} \end{tikzcd} \quad \Uparrow\beta,\ \Uparrow\alpha$$

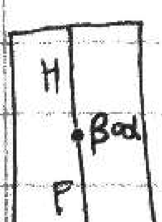
becomes 

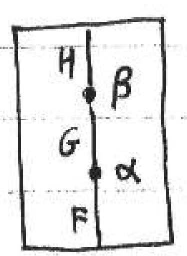
, which is just 

on top of 

.

Notice, how the top side of the bottom diagram encodes the same functor as the bottom of the top diagram, G

so 

is just defined as 

.

Two last points of order: reversal of our conventions and the interchange law.

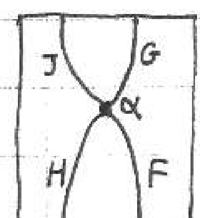
Its becoming increasingly inconvenient to keep track of the syntactic reversal of letters in composition, e.g.

$\mathcal{C} \xrightarrow{F} \mathcal{D} \xrightarrow{G} \mathcal{E}$ can be written as $\mathcal{C} \xrightarrow{GF} \mathcal{E}$.

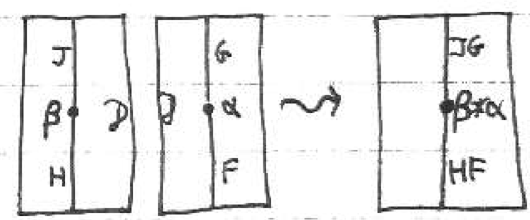
Drawing all globular diagrams and string diagrams right to left will remedy this: $\mathcal{C} \xleftarrow{F} \mathcal{D} \xleftarrow{G} \mathcal{E}$ becomes $\mathcal{C} \xleftarrow{FG} \mathcal{E}$.

In the future, we write and read string diagrams <u>right to left</u>, and still bottom to top.

This has the advantage that if $\alpha : HF \to JG$, then



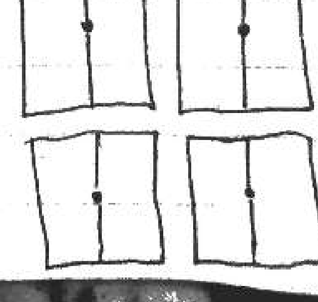
, and also horizontal composition ceases to reverse letters:



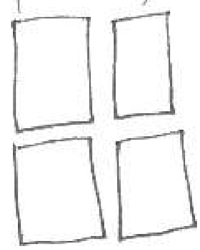
Notice, the last diagram used a single string to denote composite functors. The flexibility to denote a composite functor with one string, or "break up" the functor into its composites, can be useful.

The Interchange Law.

understanding what horizontal and vertical composition looks like leads to the question of whether or not combining these is well defined: it is precisely the interchange law that says we can either combine horizontally, then vertically, or vice versa. Thus  is well defined.



This last point about the interchange law fulfills a promise made around the beginning of the exposition: that string diagrams provide a natural algebra consistent with the rules for combining natural transformations via vertical and horizontal composition. Obviously, placing 4 rectangles next to each other in a



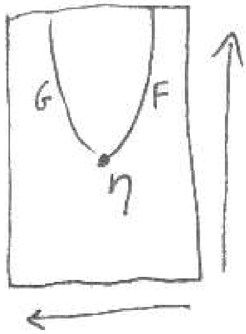
pattern, it doesnt matter whether we glue horizontally or vertically first. Thus, string diagrams make the interchange law intuitively plausible.

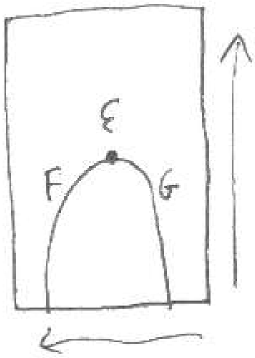
Of course, we must, and have (in the HW) proven that horizontal and vertical composition indeed satisfies the interchange law $(\alpha \circ \beta) * (\gamma \circ \theta) = (\alpha * \gamma) \circ (\beta * \theta)$ in order to properly use string diagrams.

But since we know the interchange law is satisfied by natural transformations (by straitforward computation), and since the law looks much more natural in string diagramatic form than equational, these diagrams are useful conceptual tools for reasoning about complicated categorical situations.

On a last aside, toward deeper concepts in category theory, reasoning with string diagrams is also natural in the context of monoidal categories. The collection of all categories forms a structure known as a ~~strict~~ 2-category. Strict 2-categories and monoidal categories are both examples of weak 2-categories, which are the most general structures for which string diagrams can naturally be used. Back to the case of adjunctions:

Armed with our knowledge of how horizontal and
12 vertical composition works, how the interchange law looks, and its "well-definedness" we reconsider how to express the triangle laws of an adjunction

We have 

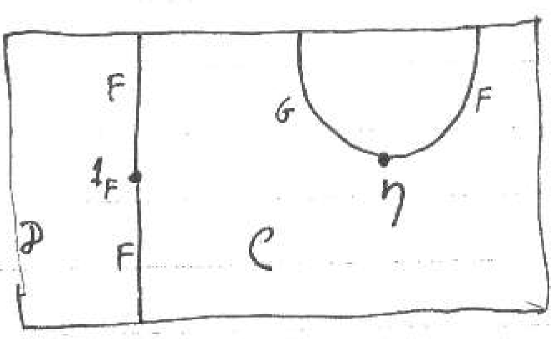
and 

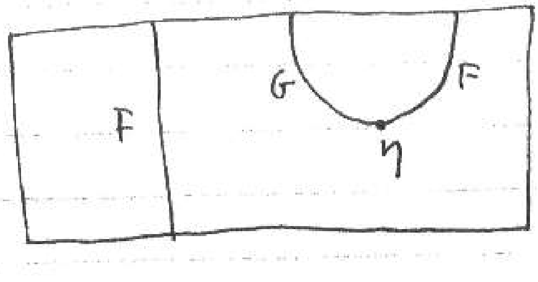
with our right to left convention.

Recall, the natural transformation $F\eta$ is defined as $1_F * \eta$, the horizontal composition of η followed by the identity natural transformation on the functor F.

Globular diagram:

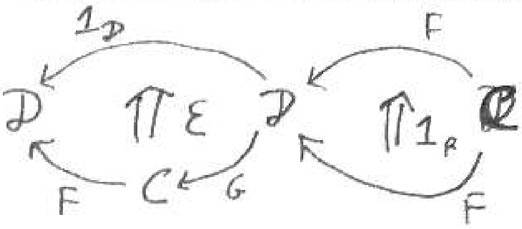
$$\begin{tikzcd} \mathcal{D} & \mathcal{C} \arrow[l, bend right, "F"'] \arrow[l, bend left, "F"] \arrow[l, phantom, "\Uparrow 1_F"] & \mathcal{C} \arrow[l, bend right, "GF"'] \arrow[l, bend left, "1_{\mathcal{C}}"] \arrow[l, phantom, "\Uparrow \eta"] \end{tikzcd}$$

string diagram 

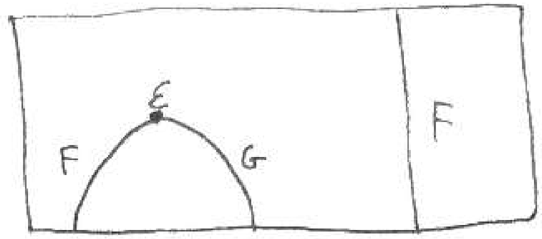
If we omit the identity node on F and the region lables: 

expresses $F\eta$, the functor from F to FGF.

Likewise, εF, has the globular diagram

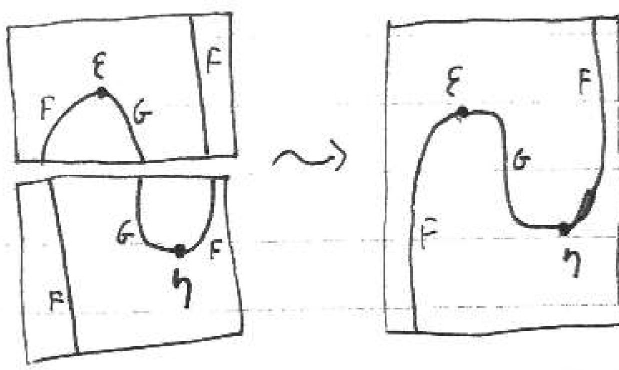


which becomes, in strings,



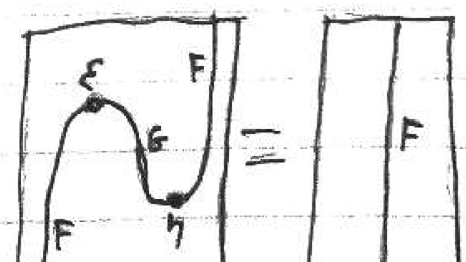
Try to visualize once more how the regions representing natural transformations became points and how the functor arrows turned perpendicular into strings as we go from the globular diagram to the string diagram.

vertically composing $\varepsilon F \circ F\eta$ gives



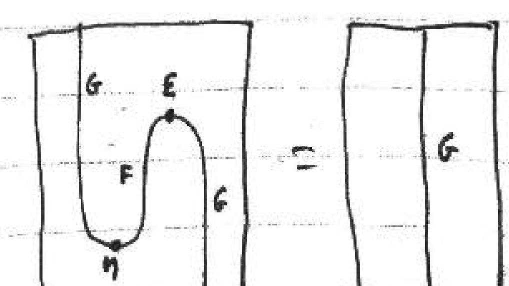
This final picture of a "squiggle" is very suggestive. Indeed, the horizontal cross sections, bottom to top, are F, then FGF, then F again.

The axiom
$$\begin{tikzcd} F \arrow[r, "F\eta"] \arrow[rr, bend right, "1_F"'] & FGF \arrow[r, "\varepsilon F"] & F \end{tikzcd}$$
claims

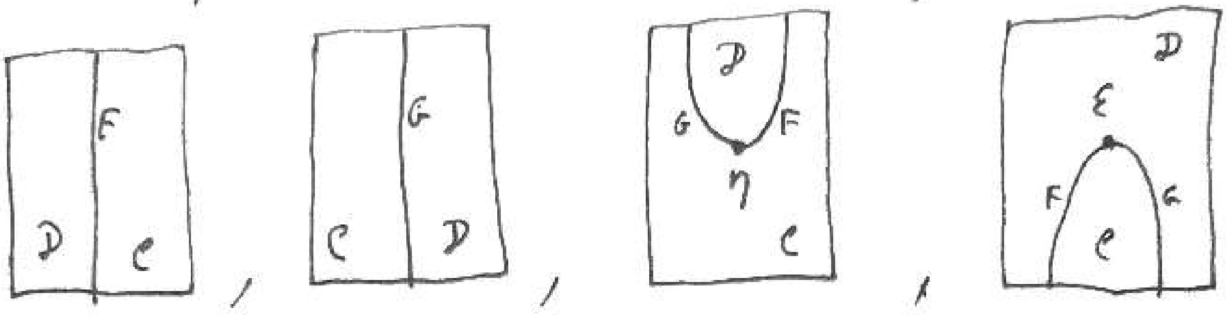


nothing more, nothing less.
Here, again, identity natural transformations and functors are omitted.

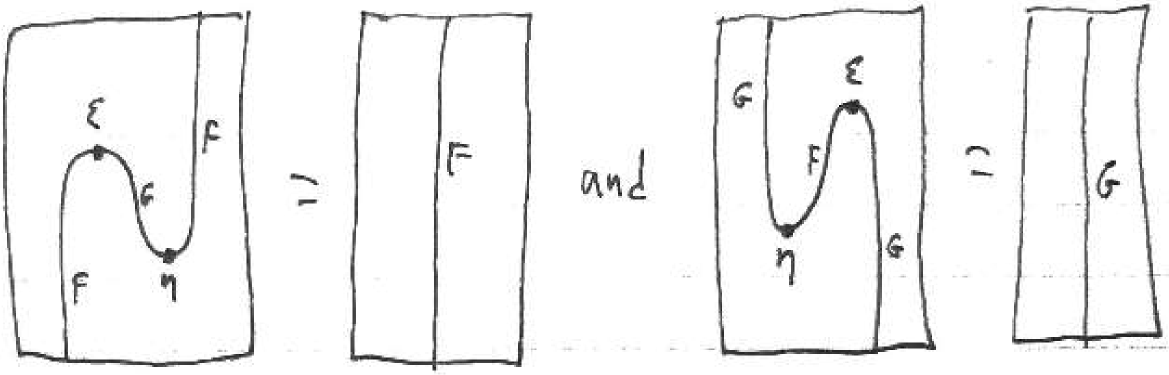
Likewise,
The dual axiom:
$$\begin{tikzcd} G \arrow[r, "\eta G"] \arrow[dr, "1_G"'] & GFG \arrow[d, "G\varepsilon"] \\ & G \end{tikzcd}$$
claims



Finally, in the language of string diagrams, we may completely rephrase the definition of an adjunction as the following information:

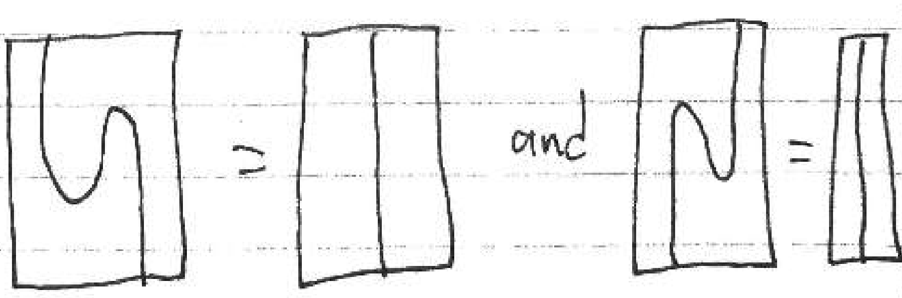


such that



Remark: Once the data of F, G, η, ε is understood, the axioms might be written in an even more concise way, with context fully determining the meaning.

Hence, if we write



the reader can infer where η and ε go based on the shape of the string.

Also, strings provide a medium for "categorical calculations", during which people often leave off lables.

We close with an application:

The final point of discussion will showcase how IS powerful string manipulation can be.

We introduce the definition of a monad, display the definition in terms of strings, and prove the following: if $\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$ ($F \dashv G$, $\perp$), then GF is a monad

Definition: A monad T is a functor on some category $\mathcal{C}$ along with natural transformations $\eta : 1_{\mathcal{C}} \to T$, $\mu : T^2 \to T$ such that these diagrams in $[\mathcal{C}, \mathcal{C}]$ commute:

$$\begin{tikzcd} T \arrow[r, "T\eta"] \arrow[dr, "1_T"'] & T^2 \arrow[d, "\mu"] \\ & T \end{tikzcd} , \quad \begin{tikzcd} T \arrow[r, "\eta T"] \arrow[dr, "1_T"'] & T^2 \arrow[d, "\mu"] \\ & T \end{tikzcd} , \quad \begin{tikzcd} T^3 \arrow[r, "T\mu"] \arrow[d, "\mu T"'] & T^2 \arrow[d, "\mu"] \\ T^2 \arrow[r, "\mu"'] & T \end{tikzcd}$$

Remark: there are formal similarities between this definition and the axioms of a monoid, with η being the monoidal unit and μ being the multiplication. The first 2 commutative triangles express that η is a right and left identity with respect to μ, and the commutative square claims that μ is associative.

One can summarize by saying "a monad is a monoid object in the category of endofunctors", which is an often repeated slogan among category theorists, and for whatever reason, also functional programmers.

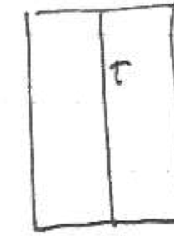
Note, to gain intuition, think about what a monad looks like if $\mathcal{C}$ is a poset category.

Hint: these two words arguably give it away, but "closure operator"

Now that we have a bit of a feel for monads, the theorem is that if $F: \mathcal{C} \to \mathcal{D}$, $G: \mathcal{D} \to \mathcal{C}$ with F left adjoint to G, then GF forms a monad on $\mathcal{C}$.

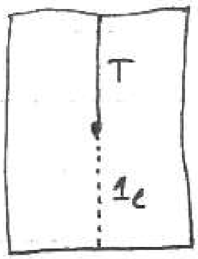
To see how this could be the case, lets recast our definition of a monad in terms of strings.

The functor $T: \mathcal{C} \to \mathcal{C}$ looks like

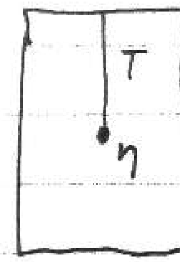


The natural transformation $\eta: 1_{\mathcal{C}} \to T$

looks like

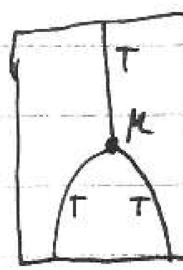


but as before, we write



,

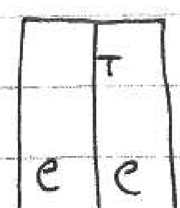
and $\mu: T^2 \to T$ looks like



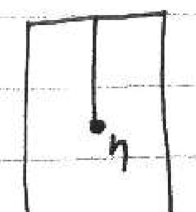
.

For string diagrams encoding a monad, every region is implicitly labled $\mathcal{C}$, and we can also assume every string is T.

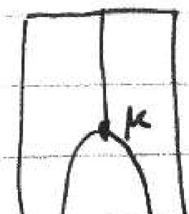
So a monad is



,

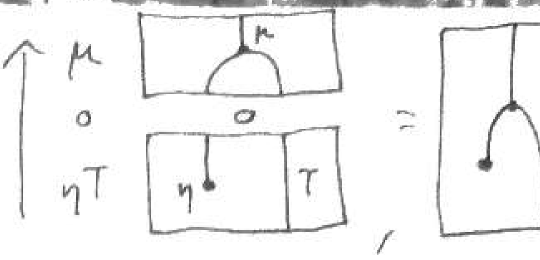


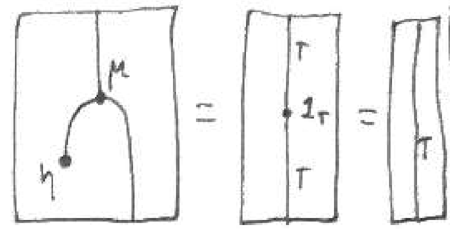
,

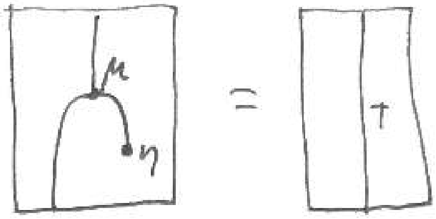


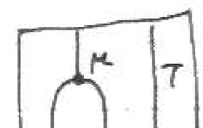
, satisfying certain rules.

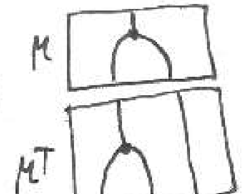
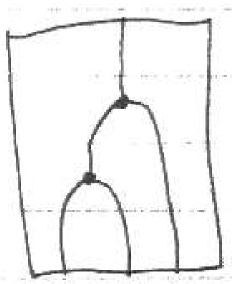
The commutative diagrams assert that $\mu \circ \eta T = 1_T$, $\mu \circ T\eta = 1_T$ and $\mu \circ \mu T = \mu \circ T\mu$. Lets see what these equations look like in strings

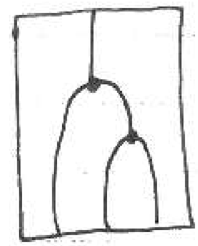
The ~~equation~~ expression $\mu \circ \eta T$ looks like ,

The equation claims $\mu \circ \eta T = 1_T$, so 

Likewise, $\mu \circ T\eta$ means .

To parse $\mu \circ \mu T$, we note μT looks like ,

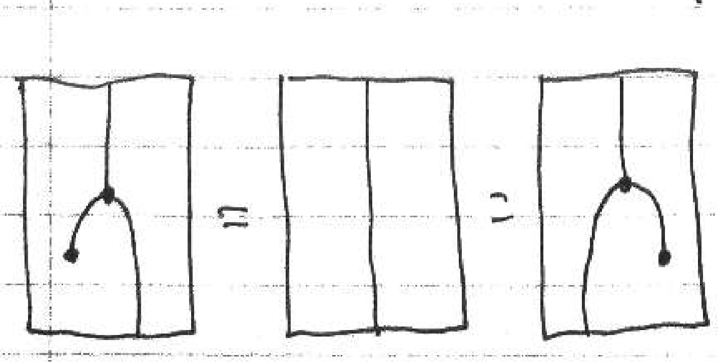
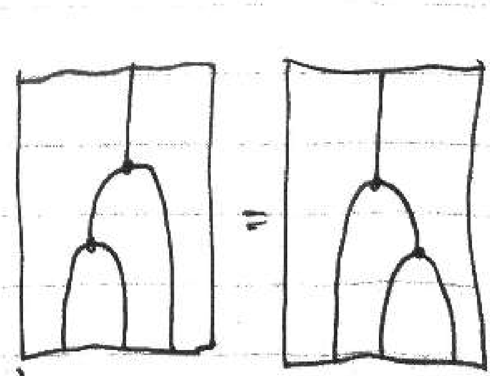
and when stacked as , we obtain .

Likewise, $\mu \circ T\mu$ looks like .

So, again, the commutative diagrams defining a monad,

$$
\begin{tikzcd}
T \arrow[r, "\eta T"] \arrow[dr, "1_T"'] & T^2 \arrow[d, "\mu"] & T \arrow[l, "T\eta"'] \arrow[dl, "1_T"] \\
 & T &
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
T^3 \arrow[r, "\mu T"] \arrow[d, "T\mu"'] & T^2 \arrow[d, "\mu"] \\
T^2 \arrow[r, "\mu"] & T
\end{tikzcd}
$$

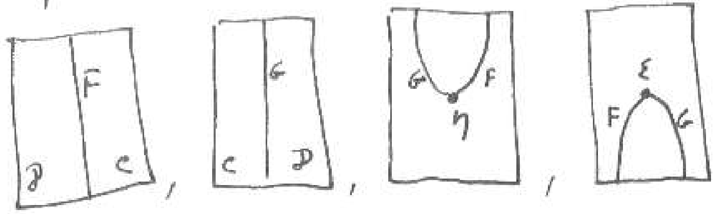
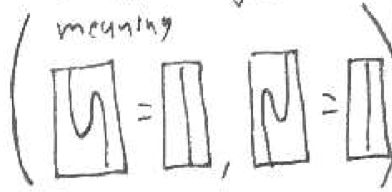
are equivalent to the equations:

 and 

(the lables can be recovered with some thought).

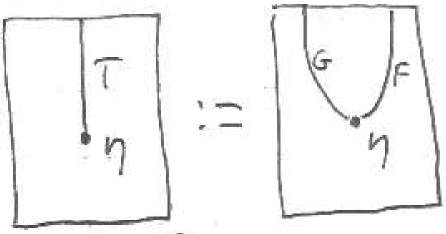
Last but not least, we will swiftly prove the theorem that the composition of a left, then right adjoint is a monad.

18

proof: Let  be an adjunction (meaning )

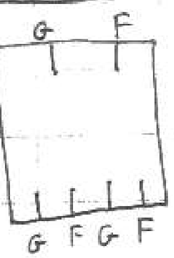
Let $T = GF$. We need an η and μ for T.

Let $\eta : 1_{\mathcal{C}} \to T$ just be the η given by the adjunction:

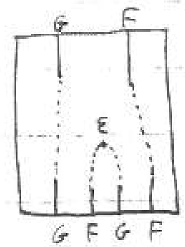


We need $\mu : T^2 \to T$, which is $GFGF \to GF$

Heres a trick. Draw a string diagram

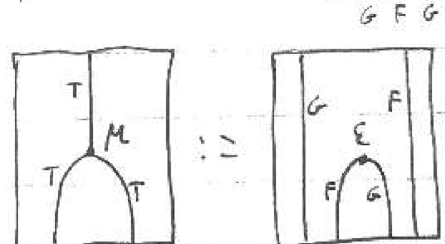


with a missing middle. now, it should be visually clear that

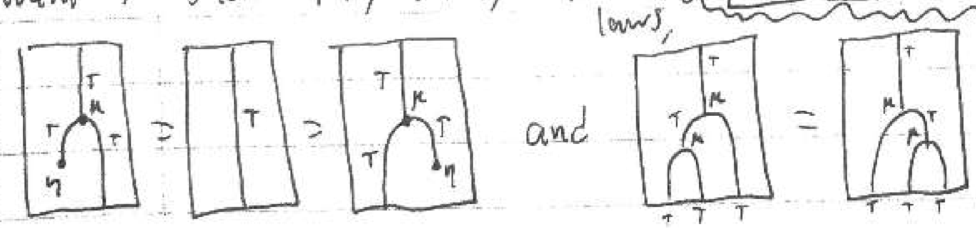


works.

So we define μ as $G\varepsilon F$:

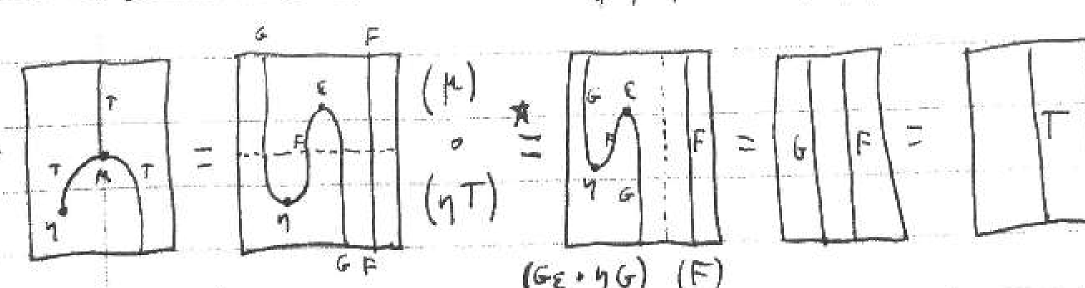


With η and μ defined, we want to show they satisfy the monad laws,

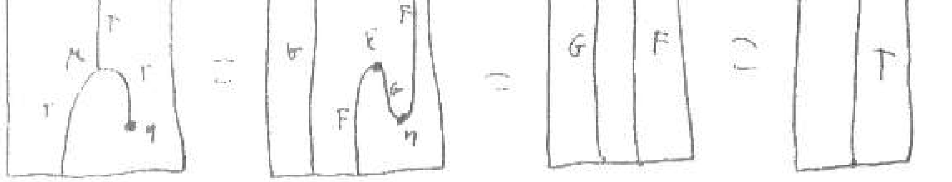


If so, then $T = GF$ is a monad on $\mathcal{C}$.

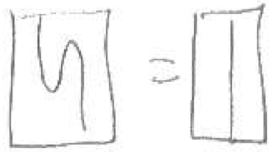
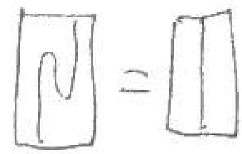
We have



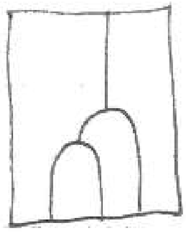
the dashed line is indicating how these strings are being grouped. This calculation implicitly used the interchange law at ☆.

Likewise 

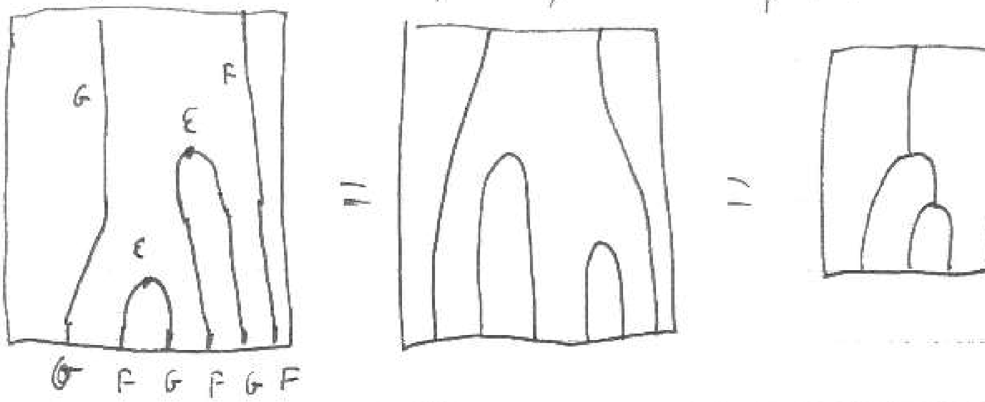
19

In both cases, we use that  or ~~rather~~ , which is true since $F \dashv G$.

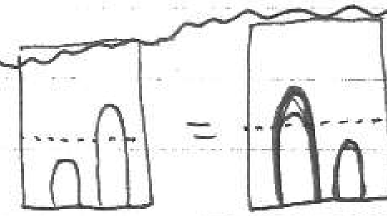
To show μ satisfies "associativity", we expand



as



Thus, we are done, and GF is a monad



* to be fully precise, the middle part are both $\varepsilon * \varepsilon$, even though the first was given as $(1_D \varepsilon)\circ(\varepsilon FG)$, and the second by $(\varepsilon 1_D)\circ(FG\varepsilon)$. Notice also, $F(G\varepsilon) = (FG)\varepsilon$. The interchange law is the ultimate source of all of these equalities, and with string diagrams, they become visually clear.

As a challenge, and also to illustrate the point, try to prove that GF is a monad without appealing to string diagrams, by using just the commutative diagrams, or (as an extra challenge), just reasoning with equations.

Conclusion: when dealing with complicated expressions involving functors, natural transformations, and compositions thereof, string diagrams form a natural and intuitive language that eases the burden of complexity and formalism associated with category theory. They are, at least, useful tools. Enjoy!