Journal of Logical and Algebraic Methods in Programming 85 (2016) 931–951

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Equational reasoning with lollipops, forks, cups, caps, snakes, and speedometers

Ralf Hinze *, Dan Marsden

Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, England, United Kingdom

ARTICLE INFO

Article history:
Received 2 June 2015
Received in revised form 9 September 2015
Accepted 18 December 2015
Available online 25 January 2016

Keywords:
String diagram
Calculational proof
Category theory
Adjunction
Resumption monad

ABSTRACT

Proofs in elementary category theory typically involve either the pasting together of commuting diagrams or calculational reasoning using chains of equalities. Much of the effort in these styles can be consumed with trivial administrative steps involving functoriality, naturality, and the handling of identities. As an alternative, we advocate the use of string diagrams, a two-dimensional form of notation which silently deals with these distracting bookkeeping steps. While originally developed to reason about functors and natural transformations, we show that string diagrams can also easily accommodate objects and arrows. Throughout the paper we emphasize how the topological freedom inherent in the notation can be exploited to aid the use of geometric intuition in the development of proofs. Indeed, drawing string diagrams is a bit of an art: good diagrammatic choices can make all the difference.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Proofs in elementary category theory typically involve either the pasting together of commuting diagrams [7] or calculational reasoning using chains of equalities [4]. The first style is sometimes referred to as “diagram chasing” since the focus of a proof is chased around a diagram; the second is also known as “squiggolling” since it often involves the use of “squiggly” symbols.

Both styles have their merits. Commutative diagrams capture an abundance of type information, and invoke a certain amount of visual intuition. Equational proofs are familiar from many other branches of science and mathematics. They are also compact and carry a clear orientation from assumptions to goals.

Unfortunately, both styles also have serious limitations. The usual equational style of reasoning forces us to abandon the vital type information that guides proof attempts and protects against errors. Further, much of the effort in these proofs can be consumed with trivial administrative steps involving functoriality, naturality, and the introduction and elimination of identities. Commuting-diagram-style proofs retain the type information, but the proof style is unfamiliar to many in other fields of mathematics and computer science, and the resulting proofs often lack motivation and a clear direction of argument. We are often led towards proofs that reason using natural transformations component-wise, again leading to distracting administration of naturality conditions.

In order to recover the best of both these approaches, we advocate the use of *string diagrams* [8], a two-dimensional form of notation, that retains the vital type information whilst permitting an equational style of reasoning. This notation

* Corresponding author.
E-mail addresses: ralf.hinze@cs.ox.ac.uk (R. Hinze), daniel.marsden@cs.ox.ac.uk (D. Marsden).

http://dx.doi.org/10.1016/j.jlamp.2015.12.004
2352-2208/© 2016 Elsevier Inc. All rights reserved.

silently deals with distracting bookkeeping steps, such as naturality and functoriality issues, leaving us to concentrate on the essentials. This is an important aspect in any choice of notation, as advocated by Backhouse [1]. The resulting diagrams and proofs are highly visual and we can often exploit topological intuition to identify suitable steps in our reasoning. Indeed, developing and drawing string diagrams is a bit of an art: good diagrammatic choices can make the difference between a tangled knot and a regular weave.

Our approach will be by example; after introducing string-diagrammatic concepts in Section 2 we will apply them to calculations in elementary category theory, involving classical concepts such as monads and distributive laws in Section 3, and adjunctions and adjoint squares in Section 5. String diagrams were initially developed to reason about functors and natural transformations. In Section 4 we show how to incorporate objects and arrows into the string-diagram notation, using algebras and homomorphisms as running examples. We present a worked-out application in Section 6, revolving around the so-called resumption monad, and conclude in Section 7.

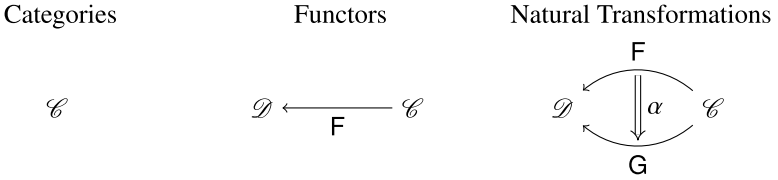
Finally, we should emphasize that we concentrate on the categorical concepts. In particular, we do not provide any applications of the constructions introduced.

2. Basics of string diagrams

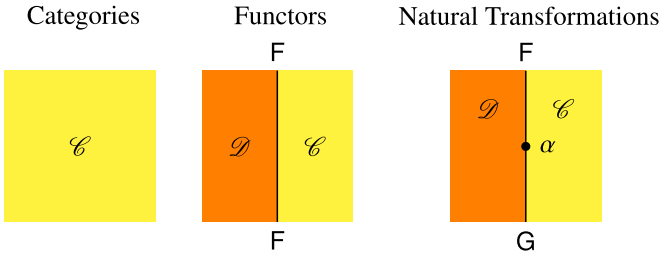
Elementary category theory is concerned with categories, functors, and natural transformations. As described in Mac Lane [7]:

> … "category" has been defined in order to be able to define "functor" and "functor" has been defined in order to be able to define "natural transformation".

The traditional notation for these entities represents categories by vertices, functors by arrows between vertices, and natural transformations as double arrows in the region between functor arrows, as illustrated in the following table:

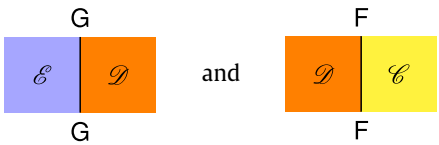


String diagrams are the *topological* or *Poincaré dual* of the traditional notation. Natural transformations are now vertices, functors are edges connecting natural transformations, and categories are regions between the functor edges:

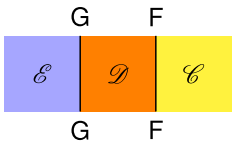


In a sense, string diagrams return the visual focus to the conceptually more significant natural transformations. This seemingly trivial change in perspective provides category theory with a new and very distinctive visual flavor. We aim to demonstrate that string diagrams provide an effective tool for equational reasoning about elementary category theory.

In order to approach practical problems, it is important to be able to use *composition* to construct more complex structures and equations between them. Given functors $\mathsf{F} : \mathscr{C} \to \mathscr{D}$ and $\mathsf{G} : \mathscr{D} \to \mathscr{E}$, in pictures:



their composite $\mathsf{G} \circ \mathsf{F} : \mathscr{C} \to \mathscr{E}$ is drawn horizontally as follows:



Notice how the graphical direction of composition follows the direction in the symbolic notation $G \circ F$. As a special case, we represent the identity functor on a category $\mathscr{D}$ as the corresponding region, without any associated edge[1]:



In this way, composing with identity functors "does nothing" in the diagrammatic notation, and so the equations between functors:

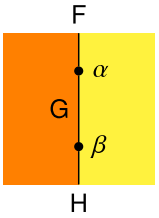
$$\mathsf{Id} \circ F = F = F \circ \mathsf{Id}$$

are handled silently by the notation.

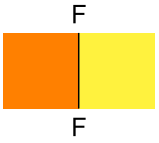
We now consider composition of natural transformations. Given natural transformations:



we can form their *vertical composite*, denoted $\beta \cdot \alpha : F \xrightarrow{\cdot} H$. In string diagrams, we adopt the convention that vertical composition is ordered from top to bottom, as in the following diagram:



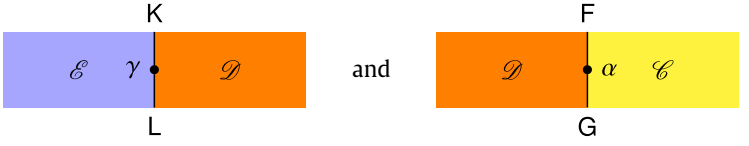
Similarly to the notation for identity functors, identity natural transformations are represented by the edge for the corresponding functor.



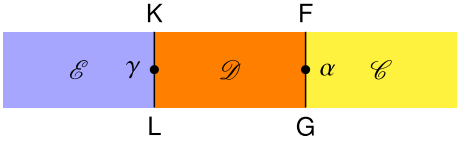
In this way, vertically composing with an identity natural transformation "does nothing" in the string notation, silently capturing the equations between natural transformations:

$$id \cdot \alpha = \alpha = \alpha \cdot id$$

There is a second form of composition for natural transformations. Consider natural transformations:



Generalizing the notion of composite for functors, we can form their *horizontal composite*, denoted $\gamma \circ \alpha : K \circ F \xrightarrow{\cdot} L \circ G$. Again the direction of composition in the string diagram parallels the symbolic notation:



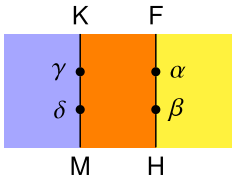
To reduce clutter, from this point on we will usually avoid labeling categories explicitly in diagrams, particularly if they are clear from the surrounding context or if they are genuinely uninteresting.

[1] There is a similar convention in one-dimensional notation where the identity arrow of an object is denoted by the name of the object itself.

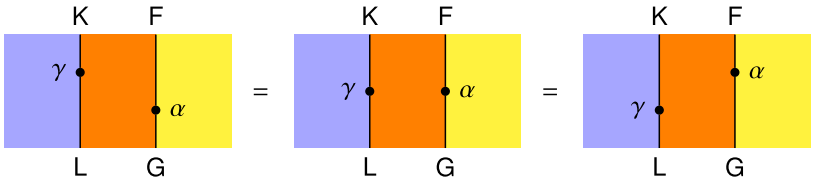
These two forms of composition are related by the important *interchange law*:

$$(\delta \cdot \gamma) \circ (\beta \cdot \alpha) = (\delta \circ \beta) \cdot (\gamma \circ \alpha)$$

Graphically, this law is subsumed by the notation, simply stating that the relative nesting of horizontal and vertical composition does not matter. The following diagram therefore denotes a well-defined natural transformation:



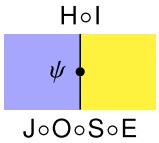
By combining the rules for identities and the interchange law appropriately, we obtain the important *sliding equations*, allowing us to move natural transformations up and down wires in an intuitive manner:



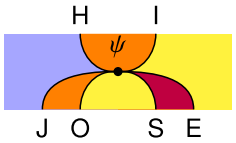
(1)

We shall see in Section 4 that the sliding equations generalize the naturality condition. But we are leaping ahead.

Finally we consider how to draw natural transformations between composite functors. For example, we could naively draw the natural transformation $\psi : \mathsf{H} \circ \mathsf{I} \dot{\to} \mathsf{J} \circ \mathsf{O} \circ \mathsf{S} \circ \mathsf{E}$ as:



This diagram is unsatisfactory as it does not make the structure of the input and output functors explicit, and may restrict how we compose and manipulate ψ in our diagrams. Alternatively, we can draw our diagram with each of the individual component functors connected as a separate input or output to ψ, resembling the legs of, well, perhaps a small spider:



This way of drawing ψ is much more flexible as it allows us to access the individual functors, H, I, J, O, S, and E, directly in our calculations. Observe that the legs are drawn as bends, rather than as straight lines. In general, edges can be arbitrary curves with the important restriction that they *must not have a horizontal tangent*.

As an intermediate summary, edges in string diagrams are implicitly directed, and we stipulate that the flow is from right to left for horizontal composition $\gamma \circ \alpha$, and from top to bottom for vertical composition $\alpha \cdot \beta$. Diagrams that differ only in the vertical position of natural transformations are identified—this is the import of the sliding equations. More generally, string diagrams that are equivalent up to *planar isotopy* denote the same natural transformation. A planar isotopy is a continuous deformation of a plane diagram that preserves cusps, crossings, and the property of having no horizontal tangents.

We now have enough machinery to start investigating some examples.

3. Monads and distributive laws

We start with diagrams involving a single category in Section 3.1. These diagrams are “monochrome” as only a single type of region is involved. We then move on to richer diagrams with multiple distinct region types in the subsequent sections.

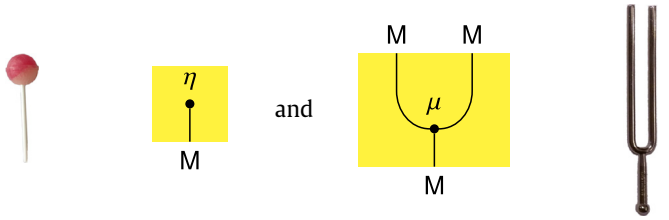
3.1. Monads

A monad consists of an endofunctor $\mathsf{M} : \mathscr{C} \to \mathscr{C}$ and natural transformations:

$$\eta : \mathsf{Id} \dot{\to} \mathsf{M}$$

$$\mu : \mathsf{M} \circ \mathsf{M} \dot{\to} \mathsf{M}$$

Drawn as a string diagram η looks a tad like a lollipop—recall that the identity functor is drawn as a region—whereas μ resembles a tuning fork:

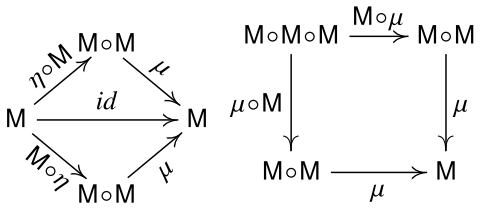


The two natural transformations are required to satisfy the following unit and associativity axioms:

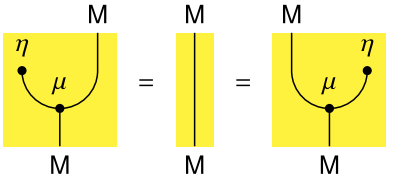
$$\mu \cdot (\eta \circ \mathsf{M}) = id = \mu \cdot (\mathsf{M} \circ \eta) \tag{2}$$

$$\mu \cdot (\mu \circ \mathsf{M}) = \mu \cdot (\mathsf{M} \circ \mu) \tag{3}$$

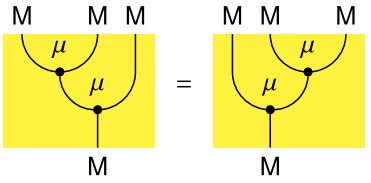
Written algebraically, these axioms are not particularly instructive. The corresponding commutative diagrams are not a lot clearer, even though they add vital type information:



The equivalent string diagrams, however, are much more revealing:



(4)



(5)

Diagrams (4) and (5) make plain the idea that monads can be seen as monoids in an appropriate way, as hinted by the suggestive names unit and multiplication for η and μ.

Alternatively, the natural transformations can be seen as constructors of binary leaf trees: η creates a leaf and μ creates a binary node. The monad laws then correspond to transformations on binary trees: the unit laws allow us to prune or to add leaves and the associative law captures a simple tree rotation.

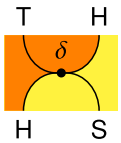
3.2. Distributive laws

Generally monads cannot be composed, but in the presence of a suitable natural transformation, commonly referred to as a *distributive law* [2], a composite monad can be constructed. As a first application of string-diagrammatic reasoning, we will now investigate this construction, proceeding in two steps. Firstly, in this section, we consider asymmetric laws, natural transformations that distribute a monad over a plain functor. Secondly, in the next section, we show how a symmetric law, a natural transformation that distributes one monad over a second monad, allows us to solve the original problem.

Given a functor $\mathsf{H} : \mathscr{C} \to \mathscr{D}$ and a pair of monads:

$$\begin{array}{ll} \mathsf{S} : \mathscr{C} \to \mathscr{C} & \mathsf{T} : \mathscr{D} \to \mathscr{D} \\ \eta : \mathsf{Id} \dot{\to} \mathsf{S} & \eta : \mathsf{Id} \dot{\to} \mathsf{T} \\ \mu : \mathsf{S} \circ \mathsf{S} \dot{\to} \mathsf{S} & \mu : \mathsf{T} \circ \mathsf{T} \dot{\to} \mathsf{T} \end{array}$$

an Eilenberg–Moore-law, EM-law for short, is a natural transformation:



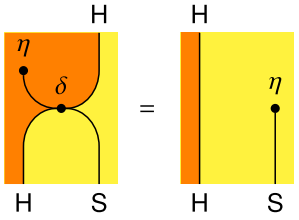
The choice of name will become apparent in Section 4.2. As an aside, when dealing with multiple monads, we overload the letters η and μ for the units and multiplications of the monads. The type information in the diagrams should resolve any ambiguity about which natural transformations are intended.

The natural transformation δ is required to satisfy the following coherence conditions:

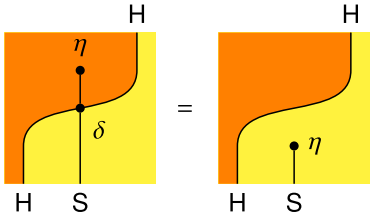
$$\delta \cdot (\eta \circ \mathsf{H}) = (\mathsf{H} \circ \eta) \tag{6}$$

$$\delta \cdot (\mu \circ \mathsf{H}) = (\mathsf{H} \circ \mu) \cdot (\delta \circ \mathsf{S}) \cdot (\mathsf{T} \circ \delta) \tag{7}$$

Written algebraically like this, these equations are not particularly memorable or well motivated. As the diagrams in equations (4) and (5) helped clarify the significance of those axioms, we should see what string diagrams reveal in this case. If we draw the string diagram corresponding to equation (6), we might produce the following picture:



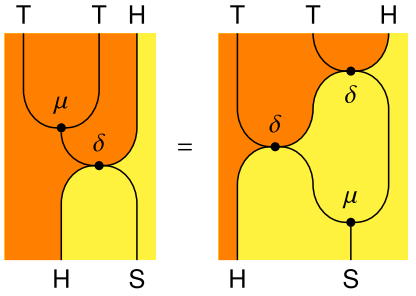
The diagram is possibly slightly clearer than the original algebraic formulation, but it is still not entirely perspicuous. We have somewhat naively drawn δ in a very symmetrical manner. In order to improve our understanding further we must think slightly less "rigidly". We can exploit the topological freedom afforded by the string diagram notation to slightly deform our diagram. This leads to the following equivalent, but much more intuitive form:



(8)

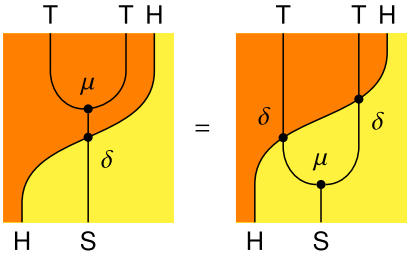
It is now clear that this axiom allows us to "drag" the unit of each monad, the lollipop, across the functor H. As an aside, observe how the string diagrams resolve the overloading of the identifier η: the regions make clear where each η lives.

If we turn our attention to the remaining axiom (7), it is equivalent to the following equation between string diagrams:



Again we have drawn the diagram naively, maintaining an obvious symmetrical structure to each of the natural transformations. This depiction is neutral in terms of its emphasis, but it is unwise to pursue this neutrality, as it obscures the significance of the axiom. And again, if we exploit the freedom of the string diagram notation, we can deform the diagram

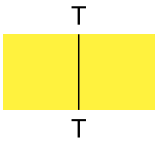
to produce a more instructive presentation:



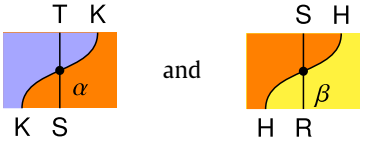
(9)

In a similar way to the unit axiom, we now see that we can also "drag" the multiplication of each monad, the fork, across the functor H. The functor H can be seen as a "static" component of our diagram. Our axiom involves dragging μ downwards whilst the functor H remains fixed. As many axioms involve the vertical movement of various natural transformations, experience shows that static components are best drawn as straight diagonals. Again as an aside, note how the regions resolve the ambiguity in naming both multiplications μ.

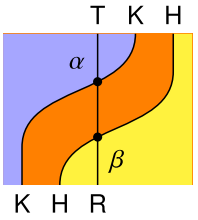
Monads and EM-laws form a category. The identity on the monad T is given by $id : \mathsf{T} \circ \mathsf{Id} \dot{\to} \mathsf{Id} \circ \mathsf{T}$:



The identity trivially satisfies the coherence conditions (8)–(9). The composition of two EM-laws:

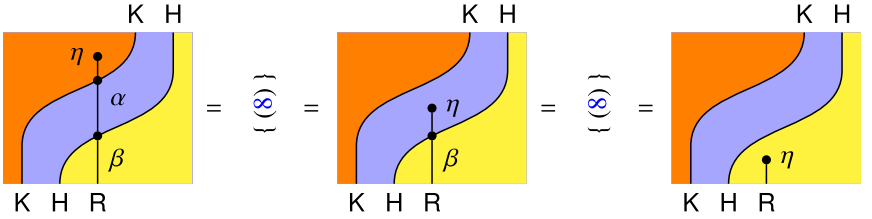


is formed by adjoining the laws vertically:

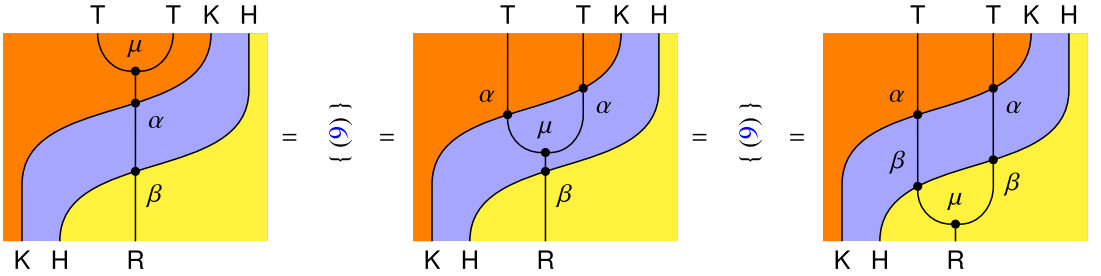


(10)

The unit axiom (8) for the composite then follows from the equalities:



We drag the lollipop twice, first across the functor K and then a second time across the functor H. We proceed in an analogous fashion to establish the multiplication axiom (9), dragging the fork twice:



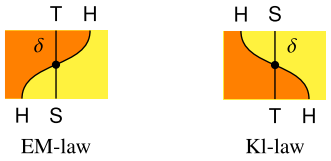
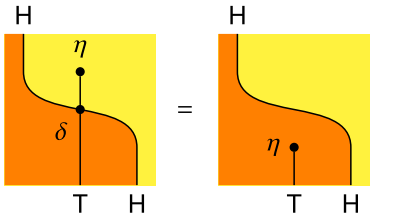
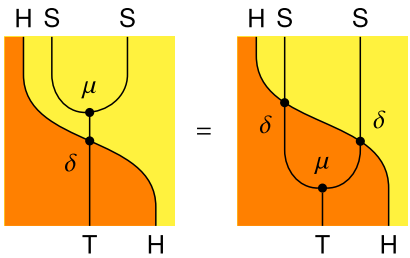


Fig. 1. Two flavors of distributive laws: EM-laws and Kl-laws.

There are, in fact, at least two ways to set up a category whose objects are monads. For completeness, we briefly sketch the "dual" approach. Observe that the static component for an EM-law is drawn as a "minor" diagonal; there is also a second type of law, whose static component is rendered as a "major" diagonal, see Fig. 1. The coherence conditions for the so-called Kleisli-law, Kl-law for short, are left-to-right mirror images of the diagrams (8) and (9):



(11)



(12)

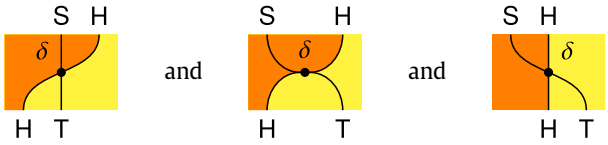
There is an identity Kl-law, and Kl-laws compose. Hence, monads and Kl-laws form a category as well.

String diagrams extend over two dimensions and hence enjoy two forms of symmetry. Consider again Fig. 1. If we reflect the diagram for an EM-law horizontally, we obtain the diagram for a Kl-law, and vice versa. Moreover, also vertical reflection turns one type of diagram into the other. Vertical reflection is meaningful if the natural transformations involved have inverses. For example, assume that $\delta : \mathsf{T} \circ \mathsf{H} \xrightarrow{\cdot} \mathsf{H} \circ \mathsf{S}$ is a natural isomorphism with inverse δ°. Then one can show that

$$\delta \text{ is an EM-law if and only if } \delta^\circ \text{ is a Kl-law} \tag{13}$$

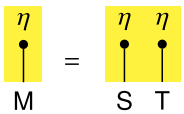
The instructive proof is left as an exercise to the reader. *Hint:* show that δ satisfies (8) if and only if δ° satisfies (11), and likewise for (9) and (12).

We observe that carefully varying how δ was drawn between diagrams allowed us to emphasize the "moves" that were being described. In this way, *artistic choices matter* in string diagram notations. Good diagrammatic choices can make the difference between a cryptic muddle of wires and an intuitive notion such as the dragging enabled by the distributive law axioms. For example, there are three potential depictions of an EM-law $\delta : \mathsf{S} \circ \mathsf{H} \to \mathsf{H} \circ \mathsf{T}$, depending on whether we wish to emphasize the role of the monads S and T, neutrality, or the role of the functor H.

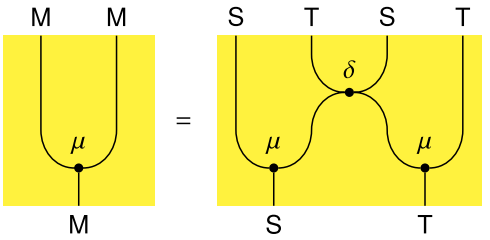


3.3. Composition of monads

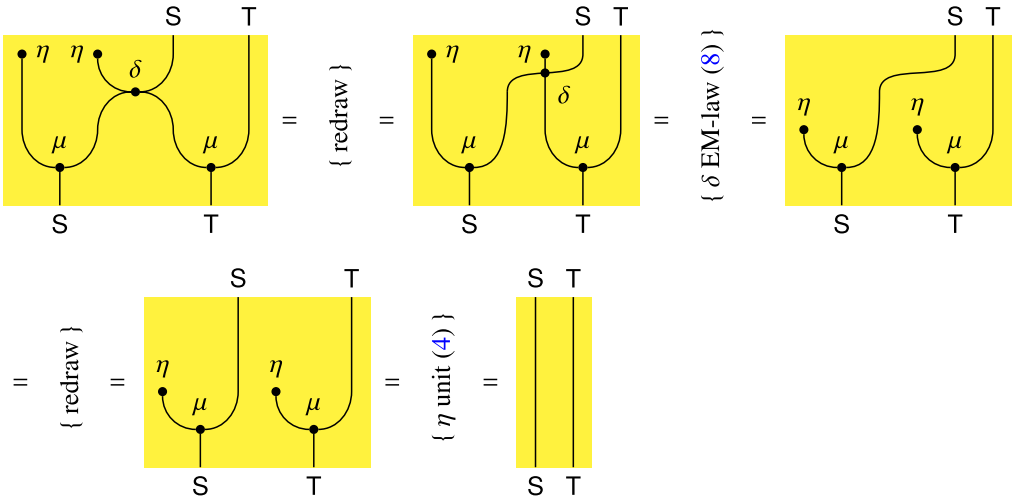
Given a so-called *distributive law* $\delta : \mathsf{T} \circ \mathsf{S} \xrightarrow{\cdot} \mathsf{S} \circ \mathsf{T}$, a natural transformation that is simultaneously an EM-law and a Kl-law, we can solve our problem of composing S and T. We define the functor of our composite monad as $\mathsf{M} = \mathsf{S} \circ \mathsf{T}$ and take the unit as:



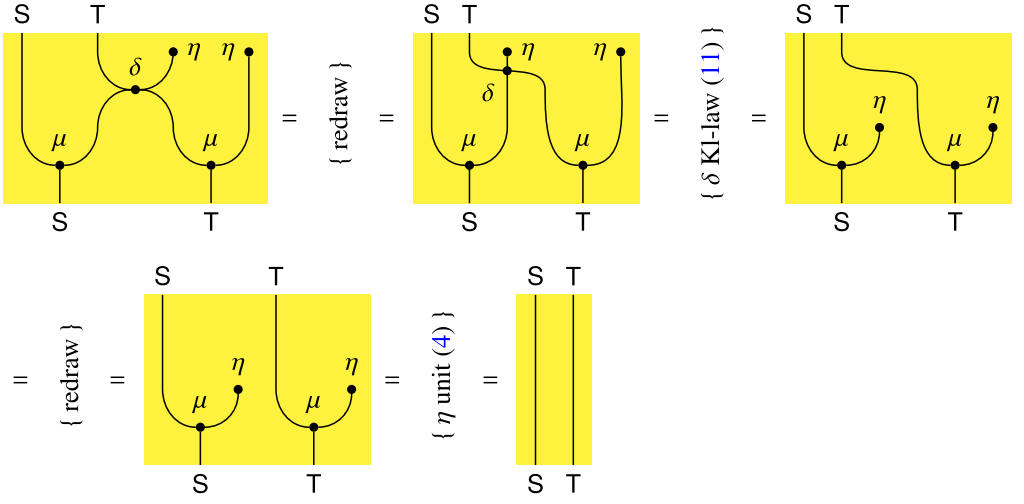
The multiplication makes essential use of the distributive law to manoeuvre the functors S and T into the correct positions:



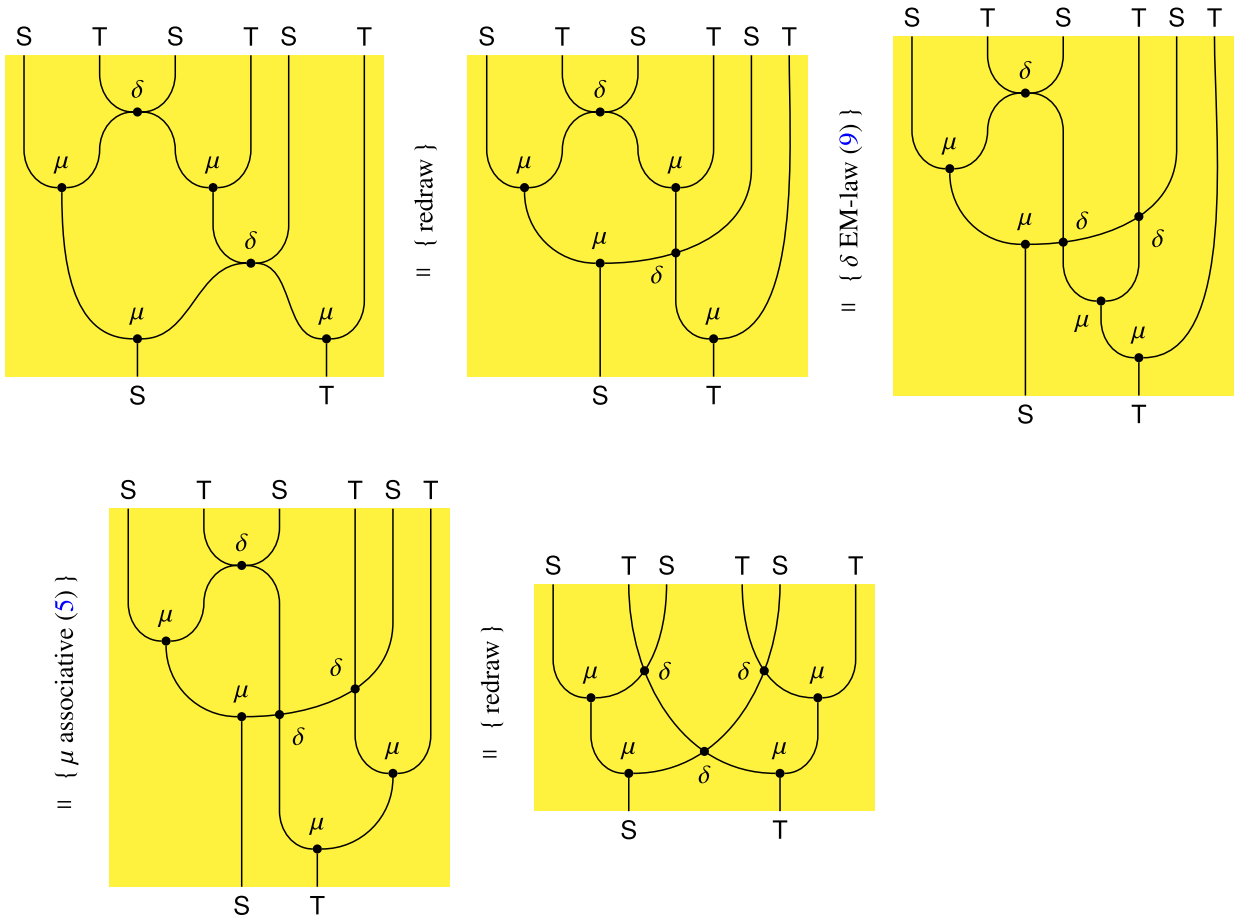
The distributive law axioms (8), (9), (11), and (12) will prove essential in establishing that this construction provides a genuine monad. We first check that the left unit law, the first equation of (4), is satisfied:



The right unit law, the second equation of (4), follows similarly:



In both proofs as a first step we redraw the diagram to emphasize the axiom to be applied. These clarifying redrawing operations become even more helpful in the proof of associativity:



We have transformed our original diagram into a symmetrical position using the properties of monads and distributive laws. These axioms are symmetric in the monads S and T and so we can now transform our final diagram using the dual axioms to complete our proof.

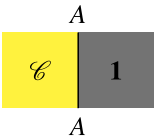
4. Recovering objects and arrows

We have seen how to reason graphically about categories, functors, and natural transformations. When doing elementary category theory, we generally need to refer to objects in categories, and to arrows between these objects. We now consider how these components can be incorporated into the string-diagram notation.

Let **1** denote the category with one object and one arrow. As usual, this category is displayed in string diagrams as a region:



We reserve this type of region for the category **1**, so that it can be easily identified in subsequent diagrams. A functor of type $\mathbf{1} \to \mathscr{C}$ then picks out an object, and corresponding identity arrow in $\mathscr{C}$. Therefore, via a slight abuse of notation, we also denote the functor that picks out the object A in $\mathscr{C}$ as $A : \mathbf{1} \to \mathscr{C}$, and draw this graphically as:



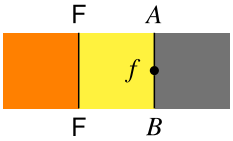
Let $\mathscr{C}$ and $\mathscr{D}$ be two arbitrary categories, and $\mathsf{F}, \mathsf{G} : \mathscr{C} \to \mathscr{D}$ two functors between them. We recall that a natural transformation $\alpha : \mathsf{F} \dot{\to} \mathsf{G}$ is a family of $\mathscr{D}$-arrows indexed by the objects in $\mathscr{C}$, so-called components $\alpha A : \mathsf{F}A \to \mathsf{G}A$, such that for each arrow $f : A \to B$ in $\mathscr{C}$:

$$\mathsf{G}f \cdot \alpha A = \alpha B \cdot \mathsf{F}f \tag{14}$$

Now consider the special case of a natural transformation between two functors $A, B : \mathbf{1} \to \mathscr{C}$. Such a natural transformation must have just one component, of type $A \to B$. The naturality condition (14) becomes trivial, and so a $\mathscr{C}$-arrow $f : A \to B$ is "the same thing" as natural transformation of type $A \dot{\to} B$. Using a similar abuse of notation to that for objects, we denote this natural transformation $f : A \dot{\to} B$. The key idea is that *objects are a special case of functors* and that *arrows are a special case of natural transformations*. We can therefore draw an arrow $f : A \to B$ in string diagram notation simply as:



Using this diagrammatic device we can now easily combine functors and arrows. The functor application $\mathsf{F} f$ is rendered:

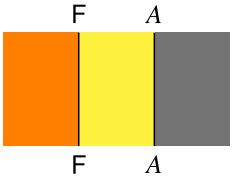


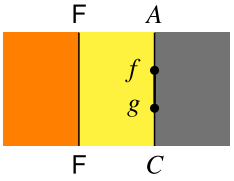
Quite pleasingly, the functor laws

$$\mathsf{F}\, id = id \tag{15}$$

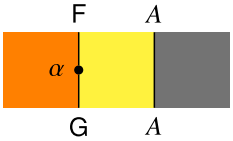
$$\mathsf{F}(f \cdot g) = \mathsf{F} f \cdot \mathsf{F} g \tag{16}$$

are built into the notation, in the sense that both sides of each equation correspond to the same diagram:

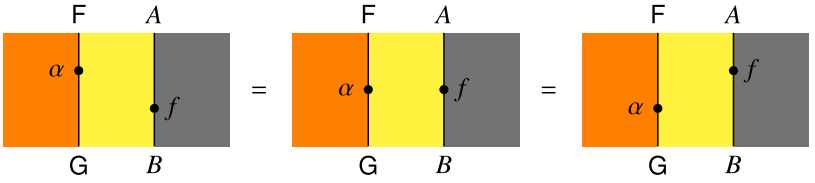




Likewise, we depict the component $\alpha\, A$ of a natural transformation:



The diagrams for $\mathsf{F} f$ and $\alpha\, A$ exhibit the symmetry between the two concepts: both consist of two parallel lines, one of which has a "bead" on it. We have noted above that the string diagrams silently deal with applications of the functor laws. The same holds true of the naturality condition (14) as diagrams that differ only in the vertical position of natural transformations are identified:



We observe that these diagrams are instances of the sliding equations (1) that we have encountered in Section 2. In each of the sliding diagrams the natural transformation on the right is replaced by an arrow, a single-component natural transformation. As an aside, the diagram in the middle above also has categorical significance: functor application, $\mathsf{F} A$, is itself functorial; αf is the action on arrows for this higher-order functor.

4.1. Categories of algebras and Eilenberg–Moore categories

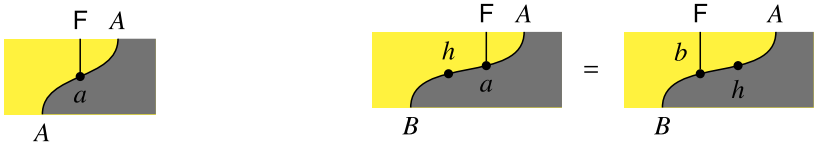
As an example of manipulating "ordinary" arrows using string diagrams, we consider algebras and their homomorphisms. Given an endofunctor

$$\mathsf{F} : \mathscr{C} \to \mathscr{C}$$

an F-algebra is a pair (A, a), where A is an object in $\mathscr{C}$ and $a : \mathsf{F}A \to A$ is an arrow in $\mathscr{C}$, which are known as the *carrier* and *action* of the algebra. An F-homomorphism between algebras (A, a) and (B, b) is an arrow $h : A \to B$ in the underlying category $\mathscr{C}$ such that

$$h \cdot a = b \cdot \mathsf{F}\,h \tag{17}$$

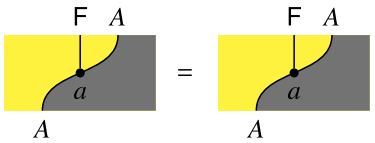
Turning to the string-diagram representation, the carrier of an algebra can be seen as static, which is why it is drawn as a "minor" diagonal.



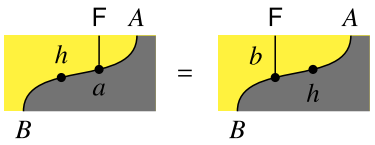
(18)

The homomorphism condition (17) has an appealing visual interpretation: it allows us to slide h along the wire across the action a, which is transmogrified into b in the process.

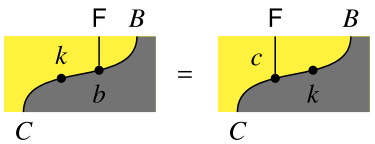
To see that the identity $id : A \to A$ is an F-algebra homomorphism, we have to show: $a \cdot \mathsf{F}\,id = id \cdot a$. In string diagrams this reduces to the tautology:



As composition is performed in the underlying category, it follows immediately that $id \cdot h = h$ and $h \cdot id = h$. If we consider a pair of homomorphisms h and k:

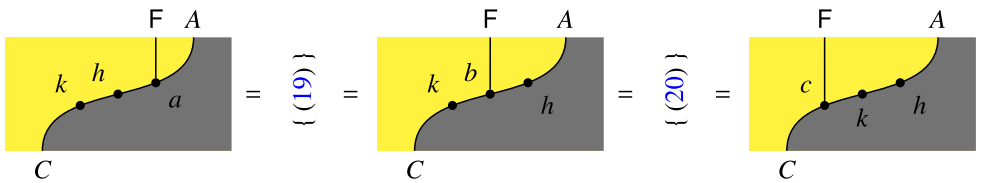


(19)



(20)

we then see that F-homomorphisms compose as:



We first slide h across a, turning it into b, and then slide k across b, turning it into c. Since F-homomorphisms compose and have an identity, it follows that F-algebras and F-homomorphisms form a category, which is denoted F-**Alg**($\mathscr{C}$).

Next we consider algebras for a monad. Given a monad

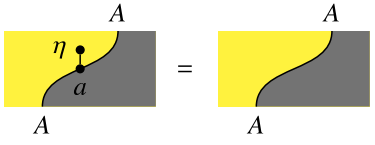
$$\mathsf{M} : \mathscr{C} \to \mathscr{C}$$

an algebra for M is an M-algebra (A, a) that satisfies the coherence conditions:

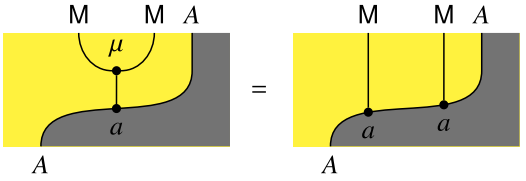
$$a \cdot \eta A = id$$

$$a \cdot \mu A = a \cdot \mathsf{M}\,a$$

Again it is instructive to turn the algebraic formulation into string diagrams.



(21)



(22)

The diagrams are reminiscent of the diagrams for EM-laws, except that unit and multiplication are now "absorbed" by the diagonal, leaving zero or two instances of the action a behind. This similarity is not a coincidence, as we shall see in the next section. The category of algebras for M and M-homomorphisms is known as the Eilenberg–Moore category of M, denoted $\mathscr{C}^{\mathsf{M}}$.

4.2. Lifting functors

A category of algebras equips a given category with additional structure. The forgetful functor $\mathsf{U_F} : \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C}) \to \mathscr{C}$ defined

$$\mathsf{U_F}\,(A, a) = A \tag{23}$$

$$\mathsf{U_F}\,(h) = h \tag{24}$$

forgets about this additional structure, mapping an algebra to its carrier and a homomorphism to its underlying arrow.

In this section we investigate how to "lift" a functor between underlying categories to a functor between categories of algebras. Given a functor $\mathsf{H} : \mathscr{C} \to \mathscr{D}$ and a pair of endofunctors

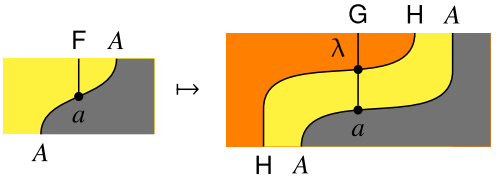
$$\mathsf{F} : \mathscr{C} \to \mathscr{C} \qquad \mathsf{G} : \mathscr{D} \to \mathscr{D}$$

a functor $\bar{\mathsf{H}} : \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C}) \to \mathsf{G}\text{-}\mathbf{Alg}(\mathscr{D})$ is a *lifting* of $\mathsf{H} : \mathscr{C} \to \mathscr{D}$ if

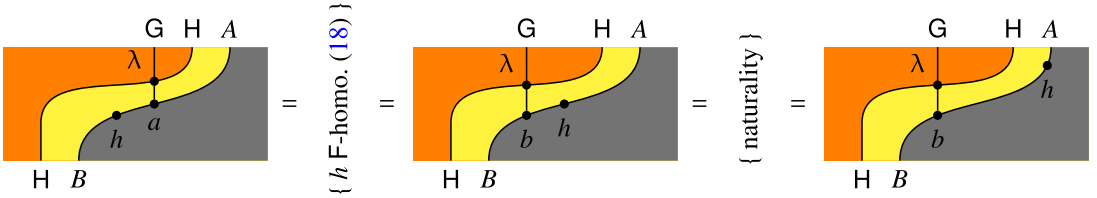
$$\mathsf{H} \circ \mathsf{U_F} = \mathsf{U_G} \circ \bar{\mathsf{H}}$$

This property determines the action of $\bar{\mathsf{H}}$ on carriers and homomorphisms. In other words, different liftings differ only in their action on the action part of an algebra.

One possible approach to fix the action is to make use of a natural transformation $\lambda : \mathsf{G} \circ \mathsf{H} \dot{\to} \mathsf{H} \circ \mathsf{F}$, called a *vanilla law* for emphasis:



Roughly speaking, H is inserted between F and A turning F into G. The diagram nicely visualizes how an F-algebra with carrier A is mapped to a G-algebra with carrier $\mathsf{H}A$. The lifting induced by λ is denoted $\mathsf{H}^{\lambda} : \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C}) \to \mathsf{G}\text{-}\mathbf{Alg}(\mathscr{D})$. Since $\mathsf{H}^{\lambda}\,h = \mathsf{H}\,h$, we know that the lifting preserves identity and composition. It remains to show that H^{λ} maps F-homomorphisms to G-homomorphisms. The proof that $\mathsf{H}^{\lambda}\,h$ is a G-homomorphism is actually straightforward, merely applying naturality and the assumption that h is an F-homomorphism:



In a sense, h simply operates "below" the functor H.

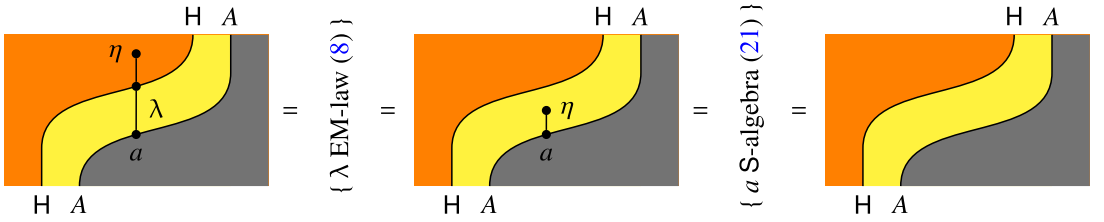
Finally, we turn to Eilenberg–Moore categories and liftings between them. Given a pair of monads:

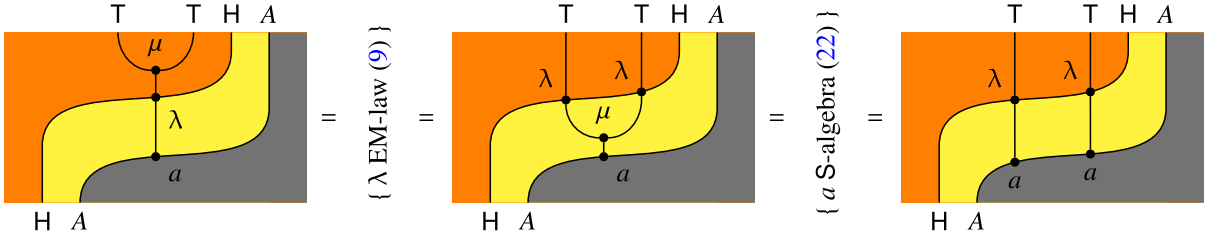
$$\begin{array}{ll} \mathsf{S} : \mathscr{C} \to \mathscr{C} & \mathsf{T} : \mathscr{D} \to \mathscr{D} \\ \eta : \mathsf{Id} \dot{\to} \mathsf{S} & \eta : \mathsf{Id} \dot{\to} \mathsf{T} \\ \mu : \mathsf{S} \circ \mathsf{S} \dot{\to} \mathsf{S} & \mu : \mathsf{T} \circ \mathsf{T} \dot{\to} \mathsf{T} \end{array}$$

we aim to lift $\mathsf{H} : \mathscr{C} \to \mathscr{D}$ to a functor $\mathscr{C}^{\mathsf{S}} \to \mathscr{D}^{\mathsf{T}}$ between corresponding Eilenberg–Moore categories. Quite pleasingly, we can re-use the machinery introduced above. However, one additional proof obligation arises: we have to show that the

lifted functor maps algebras for S to algebras for T. It turns out that this requirement is met if the natural transformation $\lambda : \mathsf{T} \circ \mathsf{H} \dot{\to} \mathsf{H} \circ \mathsf{S}$ is an EM-law—finally motivating the choice of name.

The proofs of (21) and (22) are visually appealing. They exploit the corresponding properties for the underlying S-algebra, and the EM-law axioms:





We first drag the unit and the multiplication across the functor H, and then apply the assumption that a respects η and μ.

4.3. Lifting natural transformations

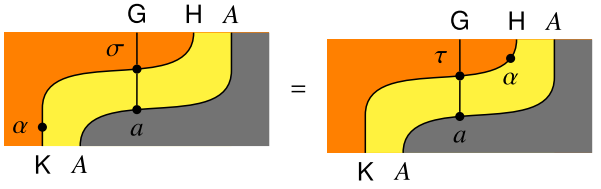
We have equipped categories with additional algebraic structure and then lifted functors to these categories. The next logical step is to lift natural transformations to the higher realm.

Let $\mathsf{H}, \mathsf{K} : \mathscr{C} \to \mathscr{D}$ be a parallel pair of functors, and let $\bar{\mathsf{H}}, \bar{\mathsf{K}} : \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C}) \to \mathsf{G}\text{-}\mathbf{Alg}(\mathscr{D})$ be a corresponding pair of liftings. A natural transformation $\bar{\alpha} : \bar{\mathsf{H}} \to \bar{\mathsf{K}}$ is a *lifting* of $\alpha : \mathsf{H} \to \mathsf{K}$ if

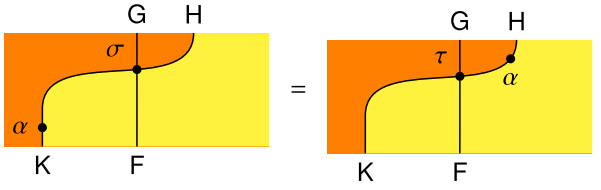
$$\alpha \circ \mathsf{U}_{\mathsf{F}} = \mathsf{U}_{\mathsf{G}} \circ \bar{\alpha} \tag{25}$$

Since the forgetful functor is faithful, a lifting is uniquely determined: $\bar{\alpha}\,(A, a) = \alpha\, A$, quite unlike the situation for functors. A straightforward calculation furthermore shows that $\bar{\alpha}$ thus defined indeed satisfies the naturality property (14). The only remaining proof obligation is to ensure that its components are G-homomorphisms. Let's see what's involved.

To this end we consider lifted functors induced by vanilla laws: assume $\bar{\mathsf{H}} = \mathsf{H}^{\sigma}$ and $\bar{\mathsf{K}} = \mathsf{K}^{\tau}$, where $\sigma : \mathsf{G} \circ \mathsf{H} \dot{\to} \mathsf{H} \circ \mathsf{F}$ and $\tau : \mathsf{G} \circ \mathsf{K} \dot{\to} \mathsf{K} \circ \mathsf{F}$. Under these assumptions the homomorphism condition "reads":



Observe that the algebra a is in a sense passive. We only need to be able to slide α across σ to obtain τ. It seems worthwhile to isolate this property by "erasing" the bottom of the previous diagrams:



(26)

If α satisfies this condition, it is called a *transformation* of vanilla laws, written $\alpha : \sigma \to \tau$.

Table 1
Liftings induced by distributive laws.

Lifting	Category of algebras	Eilenberg–Moore category
plain functor $\mathsf{H} : \mathscr{C} \to \mathscr{D}$	vanilla law $\lambda : \mathsf{G} \circ \mathsf{H} \dot{\to} \mathsf{H} \circ \mathsf{F}$ $\mathsf{H}^{\lambda} : \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C}) \to \mathsf{G}\text{-}\mathbf{Alg}(\mathscr{D})$	EM-law $\lambda : \mathsf{T} \circ \mathsf{H} \dot{\to} \mathsf{H} \circ \mathsf{S}$ $\mathsf{H}^{\lambda} : \mathscr{C}^{\mathsf{S}} \to \mathscr{D}^{\mathsf{T}}$
monad $\mathsf{M} : \mathscr{C} \to \mathscr{C}$	Kl-law $\lambda : \mathsf{F} \circ \mathsf{M} \dot{\to} \mathsf{M} \circ \mathsf{F}$ $\mathsf{M}^{\lambda} : \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C}) \to \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C})$	distributive law $\lambda : \mathsf{T} \circ \mathsf{M} \dot{\to} \mathsf{M} \circ \mathsf{T}$ $\mathsf{M}^{\lambda} : \mathscr{C}^{\mathsf{T}} \to \mathscr{C}^{\mathsf{T}}$

4.4. Lifting monads

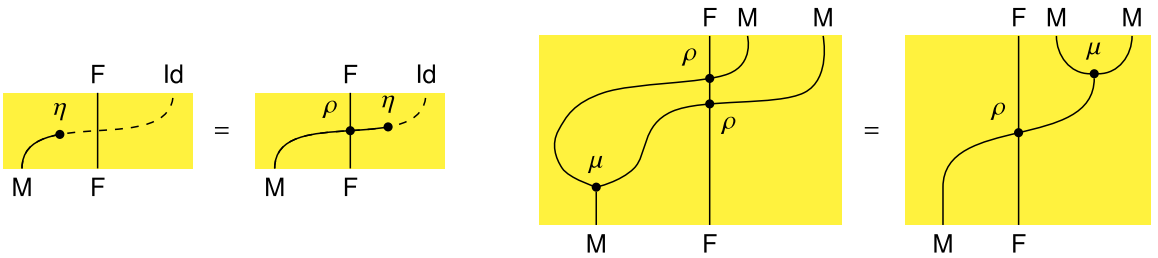
As our final example of liftings, we consider lifting monads to categories of algebras and homomorphisms. Consider the monad $\mathsf{M} : \mathscr{C} \to \mathscr{C}$ and the endofunctor $\mathsf{F} : \mathscr{C} \to \mathscr{C}$. We already know that given a natural transformation $\rho : \mathsf{F} \circ \mathsf{M} \dot{\to} \mathsf{M} \circ \mathsf{F}$, we can lift the endofunctor M to the category $\mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C})$. We shall see that in order to also be able to lift the unit and multiplication we additionally need to assume that ρ is a Kl-law.

It seems natural to stipulate that unit and multiplication of M^{ρ} are liftings of M's unit and multiplication:

$$\eta \circ \mathsf{U}_{\mathsf{F}} = \mathsf{U}_{\mathsf{F}} \circ \bar{\eta} \quad \text{and} \quad \mu \circ \mathsf{U}_{\mathsf{F}} = \mathsf{U}_{\mathsf{F}} \circ \bar{\mu}$$

We have seen in Section 4.3 that this uniquely determines $\bar{\eta}$ and $\bar{\mu}$, and that both are indeed natural transformations. By the same token—the functor U_{F} is faithful—the liftings satisfy the monad laws. It remains to show that the components of $\bar{\eta} : \mathsf{Id} \dot{\to} \mathsf{M}^{\rho}$ and $\bar{\mu} : \mathsf{M}^{\rho} \circ \mathsf{M}^{\rho} \dot{\to} \mathsf{M}^{\rho}$ are F-homomorphisms.

We have discussed in Section 4.3 that this is the case if η and μ are transformations of laws (26), in our case $\eta : id \to \rho$ and $\mu : \rho \mid \rho \to \rho$ where "$\mid$" denotes the vertical composition of two Kl-laws (10). It turns out that the two conditions are precisely the axioms of Kl-laws (11) and (12), repeated below, but drawn slightly skewed and with explicit identity functors for emphasis:



Quite interestingly, (11) and (12) have two readings: the equations can be seen as the defining properties of a Kl-law or as the defining properties of two transformations of laws, namely unit and multiplication. (For the latter reading to make sense, the two monads S and T have to be identified: $\mathsf{S} = \mathsf{T} = \mathsf{M}$.)

Now, suppose we wish to lift the monad (M, η, μ) to the Eilenberg–Moore category $\mathscr{C}^{\mathsf{T}}$. Our previous findings show that $\rho : \mathsf{T} \circ \mathsf{M} \dot{\to} \mathsf{M} \circ \mathsf{T}$ has to be both an EM-law and a Kl-law. In other words, ρ has to be a distributive law of the monad M over the monad T.

Table 1 overviews the different types of liftings induced by different types of distributive laws.

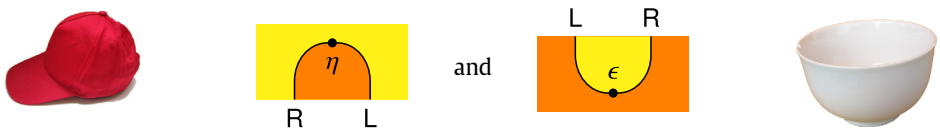
5. Adjunctions and adjoint squares

5.1. Adjunctions

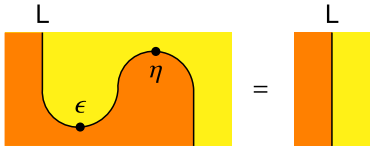
Adjunctions [6] are one of the most significant concepts in category theory, and they have a pleasing representation in string diagrams. An adjunction $\mathsf{L} \dashv \mathsf{R}$ is typically described as a pair of functors $\mathsf{L} : \mathscr{C} \to \mathscr{D}$ and $\mathsf{R} : \mathscr{D} \to \mathscr{C}$ such that there is a natural bijection between collections of arrows:

$$\frac{\mathsf{L}A \to B \text{ in } \mathscr{D}}{A \to \mathsf{R}B \text{ in } \mathscr{C}} \tag{27}$$

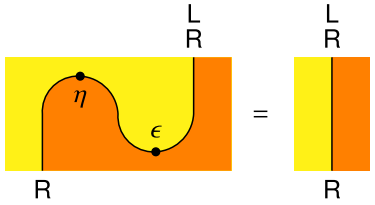
In this section, we consider an equivalent definition of adjunction that will be more convenient in our graphical calculations. Instead of a natural bijection between two sets of arrows, we require a pair of natural transformations, referred to as the *unit* and *counit* of the adjunction:



The unit resembles a cap, the counit a cup. They are required to satisfy the following *snake equations*, which intuitively allow us to straighten out bends in our string diagrams ("pull a string straight"):

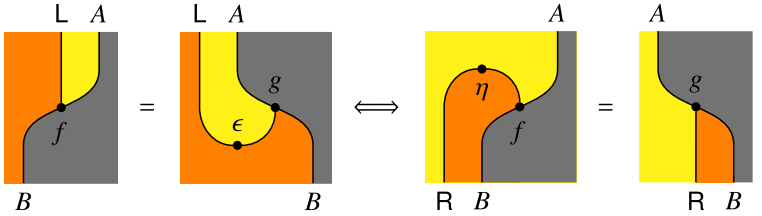


(28)



(29)

Using the units we can, in fact, express the natural bijection (27) as an equivalence:



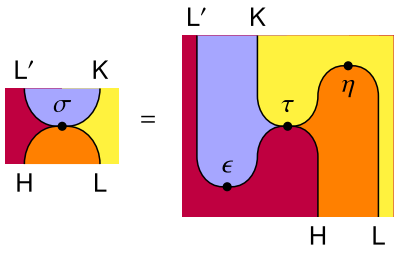
(30)

The equivalence is established by "bending wires", where a double bend is pulled straight using the snake equations (28) and (29). For example, for the left-to-right direction we place a cap on L bending the functor down to the left.

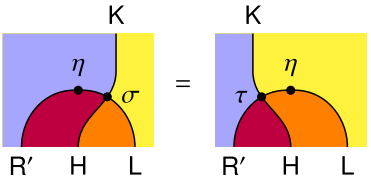
5.2. Adjoint squares

Monads and EM-laws form a category. In a similar way we can also form a category whose objects are adjunctions. The purpose of this section is to investigate this construction, in particular, the definition of arrows between adjunctions, emphasizing as before string-diagrammatic representations and proofs.

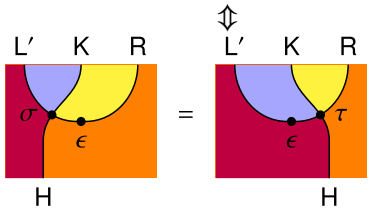
Let $\mathsf{L} \dashv \mathsf{R}$ and $\mathsf{L}' \dashv \mathsf{R}'$ be two adjunctions. Generalizing the one-to-one correspondence between arrows (27), we first show that natural transformations of type $\mathsf{L}' \circ \mathsf{K} \dot{\to} \mathsf{H} \circ \mathsf{L}$ are in one-to-one correspondence with natural transformations of type $\mathsf{K} \circ \mathsf{R} \dot{\to} \mathsf{R}' \circ \mathsf{H}$:



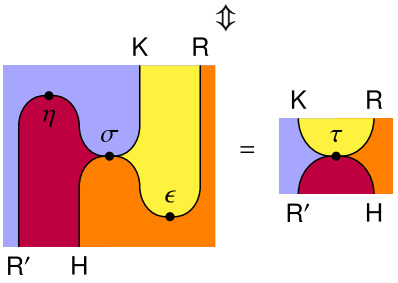
(31)



(32)



(33)



(34)

The equivalence of the four equations is established by bending wires and applying the snake equations (28) and (29). For example, the equivalence on top is shown by placing a cap on both "L′"s in (31) and pulling the string straight on the right-hand side using (29). Each of the diagrams in equations (32) and (33) loosely resembles a speedometer with the functors H and K acting as the needle. As the needle moves between the conjugates σ and τ it transmogrifies one unit into the other, and likewise for the counits.

The data $(\mathsf{H}, \sigma \dashv \tau, \mathsf{K}) : (\mathsf{L}, \mathsf{R}, \epsilon, \eta) \to (\mathsf{L}', \mathsf{R}', \epsilon, \eta)$ is called an *adjoint square*; σ and τ are sometimes called *conjugates* or *mates* as σ uniquely determines τ and vice versa, see (31) and (34). Fig. 2 summarizes our notation and terminology

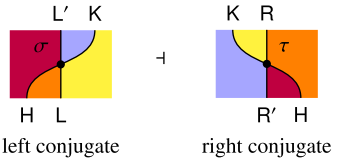


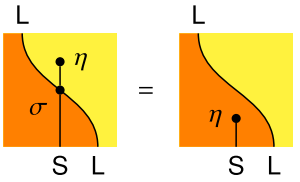
Fig. 2. Conjugates.

for future reference. Observe that the left conjugate is drawn with a "minor" diagonal, while the right conjugate features a "major" diagonal. Like EM-laws, we can compose conjugates vertically: we simply compose left and right conjugates separately. Unlike EM-laws, we can also compose conjugates horizontally, because adjunctions compose (unlike monads). The interested reader—we hope this is you, José—is invited to work out the details.

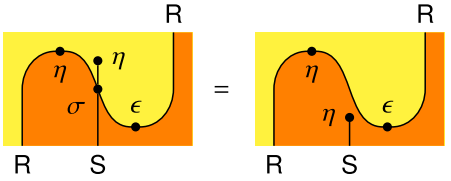
Let us conclude the section with an example property that illustrates the power of conjugates. Given a pair of monads, S and T, and an adjoint square $(\mathsf{S}, \sigma \dashv \tau, \mathsf{T})$ we aim to show:

$$\sigma \text{ is a Kl-law if and only if } \tau \text{ is an EM-law} \tag{35}$$

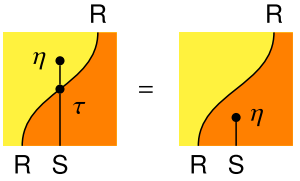
For the unit axioms, (8) and (11), we reason:



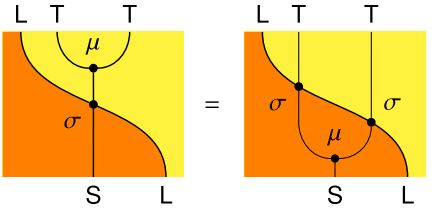
$\Longleftrightarrow$ { wire bending and snake equation (28) }



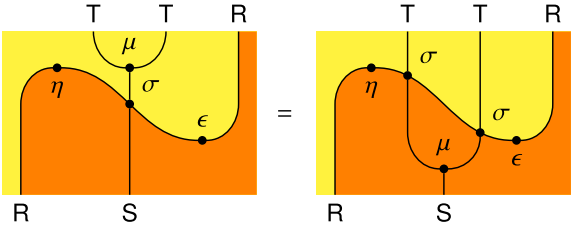
$\Longleftrightarrow$ { conjugates (34) and snake equation (29) }



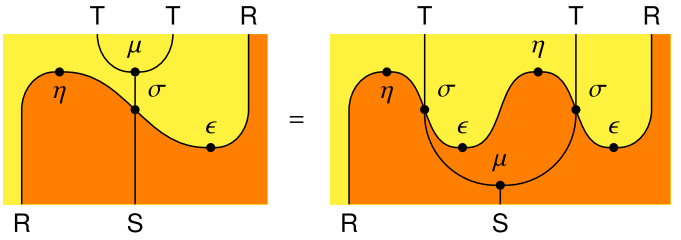
The proof rests on the fact that wire bending is a reversible operation, by virtue of the snake equations (28) and (29). For the multiplication axioms, (9) and (12), we proceed in a similar manner:



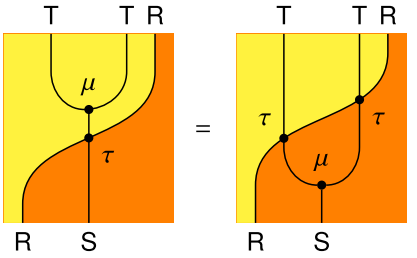
$\Longleftrightarrow$ { wire bending and snake equation (28) }



$\Longleftrightarrow$ { snake equation (28) }

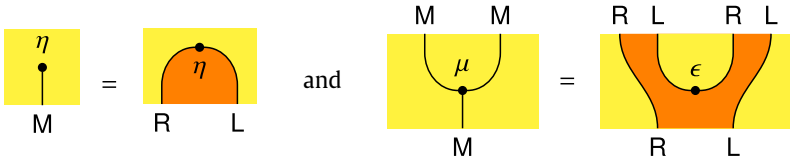


$\Longleftrightarrow$ { conjugates (34) }



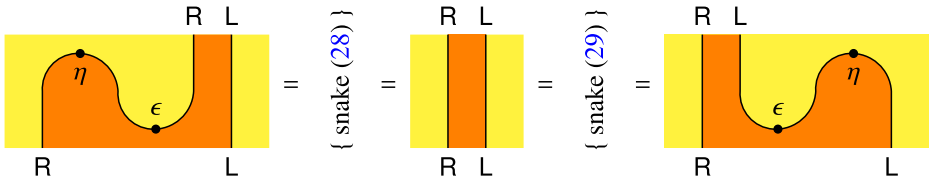
5.3. Huber's construction

Monads and adjunctions are intimately related [5]; in fact every adjunction $\mathsf{L} \dashv \mathsf{R}$ induces a monad $\mathsf{M} = \mathsf{R} \circ \mathsf{L}$, with the unit given by the unit of the adjunction, and the multiplication given by the composite on the right:

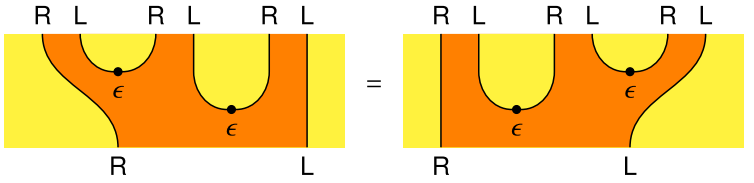


The diagrams on the right-hand sides feature an internal region which is contracted on the left-hand sides as $\mathsf{R} \circ \mathsf{L}$ becomes M.

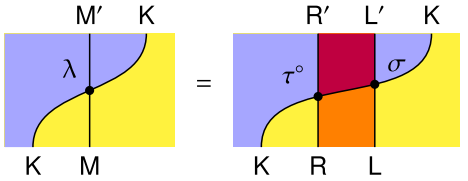
We check the monad unit axioms (4) as follows:



The associativity law (5) requires no proof as the diagrams for the left- and the right-hand side are identified:

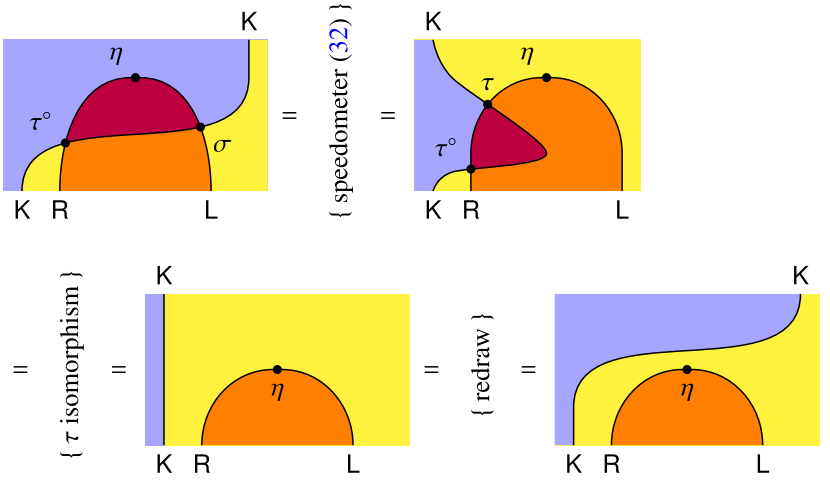


Alas, given an adjoint square we cannot horizontally compose the two conjugates to form an EM-law, simply because they do not match up: the right conjugate has the shape of a Kl-law. Fortunately, an additional assumption saves the day. If τ is an isomorphism, we can construct an EM-law as follows:

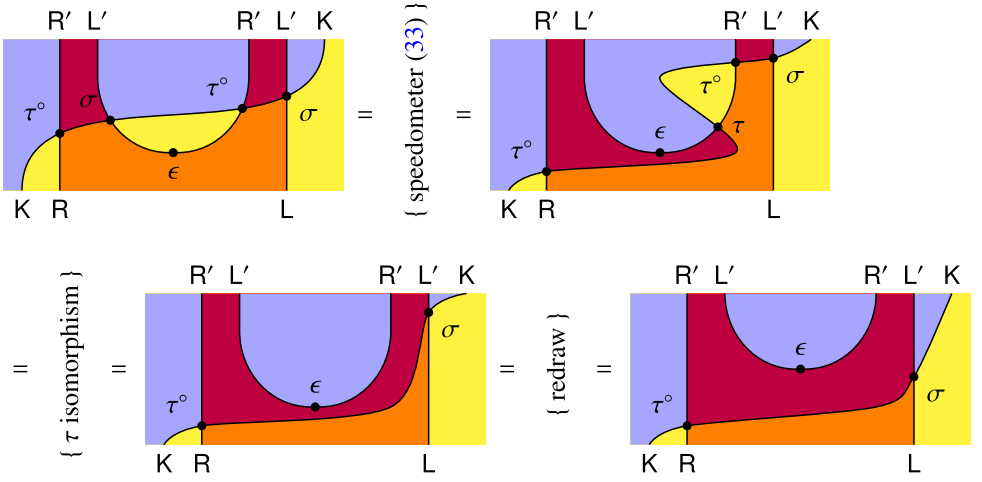


The two laws are composed horizontally. Observe that the diagram on the right-hand side features four different categories, two of which are contracted to a wire in the diagram on the left-hand side.

We have to show that λ thus defined satisfies the coherence conditions. For the unit axiom (8) we reason:



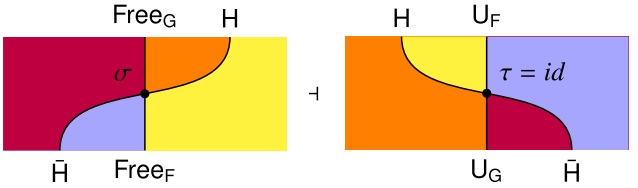
For the multiplication axiom (9) we have the following chain of equalities:



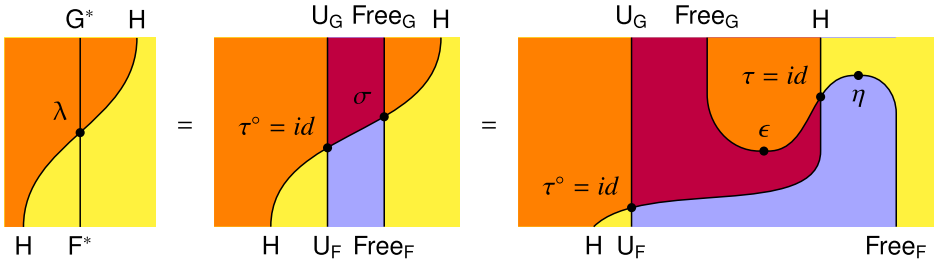
5.4. Example: the free monad of a functor

Let $\mathsf{F} : \mathscr{C} \to \mathscr{C}$ be an endofunctor. The category F-**Alg**$(\mathscr{C})$ of F-algebras and F-homomorphisms makes for an instructive example of the previous material. For many interesting cases, the forgetful functor $\mathsf{U_F} : \mathsf{F}$-**Alg**$(\mathscr{C}) \to \mathscr{C}$ has a left adjoint, denoted $\mathsf{Free_F} : \mathscr{C} \to \mathsf{F}$-**Alg**$(\mathscr{C})$. Their composite $\mathsf{F}^* = \mathsf{Free_F} \circ \mathsf{U_F}$ is then a monad, known as the *free monad of the functor* F.

Recall that the functor $\bar{\mathsf{H}} : \mathsf{F}$-**Alg**$(\mathscr{C}) \to \mathsf{G}$-**Alg**$(\mathscr{D})$ is a *lifting* of $\mathsf{H} : \mathscr{C} \to \mathscr{D}$ if $\mathsf{H} \circ \mathsf{U_F} = \mathsf{U_G} \circ \bar{\mathsf{H}}$. Now, when both forgetful functors are part of an adjoint situation, a lifting is a special case of adjoint square, the right conjugate of which is manifestly the identity:



Since the right conjugate has an inverse, the adjoint square induces an EM-law $\lambda : \mathsf{G}^* \circ \mathsf{H} \dot{\to} \mathsf{H} \circ \mathsf{F}^*$:



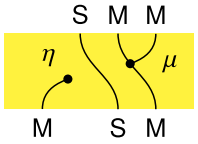
This law will play a prominent role in the next section.

6. Example: the resumption monad

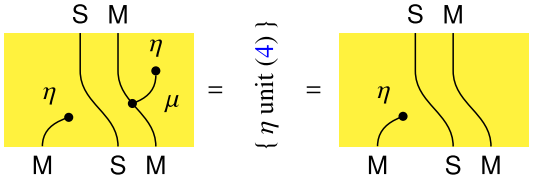
We now consider our final example, involving much of what has gone before, including monads and adjunctions.

Let $\mathsf{M} : \mathscr{C} \to \mathscr{C}$ be a monad and $\mathsf{S} : \mathscr{C} \to \mathscr{C}$ be an endofunctor; we aim to show that $\mathsf{M} \circ (\mathsf{S} \circ \mathsf{M})^*$ is a monad. This construction is known as the *resumption monad* [3].

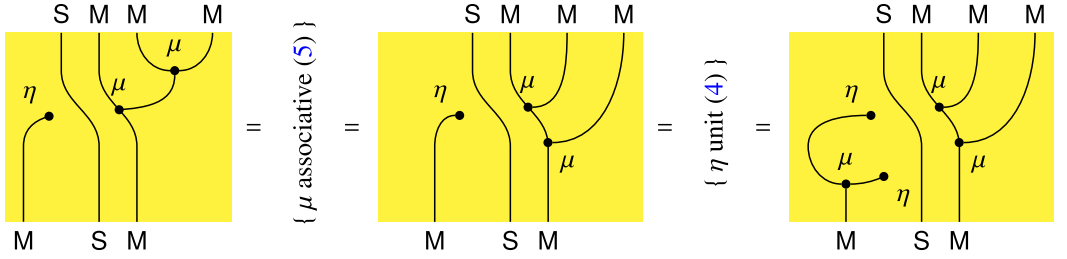
We generalize slightly: given a Kl-law $\rho : \mathsf{F} \circ \mathsf{M} \dot{\to} \mathsf{M} \circ \mathsf{F}$ we show that $\mathsf{M} \circ \mathsf{F}^*$ is a monad. To see that the resumption monad is an instance of the more general set-up, set $\mathsf{F} = \mathsf{S} \circ \mathsf{M}$ and $\rho = \eta \circ \mathsf{S} \circ \mu : (\mathsf{S} \circ \mathsf{M}) \circ \mathsf{M} \dot{\to} \mathsf{M} \circ (\mathsf{S} \circ \mathsf{M})$, graphically:



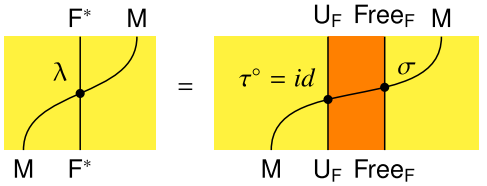
We must check this composite is a Kl-law. First, we check the unit axiom (11):



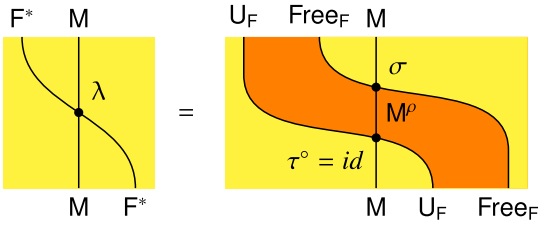
Secondly, the multiplication axiom (12):



Returning to the generalized problem, using the Kl-law $\rho : \mathsf{F} \circ \mathsf{M} \dot{\to} \mathsf{M} \circ \mathsf{F}$ we can define the lifting $\mathsf{M}^\rho : \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C}) \to \mathsf{F}\text{-}\mathbf{Alg}(\mathscr{C})$, which in turn induces an EM-law λ, see Section 5.4:

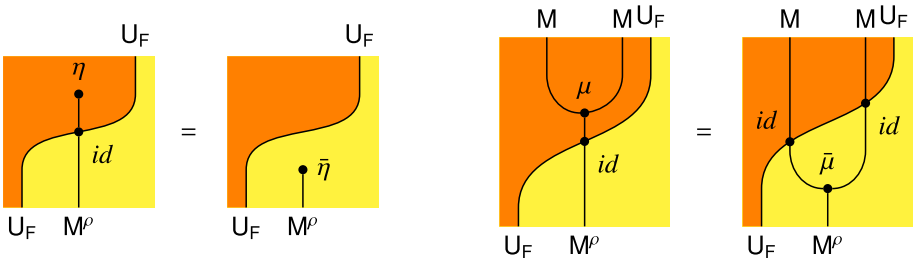


To be able to apply the construction of Section 3.3 we need to show that λ is simultaneously a Kl-law and therefore a fully-fledged distributive law.



Of course, this statement only makes sense if M^ρ is actually a monad, which is why we require ρ to be a Kl-law, see Table 1—note that we lift a monad to a category of algebras.

We first establish that $\tau = id : \mathsf{M} \circ \mathsf{U_F} \dot{\to} \mathsf{U_F} \circ \mathsf{M}^\rho$ is an EM-law. The coherence conditions (8) and (9):



simplify to $\eta \circ \mathsf{U_F} = \mathsf{U_F} \circ \bar{\eta}$ and $\mu \circ \mathsf{U_F} = \mathsf{U_F} \circ \bar{\mu}$. In other words, $\bar{\eta}$ and $\bar{\mu}$ have to be liftings, which they are. Since $\tau = id$ is an EM-law, we know that both τ's inverse τ° (13) and τ's conjugate σ (35) are Kl-laws. Since furthermore Kl-laws compose, it follows that $\lambda : \mathsf{F}^* \circ \mathsf{M} \dot{\to} \mathsf{M} \circ \mathsf{F}^*$ is also a Kl-law, which completes the proof.

7. Conclusion

String diagrams provide category theory with a new and very distinctive visual flavor. We hope to have amply demonstrated that string diagrams are an effective tool for equational reasoning about elementary category theory, as they silently deal with distracting bookkeeping steps, such as naturality and functoriality properties. It also has become clear that drawing string diagrams is a bit of an art: good diagrammatic choices can make all the difference. Many if not most diagrams that appear in the paper went through several iterations. First experiments with the two-dimensional notation invariably lead to very symmetrical drawings; early attempts of the authors were no exception. However, the full potential of string diagrams is only unlocked if the topological freedom inherent in the notation is exploited to aid the use of geometric intuition in the development of proofs.

As a final remark, we hope that you, José, have enjoyed reading the paper as much as we have enjoyed writing and drawing it.

Acknowledgements

This work has been partially funded by EPSRC grant numbers EP/J010995/1, EP/I03808X/1 and AFSOR grant number FA9550-14-1-0079 “Algorithmic and Logical Aspects when Composing Meanings”. The authors would like to thank Jeremy Gibbons for carefully proof-reading a draft of this article and for suggesting numerous improvements. We would also like to thank the anonymous reviewers for pointing out a couple of errors and infelicities. Thanks are furthermore due to Till Tantau for creating TikZ, without which we wouldn’t have been able to convey the beauty of string diagrams.

References

[1] R. Backhouse, Making formality work for us, EATCS Bull. 38 (June 1989) 219–249.
[2] J. Beck, Distributive laws, in: H. Appelgate, M. Barr, J. Beck, F.W. Lawvere, F. Linton, E. Manes, M. Tierney, F. Ulmer (Eds.), Seminar on Triples and Categorical Homotopy Theory, in: Lecture Notes in Mathematics, vol. 80, Springer, 1969, pp. 119–140.
[3] P. Cenciarelli, E. Moggi, A syntactic approach to modularity in denotational semantics, in: Category Theory and Computer Science, The Netherlands, Amsterdam, 1993.
[4] M.M. Fokkinga, L. Meertens, Adjunctions, Tech. Rep. Memoranda Inf 94-31, University of Twente, Enschede, Netherlands, June 1994.
[5] P.J. Huber, Homotopy theory in general categories, Math. Ann. 144 (1961) 361–385, http://dx.doi.org/10.1007/BF01396534.
[6] D.M. Kan, Adjoint functors, Trans. Am. Math. Soc. 87 (2) (1958) 294–329.
[7] S. Mac Lane, Categories for the Working Mathematician, 2nd edition, Graduate Texts in Mathematics, Springer-Verlag, Berlin, 1998.
[8] R. Penrose, Applications of negative dimensional tensors, in: D. Welsh (Ed.), Proceedings of a Conference Held at the Mathematical Institute, 7–10 July 1969, in: Combinatorial Mathematics and Its Applications, Academic Press, Oxford, 1971, pp. 221–244.