

# Monads

*category theory for  
effects in typed functional programming*

*monoids: a categorical generalisation  
algebras: a categorical generalisation*

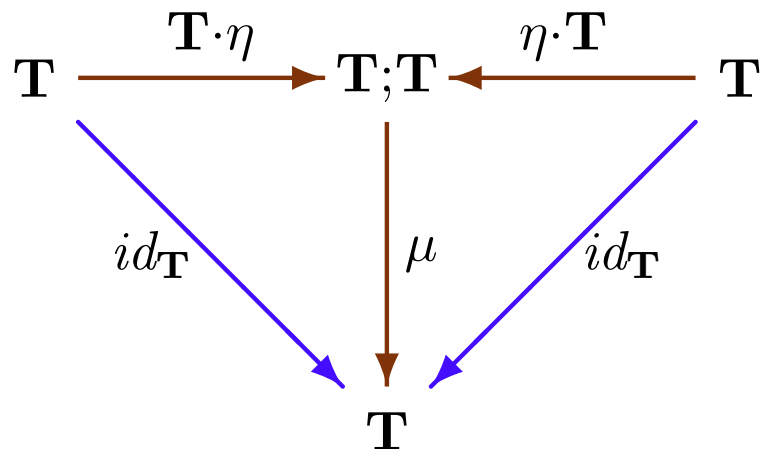
# Monads

A *monad* in a category  $\mathbf{K}$  is a triple:

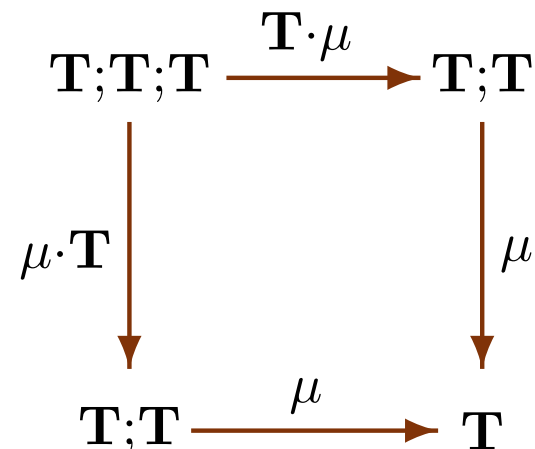
$$\langle \mathbf{T}: \mathbf{K} \rightarrow \mathbf{K}, \eta: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{T}, \mu: \mathbf{T}; \mathbf{T} \rightarrow \mathbf{T} \rangle$$

such that for each  $X \in |\mathbf{K}|$

- $\eta_{\mathbf{T}(X)}; \mu_X = \text{id}_{\mathbf{T}(X)} = \mathbf{T}(\eta_X); \mu_X$



- $\mu_{\mathbf{T}(X)}; \mu_X = \mathbf{T}(\mu_X); \mu_X$



## Trivial examples

- *Identity* monad
- *Terminal* monad
- Monads in partial orders: *closure operators*
- . . .

## Simple Examples

Examples of monads in **Set**

- *Partiality* monad:

- $\mathbf{P}(X) = X + \{\perp\};$
- $\eta_X^{\mathbf{P}}(x) = x;$
- $\mu_X^{\mathbf{P}}(x) = x$  for  $x \in X$ ,  $\mu_X^{\mathbf{P}}(x) = \perp$  for  $x \notin X$ .

- *Exceptions* monad;

- $\mathcal{E}(X) = X + E;$
- $\eta_X^{\mathcal{E}}(x) = x;$
- $\mu_X^{\mathcal{E}}(x) = x$  for  $x \in X$ ,  $\mu_X^{\mathcal{E}}(e) = e$  for  $e \in E$ .

- *Nondeterminism* monad:

- $\mathcal{P}(X) = 2^X;$
- $\eta_X^{\mathcal{P}}(x) = \{x\};$
- $\mu_X^{\mathcal{P}}(U) = \bigcup U$  for  $U \in 2^{2^X}$ .

## Typical examples

- *List* monad:

- $\mathcal{L}(X) = X^*$ ;
- $\eta_X^{\mathcal{L}}(x) = \langle x \rangle$ ;
- $\mu_X^{\mathcal{L}}(\langle l_1, \dots, l_n \rangle) = \text{append}(l_1, \dots \text{append}(l_{n-1}, l_n) \dots)$ .

Examples of monads in **Set**

- *Term* monad:

- $\mathcal{T}_{\Sigma}(X) = T_{\Sigma}(X)$ ;
- $\eta_X^{\mathcal{T}_{\Sigma}}(x) = x$ ;
- $\mu_X^{\mathcal{T}_{\Sigma}}(t) = t[id_{T_{\Sigma}(X)}]$  for  $t \in T_{\Sigma}(T_{\Sigma}(X))$ .

## Difficult(?) examples

Examples of monads in **Set**

- *Side-effects* monad:

- $\mathcal{S}(X) = (X \times S)^S$ ;
- $\eta_X^{\mathcal{S}}(x)(s) = \langle x, s \rangle$ ;
- $\mu_X^{\mathcal{S}}(f)(s) = g(s')$  where  $f(s) = \langle g, s' \rangle$ , for  $f \in ((X \times S)^S \times S)^S$ .

- *Continuation* monad:

- $\mathcal{K}(X) = A^{(A^X)}$ ;
- $\eta_X^{\mathcal{K}}(x)(k) = k(x)$ ;
- $\mu_X^{\mathcal{K}}(f)(k) = f(\lambda g \in A^{(A^X)} . g(k))$ , for  $f \in A^{(A^{(A^{(A^X)})})}$ .

## Instead of more examples

*Adjunctions give rise to monads*

**Fact:** For any adjunction  $\langle \mathbf{F}, \mathbf{G}, \eta, \varepsilon \rangle: \mathbf{K} \rightarrow \mathbf{K}'$ , we have the monad:

$$\langle \mathbf{T}: \mathbf{K} \rightarrow \mathbf{K}, \eta^{\mathbf{T}}: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{T}, \mu^{\mathbf{T}}: \mathbf{T};\mathbf{T} \rightarrow \mathbf{T} \rangle$$

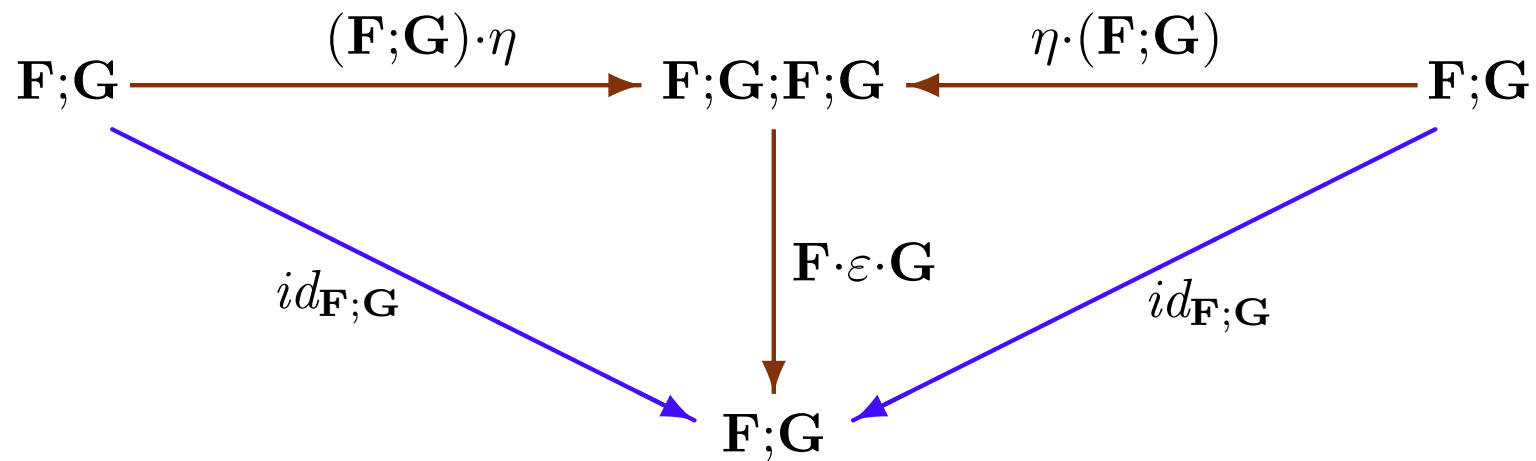
given by:

- $\mathbf{T} = \mathbf{F};\mathbf{G}$
- $\eta^{\mathbf{T}} = \eta: \text{Id}_{\mathbf{K}} \rightarrow \mathbf{F};\mathbf{G}$
- $\mu^{\mathbf{T}} = \mathbf{F} \cdot \varepsilon \cdot \mathbf{G}: \mathbf{F};(\mathbf{G};\mathbf{F});\mathbf{G} \rightarrow \mathbf{F};\mathbf{G}$   
(i.e.  $\mu_X^{\mathbf{T}} = \mathbf{G}(\varepsilon_{\mathbf{F}(X)}): \mathbf{G}(\mathbf{F}(\mathbf{G}(\mathbf{F}(X)))) \rightarrow \mathbf{G}(\mathbf{F}(X))$ )

## Proof

unit laws:

- $(G \cdot \eta); (\varepsilon \cdot G) = id_G$  implies  $(F \cdot (G \cdot \eta)); (F \cdot \varepsilon \cdot G) = id_{F;G}$
- $(\eta \cdot F); (F \cdot \varepsilon) = id_F$  implies  $((\eta \cdot F) \cdot G); (F \cdot \varepsilon \cdot G) = id_{F;G}$





## Proof cntd.

associativity:

$$\begin{array}{ccc}
 \mathbf{F};\mathbf{G};\mathbf{F};\mathbf{G};\mathbf{F};\mathbf{G} & \xrightarrow{(\mathbf{F};\mathbf{G}) \cdot (\mathbf{F} \cdot \varepsilon \cdot \mathbf{G})} & \mathbf{F};\mathbf{G};\mathbf{F};\mathbf{G} \\
 \downarrow (\mathbf{F} \cdot \varepsilon \cdot \mathbf{G}) \cdot (\mathbf{F};\mathbf{G}) & & \downarrow \mathbf{F} \cdot \varepsilon \cdot \mathbf{G} \\
 \mathbf{F};\mathbf{G};\mathbf{F};\mathbf{G} & \xrightarrow{\mathbf{F} \cdot \varepsilon \cdot \mathbf{G}} & \mathbf{F};\mathbf{G}
 \end{array}$$

Follows by the commutativity of the diagrams below:

$$\begin{array}{ccc}
 \mathbf{G};\mathbf{F};\mathbf{G};\mathbf{F} & \xrightarrow{\mathbf{G} \cdot (\mathbf{F} \cdot \varepsilon)} & \mathbf{G};\mathbf{F} \\
 \downarrow (\varepsilon \cdot \mathbf{G}) \cdot \mathbf{F} & & \downarrow \varepsilon \\
 \mathbf{G};\mathbf{F} & \xrightarrow{\varepsilon} & \text{Id}_{\mathbf{K}}
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathbf{F}(\mathbf{G}(\mathbf{F}(\mathbf{G}(\mathbf{X})))) & \xrightarrow{\varepsilon_{\mathbf{F}(\mathbf{G}(\mathbf{X}))}} & \mathbf{F}(\mathbf{G}(\mathbf{X})) \\
 \downarrow \mathbf{F}(\mathbf{G}(\varepsilon_{\mathbf{X}})) & & \downarrow \varepsilon_{\mathbf{X}} \\
 \mathbf{F}(\mathbf{G}(\mathbf{X})) & \xrightarrow{\varepsilon_{\mathbf{X}}} & \mathbf{X}
 \end{array}$$

# Algebras

Check this out for the term monad

Given a monad  $\langle \mathbf{T}, \eta, \mu \rangle$  in  $\mathbf{K}$ :

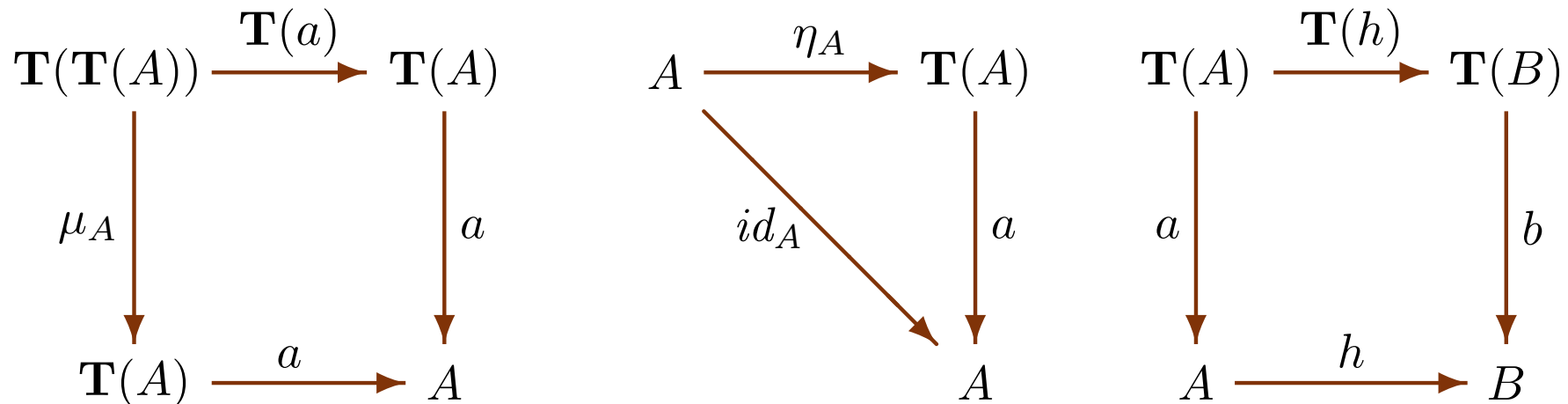
The category  $\mathbf{Alg}(\mathbf{T})$  of  $\mathbf{T}$ -algebras and  $\mathbf{T}$ -homomorphisms:

- $\mathbf{T}$ -algebras:

$\langle A \in |\mathbf{K}|, a: \mathbf{T}(A) \rightarrow A \rangle$  such that  $\mathbf{T}(a);a = \mu_A;a$  and  $\eta_A;a = id_A$

- $\mathbf{T}$ -homomorphism from  $\langle A, a: \mathbf{T}(A) \rightarrow A \rangle$  to  $\langle B, b: \mathbf{T}(B) \rightarrow B \rangle$ :

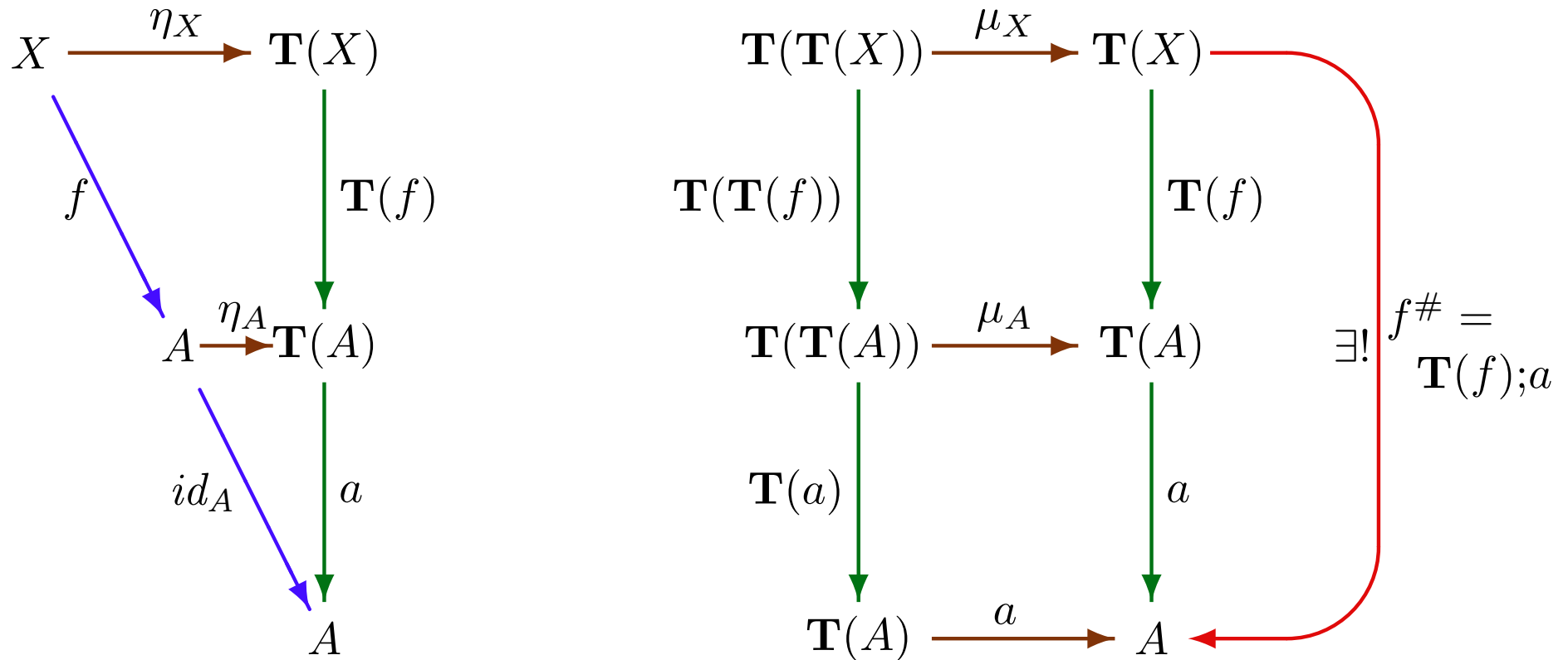
$h: A \rightarrow B$  such that  $\mathbf{T}(h);b = a;h$



## Monadic adjunction

Let  $\mathbf{G}^{\mathbf{T}}: \mathbf{Alg}(\mathbf{T}) \rightarrow \mathbf{K}$  be the obvious projection:  $\mathbf{G}^{\mathbf{T}}(\langle A, a \rangle) = A, \dots$

For  $X \in |\mathbf{K}|$ ,  $\mathbf{F}^{\mathbf{T}}(X) = \langle \mathbf{T}(X), \mu_X: \mathbf{T}(\mathbf{T}(X)) \rightarrow \mathbf{T}(X) \rangle$  with unit  $\eta_X: X \rightarrow \mathbf{G}^{\mathbf{T}}(\mathbf{F}^{\mathbf{T}}(X))$  is free over  $X$  w.r.t.  $\mathbf{G}$ :



## All monads arise from adjunctions

Given a monad  $\langle \mathbf{T}, \eta, \mu \rangle$  in  $\mathbf{K}$  we have an adjunction

$$\langle \mathbf{F}^{\mathbf{T}}, \mathbf{G}^{\mathbf{T}}, \eta, \varepsilon^{\mathbf{T}} \rangle : \mathbf{K} \rightarrow \mathbf{Alg}(\mathbf{T})$$

$$\varepsilon_{\langle A, a \rangle} : \mathbf{F}^{\mathbf{T}}(\mathbf{G}^{\mathbf{T}}(\langle A, a \rangle)) \rightarrow \langle A, a \rangle \text{ is } a : \mathbf{T}(A) \rightarrow A.$$

**Fact:**  $\langle \mathbf{T}, \eta, \mu \rangle$  is the monad associated with  $\langle \mathbf{F}^{\mathbf{T}}, \mathbf{G}^{\mathbf{T}}, \eta, \varepsilon^{\mathbf{T}} \rangle$ .

**Fact:** Given an adjunction  $\langle \mathbf{F}, \mathbf{G}, \eta, \varepsilon \rangle : \mathbf{K} \rightarrow \mathbf{K}'$ , let  $\langle \mathbf{T} = \mathbf{F};\mathbf{G}, \eta, \mu^{\mathbf{T}} = \mathbf{F} \cdot \varepsilon \cdot \mathbf{G} \rangle$  be the monad it yields, and then let  $\langle \mathbf{F}^{\mathbf{T}}, \mathbf{G}^{\mathbf{T}}, \eta, \varepsilon^{\mathbf{T}} \rangle : \mathbf{K} \rightarrow \mathbf{Alg}(\mathbf{T})$  be the adjunction for  $\mathbf{T}$ . Then there is a unique *comparison functor*  $\Phi : \mathbf{K}' \rightarrow \mathbf{Alg}(\mathbf{T})$  such that  $\Phi; \mathbf{G}^{\mathbf{T}} = \mathbf{G}$  and  $\mathbf{F}; \Phi = \mathbf{F}^{\mathbf{T}}$ .

$$\Phi(A') = \langle \mathbf{G}(A'), \mathbf{G}(\varepsilon_{A'}) \rangle : \mathbf{G}(\mathbf{F}(\mathbf{G}(A'))) \rightarrow \mathbf{G}(A')$$

## Free algebras

Given a monad  $\langle \mathbf{T}, \eta, \mu \rangle$  in  $\mathbf{K}$ :

The *Kleisli category*  $\mathbf{Kl}(\mathbf{T})$  for  $\mathbf{T}$ :

- objects:  $|\mathbf{Kl}(\mathbf{T})| = |\mathbf{K}|$
- morphisms:  $f: X \rightarrow Y$  in  $\mathbf{Kl}(\mathbf{T})$  are morphisms  $f: X \rightarrow \mathbf{T}(Y)$  in  $\mathbf{K}$
- composition: given  $f: X \rightarrow Y$ ,  $g: Y \rightarrow Z$  in  $\mathbf{Kl}(\mathbf{T})$ ,  $f;g: X \rightarrow Z$  in  $\mathbf{Kl}(\mathbf{T})$  is  $f;\mathbf{T}(g);\mu_Y: X \rightarrow \mathbf{T}(Z)$  in  $\mathbf{K}$ .

$$X \xrightarrow{f} \mathbf{T}(Y) \xrightarrow{\mathbf{T}(g)} \mathbf{T}(\mathbf{T}(Z)) \xrightarrow{\mu_Y} \mathbf{T}(Z)$$

**Again:** there is an adjunction  $\langle \mathbf{F}^{\mathbf{T}}, \mathbf{G}^{\mathbf{T}}, \eta, \varepsilon^{\mathbf{T}} \rangle: \mathbf{K} \rightarrow \mathbf{Kl}(\mathbf{T})$  which gives rise to the monad  $\langle \mathbf{T}, \eta, \mu \rangle$ , and for any adjunction  $\langle \mathbf{F}, \mathbf{G}, \eta, \varepsilon \rangle: \mathbf{K} \rightarrow \mathbf{K}'$  which also gives rise to this monad, we have a comparison functor  $\Psi: \mathbf{K}' \rightarrow \mathbf{Kl}(\mathbf{T})$  such that  $\Psi;\mathbf{G}^{\mathbf{T}} = \mathbf{G}$  and  $\mathbf{F};\Psi = \mathbf{F}^{\mathbf{T}}$ .

View  $\mathbf{Kl}(\mathbf{T})$  as the image of  $\mathbf{F}^{\mathbf{T}}$  in  $\mathbf{Alg}(\mathbf{T})$

## Triples

A *triple* in  $\mathbf{K}$ :

$$\langle T, \eta, (-)^* \rangle$$

where

- $T: |\mathbf{K}| \rightarrow |\mathbf{K}|$ ,
- $\eta_A: A \rightarrow T(A)$  for all  $A \in |\mathbf{K}|$ ,
- $f^*: T(A) \rightarrow T(B)$  for all  $f: A \rightarrow T(B)$

are such that

- $\eta_A^* = id_{T(A)}$  for all  $A \in |\mathbf{K}|$
- $\eta_A; f^* = f$  for all  $f: A \rightarrow T(B)$
- $f^*; g^* = (f; g^*)^*$  for all  $f: A \rightarrow T(A)$ ,

Triples and monads are the same concepts

## Triples as monads, monads as triples

Given a monad  $\langle \mathbf{T}, \eta, \mu \rangle$  in  $\mathbf{K}$ , put:

- $T(A) = \mathbf{T}(A)$  for  $A \in |\mathbf{K}|$ ,
- $\eta_A = \eta_A: A \rightarrow T(A)$  for  $A \in |\mathbf{K}|$ ,
- $f^* = \mathbf{T}(f); \mu_A: T(A) \rightarrow T(B)$  for  $f: A \rightarrow T(B)$ .

This yields a triple  $\langle T, \eta, (-)^* \rangle$ .

*“Triple” the monads given as examples*

Given a triple  $\langle T, \eta, (-)^* \rangle$  in  $\mathbf{K}$ , put:

- $\mathbf{T}(A) = T(A)$  for  $A \in |\mathbf{K}|$ , and  $\mathbf{T}(f) = (f; \eta_B)^*$  for  $f: A \rightarrow B$ ,
- $\eta_A = \eta_A: A \rightarrow T(A)$  for  $A \in |\mathbf{K}|$ ,
- $\mu_A = id_{T(A)}^*: T(T(A)) \rightarrow T(A)$  for  $A \in |\mathbf{K}|$ ,

This yields a monad  $\langle T, \eta, \mu \rangle$ .

## Further monadic concepts

- Most importantly:

### *Functional programming with effects*

Given a triple  $\langle T, \eta, (-)^* \rangle$ :

- **return**  $--$ :  $\alpha \rightarrow T\alpha$  is  $\eta_\alpha$
- $-->>=$   $--$ :  $T\alpha \rightarrow (\alpha \rightarrow T\beta) \rightarrow T\beta$  is given by  $x >>= f = f^*(x)$
- **do**-notation from  $-->>=$   $--$  and  $\lambda$ -notation
- *Strong* (context-preserving) monads
- Monad *composition* and *distributivity laws* for monads
- Monad *transformers*
- ...