String Diagrams for Elementary Category Theory
3: Adjunctions

Dan Marsden
Based on joint work with Ralf Hinze

April 4, 2023

Adjunctions

Given a functor $F : \mathcal{C} \to \mathcal{D}$, is there a way of traveling back in the other direction?

1. If we want to get back to exactly where we started, We could just ask for an inverse functor

$$F \circ F^{\circ} = \mathrm{Id} \qquad \text{and} \qquad \mathrm{Id} = F^{\circ} \circ F.$$

Adjunctions

Given a functor $F : \mathcal{C} \to \mathcal{D}$, is there a way of traveling back in the other direction?

1. If we want to get back to exactly where we started, We could just ask for an inverse functor

$$F \circ F^{\circ} = \mathrm{Id} \qquad \text{and} \qquad \mathrm{Id} = F^{\circ} \circ F.$$

2. If just want to to get back to somewhere isomorphic to our starting point, We could require a functor $G : \mathcal{D} \to \mathcal{C}$ such there are natural isomorphisms

$$F \circ G \stackrel{\cdot}{\cong} \mathrm{Id} \qquad \text{and} \qquad \mathrm{Id} \stackrel{\cdot}{\cong} G \circ F.$$

Adjunctions

Given a functor $F : \mathcal{C} \to \mathcal{D}$, is there a way of traveling back in the other direction?

1. If we want to get back to exactly where we started, We could just ask for an inverse functor

$$F \circ F^{\circ} = \mathrm{Id} \qquad \text{and} \qquad \mathrm{Id} = F^{\circ} \circ F.$$

2. If just want to to get back to somewhere isomorphic to our starting point, We could require a functor $G : \mathcal{D} \to \mathcal{C}$ such there are natural isomorphisms

$$F \circ G \mathrel{\dot{\cong}} \mathrm{Id} \qquad \text{and} \qquad \mathrm{Id} \mathrel{\dot{\cong}} G \circ F.$$

3. If we only require that there is a morphism relating the start and end points of a round trip, there are two sensible choices:

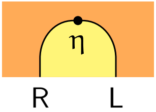
$$F \circ G \mathrel{\dot{\to}} \mathrm{Id} \qquad \text{and} \qquad \mathrm{Id} \mathrel{\dot{\to}} G \circ F$$
$$F \circ G \mathrel{\dot{\leftarrow}} \mathrm{Id} \qquad \text{and} \qquad \mathrm{Id} \mathrel{\dot{\leftarrow}} G \circ F$$

Adjunctions

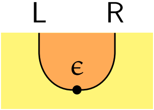
Convenient graphical definition

Definition

Functor $\mathsf{L} : \mathcal{D} \to \mathcal{C}$ is left adjoint to functor $\mathsf{R} : \mathcal{C} \to \mathcal{D}$, written $\mathsf{L} \dashv \mathsf{R} : \mathcal{C} \rightharpoonup \mathcal{D}$, if there exist unit (**cap**) and counit (**cup**) natural transformations:



and



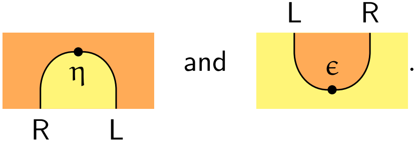
.

Adjunctions

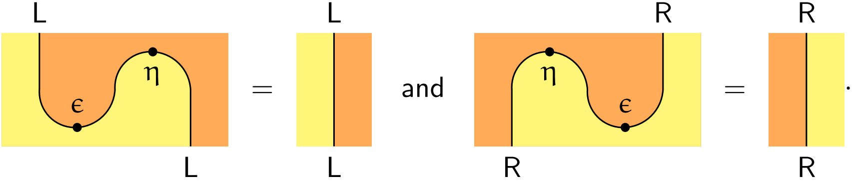
Convenient graphical definition

Definition

Functor $\mathsf{L} : \mathcal{D} \to \mathcal{C}$ is left adjoint to functor $\mathsf{R} : \mathcal{C} \to \mathcal{D}$, written $\mathsf{L} \dashv \mathsf{R} : \mathcal{C} \rightharpoonup \mathcal{D}$, if there exist unit (**cap**) and counit (**cup**) natural transformations:



such that the following **snake equations** hold:



Adjoints

Example

Let **Mon** be the category of monoids and monoid homomorphisms.

- There is a forgetful functor $\mathsf{U} : \mathbf{Mon} \to \mathbf{Set}$ such that $\mathsf{U}\,(X, 1, \times) \mapsto X$.

Adjoints

Example

Let **Mon** be the category of monoids and monoid homomorphisms.

- There is a forgetful functor $\mathsf{U} : \mathbf{Mon} \to \mathbf{Set}$ such that $\mathsf{U}\,(X, 1, \times) \mapsto X$.
- There is a functor $\mathsf{F} : \mathbf{Set} \to \mathbf{Mon}$ mapping set X to the monoid with underlying set $\mathsf{L}\,(X)$, unit the empty list, and multiplication list concatenation.

Adjoints

Example

Let **Mon** be the category of monoids and monoid homomorphisms.

- There is a forgetful functor $\mathsf{U} : \mathbf{Mon} \to \mathbf{Set}$ such that $\mathsf{U}\,(X, 1, \times) \mapsto X$.
- There is a functor $\mathsf{F} : \mathbf{Set} \to \mathbf{Mon}$ mapping set X to the monoid with underlying set $\mathsf{L}\,(X)$, unit the empty list, and multiplication list concatenation.
- There is a unit $\eta : \mathsf{Id} \stackrel{.}{\to} UF$ with component at set X

$$\eta\, X\, x = [x].$$

Adjoints

Example

Let **Mon** be the category of monoids and monoid homomorphisms.

- There is a forgetful functor $\mathsf{U} : \mathbf{Mon} \to \mathbf{Set}$ such that $\mathsf{U}\,(X, 1, \times) \mapsto X$.
- There is a functor $\mathsf{F} : \mathbf{Set} \to \mathbf{Mon}$ mapping set X to the monoid with underlying set $\mathsf{L}\,(X)$, unit the empty list, and multiplication list concatenation.
- There is a unit $\eta : \mathsf{Id} \dot{\to} UF$ with component at set X

$$\eta\, X\, x = [x].$$

- There is a counit $\epsilon : FU \dot{\to} \mathsf{Id}$, with component at monoid $(X, 1, \times)$

$$\epsilon\,(X, 1, \times)\,[x_1, ..., x_n] = 1 \times x_1 \times ... \times x_n.$$

Adjoints are canonical

Right adjoints are determined up to isomorphism

Let $\mathsf{L}_1 \dashv \mathsf{R}_1, \mathsf{L}_2 \dashv \mathsf{R}_2 : \mathcal{C} \rightharpoonup \mathcal{D}$ be two parallel adjunctions. Then

$$\mathsf{L}_1 \cong \mathsf{L}_2 \iff \mathsf{R}_1 \cong \mathsf{R}_2.$$

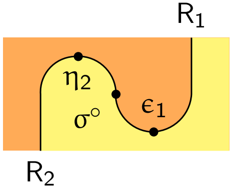
Adjoints are canonical

Right adjoints are determined up to isomorphism

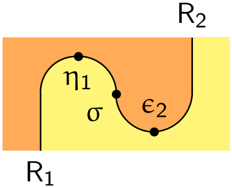
Let $L_1 \dashv R_1, L_2 \dashv R_2 : \mathcal{C} \rightharpoonup \mathcal{D}$ be two parallel adjunctions. Then

$$L_1 \cong L_2 \iff R_1 \cong R_2.$$

If σ witnesses the isomorphism of the left adjoints, then the natural isomorphism for the right adjoints is given by two "fake" snakes:



and



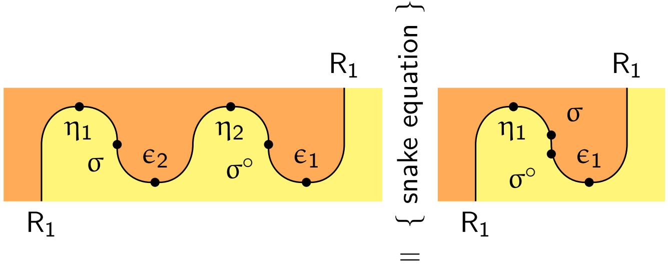
.

Adjoints are canonical

Right adjoints are determined up to isomorphism

Let $L_1 \dashv R_1, L_2 \dashv R_2 : \mathcal{C} \rightharpoonup \mathcal{D}$ be two parallel adjunctions. Then

$$L_1 \cong L_2 \iff R_1 \cong R_2.$$

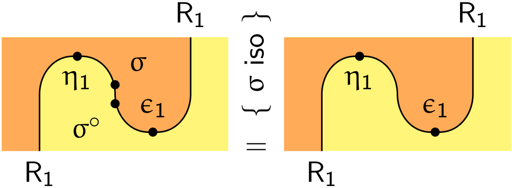


Adjoints are canonical

Right adjoints are determined up to isomorphism

Let $\mathsf{L}_1 \dashv \mathsf{R}_1, \mathsf{L}_2 \dashv \mathsf{R}_2 : \mathcal{C} \rightharpoonup \mathcal{D}$ be two parallel adjunctions. Then

$$\mathsf{L}_1 \cong \mathsf{L}_2 \iff \mathsf{R}_1 \cong \mathsf{R}_2.$$

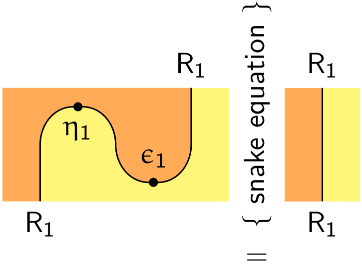


Adjoints are canonical

Right adjoints are determined up to isomorphism

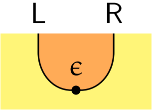
Let $\mathsf{L}_1 \dashv \mathsf{R}_1, \mathsf{L}_2 \dashv \mathsf{R}_2 : \mathcal{C} \rightharpoonup \mathcal{D}$ be two parallel adjunctions. Then

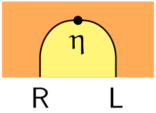
$$\mathsf{L}_1 \cong \mathsf{L}_2 \iff \mathsf{R}_1 \cong \mathsf{R}_2.$$



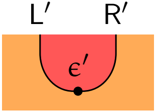
Adjoints compose

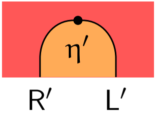
Given a pair of adjunctions with units and counits,





and



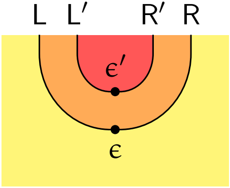


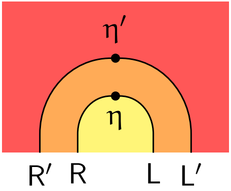
,

then $\mathsf{L} \circ \mathsf{L}' \dashv \mathsf{R}' \circ \mathsf{R}$.

Adjoints compose

We can build candidate cups and caps by nesting:

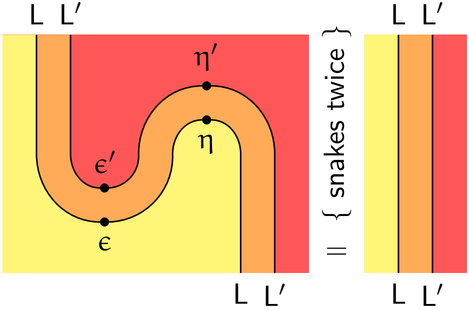




.

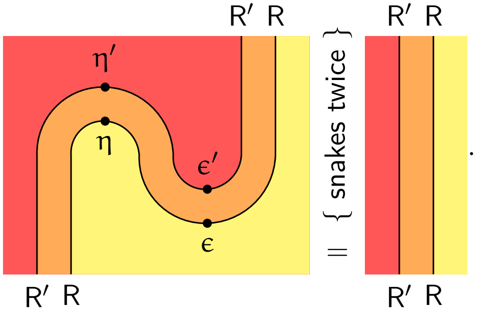
Adjoints compose

The first equation is shown as follows:



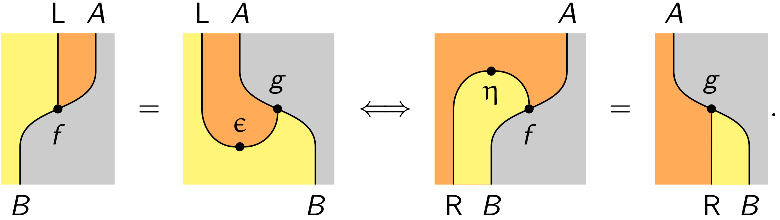
Adjoints compose

The second snake equation follows similarly:



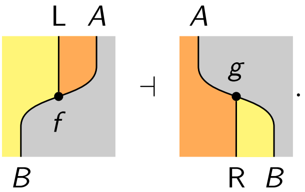
Adjunctions and bending wires

For an adjunction $\mathsf{L} \dashv \mathsf{R} : \mathcal{C} \rightharpoonup \mathcal{D}$, by "bending wires" using the cup and cap, we have the following relationship:



Adjunctions and bending wires

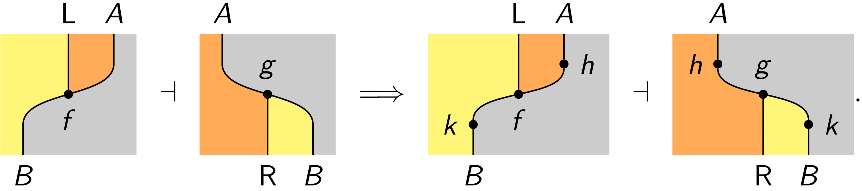
This “wire bending” establishes a bijection between morphisms of types $\mathsf{L}\,A \to B$ and $A \to \mathsf{R}\,B$. We shall write $f\,(\mathsf{L} \dashv \mathsf{R})\,g$ if f and g are related by this bijection, or $f \dashv g$ if the adjunction is clear from the context. Pictorially:



We say f is the **left transpose** of g, and g is the **right transpose** of f.

Adjunctions and bending wires

The “wire bending” relationship is natural in the sense that:



Adjunctions and bending wires

We have shown that taking transposes yields a *natural* bijection between collections of arrows:

$$\mathsf{L}\,A \to B : \mathcal{C} \quad \cong \quad A \to \mathsf{R}\,B : \mathcal{D}.$$

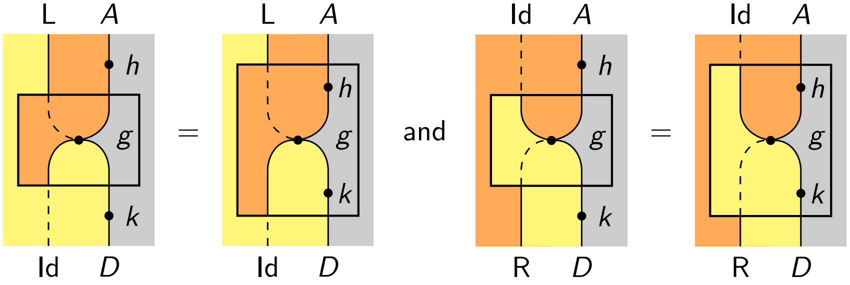
The maps witnessing the bijection, written $\lfloor - \rfloor$ and $\lceil - \rceil$, are called **adjoint transpositions**:

$$\frac{f : \mathsf{L}\,A \to B : \mathcal{C}}{\lfloor f \rfloor : A \to \mathsf{R}\,B : \mathcal{D}} \qquad\qquad \frac{\lceil g \rceil : \mathsf{L}\,A \to B : \mathcal{C}}{g : A \to \mathsf{R}\,B : \mathcal{D}}.$$

In fact, having such a bijection is equivalent to our previous definition of adjunction.

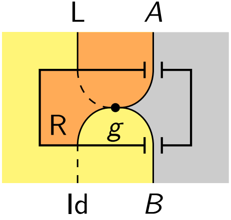
Transposes yield cups caps and snake equations

We can introduce a box notation to denote transposition maps in string diagrams:

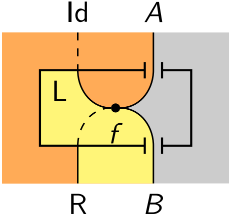


Transposes yield cups caps and snake equations

A more suggestive rendition highlights the naturality property:

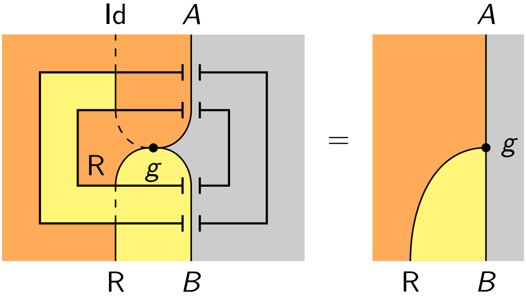


and



Transposes yield cups caps and snake equations

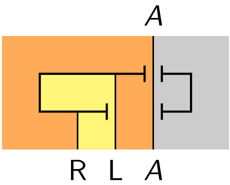
Pictorially, the fact that transpositions are mutually inverse appears as:



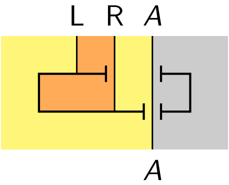
(and the similar equation when the transpositions are applied in the other order).

Transposes yield cups caps and snake equations

We can then define the components of putative units and counit *componentwise* as follows:

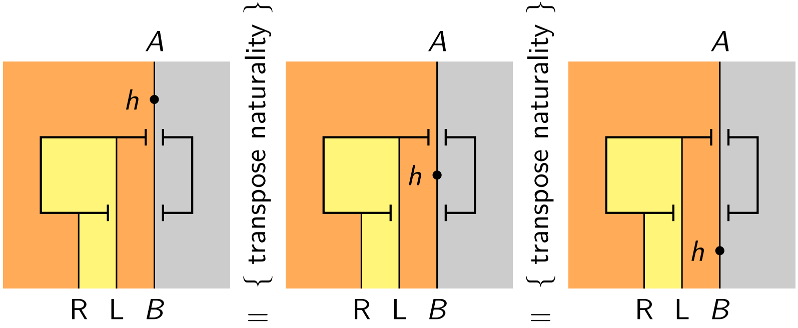


and



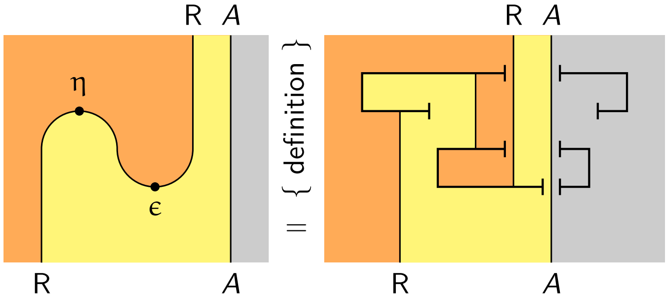
Transposes yield cups caps and snake equations

Naturality of the unit is easy to verify:

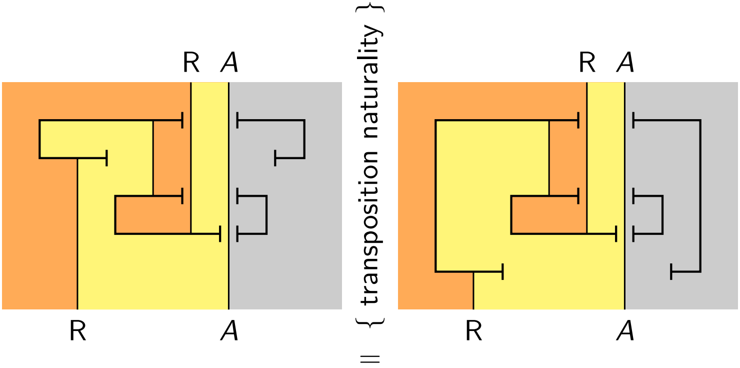


The naturality of the counit is verified similarly.

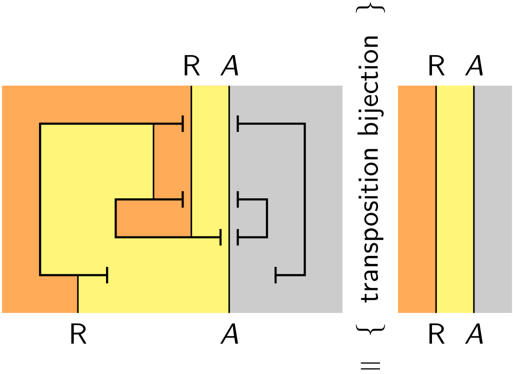
Transposes yield cups caps and snake equations



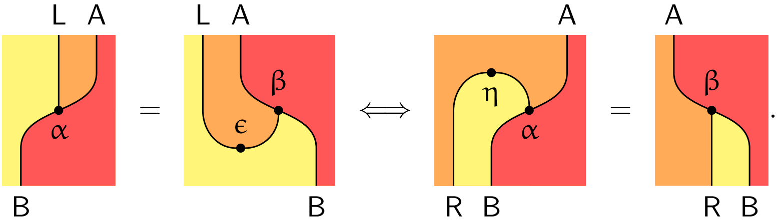
Transposes yield cups caps and snake equations



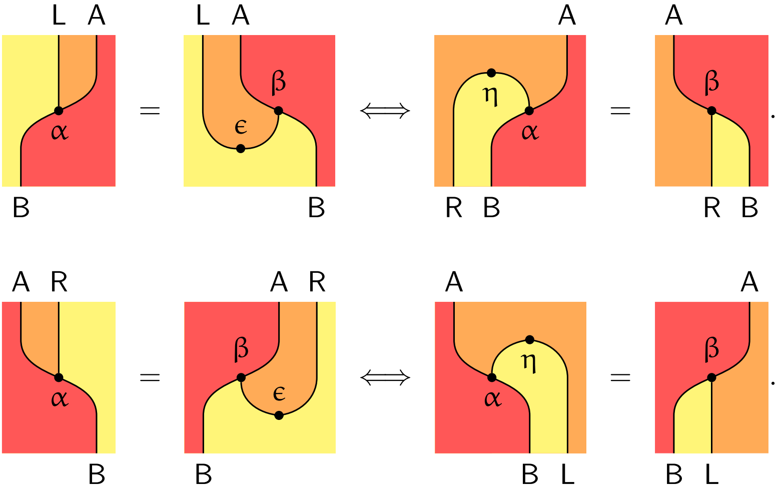
Transposes yield cups caps and snake equations



Lifting adjunctions

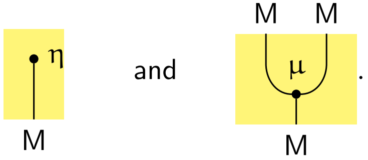


Lifting adjunctions

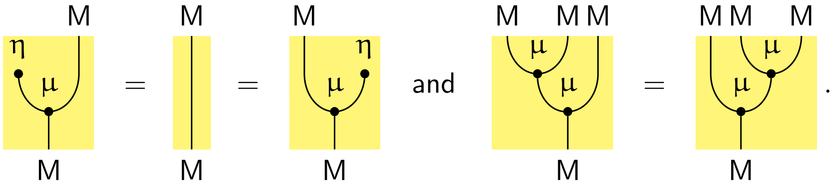


Huber's construction

Recall a monad on $\mathcal{C}$ consists of an endofunctor $M : \mathcal{C} \to \mathcal{C}$, and unit and multiplication natural transformations:

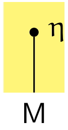


such that:

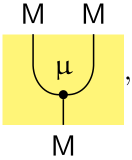


Huber's construction

Now assume we have a monad with unit and multiplication:

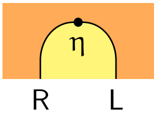


and

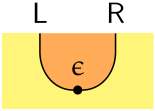


,

and an adjunction $L \dashv R : \mathcal{C} \rightharpoonup \mathcal{D}$ with unit and counit:



and

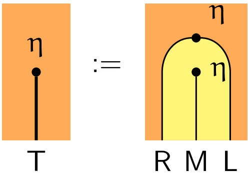


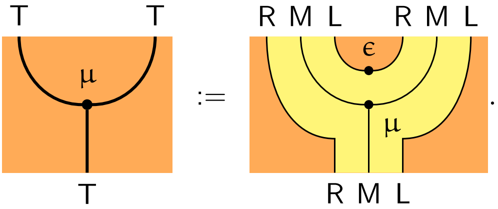
.

Is there a natural way to build a monad on $T := R \circ M \circ L$?

Huber's construction

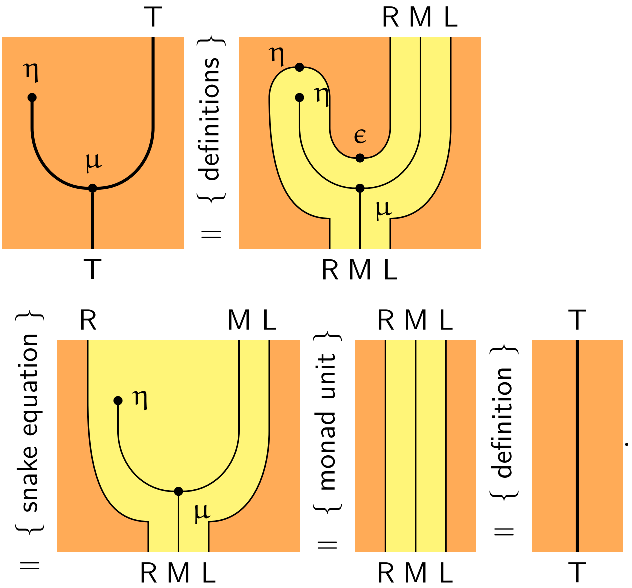
It seems natural to form the composites:



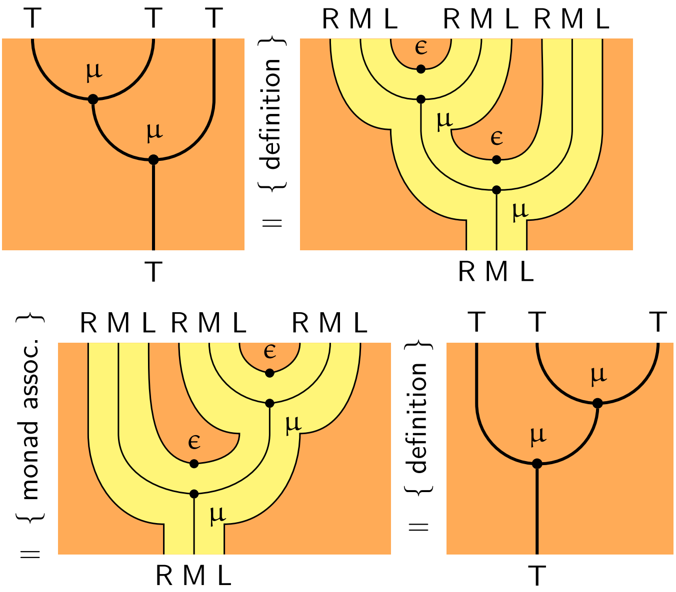


Huber's construction

To verify one of the monad unit axioms, we calculate:

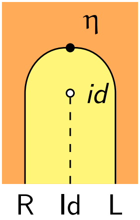


Huber's construction

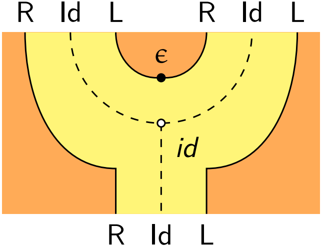


Huber's construction

Special case, every adjunction induces a monad with unit and multiplication:



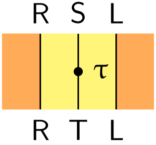
and



.

Huber on monad morphisms

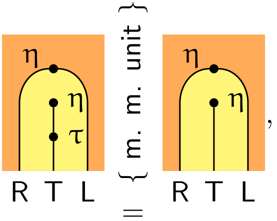
Given a monad morphism $\tau : S \to T$, we might expect the composite



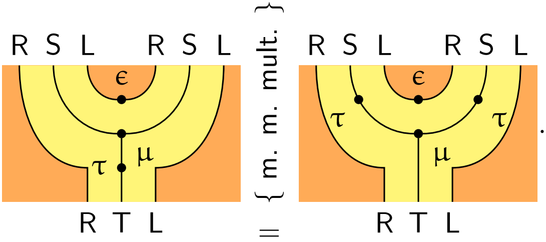
is also a monad morphism.

Huber on monad morphisms

Verifying the unit axiom:



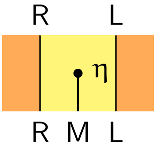
and for the multiplication axiom:



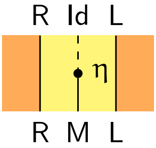
Huber on monad morphisms

Example

Units As a special case, we get a monad morphism
$R \circ L \to R \circ M \circ L$:



explicitly



Next time

Applications!