Rolling transformers

Rik van Geldrop and Jaap van der Woude

Technische Universiteit
Eindhoven
University of Technology

why

just for fun and old times' sake

monad transformer $\mathbb{T}.\mathsf{M} \in \mathcal{M} \to \mathcal{M}$

- understand (notion and literature) $\mathit{lift} \in \mathsf{M} \xrightarrow{\bullet} \mathbb{T}.\mathsf{M}.\mathsf{H}$
- teach
- prove
- not yet program design

flat monads	: state, error, writer and reader	adjunction composition
inductive monads	: list, tree, free	???

how / what

we got a blueprint for construction (and proof and code and functoriality) of a

- free monad monad transformer (and binary tree monad transformer)
- rosetree monad transformer
- list monad transformer

we used

- rolling rule — categorical version of $\mu(fg) = f.\mu(gf) \Leftarrow \mu(gf)$ exists
- parameterised fixpoints of bifunctors — $\mathsf{in}_{\mathsf{F}} \in \mu(\mathsf{F}.A) \leftarrow \mathsf{F}.A.(\mu\mathsf{F}.A)$
- monads / extension structures
- monad morphisms

monad / extension structure

In Haskell a monad is defined by its Kleisli splitting (though they hide it deeply).

extension structure — Manes 1976

$(\mathsf{H}, \upsilon, \lhd)$ with $\mathsf{H} \in \mathcal{C} \to \mathcal{C}$ a type function, $\upsilon_X \in X \to \mathsf{H}.X$ and $\lhd_{X,Y} \in (X \to \mathsf{H}.Y) \longrightarrow (\mathsf{H}.X \to \mathsf{H}.Y)$ object-indexed

$$\lhd.\upsilon = id \qquad \text{ES-identity}$$

$$\lhd.f \circ \upsilon = f \qquad \text{ES-extension}$$

$$\lhd.(\lhd.f \circ g) = \lhd.f \circ \lhd.g \qquad \text{ES-composition}$$

the dnib $\lhd$ is usually declared as `>>= : H X -> (X -> H Y) -> H Y` thus obfuscating functoriality of $(\mathsf{H}, \lhd) \in \mathcal{K}_{\mathsf{H}} \to \mathcal{C}$.

monad / extension structure

Note that $\mathsf{H}.f = \lhd.(\upsilon \circ f)$ turns the object function H into a functor.

The Kleisli splitting is the adjunction $((\mathsf{I}, \upsilon \circ), (\mathsf{H}, \lhd), \upsilon)$.

the adjunction leads to the monad $(\mathsf{H}, \upsilon, \lhd.id)$ and
every monad($(\mathsf{H}, \upsilon, \theta)$ defines an extension structure via $\lhd.f = \theta \circ \mathsf{H}.f$

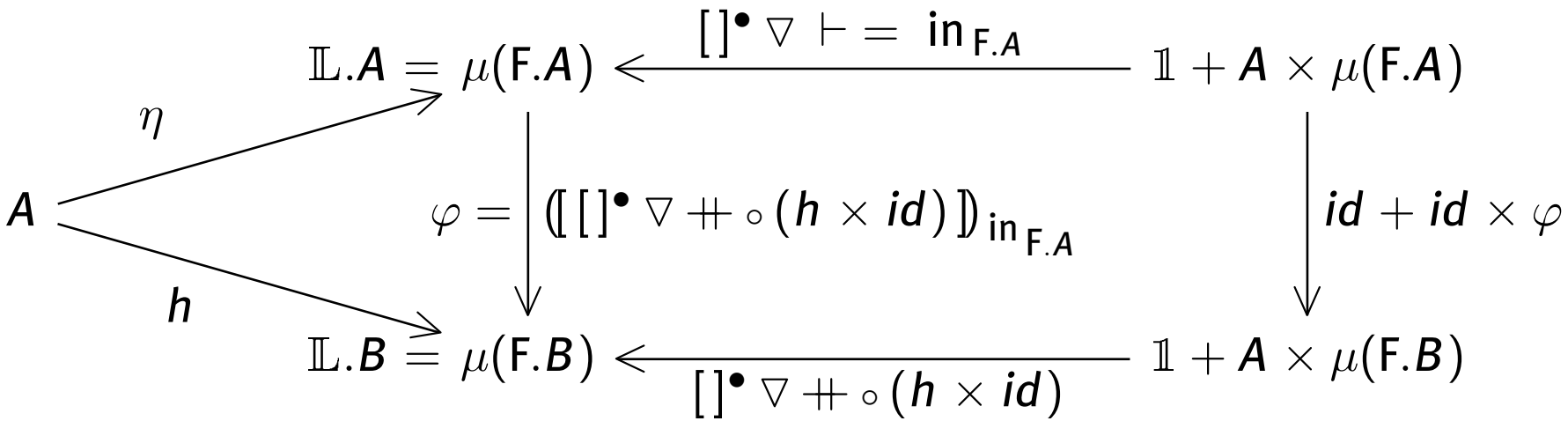
Monads form a category where the arrows are monad morphisms. The extension structure form is

monad morphism

Let $(\mathsf{H}, \upsilon, \lhd)$ and $(\mathsf{H}', \upsilon', \lhd')$ be extension structures.
A monad morphism is a natural transformation $\sigma \in \mathsf{H} \xrightarrow{\bullet} \mathsf{H}'$ with
$\sigma \circ \upsilon = \upsilon'$ and $\sigma \circ \lhd .f = \lhd'.(\sigma \circ f) \circ \sigma$

list monad

$$\mathsf{F}.A.X \;=\; \mathbb{1} + A \times X$$



the monoid operator $+\!\!+ \in \mu(\mathsf{F}.B) \times \mu(\mathsf{F}.B) \to \mu(\mathsf{F}.B)$ is defined by

$$(+\!\!+ ls) \;=\; (\!|\, ls^\bullet \triangledown \vdash \,|\!)_{\mathsf{in}_{\mathsf{F}.A}}$$

The list monad is given as $(\mathbb{L}, \eta, \lhd^{\mathbb{L}})$ where

$$\begin{aligned} \eta &= \vdash \circ (id \vartriangle [\,]^\bullet) = [\_] = (\vdash [\,]) \\ \lhd^{\mathbb{L}}.h &= (\!|\, [\,]^\bullet \triangledown +\!\!+ \circ (h \times id)\,|\!)_{\mathsf{in}_{\mathsf{F}.A}} \end{aligned}$$

rolling

For a μF-monad transformer we have to weave $(\mathsf{H}, \upsilon, \lhd)$ with F-levels in $\mu\mathsf{F}$.

So we look at $\mu(\mathsf{F}.A \circ \mathsf{H})$ and/or $\mu(\mathsf{H} \circ \mathsf{F}.A)$

rolling rule

Assume that an FH-initial algebra $\mathsf{in}_{\mathsf{FH}} \in \mu\mathsf{FH} \leftarrow \mathsf{FH}.\mu\mathsf{FH}$ exists,
then $\mathsf{H}.\mathsf{in}_{\mathsf{FH}} \in \mathsf{H}.\mu\mathsf{FH} \leftarrow \mathsf{HFH}.\mu\mathsf{FH}$ is an initial HF-algebra.
$(\!|f|\!)_{\mathsf{H}.\,\mathsf{in}_{\mathsf{FH}}} = f \circ \mathsf{H}.(\!|\mathsf{F}.f|\!)_{\mathsf{in}_{\mathsf{FH}}}$

We chose $\mathsf{in}_{\mathsf{HF}} = \mathsf{H}.\mathsf{in}_{\mathsf{FH}}$

$\mathsf{FH}.A.X = \mathsf{F}.A.(\mathsf{H}.X)$ and $\mathsf{HF}.A.X = \mathsf{H}.(\mathsf{F}.A.X)$ are binary functors,
we look at their parameterised fixpoints with the choice above.

parameterised fixpoints

Let $\mathsf{F} \in \mathcal{A} \times \mathcal{C} \to \mathcal{C}$ be a binary functor such that for all A the initial $\mathsf{F}.A$-algebra $\mathsf{in}_{\mathsf{F}.A} \in \mu(\mathsf{F}.A) \leftarrow (\mathsf{F}.A).\mu(\mathsf{F}.A)$ exists.
Then it generates a functor $\mathbb{P}_{\mathsf{F}} \in \mathcal{A} \to \mathcal{C}$ given by

$$\mathbb{P}_{\mathsf{F}}.A \;=\; \mu(\mathsf{F}.A) \quad \text{and} \quad \mathbb{P}_{\mathsf{F}}.f \;=\; (\![\mathsf{in}_{\mathsf{F}.B} \circ \mathsf{F}.f.id]\!)_{\mathsf{in}_{\mathsf{F}.A}} \quad \text{for } f \in A \to B$$

For instance $\mathsf{F}.A.X \;=\; \mathbb{1} + A \times X$, the functor with lists as fixpoint then
$\mathbb{P}_{\mathsf{F}}.A \;=\; \mathbb{L}.A \;=\; \texttt{[A]}$ and $\mathbb{P}_{\mathsf{F}}.f \;=\; \mathbb{L}.f \;=\; (\![[\,]^{\bullet} \triangledown \vdash \circ (f \times id)]\!) \;=\; \texttt{map f}$

What does that mean for FH and HF ?

$$\mathbb{P}_{\mathsf{FH}}.A \;=\; \mu(\mathsf{FH}.A) \quad \text{and} \quad \mathbb{P}_{\mathsf{FH}}.f \;=\; (\![\mathsf{in}_{\mathsf{FH}.B} \circ \mathsf{FH}.f.id]\!)_{\mathsf{FH}} \;=\; (\![\mathsf{in}_{\mathsf{FH}.B} \circ \mathsf{F}.f.id]\!)_{\mathsf{FH}}$$
$$\mathbb{P}_{\mathsf{HF}}.A \;=\; \mathsf{H}.\mu(\mathsf{FH}.A) \quad \text{and} \quad \mathbb{P}_{\mathsf{HF}}.f \;=\; (\![\mathsf{H}.(\mathsf{in}_{\mathsf{FH}.B} \circ \mathsf{F}.f.id)]\!)_{\mathsf{HF}} \;=\; \mathsf{H}.(\mathbb{P}_{\mathsf{FH}}.f)$$

parameterised fixpoints

The functors $\mathbb{P}_{\mathsf{F}}$ and $\mathbb{P}_{\mathsf{HF}}$, $\mathbb{P}_{\mathsf{FH}}$ are related via natural transformations:

lemma

$$\mathit{lift}_{\mathsf{HF}} = (\![\upsilon \circ \mathsf{in}_{\mathsf{FH}}]\!)_{\mathsf{F}} \quad \in \mathbb{P}_{\mathsf{F}} \xrightarrow{\bullet} \mathbb{P}_{\mathsf{HF}} \text{ and}$$
$$\mathit{lift}_{\mathsf{FH}} = (\![\mathsf{in}_{\mathsf{FH}} \circ \mathsf{F}.\mathit{id}.\upsilon]\!)_{\mathsf{F}} \in \mathbb{P}_{\mathsf{F}} \xrightarrow{\bullet} \mathbb{P}_{\mathsf{FH}}$$

So both $\mathbb{P}_{\mathsf{HF}}$ and $\mathbb{P}_{\mathsf{FH}}$ may serve as monad transformer.
We chose $\mathbb{P}_{\mathsf{HF}}$ since the images are H-images which is Kleisli-friendly

Note that

$$(\![\lhd.\alpha]\!)_{\mathsf{HF}} \circ \mathit{lift}_{\mathsf{HF}} = (\![\alpha]\!)_{\mathsf{F}}$$

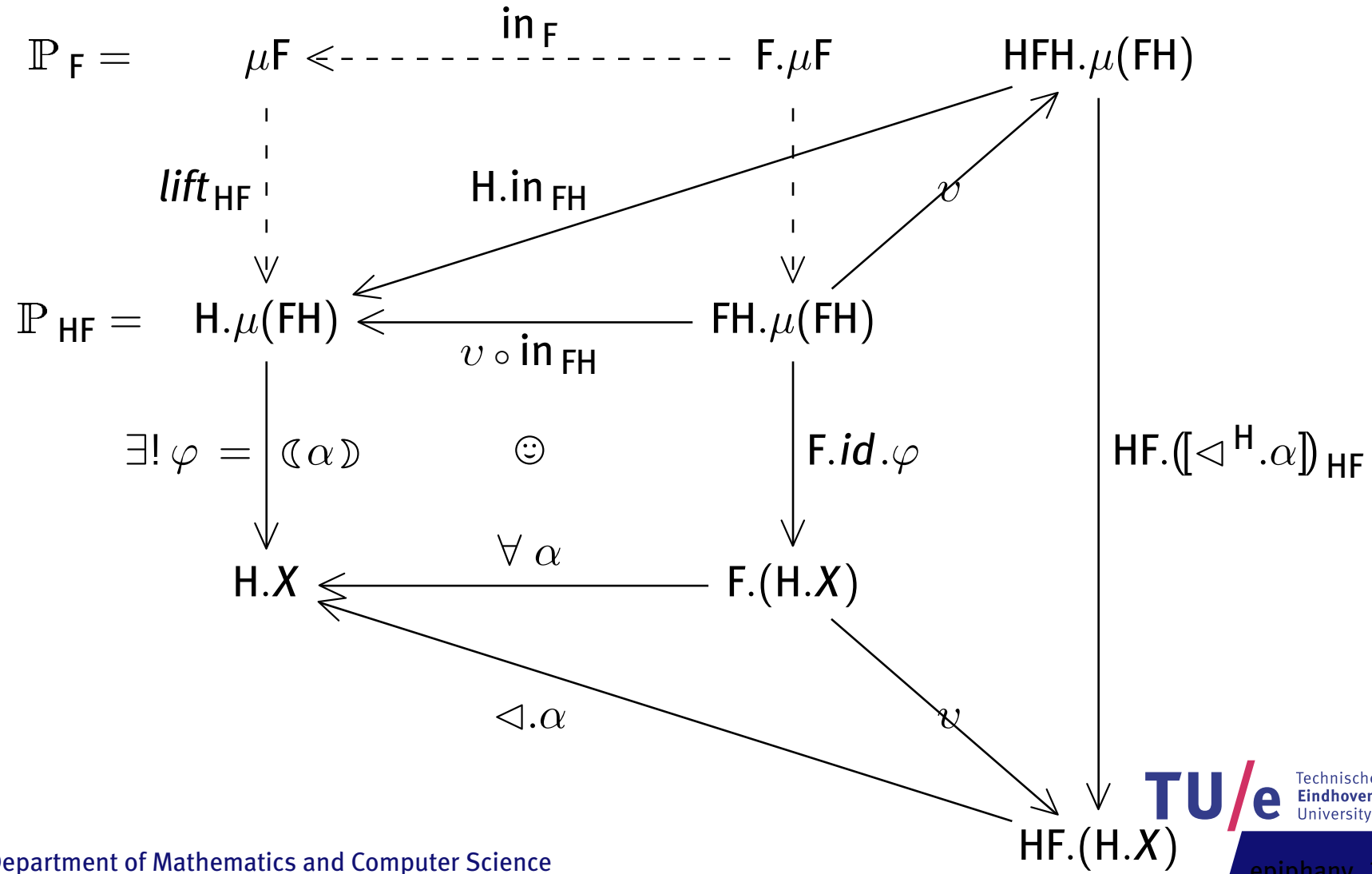
proof

$$
\begin{array}{ll}
& (\!|\lhd.\alpha|\!)_{\mathsf{HF}} \circ (\!|\upsilon \circ \mathsf{in}_{\mathsf{FH}}|\!)_{\mathsf{F}} \;=\; (\!|\alpha|\!)_{\mathsf{F}} \\
\Leftarrow & \quad \{ \text{ fusion } \} \\
& (\!|\lhd.\alpha|\!)_{\mathsf{HF}} \circ \upsilon \circ \mathsf{in}_{\mathsf{FH}} \;=\; \alpha \circ \mathsf{F}.(\!|\lhd.\alpha|\!)_{\mathsf{HF}}
\end{array}
$$

and $\upsilon \circ \mathsf{in}_{\mathsf{FH}}$ is F-initial

$$
\begin{array}{ll}
& (\!|\lhd.\alpha|\!)_{\mathsf{HF}} \circ \upsilon \circ \mathsf{in}_{\mathsf{FH}} \\
= & \quad \{ \;\upsilon\; \text{is natural } \} \\
& (\!|\lhd.\alpha|\!)_{\mathsf{HF}} \circ \mathsf{H}.\mathsf{in}_{\mathsf{FH}} \circ \upsilon \\
= & \quad \{ \text{ cata characterisation } \} \\
& \lhd.\alpha \circ \mathsf{HF}.(\!|\lhd.\alpha|\!)_{\mathsf{HF}} \circ \upsilon \\
= & \quad \{ \;\upsilon\; \text{is natural } \} \\
& \lhd.\alpha \circ \upsilon \circ \mathsf{F}.(\!|\lhd.\alpha|\!)_{\mathsf{HF}} \\
= & \quad \{ \text{ ES-unit } \} \\
& \alpha \circ \mathsf{F}.(\!|\lhd.\alpha|\!)_{\mathsf{HF}}
\end{array}
$$

moon $\quad (\!(\alpha)\!) = ([\![\alpha]\!])_{\upsilon \circ \mathsf{in}_{\mathsf{FH}}} = ([\![\lhd.\alpha]\!])_{\mathsf{HF}}$



list monad transformer

Define the list monad transformer following the structure of the list monad

With $\mathsf{F}.A.X = \mathbb{1} + A \times X$ and monad $(\mathsf{H}, \upsilon, \lhd)$ we set

$$\mathbb{LT}.A = \mathbb{P}_{\mathsf{HF}}.A \text{ and } \mathbb{K}.A = \mathbb{P}_{\mathsf{FH}}.A$$

$$\mathsf{in}_{\mathsf{FH}} = \overline{[\,]}^{\bullet} \triangledown \overline{\vdash} \in \mathbb{K}.A \leftarrow \mathbb{1} + A \times \mathsf{H}.(\mathbb{K}.A)$$

$$\upsilon \circ \mathsf{in}_{\mathsf{FH}} = (\upsilon \circ \overline{[\,]}^{\bullet}) \triangledown (\upsilon \circ \overline{\vdash}) = \in \mathbb{LT}.A \leftarrow \mathbb{1} + A \times \mathbb{LT}.A$$

is an initial F-algebra for H-images

denote it by $\ll\gg^{\bullet} \triangledown \oplus$ (kind of LTempty and LTcons)

Like $+\!\!+$ based on $\vdash$ define the operator $\oplus\!\!\!\oplus$ based on $\oplus$ by

$$(\oplus\!\!\!\oplus\, ls) = (\!| ls^{\bullet} \triangledown \oplus |\!)$$

it forms a monoid with unit $\ll\gg$

titel

List monad Transformer in H

Assume that $\mathbb{1} + A \times H$ has an initial algebra for all A. Then

$(\mathbb{L}\mathbb{T}, \eta', \lhd')$ is a monad

$lift \in (\mathbb{L}, (\vdash [\,]), \lhd^{\mathbb{L}}) \xrightarrow{\bullet} (\mathbb{L}\mathbb{T}, \eta', \lhd')$ is a monad morphism

where

$$\eta' = (\oplus \ll\gg)$$

$$\lhd'.f = (\!| \ll\gg^{\bullet} \triangledown \oplus \circ (f \times id) |\!)$$

$$lift = (\!| \upsilon \circ \mathsf{in}_{\mathsf{FH}} |\!)_{\mathsf{F}} = (\!| \ll\gg^{\bullet} \triangledown \oplus |\!)_{\mathsf{F}}$$

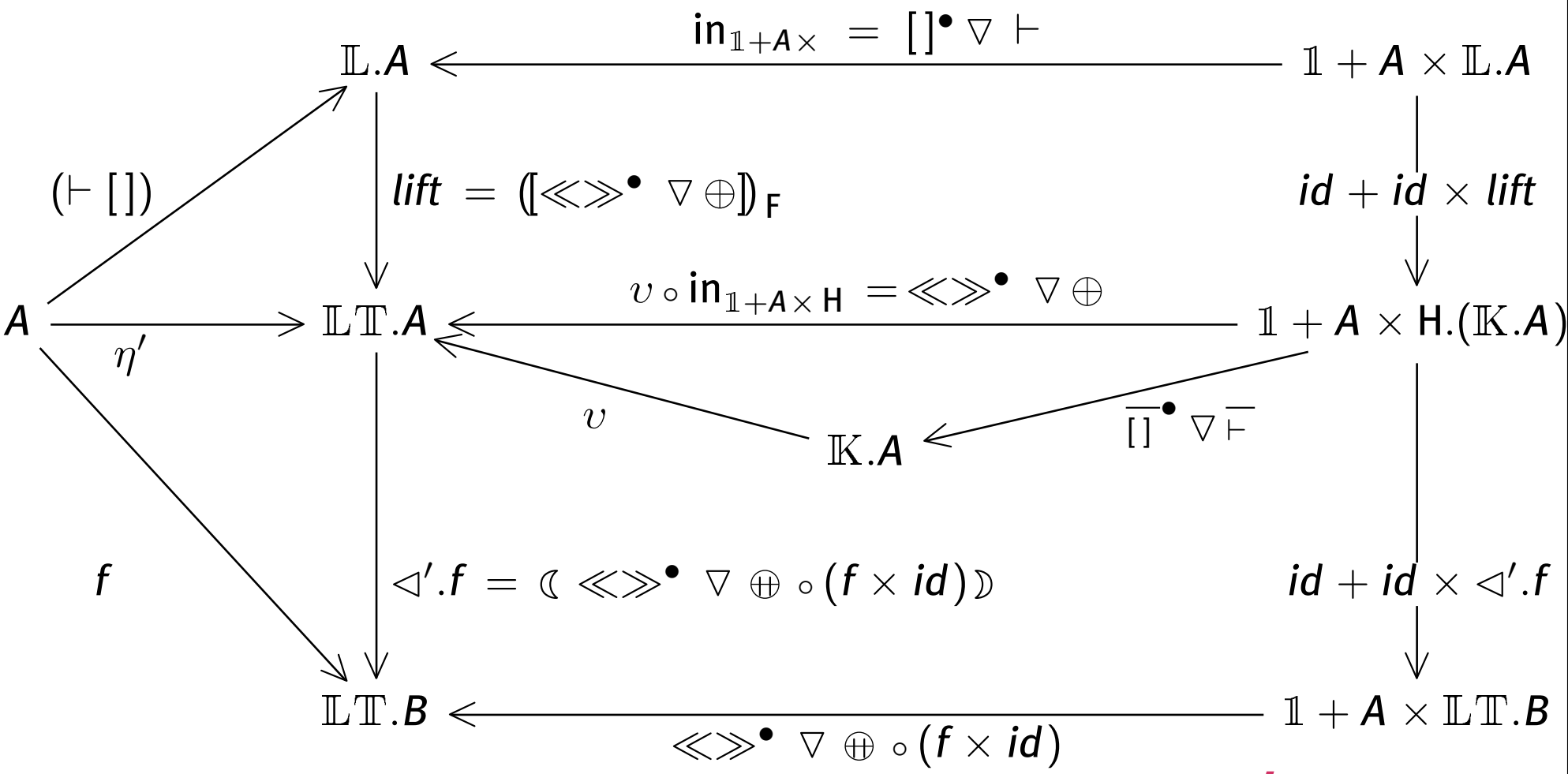
Compare the list monad given by

$(\mathbb{L}, \eta, \lhd^{\mathbb{L}})$ where

$$\eta = \vdash \circ (id \vartriangle [\,]^{\bullet}) = (\vdash [\,])$$

$$\lhd^{\mathbb{L}}.f = (\!| [\,]^{\bullet} \triangledown +\!\!+ \circ (f \times id) |\!)_{\mathsf{in}_{\mathsf{F}.A}}$$

list monad transformer diagram



code (liftless)

```
module ListMonadTransformer where
import Control.Monad

newtype  Lmt h a = L {unL:: h (Kmt h a)}
data     Kmt h a = Kempty | Kcons (a,h (Kmt h a))
lempty :: (Monad h) => h (Kmt h a)
lempty = return Kempty
lcons  :: (Monad h) => (a,h (Kmt h a)) -> h (Kmt h a)
lcons  = return . Kcons

moonl :: (Monad h) => h b -> ((a, h b) -> h b) -> h (Kmt h a) -> h b
moonl y op = (>>= (\x -> case x of
                           Kempty      -> y
                           Kcons (a,p) -> op (a, moonl y op p)))
catt :: (Monad h) => (Lmt h a,Lmt h a) -> Lmt h a
catt (ll,lr) = L (moonl (unL lr) lcons (unL ll))

lhdl :: (Monad h) => (a -> Lmt h b) -> Lmt h a -> Lmt h b
lhdl f l = L (moonl lempty (\(x,q) -> unL (catt (f x,L q))) (unL l))

retl :: (Monad h) => a -> Lmt h a
retl x = L (lcons (x,lempty))
```

proof obligations

In the draft paper we took care of

η', $\lhd'$ satisfy the three ES-laws

lift is a monad morphism

$(\mathbb{LT}.A, \oplus, \ll\gg)$ is a monoid

finally (too recent for the draft

$$\sigma \in \mathsf{H} \xrightarrow{\bullet} \mathsf{H}' \;\Rightarrow\; (\![\mathsf{in}_{\mathsf{H}'\,\mathsf{F}} \circ \sigma]\!)_{\mathsf{HF}} \in \mathbb{LT}_{\mathsf{H}} \xrightarrow{\bullet} \mathbb{LT}_{\mathsf{H}'}$$

which leads to functoriality of $\mathsf{H} \mapsto \mathbb{LT}_{\mathsf{H}}$