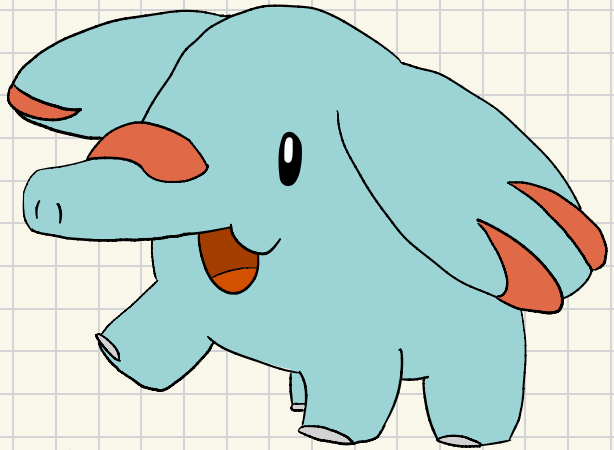
The Stone Spectrum of a Monad

Richard Garner
Alyssa Renata (me!)
Nicolas Wu



29 March 2026 PSSL 112 Nottingham

Part 1: Motivation

Monads encode Effectful Operations

Monads encode Effectful Operations

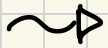
code.txt

```
x = 15;
y = x+1;
return y==17;
```

Monads encode Effectful Operations

code.txt

```
x = 15;
y = x+1;
return y==17;
```



algebraic operations

$put_{x,15}($

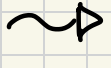
)

unary operation

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations

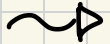
$put_{x,15}($
  $get_x(\lambda m.$
)
)

N-ary operation

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations

$put_{x,15}($
  $get_x(\lambda m.$
    $put_{y,m+1}($
)
)
)

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```

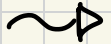
algebraic operations

$$put_{x,15}(\ get_x(\lambda m.\ put_{y,m+1}(\ get_y(\lambda n.\ \ )\ )\ )\ )$$

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations

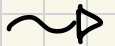
$$put_{x,15}(\ get_x(\lambda m.\ put_{y,m+1}(\ get_y(\lambda n.\ \begin{cases} \text{true} & n = 17 \\ \text{false} & n \neq 17 \end{cases})\ )\ )\ )$$

"true" and "false" are variables in the algebraic sense

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



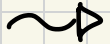
algebraic operations

$$
\begin{aligned}
&\mathrm{put}_{x,15}(\\
&\quad \mathrm{get}_x(\lambda m.\\
&\qquad \mathrm{put}_{y,m+1}(\\
&\qquad\quad \mathrm{get}_y\left(\lambda n. \begin{cases} \text{true} & n = 17 \\ \text{false} & n \neq 17 \end{cases}\right)\\
&\qquad)\\
&\quad)\\
&)
\end{aligned}
$$

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



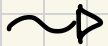
algebraic operations

$$put_{x,15}\Big($$
$$\quad get_x(\lambda m.\ put_{y,15+1}\Big($$
$$\qquad get_y\Big(\lambda n.\ \begin{cases}\text{true} & n = 17\\ \text{false} & n \neq 17\end{cases}\Big)$$
$$\quad \Big)$$
$$\Big)$$

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations

$$put_{x,15}\Big($$

$$put_{y,16}\Big($$

$$get_y\Big(\lambda n.\ \begin{cases}\text{true} & n = 17\\ \text{false} & n \neq 17\end{cases}\Big)$$

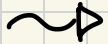
$$\Big)$$

$$\Big)$$

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations

$put_{x,15}($

$put_{y,16}($

$get_y(n.$ {true n = 17 {false n ≠ 17 } 16 }

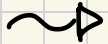
)

)

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations

$put_{x,15}($

$put_{y,16}($

$get_y(\lambda a.$ {true if $a = 17$ / false if $a \neq 17$})

16

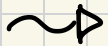
)

)

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations + equations

$put_{x,15}($

$put_{y,16}($

false

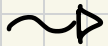
)

)

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations + equations =: $\mathbb{T}$ alg. theory

$put_{x,15}($

$put_{y,16}($

false

)

)

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```

algebraic operations + equations =: $\mathbb{T}$ alg. theory

$put_{x,15}($

$put_{y,16}($

false

)

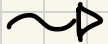
)

- But of course, algebraic theories correspond to monads T

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations + equations =: $\mathbb{T}$ alg. theory

$put_{x,15}($

$put_{y,16}($

false

)

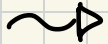
)

- But of course, algebraic theories correspond to monads T
- If Y = output values, TY = computations returning values in Y

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations + equations =: T alg. theory

$put_{x,15}($

$put_{y,16}($

)

)

false

} computation

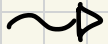
$T\{true, false\}$

- But of course, algebraic theories correspond to monads T
- If Y = output values, TY = computations returning values in Y

Monads encode Effectful Operations

code.txt

```
X = 15;
y = x+1;
return y==17;
```



algebraic operations + equations =: T alg. theory

$put_{x,15}($

$put_{y,16}($

)

)

false

} computation $T\{true, false\}$

- But of course, algebraic theories correspond to monads T
- If Y = output values, TY = computations returning values in Y
- Kleisli map $X \to TY$ = program $X \to Y$

What is being effected?

Let $\mathcal{E}$ be category with <u>copowers</u> $\coprod_X E$ for all $X \in \mathrm{Set}$.

What is being effected?

Let $\mathcal{E}$ be category with <u>copowers</u> $\coprod_X E$ for all $X \in \mathrm{Set}$.

↑ Set, Top, Loc, Gr. topoi

What is being effected?

Let $\mathcal{E}$ be category with copowers $\coprod_X E$ for all $X \in \mathrm{Set}$.

↑ Set, Top, Loc, Gr. topoi

Defn A T-Comodel in $\mathcal{E}$ is a copower-preserving functor

[Power, Shkaravska '04] $W : Kl(T) \to \mathcal{E}$

What is being effected?

Let $\mathcal{E}$ be category with copowers $\coprod_X E$ for all $X \in \mathbf{Set}$.

↖ Set, Top, Loc, Gr. topoi

Defn A T-Comodel in $\mathcal{E}$ is a copower-preserving functor

[Power, Shkaravska '04] $W : Kl(T) \to \mathcal{E}$ i.e. $WX \cong \coprod_X W1$

What is being effected?

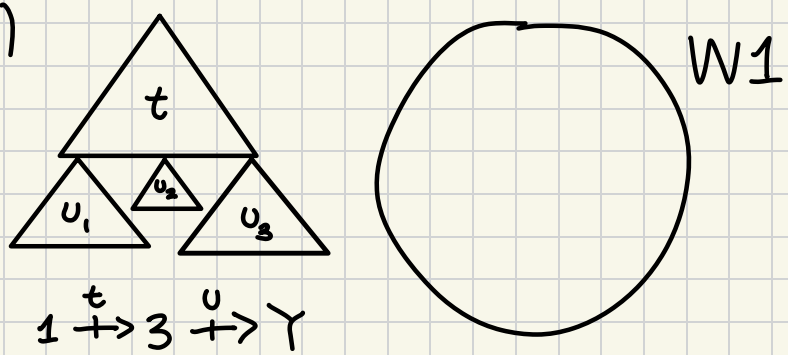
Let $\mathcal{E}$ be category with copowers $\coprod_X E$ for all $X \in \mathrm{Set}$.

↖ Set, Top, Loc, Gr. topoi

Defn A T-Comodel in $\mathcal{E}$ is a copower-preserving functor

[Power, Shkaravska '04] $W : Kl(T) \to \mathcal{E}$ i.e. $WX \cong \coprod_X W1$

Example ($\mathcal{E} = \mathrm{Set}$)



What is being effected?

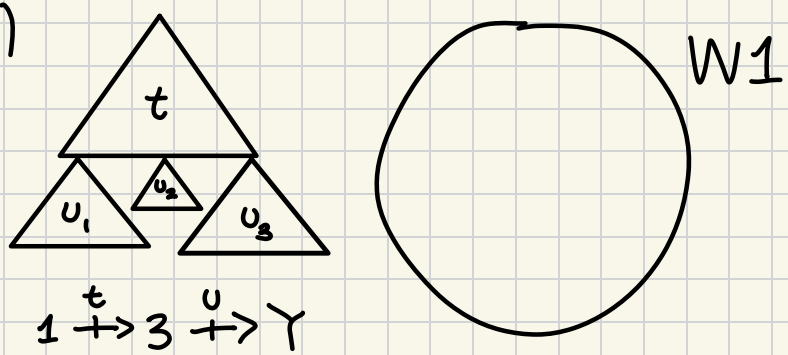
Let $\mathcal{E}$ be category with copowers $\coprod_X E$ for all $X \in \mathrm{Set}$.

↖ Set, Top, Loc, Gr. topoi

Defn A T-Comodel in $\mathcal{E}$ is a copower-preserving functor

[Power, Shkaravska '04] $W : Kl(T) \to \mathcal{E}$ i.e. $WX \cong \coprod_X W1$

Example ($\mathcal{E} = \mathrm{Set}$)



Comodels are Transition Systems!

Classification of Comodels

Classification of Comodels

[Garner '21]

$$\mathrm{Comod}_T(\mathrm{Set}) := [\mathrm{Kl}(T), \mathrm{Set}]_{\perp\!\!\!\perp}$$

$$\simeq [\mathbb{B}T, \mathrm{Set}]$$ Presheaf Cat!

Classification of Comodels

[Garner '21]

$\mathsf{Comod}_T(\mathsf{Set}) := [\mathcal{K}\ell(T), \mathsf{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{BT}, \mathsf{Set}]$ Presheaf Cat!

$\mathbb{BT}$ (Behaviour Category) is a transition system

Classification of Comodels

[Garner '21]

$\mathrm{Comod}_T(\mathrm{Set}) := [\mathbb{K}\ell(T), \mathrm{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{B}T, \mathrm{Set}]$ Presheaf Cat!

$\mathbb{B}T$ (Behaviour Category) is a transition system

$\mathbb{B}_\circ T$ is terminal comodel (states are observable behaviours ala Coalgebra)

Classification of Comodels

[Garner '21]

$\mathrm{Comod}_T(\mathbf{Set}) := [\mathcal{K}\ell(T), \mathbf{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{B}T, \mathbf{Set}]$ Presheaf Cat!

$\mathbb{B}T$ (Behaviour Category) is a transition system

$\mathbb{B}_0T$ is terminal comodel (states are observable behaviours ala Coalgebra)

$\mathbb{B}_1T$ is $\sum_{\beta \in \mathbb{B}_0T} T1/\sim_\beta$ (transitions are computations, quotienting ensures composition)

Classification of Comodels

[Garner '21]

$\mathrm{Comod}_T(\mathrm{Set}) := [\mathcal{K}\ell(T), \mathrm{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{B}T, \mathrm{Set}]$ Presheaf Cat!

$\mathbb{B}T$ (Behaviour Category) is a transition system

$\mathbb{B}_0T$ is terminal comodel (states are observable behaviours ala Coalgebra)

$\mathbb{B}_1T$ is $\sum_{\beta \in \mathbb{B}_0T} T1/\sim_\beta$ (transitions are computations, quotienting ensures composition)

not what you think, unless...?

Classification of Comodels

[Garner '21]

$\mathsf{Comod}_T(\mathsf{Set}) := [\mathcal{K}\ell(T), \mathsf{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{B}T, \mathsf{Set}]$ Presheaf Cat!

$\mathbb{B}T$ (Behaviour Category) is a transition system

$\mathbb{B}_0T$ is terminal comodel (states are observable behaviours ala Coalgebra)

$\mathbb{B}_1T$ is $\sum_{\beta \in \mathbb{B}_0T} T1/\sim_\beta$ (transitions are computations, quotienting ensures composition)

not what you think, unless...?

Observation B Bool. Alg. $\rightsquigarrow$ T_B Monad of B-partitions

B-tests / if $b \in B$ then... else...

Classification of Comodels

[Garner '21]

$\mathsf{Comod}_T(\mathsf{Set}) := [\mathcal{K}\ell(T), \mathsf{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{B}T, \mathsf{Set}]$ Presheaf Cat!

$\mathbb{B}T$ (Behaviour Category) is a transition system

$\mathbb{B}_0T$ is terminal comodel (states are observable behaviours ala Coalgebra)

$\mathbb{B}_1T$ is $\sum_{\beta \in \mathbb{B}_0T} T1/\sim_\beta$ (transitions are computations, quotienting ensures composition)

not what you think, unless...?

Observation B Bool. Alg. $\rightsquigarrow$ T_B Monad of B-partitions

$\mathbb{B}_0T_B$ is the set of ultrafilters

B-tests / if $b \in B$ then... else...

Classification of Comodels

[Garner '21]

$\mathsf{Comod}_T(\mathsf{Set}) := [\mathcal{K}\ell(T), \mathsf{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{B}T, \mathsf{Set}]$ Presheaf Cat!

$\mathbb{B}T$ (Behaviour Category) is a transition system

$\mathbb{B}_0T$ is terminal comodel (states are observable behaviours ala Coalgebra)

$\mathbb{B}_1T$ is $\sum_{\beta \in \mathbb{B}_0T} T1/\sim_\beta$ (transitions are computations, quotienting ensures composition)

not what you think, unless...?

Observation B Bool. Alg. $\rightsquigarrow$ T_B Monad of B-partitions

B-tests / if $b \in B$ then... else...

$\mathbb{B}_0T_B$ is the set of ultrafilters

$\mathbb{B}_1T_B$ (identity transitions only)

Classification of Comodels

[Garner '21]

$\mathsf{Comod}_T(\mathsf{Set}) := [\mathcal{K}\ell(T), \mathsf{Set}]_{\perp\!\!\!\perp} \simeq [\mathbb{B}T, \mathsf{Set}]$ Presheaf Cat!

$\mathbb{B}T$ (Behaviour Category) is a transition system

$\mathbb{B}_0T$ is terminal comodel (states are observable behaviours ala Coalgebra)

$\mathbb{B}_1T$ is $\sum_{\beta \in \mathbb{B}_0T} T1/\sim_\beta$ (transitions are computations, quotienting ensures composition)

not what you think, unless...?

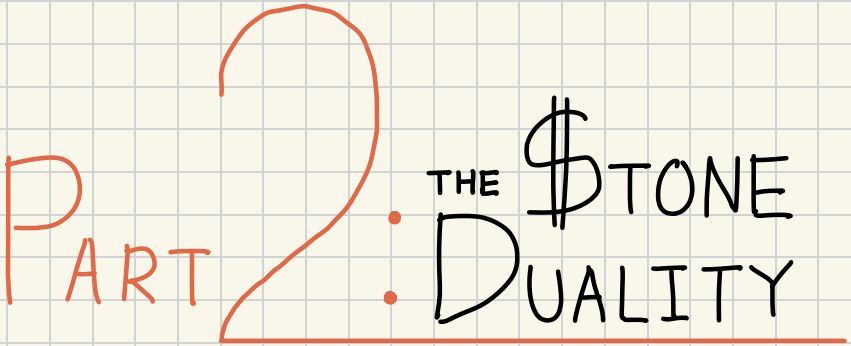
Observation: B Bool. Alg. $\rightsquigarrow$ T_B Monad of B-partitions

$\mathbb{B}_0T_B$ is the set of ultrafilters

$\mathbb{B}_1T_B$ (identity transitions only)

B-tests / if $b \in B$ then... else...

Q: How to get Topology as well?


PART 2: THE $TONE DUALITY

The Topological Behaviour Category

Q: How to get Topology as well?

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_T(\mathrm{Top})$!

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_{\mathbb{T}}(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0\mathbb{T}$ and $\mathbb{B}_1\mathbb{T}$ ⇝ Internal Category

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_{\mathbb{T}}(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0\mathbb{T}$ and $\mathbb{B}_1\mathbb{T}$ ⇝ Internal Category

Theorem

$$\mathrm{Mnd}_{\omega}(\mathrm{Set}) \simeq \mathrm{TopCat}^{op}$$

with $\mathbb{B}$ and Γ_{ω}

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_{\mathbb{T}}(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0\mathbb{T}$ and $\mathbb{B}_1\mathbb{T}$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_{\omega}(\mathrm{Set})$

$\mathbb{B} \dashv \Gamma_{\omega}$

~~TopCat^{op}~~

$\mathrm{TopRetro}^{op}$

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_{\mathbb{T}}(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0\mathbb{T}$ and $\mathbb{B}_1\mathbb{T}$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_{\omega}(\mathrm{Set})$

$\mathbb{B} \dashv \Gamma_{\omega}$

$\mathrm{Top}\mathrm{Cat}^{\mathrm{op}}$

$\mathrm{TopRetro}^{\mathrm{op}}$

$S \xrightarrow{\varphi} T$

$\mathbb{B}\varphi$

$\mathbb{B}T \rightsquigarrow \mathbb{B}S$

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_T(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0T$ and $\mathbb{B}_1T$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_\omega(\mathrm{Set})$

$\mathbb{B} \dashv \Gamma_\omega$

TopCat^{op}

$\mathrm{TopRetro}^{op}$

$S \xrightarrow{\varphi} T$

$\mathbb{B}\varphi$

$\mathbb{B}T \rightsquigarrow \mathbb{B}S$

$\beta \longmapsto \gamma$

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_T(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0T$ and $\mathbb{B}_1T$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_\omega(\mathrm{Set})$

$\mathbb{B} \dashv \Gamma_\omega$

$\mathrm{Top}\mathrm{Cat}^{op}$

$\mathrm{TopRetro}^{op}$

$S \xrightarrow{\varphi} T$

$\mathbb{B}\varphi$

$\mathbb{B}T \rightsquigarrow \mathbb{B}S$

γ'

$\beta \longmapsto \gamma$

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_T(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0T$ and $\mathbb{B}_1T$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_\omega(\mathrm{Set})$

$\mathbb{B} \dashv \Gamma_\omega$

TopCat^{op}

$\mathrm{TopRetro}^{op}$

$S \xrightarrow{\varphi} T$

$\mathbb{B}\varphi$

$\mathbb{B}T \rightsquigarrow \mathbb{B}S$

$\beta' \dashrightarrow \gamma'$

$\beta \longmapsto \gamma$

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_{\mathbb{T}}(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0\mathbb{T}$ and $\mathbb{B}_1\mathbb{T}$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_{\omega}(\mathrm{Set})$

$\mathbb{B} \dashv \Gamma_{\omega}$

TopCat^{op}

$\mathrm{TopRetro}^{op}$

$S \xrightarrow{\varphi} T$ $\quad (\Gamma_{\omega}\mathbb{C})n = \{$ global sections (to source map) $\mathbb{C}_0 \to \coprod_n \mathbb{C}_1 \}$

$\mathbb{B}\varphi$

$\mathbb{B}T \rightsquigarrow \mathbb{B}S$

$B' \dashrightarrow y'$

$B \longmapsto y$

$\mathbb{C}$

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_{\mathbb{T}}(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0\mathbb{T}$ and $\mathbb{B}_1\mathbb{T}$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_{\omega}(\mathrm{Set})$

$\mathbb{B}\left(\dashv\right)\Gamma_{\omega}$

$\mathrm{Top}\mathrm{Cat}^{\mathrm{op}}$

$\mathrm{TopRetro}^{\mathrm{op}}$

$S \xrightarrow{\varphi} T$ $\quad (\Gamma_{\omega}\mathbb{C})n = \{$ global sections (to source map) $\mathbb{C}_0 \to \coprod_n \mathbb{C}_1 \}$

$\mathbb{B}\varphi$

$\mathbb{B}T \rightsquigarrow \mathbb{B}S$

$B' \dashrightarrow y'$

$B \longmapsto y$

$\mathbb{C}$

Remark

Indeed,

$\mathbb{B}_0\mathbb{T}_B = \mathbb{B}_1\mathbb{T}_B \cong \mathrm{Spec}(B)$

The Topological Behaviour Category

Q: How to get Topology as well?

A: Classify $\mathrm{Comod}_T(\mathrm{Top})$!

suffice to put correct topology on $\mathbb{B}_0T$ and $\mathbb{B}_1T$ ⇝ Internal Category

Theorem

$\mathrm{Mnd}_\omega(\mathrm{Set})$

$\mathbb{B} \dashv \Gamma_\omega$

$\mathrm{Top}\mathrm{Cat}^{op}$

$\mathrm{TopRetro}^{op}$

$S \xrightarrow{\varphi} T$ $\quad (\Gamma_\omega \mathbb{C})n = \{$ global sections (to source map) $\mathbb{C}_0 \to \coprod_n \mathbb{C}_1 \}$

$\mathbb{B}\varphi$

$\mathbb{B}T \rightsquigarrow \mathbb{B}S$

$B' \dashrightarrow y'$

$B \longmapsto y$

$\mathbb{C}$

Remark

Indeed,

$\mathbb{B}_0T_B = \mathbb{B}_1T_B \cong \mathrm{Spec}(B)$

In fact,

$\mathbb{B}_0T$ is always Stone!

(for T finitary)

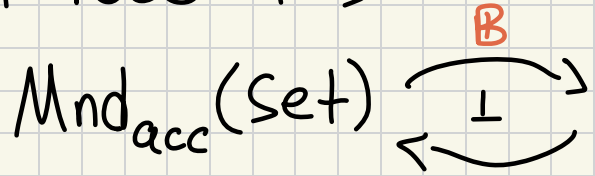
The Localic Behaviour Category

- We can redo this for $\mathrm{Comod}_T(\mathrm{Loc})$, to get

$$\mathrm{Mnd}_{\mathrm{acc}}(\mathrm{Set}) \underset{\mathbb{I}}{\overset{\mathbb{B}}{\rightleftarrows}} \mathrm{Loc}\mathrm{Retro}^{\mathrm{op}}$$

The Localic Behaviour Category

- We can redo this for $\mathrm{Comod}_T(\mathrm{Loc})$, to get



there are infinitary monads T with point-less, non-trivial $\mathbb{B}_0 T$

The Localic Behaviour Category

- We can redo this for $\mathrm{Comod}_T(\mathrm{Loc})$, to get

$$\mathrm{Mnd}_{acc}(\mathrm{Set}) \underset{\mathbb{I}}{\overset{\mathbb{B}}{\rightleftarrows}} \mathrm{LocRetro}^{op}$$

there are infinitary monads T with point-less, non-trivial $\mathbb{B}_0 T$

e.g. (1) T state monad on memory configurations $\mathbb{R} \rightarrowtail \mathbb{N}$

The Localic Behaviour Category

- We can redo this for $\mathrm{Comod}_T(\mathrm{Loc})$, to get

$$\mathrm{Mnd}_{acc}(\mathrm{Set}) \underset{\mathbb{I}}{\overset{\mathbb{B}}{\rightleftarrows}} \mathrm{LocRetro}^{op}$$

there are infinitary monads T with point-less, non-trivial $\mathbb{B}_0 T$

e.g. (1) T state monad on memory configurations $\mathbb{R} \rightarrowtail \mathbb{N}$

(2) atomless CBA $B \rightsquigarrow T_B'$ (T_B + infinitary tests)

The Localic Behaviour Category

- We can redo this for $\mathrm{Comod}_T(\mathrm{Loc})$, to get

$$\mathrm{Mnd}_{acc}(\mathrm{Set}) \underset{\mathbb{I}}{\overset{\mathbb{B}}{\rightleftarrows}} \mathrm{LocRetro}^{op}$$

there are infinitary monads T with point-less, non-trivial $\mathbb{B}_0 T$

e.g. (1) T state monad on memory configurations $\mathbb{R} \rightarrowtail \mathbb{N}$

(2) atomless CBA $B \rightsquigarrow T_B'$ (T_B + infinitary tests)

- This clarified $\mathbb{B}_1 T \xrightarrow{\mathrm{src}} \mathbb{B}_0 T$ as a sheaf generated by T1,

The Localic Behaviour Category

- We can redo this for $\mathrm{Comod}_T(\mathrm{Loc})$, to get

$$\mathrm{Mnd}_{acc}(\mathrm{Set}) \underset{\mathbb{1}}{\overset{\mathbb{B}}{\rightleftarrows}} \mathrm{LocRetro}^{op}$$

there are infinitary monads T with point-less, non-trivial $\mathbb{B}_0 T$

e.g. (1) T state monad on memory configurations $\mathbb{R} \rightarrowtail \mathbb{N}$

(2) atomless CBA $B \rightsquigarrow T'_B$ (T_B + infinitary tests)

- This clarified $\mathbb{B}_1 T \xrightarrow{\mathrm{src}} \mathbb{B}_0 T$ as a sheaf generated by $T1$, which let us prove:

theorem $\mathrm{CCMnd}_{acc}(\mathrm{Set}) \simeq \mathrm{AmpleLocRetro}^{op}$

- Strongly zero-dim AKA "superStone"
- local homeomorphism src

[uses a lot of technology from Garner '24 + '25]

If interested, Check out our preprint!

"Stone Duality for Monads"

ArXiv:2603.25710

If interested, Check out our preprint!

"Stone Duality for Monads"

ArXiv:2603.25710

(If not, check it out anyway? 🥺)

PART 3: TOWARDS ZARISKI & GROTHENDIECK

Why I'm *really* doing this

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!
- Apparently, people care about this because $\mathbb{F}_1$-algebraic geometry.

see e.g. [Durov '07]

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!
- Apparently, people care about this because $\mathbb{F}_1$-algebraic geometry.

see e.g. [Durov '07]

For me:

- If monad encodes program syntax, then monadic scheme encodes Syntax varying continuously over semantics (State space of the environment).

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!
- Apparently, people care about this because $\mathbb{F}_1$-algebraic geometry.

see e.g. [Durov '07]

For me:

- If monad encodes program syntax, then monadic scheme encodes Syntax varying continuously over semantics (State space of the environment).

example

loop_kitty.py

```
with open("cute_kitty.gif") as kitty:
    contents = kitty.read()
    kitty.write(loop(contents,50))
```

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!
- Apparently, people care about this because $\mathbb{F}_1$-algebraic geometry.

see e.g. [Durov '07]

For me:

- If monad encodes program syntax, then monadic scheme encodes Syntax varying continuously over semantics (State space of the environment).

example

loop_kitty.py

```
with open("cute_kitty.gif") as kitty:
    contents = kitty.read()
    kitty.write(loop(contents,50))
#kitty not in scope anymore
```

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!
- Apparently, people care about this because $\mathbb{F}_1$-algebraic geometry.

see e.g. [Durov '07]

For me:

- If monad encodes program syntax, then monadic scheme encodes Syntax varying continuously over semantics (State space of the environment).

example

loop_kitty.py

```
with open("cute_kitty.gif") as kitty:
    contents = kitty.read()
    kitty.write(loop(contents,50))
#kitty not in scope anymore
```

more drastic examples include multi-language programs, e.g. inline assembly in C.

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!
- Apparently, people care about this because $\mathbb{F}_1$-algebraic geometry.

see e.g. [Durov '07]

For me:

- If monad encodes program syntax, then monadic scheme encodes Syntax varying continuously over semantics (State space of the environment).

example ↳ cohomology facilitates local-to-global program reasoning?!

loop_kitty.py

```
with open("cute_kitty.gif") as kitty:
    contents = kitty.read()
    kitty.write(loop(contents,50))
#kitty not in scope anymore
```

more drastic examples include multi-language programs, e.g. inline assembly in C.

Why I'm *really* doing this

- To get a Scheme theory replacing rings by Monads!
- Apparently, people care about this because $\mathbb{F}_1$-algebraic geometry.

see e.g. [Durov '07]

For me:

- If monad encodes program syntax, then monadic scheme encodes syntax varying continuously over semantics (State space of the environment).

example ↳ cohomology facilitates local-to-global program reasoning?!

loop_kitty.py

```
with open("cute_kitty.gif") as kitty:
    contents = kitty.read()
    kitty.write(loop(contents,50))
#kitty not in scope anymore
```

Leave this for future work. For now, let's compare

ZSpec(R) vs BT

The free local ring lives over $\mathbb{Z}\mathrm{Spec}(R)$

The free local ring lives over $\mathbb{Z}\mathrm{Spec}(R)$

The affine scheme $\mathcal{O}R \in \mathrm{Sh}(\mathbb{Z}\mathrm{Spec}(R))$ admits a universal property [Cole '16 but technically '81 ??]

The free local ring lives over $\mathbb{Z}\mathrm{Spec}(R)$

The affine scheme $\mathcal{O}R \in \mathrm{Sh}(\mathbb{Z}\mathrm{spec}(R))$ admits a universal property [Cole '16 but technically '81 ??]

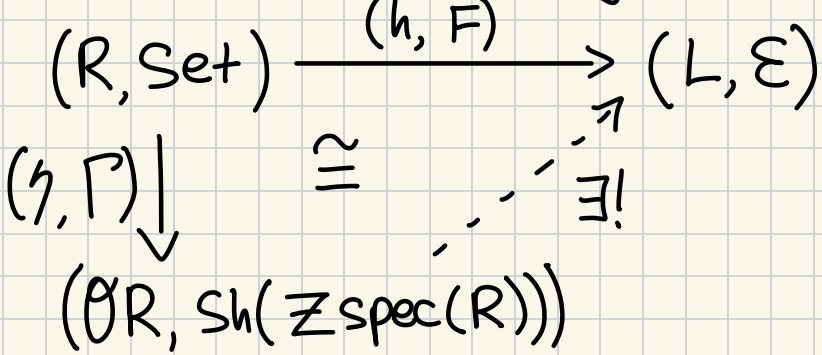
$$\begin{array}{ccc} (R, \mathrm{Set}) & \xrightarrow{(h,\, F)} & (L, \mathcal{E}) \\ (\mathcal{O}, \Gamma)\big\downarrow & & \\ (\mathcal{O}R, \mathrm{Sh}(\mathbb{Z}\mathrm{spec}(R))) & & \end{array}$$

i.e. $F : \mathcal{E} \leftarrow \mathrm{Set}$, $h : F^{*}R \to L$

in Ringed Gr Top

The free local ring lives over $\mathbb{Z}\mathrm{Spec}(R)$

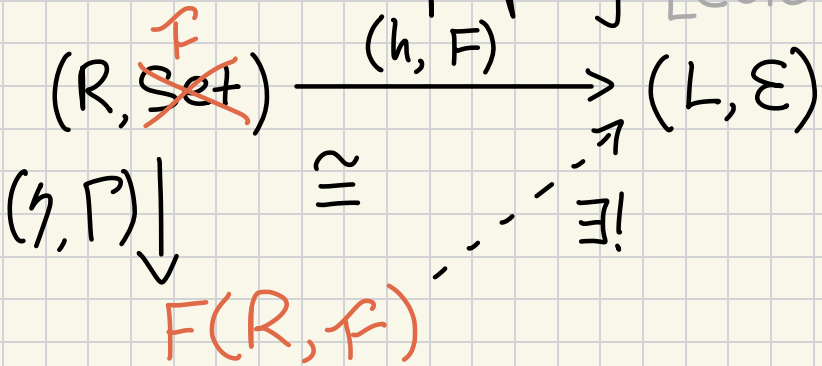
The affine scheme $\mathcal{O}R \in \mathrm{Sh}(\mathbb{Z}\mathrm{spec}(R))$ admits a universal property [Cole '16 but technically '81 ??]



i.e. $F : \mathcal{E} \leftarrow \mathrm{Set}$, $h : F^{*}R \to L$ in Ringed Gr Top

The free local ring lives over $\mathbb{Z}$Spec(R)

The affine scheme $\mathcal{O}R \in \mathrm{Sh}(\mathbb{Z}\mathrm{spec}(R))$ admits a universal property [Cole '16 but technically '81 ??]



i.e. $F : \mathcal{E} \leftarrow \mathrm{Set}$, $h : F^{*}R \to L$ in Ringed Gr Top

Theorem

$$\mathrm{Ringed\,Gr\,Top} \underset{\longleftarrow}{\overset{F}{\longrightarrow}} \mathrm{Loc\,Ringed\,Gr\,Top} \quad (\dashv)$$

The free comodel lives over B.T

The free comodel lives over $\mathbb{B}_\bullet T$

Remark

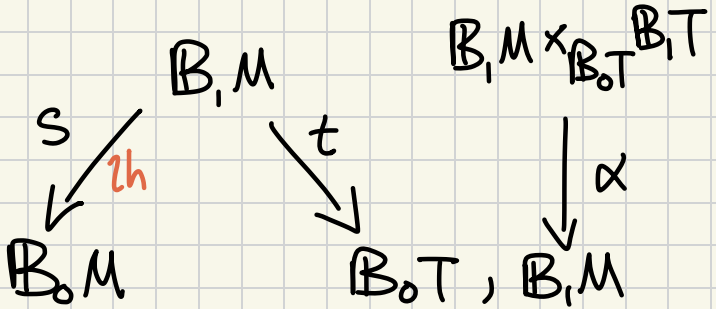
Comodels are special cases of T-actions $M \circ T \to M$,
Since $\mathrm{Act}_T(\mathcal{E}) \cong [Kl(T), \mathcal{E}] \hookleftarrow [Kl(T), \mathcal{E}]_{LL} =: \mathrm{Comod}_T(\mathcal{E})$.

The free comodel lives over $\mathbb{B}_\bullet T$

Remark

Comodels are special cases of T-actions $M \circ T \to M$,
Since $\mathrm{Act}_T(\mathcal{E}) \cong [K\wr(T), \mathcal{E}] \hookleftarrow [K\wr(T), \mathcal{E}]_{\perp\!\!\!\perp} =: \mathrm{Comod}_T(\mathcal{E})$.

The construction $\mathbb{B}$ can be generalized to $\mathbb{B}M$, a fibered $\mathbb{B}T$-space:



The free comodel lives over $\mathbb{B}_0 T$

Remark

Comodels are special cases of T-actions $M \circ T \to M$,
Since $\mathrm{Act}_T(\mathcal{E}) \cong [K\ell(T), \mathcal{E}] \hookleftarrow [K\ell(T), \mathcal{E}]_{\amalg} =: \mathrm{Comod}_T(\mathcal{E})$.

The construction $\mathbb{B}$ can be generalized to $\mathbb{B}M$, a fibered $\mathbb{B}T$-space:

$$\text{fibered } \mathbb{B}T\text{-space} := \left\{ \left(\begin{array}{ccccc} & & E & & E \times_{\mathbb{B}_0 T} \mathbb{B}_1 T \\ & S \swarrow & & t \searrow & \downarrow \alpha \\ B & & & & \mathbb{B}_0 T, \quad E \end{array} \right) \right\}$$

The free comodel lives over $\mathbb{B}_0 T$

Remark

Comodels are special cases of T-actions $M \circ T \to M$,
Since $\mathrm{Act}_T(\mathcal{E}) \cong [Kl(T), \mathcal{E}] \hookleftarrow [Kl(T), \mathcal{E}]_{!!} =: \mathrm{Comod}_T(\mathcal{E})$.

The construction $\mathbb{B}$ can be generalized to $\mathbb{B}M$, a fibered $\mathbb{B}T$-space:

fibered $\mathbb{B}T$-space $:= \simeq T$-Comodelled LocTop $= \left\{ \left(\begin{array}{ccccc} & & E & & E \times_{\mathbb{B}_0 T} \mathbb{B}_1 T \\ & s \swarrow & & t \searrow & \downarrow \alpha \\ \mathbb{B} & & & & \mathbb{B}_0 T, \; E \end{array} \right) \right\}$

The free comodel lives over $\mathbb{B}_\circ T$

Remark

Comodels are special cases of T-actions $M \circ T \to M$,
Since $\mathrm{Act}_T(\mathcal{E}) \cong [Kl(T), \mathcal{E}] \hookleftarrow [Kl(T), \mathcal{E}]_{!!} =: \mathrm{Comod}_T(\mathcal{E})$.

The construction $\mathbb{B}$ can be generalized to $\mathbb{B}M$, a fibered $\mathbb{B}T$-space:

fibered $\mathbb{B}T$-space $:=$

$$\left\{ \left(\begin{array}{ccccc} E & & & & E \times_{B_\circ T} B_! T \\ & \searrow^{t} & & & \downarrow \alpha \\ {\scriptstyle s} \swarrow & & {\scriptstyle lh} & & \\ B & & & B_\circ T, & E \end{array} \right) \right\}$$

$\simeq$ T-Comodelled LocTop

$\subseteq$ T-Comodelled GrTop

$$\| \quad \{(\mathcal{E}, W : Kl(T) \to \mathcal{E})\}$$

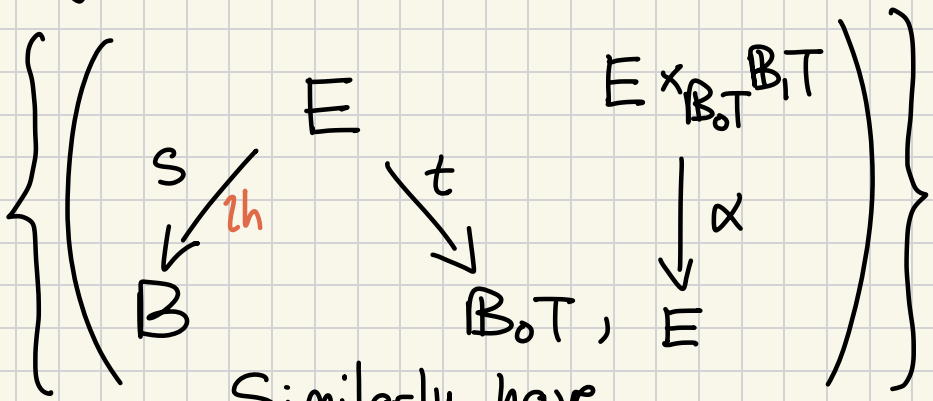
The free comodel lives over $\mathbb{B}_0 T$

Remark

Comodels are special cases of T-actions $M \circ T \to M$,
Since $\mathrm{Act}_T(\mathcal{E}) \cong [K\ell(T), \mathcal{E}] \hookleftarrow [K\ell(T), \mathcal{E}]_{\amalg} =: \mathrm{Comod}_T(\mathcal{E})$.

The construction $\mathbb{B}$ can be generalized to $\mathbb{B}M$, a fibered $\mathbb{B}T$-space:

fibered $\mathbb{B}T$-space $:=$



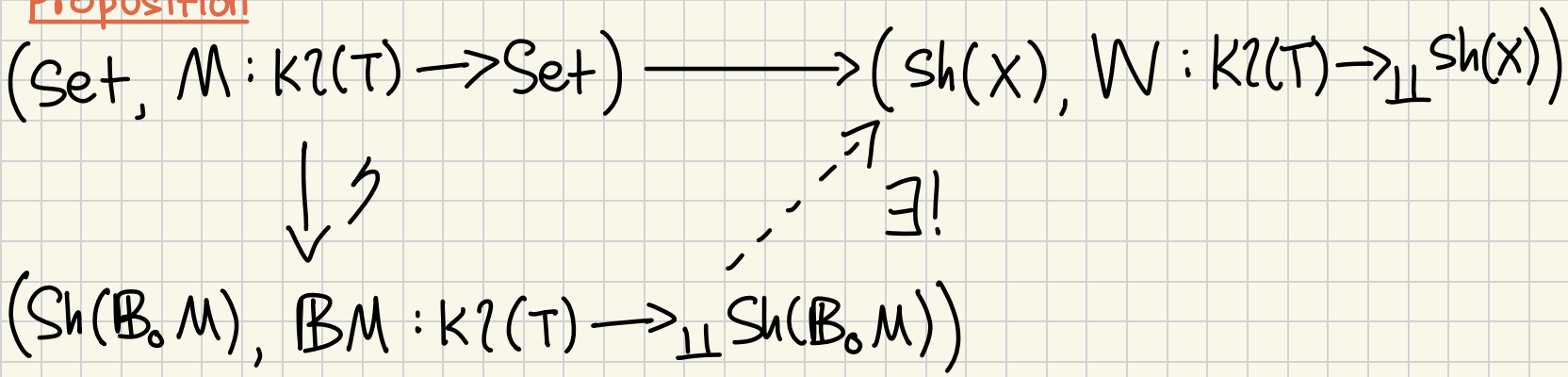
$\simeq$ T-Comodelled LocTop
$\subseteq$ T-Comodelled GrTop
$=$ $\{(\mathcal{E}, W : K\ell(T) \to \mathcal{E})\}$ $\hookrightarrow$ T-Actioned GrTop

Similarly, have

The free comodel lives over $B_0 M$

The T-comodel BM admits a universal property

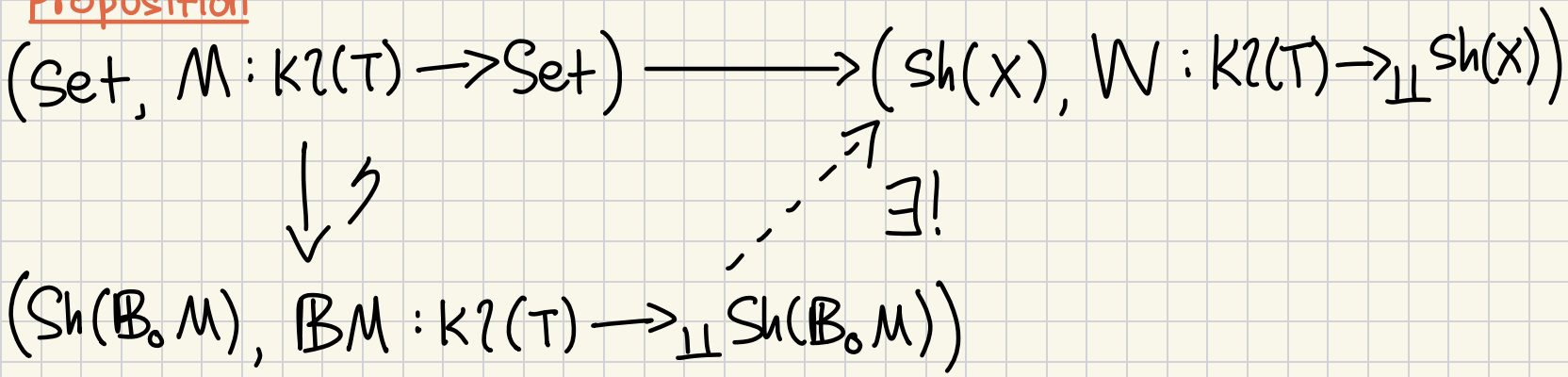
Proposition



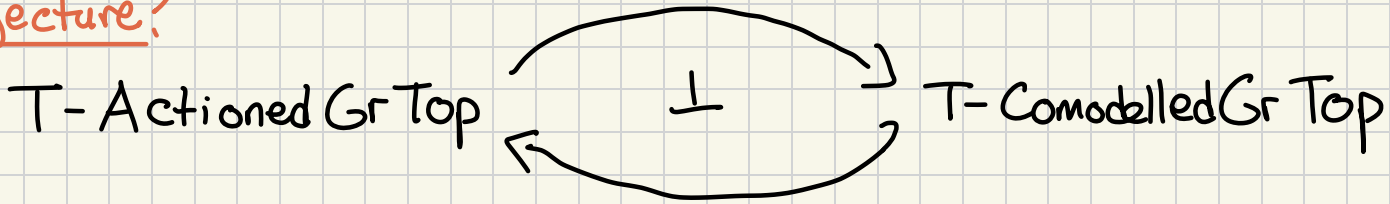
The free comodel lives over $\mathbb{B}_0 T$ M

The T-comodel $\mathbb{B}M$ admits a universal property

Proposition



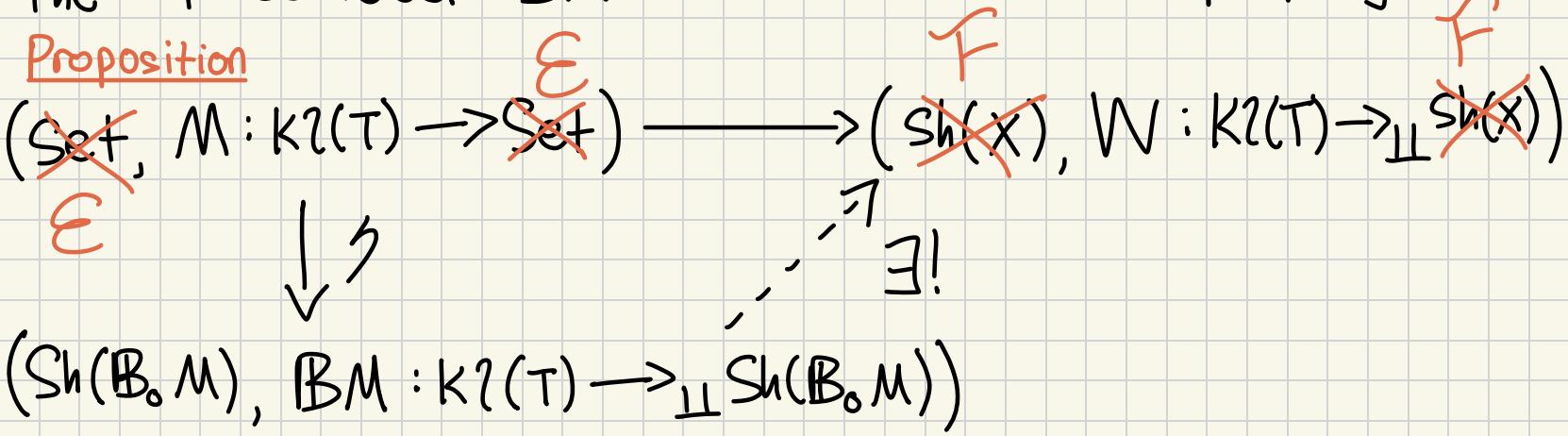
Conjecture?



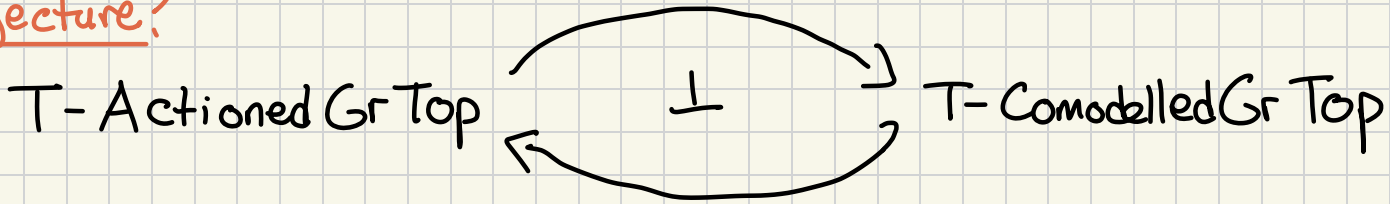
The free comodel lives over $\mathbb{B}_0 T$ M

The T-comodel $\mathbb{B}M$ admits a universal property

Proposition



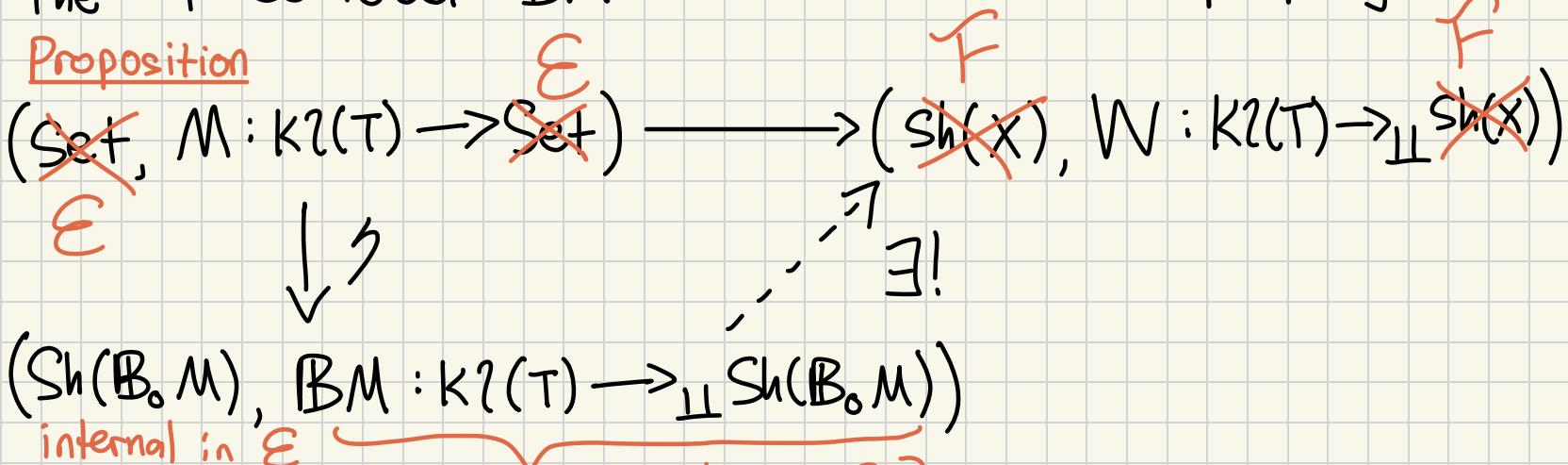
Conjecture?



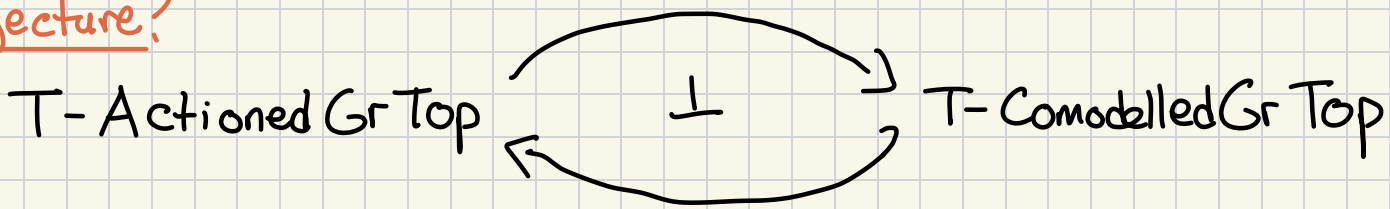
The free comodel lives over $B_0 T$ M

The T-comodel BM admits a universal property

Proposition



Conjecture?



fin.