

String Diagrams for Elementary Category Theory

4: Putting it all together

Dan Marsden

Based on joint work with Ralf Hinze

April 5, 2023

Free monads

Free algebras

Definition (Free algebra functor)

In Lecture 2 we introduced the category of algebras for an endofunctor Σ , which equips a given category \mathcal{C} with additional structure. There is a forgetful functor, the **underlying functor**,

$$\begin{aligned} U^\Sigma : \Sigma\text{-Alg}(\mathcal{C}) &\rightarrow \mathcal{C} & U^\Sigma(A, a) &:= A, \\ & & U^\Sigma(h) &:= h, \end{aligned}$$

Free monads

Free algebras

Definition (Free algebra functor)

In Lecture 2 we introduced the category of algebras for an endofunctor Σ , which equips a given category \mathcal{C} with additional structure. There is a forgetful functor, the **underlying functor**,

$$\begin{aligned} U^\Sigma : \Sigma\text{-Alg}(\mathcal{C}) &\rightarrow \mathcal{C} & U^\Sigma(A, a) &:= A, \\ & & U^\Sigma(h) &:= h, \end{aligned}$$

If it exists, the left adjoint to the forgetful functor sends an object to the **free algebra**:

$$\begin{array}{ccc} \Sigma\text{-Alg}(\mathcal{C}) & \begin{array}{c} \xleftarrow{\text{Free}^\Sigma} \\ \perp \\ \xrightarrow{U^\Sigma} \end{array} & \mathcal{C} \end{array} \qquad \begin{array}{l} \text{Free}^\Sigma A =: (\Sigma^* A, \text{in } A) \\ \text{Free}^\Sigma f =: \Sigma^* f \end{array} .$$

Free monads

Free algebra intuitions and terminology

We can think of the endofunctor Σ a signature of operations. For example if $\Sigma(X) = X \times X + 1$, then a Σ -algebra on A is a function of the form:

$$A \times A + 1 \rightarrow A$$

Free monads

Free algebra intuitions and terminology

We can think of the endofunctor Σ a signature of operations. For example if $\Sigma(X) = X \times X + 1$, then a Σ -algebra on A is a function of the form:

$$A \times A + 1 \rightarrow A$$

Which is equivalent to two functions:

$$A \times A \rightarrow A \quad \text{and} \quad 1 \rightarrow A$$

Free monads

Free algebra intuitions and terminology

We can think of the endofunctor Σ a signature of operations. For example if $\Sigma(X) = X \times X + 1$, then a Σ -algebra on A is a function of the form:

$$A \times A + 1 \rightarrow A$$

Which is equivalent to two functions:

$$A \times A \rightarrow A \quad \text{and} \quad 1 \rightarrow A$$

As a function of type $1 \rightarrow A$ encodes an element of A , this is equivalent to choosing a binary function $A \times A \rightarrow A$, which we shall denote $+$ and an element of A , which we shall denote 0 .

Free monads

Free algebra intuitions and terminology

Example

For our example Σ , the following are algebras:

- ▶ The natural numbers, with $+$ interpreted by addition, and 0 by zero.
- ▶ The natural numbers, with $+$ interpreted by *multiplication*, and 0 by 33 .
- ▶ The set of all strings of natural numbers, with $+$ concatenation, and 0 the empty string.
- ▶ The set of all strings of natural numbers, with $+$ concatenation, and 0 the string $[33]$.

Free monads

Free algebra intuitions and terminology

For our example Σ , the forgetful functor does have a left adjoint. $\Sigma^* A$ consists of all terms of the form:

$$0, a, (0 + a) + 0, a + a, \dots$$

The free algebra structure map

$$\text{in } A : \Sigma \Sigma^* A \rightarrow \Sigma^* A$$

picks out as constant element the term 0, and the binary operation is formal addition:

$$(s, t) \mapsto s + t$$

Free monads

Free algebra intuitions and terminology

Further structure:

- ▶ The unit of the adjunction, $\text{var } A : A \rightarrow \Sigma^* A$, turns an element of A into the term for the corresponding variable.
- ▶ The counit $\epsilon(A, a) : \text{Free}^\Sigma A \rightarrow (A, a)$ evaluates a term using the operations of a given algebra.

This intuitive pattern repeats for different choices of Σ .

Free monads

Folds

We introduce shorthand for the evaluation map:

$$U^\Sigma (\epsilon (A, a)).$$

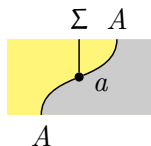
This is an arrow in the underlying category, which we shall denote $\langle\langle a \rangle\rangle$, and pronounce “**fold** a ”:

$$\frac{a : \Sigma A \rightarrow A}{\langle\langle a \rangle\rangle : \Sigma^* A \rightarrow A}.$$

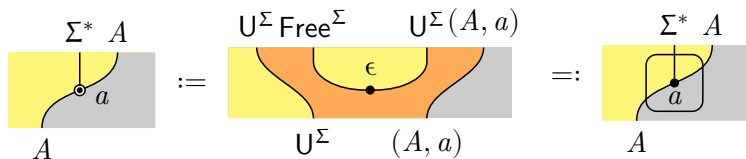
Free monads

Folds

For an algebra:



we can depict $\llbracket a \rrbracket$ as follows:



Free monads

The free / forgetful adjunction

The universal property of the free / forgetful adjunction can be unpacked in terms of the base category as

The diagram shows four string diagrams representing the universal property of the free/forgetful adjunction. Each diagram consists of a yellow region on top and a gray region on the bottom, separated by a curved boundary. The top region is labeled $\Sigma^* A$ and the bottom region is labeled B .

- Diagram 1: A vertical line with a dot at the bottom (in the gray region) extends upwards into the yellow region. The label h is placed near the dot.
- Diagram 2: A vertical line with a dot at the bottom (in the gray region) extends upwards into the yellow region. The label b is placed near the dot. A small circle is drawn around the dot.
- Diagram 3: A vertical line with a dot at the bottom (in the gray region) extends upwards into the yellow region. The label var is placed near the dot.
- Diagram 4: A vertical line with a dot at the bottom (in the gray region) extends upwards into the yellow region. The label g is placed near the dot.

The diagrams are connected by equals signs ($=$) and a double-headed arrow (\Leftrightarrow), indicating the following relationships:

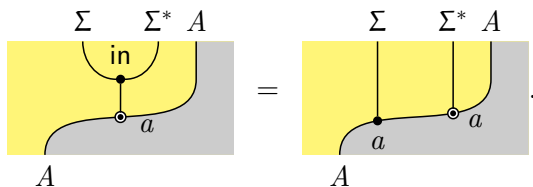
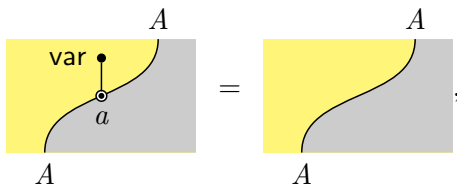
$$\text{Diagram 1} = \text{Diagram 2} \Leftrightarrow \text{Diagram 3} = \text{Diagram 4}$$

for all Σ -homomorphisms $h : \text{Free}^{\Sigma} A \rightarrow (B, b)$ and all arrows $g : A \rightarrow B$.

Free monads

The free / forgetful adjunction

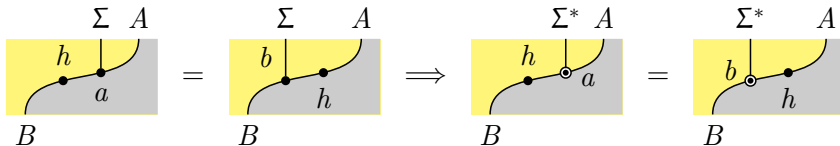
We have the following two **computation rules**:



Free monads

The free / forgetful adjunction

Naturality of the counit gives rise to the *elevation rule*:



Free monads

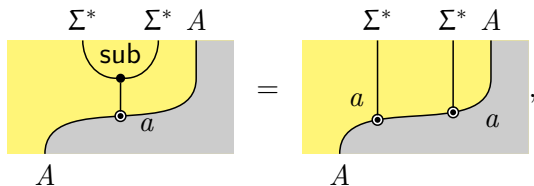
Algebras of endofunctors and monads

If we apply Huber's construction to the adjunction $\text{Free}^\Sigma \dashv \text{U}^\Sigma$, we obtain the so-called **free monad of a functor**: $(\Sigma^*, \text{var}, \text{sub})$ where $\text{sub} = \langle\langle \text{in} \rangle\rangle$.

Free monads

Algebras of endofunctors and monads

Substitution plays nicely with evaluation. Combining the second computational rule with elevation gives:



Combining this observation with the first computation rule, it tells us that for every Σ -algebra (A, a) , $(\Sigma^* A, \langle a \rangle)$ is an Eilenberg–Moore algebra. That this mapping preserves homomorphisms follows from the elevation rule.

Free monads

Algebras of endofunctors and monads

► Fold yields a functor $\text{Up} : \Sigma\text{-Alg}(\mathcal{C}) \rightarrow \mathcal{C}^{\Sigma^*}$.

Free monads

Algebras of endofunctors and monads

- ▶ Fold yields a functor $\text{Up} : \Sigma\text{-Alg}(\mathcal{C}) \rightarrow \mathcal{C}^{\Sigma^*}$.
- ▶ It turns out that every Σ^* Eilenberg–Moore algebra arises this way.

Free monads

Algebras of endofunctors and monads

- ▶ Fold yields a functor $\text{Up} : \Sigma\text{-Alg}(\mathcal{C}) \rightarrow \mathcal{C}^{\Sigma^*}$.
- ▶ It turns out that every Σ^* Eilenberg–Moore algebra arises this way.
- ▶ Categorically, the category of Σ -algebras is isomorphic to the Eilenberg–Moore category of the free monad Σ^* :

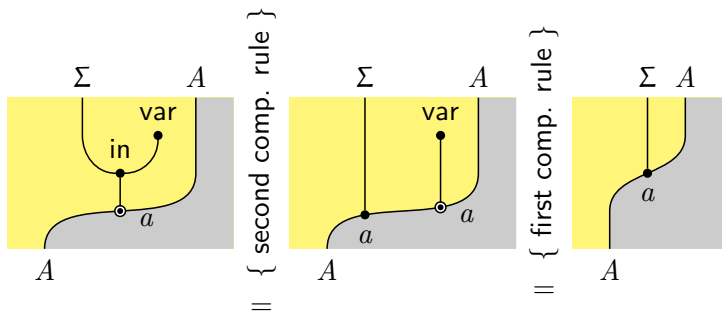
$$\text{Up} : \Sigma\text{-Alg}(\mathcal{C}) \cong \mathcal{C}^{\Sigma^*} : \text{Dn}$$

To see this, we first define $\text{Dn} : \mathcal{C}^{\Sigma^*} \rightarrow \Sigma\text{-Alg}(\mathcal{C})$.

Free monads

Algebras of endofunctors and monads

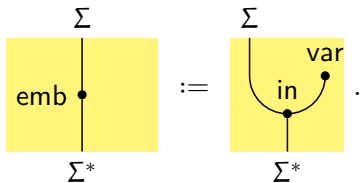
We can map Eilenberg–Moore to Σ -algebras exploiting the computation rules, reversing Up in the process:



Free monads

Algebras of endofunctors and monads

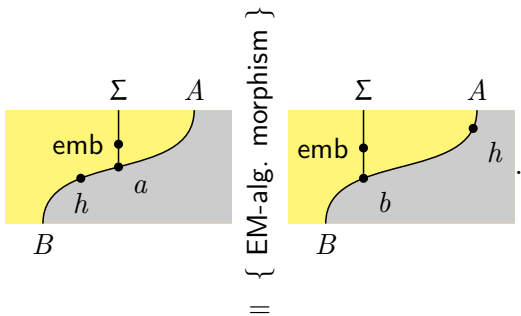
So precomposing with the following map takes Eilenberg–Moore to Σ -algebras:



Free monads

Algebras of endofunctors and monads

That precomposing with emb preserves homomorphisms is straightforward:



Free monads

Algebras of endofunctors and monads

We have two identity on morphisms functors:

$$\text{Up}(A, a : \Sigma A \rightarrow A) = (A, \langle\langle a \rangle\rangle)$$

$$\text{Up } h = h,$$

$$\text{Dn}(B, b : \Sigma^* B \rightarrow B) = (B, b \cdot \text{emb } B)$$

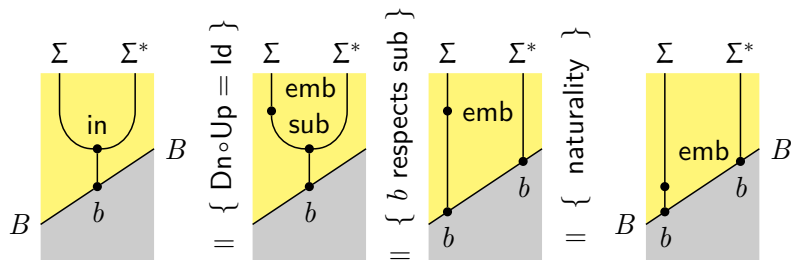
$$\text{Dn } h = h.$$

$\text{Dn} \circ \text{Up} = \text{Id}$ by design. It remains to show $\text{Up} \circ \text{Dn} = \text{Id}$. We need to prove that $\langle\langle b \cdot \text{emb } B \rangle\rangle = b$.

Free monads

Algebras of endofunctors and monads

We would like to use the free algebra universal property. To do so, we must show that Eilenberg–Moore algebra $b : \Sigma^* B \rightarrow B$ is a Σ -homomorphism $(\Sigma^* B, \text{in } B) \rightarrow (B, b \cdot \text{emb } B)$:



Free monads

Algebras of endofunctors and monads

Therefore we can then appeal to the universal property:

$$\begin{array}{c} \Sigma^* \quad B \\ \text{---} \\ \bullet \\ | \\ \text{---} \\ B \end{array} = \begin{array}{c} \Sigma^* \quad B \\ \text{---} \\ \circ \\ | \\ \text{---} \\ B \\ b \cdot \text{emb } B \end{array} \iff \begin{array}{c} B \\ \text{---} \\ \bullet \\ | \\ \text{---} \\ B \\ \text{var} \\ \bullet \\ | \\ \text{---} \\ B \\ b \end{array} = \begin{array}{c} B \\ \text{---} \\ \bullet \\ | \\ \text{---} \\ B \end{array}$$

Applying the first computation rule completes the proof of the isomorphism

$$\text{Up} : \Sigma\text{-Alg}(\mathcal{C}) \cong \mathcal{C}^{\Sigma^*} : \text{Dn}.$$

The resumption monad

The challenge

The challenge

- ▶ Let $M : \mathcal{C} \rightarrow \mathcal{C}$ be a monad and $F : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor; we aim to show that $M \circ (F \circ M)^*$ is a monad.

The resumption monad

The challenge

The challenge

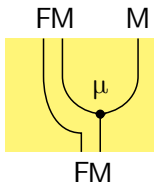
- ▶ Let $M : \mathcal{C} \rightarrow \mathcal{C}$ be a monad and $F : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor; we aim to show that $M \circ (F \circ M)^*$ is a monad.
- ▶ In fact, we shall generalise, and show that given a right monad action $\alpha : \Sigma \circ M \rightarrow \Sigma$, $M \circ \Sigma^*$ is a monad.

The resumption monad

The challenge

The challenge

- ▶ Let $M : \mathcal{C} \rightarrow \mathcal{C}$ be a monad and $F : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor; we aim to show that $M \circ (F \circ M)^*$ is a monad.
- ▶ In fact, we shall generalise, and show that given a right monad action $\alpha : \Sigma \circ M \rightarrow \Sigma$, $M \circ \Sigma^*$ is a monad.
- ▶ We recover the original result with $\Sigma = F \circ M$ and the monad action:



The resumption monad

The plan

We have the following structure available to us:

$$\Sigma\text{-Alg}(\mathcal{C}) \begin{array}{c} \xleftarrow{\text{Free}^\Sigma} \\ \perp \\ \xrightarrow{\text{U}^\Sigma} \end{array} \mathcal{C} \begin{array}{c} \curvearrowright \\ \text{M} \end{array} .$$

The resumption monad

The plan

We have the following structure available to us:

$$\Sigma\text{-Alg}(\mathcal{C}) \begin{array}{c} \xleftarrow{\text{Free}^\Sigma} \\ \perp \\ \xrightarrow{\text{U}^\Sigma} \end{array} \mathcal{C} \begin{array}{c} \curvearrowright \\ \text{M} \end{array} .$$

- ▶ This looks close to the Huber situation, but M lives at the “wrong end”.

The resumption monad

The plan

We have the following structure available to us:

$$\Sigma\text{-Alg}(\mathcal{C}) \begin{array}{c} \xleftarrow{\text{Free}^\Sigma} \\ \perp \\ \xrightarrow{\text{U}^\Sigma} \end{array} \mathcal{C} \begin{array}{c} \curvearrowright \\ \text{M} \end{array} .$$

- ▶ This looks close to the Huber situation, but M lives at the “wrong end”.
- ▶ If we have a monad $\overline{M} : \Sigma\text{-Alg}(\mathcal{C}) \rightarrow \Sigma\text{-Alg}(\mathcal{C})$, we could apply Huber’s construction, giving a monad $\text{U}^\Sigma \circ \overline{M} \circ \text{Free}^\Sigma$.

The resumption monad

The plan

We have the following structure available to us:

$$\Sigma\text{-Alg}(\mathcal{C}) \begin{array}{c} \xleftarrow{\text{Free}^\Sigma} \\ \perp \\ \xrightarrow{\text{U}^\Sigma} \end{array} \mathcal{C} \begin{array}{c} \curvearrowright \\ \text{M} \end{array}.$$

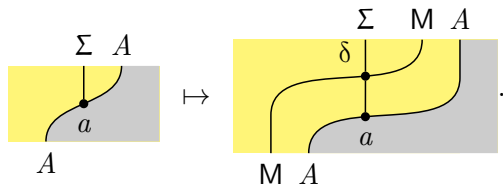
- ▶ This looks close to the Huber situation, but M lives at the “wrong end”.
- ▶ If we have a monad $\overline{M} : \Sigma\text{-Alg}(\mathcal{C}) \rightarrow \Sigma\text{-Alg}(\mathcal{C})$, we could apply Huber’s construction, giving a monad $\text{U}^\Sigma \circ \overline{M} \circ \text{Free}^\Sigma$.
- ▶ If additionally, $\text{U}^\Sigma \circ \overline{M} = M \circ \text{U}^\Sigma$, then we have:

$$\text{U}^\Sigma \circ \overline{M} \circ \text{Free}^\Sigma = M \circ \text{U}^\Sigma \circ \text{Free}^\Sigma = M \circ \Sigma^*.$$

The resumption monad

How to build a suitable \overline{M}

A natural transformation $\delta : \Sigma \circ M \rightarrow M \circ \Sigma$ induces a functor \overline{M} with action:



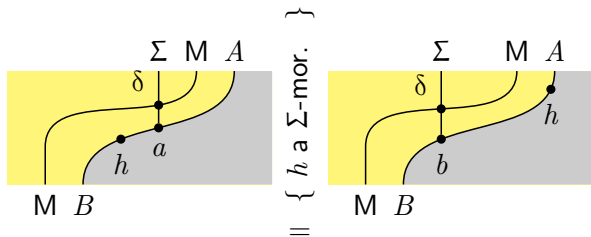
With:

$$U^\Sigma \circ \overline{M}(A, a) = U^\Sigma(M(A), M(a) \cdot \delta_A) = M(A) = M \circ U^\Sigma(A, a).$$

The resumption monad

How to build a suitable \bar{M}

This operation preserves homomorphisms as:



Preservation of identities and composition is then immediate as M does.

The resumption monad

Building a suitable δ

Given a right monad action $\alpha : \Sigma \circ M \rightarrow \Sigma$, we can build a suitable δ as the composite:

$$\begin{array}{ccc} \Sigma & & M \\ \delta & & \\ M & & \Sigma \end{array} := \begin{array}{ccc} \Sigma & & M \\ \eta & & \alpha \\ M & & \Sigma \end{array} .$$

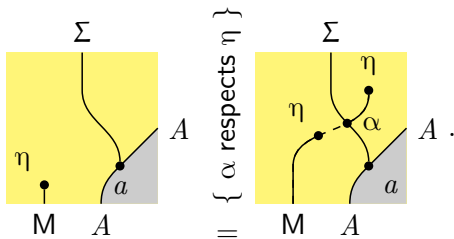
The resumption monad

Lifted unit and multiplication

To show that each component of $\bar{\eta} : \text{Id} \dot{\rightarrow} \bar{M}$,

$$\bar{\eta}(A, a) : (A, a) \rightarrow (M A, M a \cdot \delta A),$$

is a homomorphism, we simply apply the unit action law:



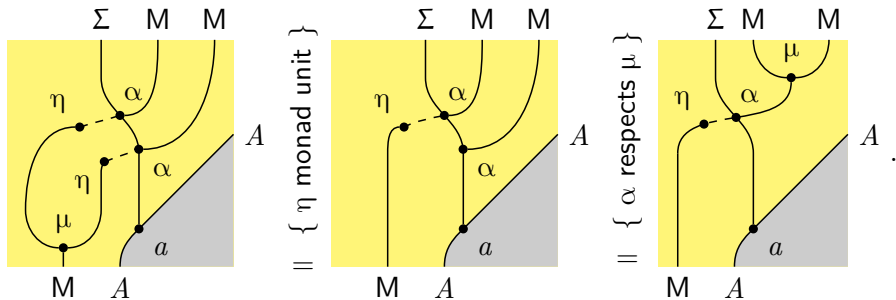
The resumption monad

Lifted unit and multiplication

Likewise, to establish that each component of $\bar{\mu} : \bar{M} \circ \bar{M} \rightarrow \bar{M}$,

$\bar{\mu}(A, a) : (M(M A), M(M a) \cdot M(\delta A) \cdot \delta(M A)) \rightarrow (M A, M a \cdot \delta A)$,

is a homomorphism, we reason



The resumption monad

Invoking Huber's construction

By construction, $\bar{\eta}$ and $\bar{\mu}$ satisfy the following equations:

The image displays two equations involving string diagrams for the resumption monad. Each diagram consists of a rectangular box divided into two vertical sections: a yellow section on the left and an orange section on the right. Vertical lines represent the monad's multiplication and comultiplication, and curved lines represent the multiplication and comultiplication of the monad.

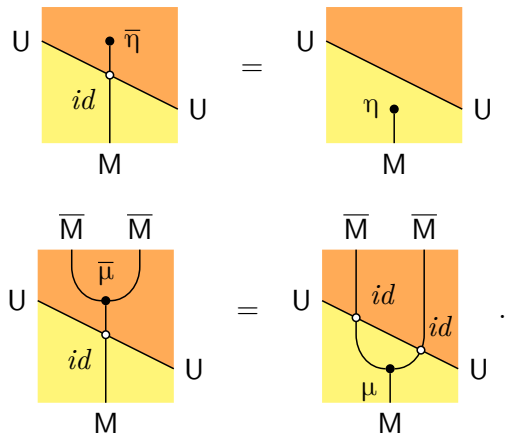
The first equation shows the naturality of the unit η . The left side features a vertical line labeled $\bar{\eta}$ with a dot at its base, positioned in the orange section. The top of the box is labeled U and the bottom is labeled U on the left and \bar{M} on the right. The right side features a vertical line labeled η with a dot at its base, positioned in the yellow section. The top of the box is labeled U and the bottom is labeled M on the left and U on the right.

The second equation shows the naturality of the multiplication μ . The left side features a curved line labeled $\bar{\mu}$ with a dot at its base, positioned in the orange section. The top of the box is labeled U on the left and \bar{M} on the right, and the bottom is labeled U on the left and \bar{M} on the right. The right side features a curved line labeled μ with a dot at its base, positioned in the yellow section. The top of the box is labeled M on the left and M on the right, and the bottom is labeled M on the left and U on the right.

The resumption monad

Invoking Huber's construction

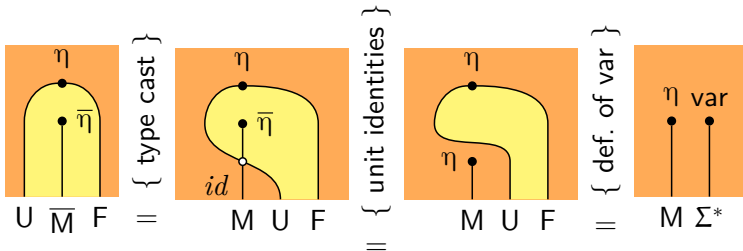
By adding explicit identity natural transformations, we get a more satisfactory rendition:



The resumption monad

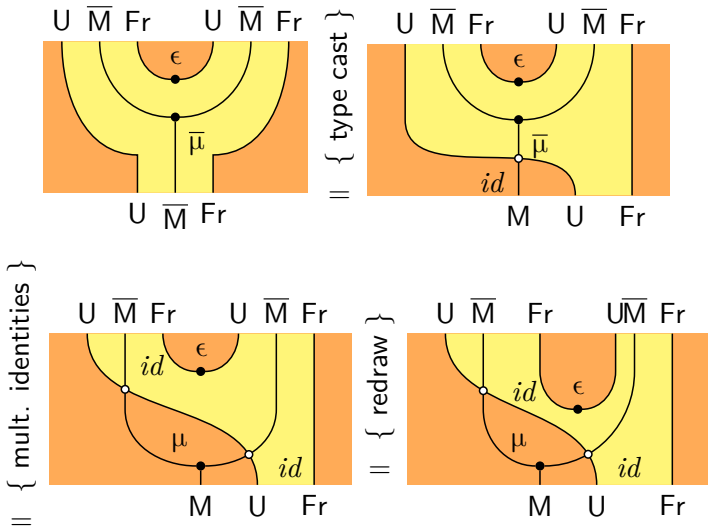
Invoking Huber's construction

We are now in a position to put the unit of the composite monad in concrete terms:



The resumption monad

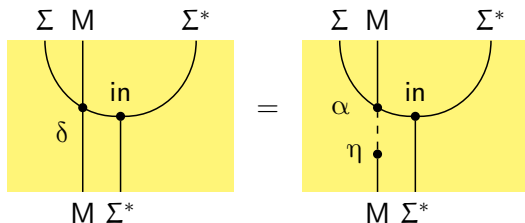
Invoking Huber's construction



The resumption monad

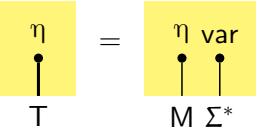
Explicit description

The counit can be described in terms of a fold for the following algebra:

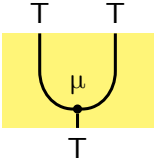


The resumption monad

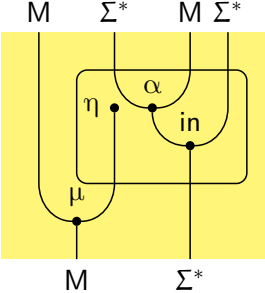
Explicit description



and



=



Further directions

Further elementary category theory with nice graphical perspectives, e.g.

- ▶ Universals and the Yoneda Lemma.
- ▶ Lots about distributive laws.
- ▶ Kan extensions and codensity monads.

Further directions

Other settings e.g.

- ▶ Monoidal categories, braided monoidal categories, symmetric monoidal categories...
- ▶ Double categories.
- ▶ Higher categories, and combinations of structures such as monoidal 2-categories.

Further directions

Applications e.g.

- ▶ Quantum theory / computation.
- ▶ Control theory.
- ▶ Linear algebra.
- ▶ Natural language semantics.
- ▶ Analog and digital electronics.

Further directions

Theory:

- ▶ PROs, PROPs, ...
- ▶ Coherence theorems.
- ▶ Expressivity, soundness and completeness results.

Further directions

Tools:

- ▶ Proof assistants.
- ▶ Diagramming tools.
- ▶ Diagramming libraries.