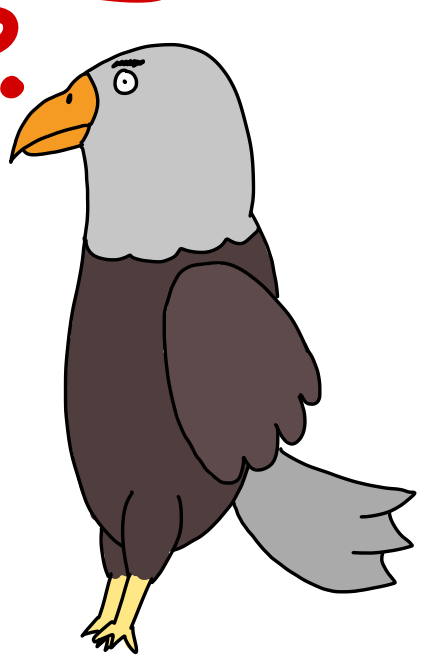
Stone Duality for Monads

Richard Garner[1]
Alyssa Renata[2]
Nicolas Wu[2]

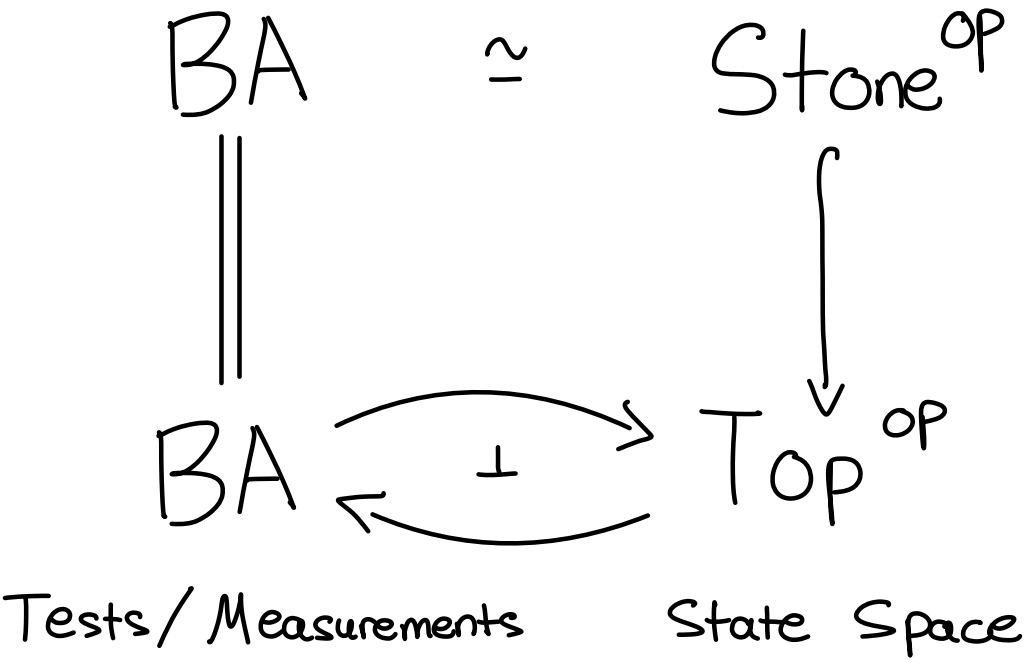
[1] Macquarie University, Sydney, Australia
[2] Imperial College London, United Kingdom



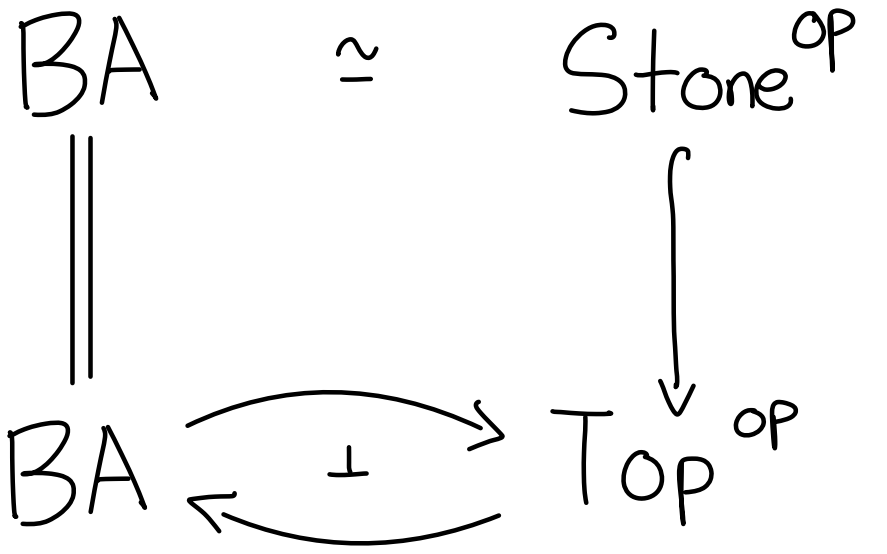
30 July 2026

TACL@Kraków

Stone Duality (Classically)



Stone Duality (Classically)



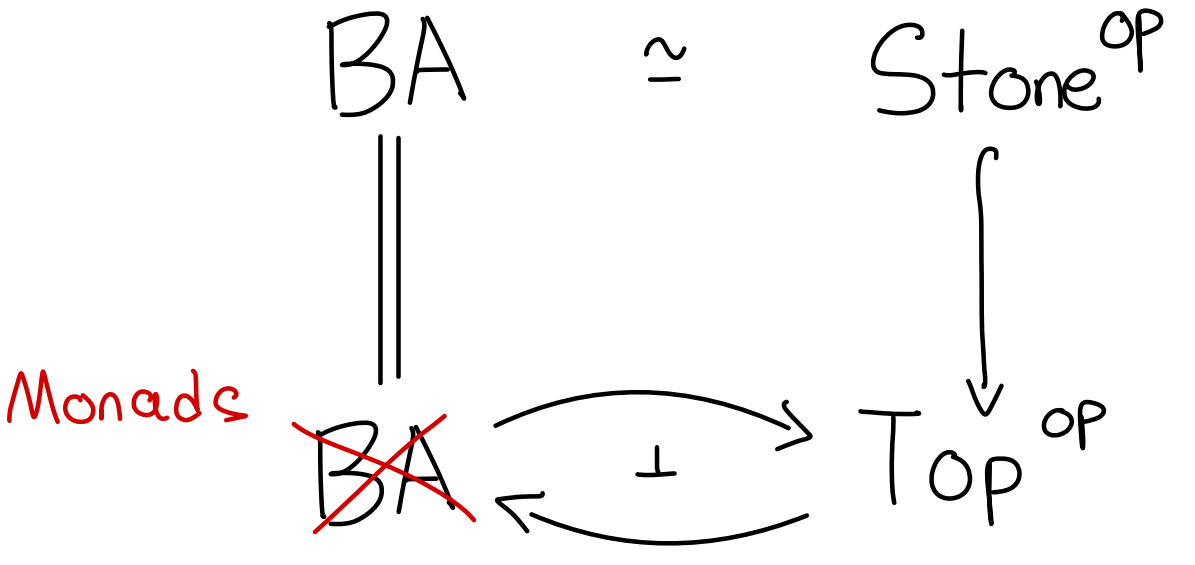
Tests / Measurements

State Space

Good science...

What about bad science?

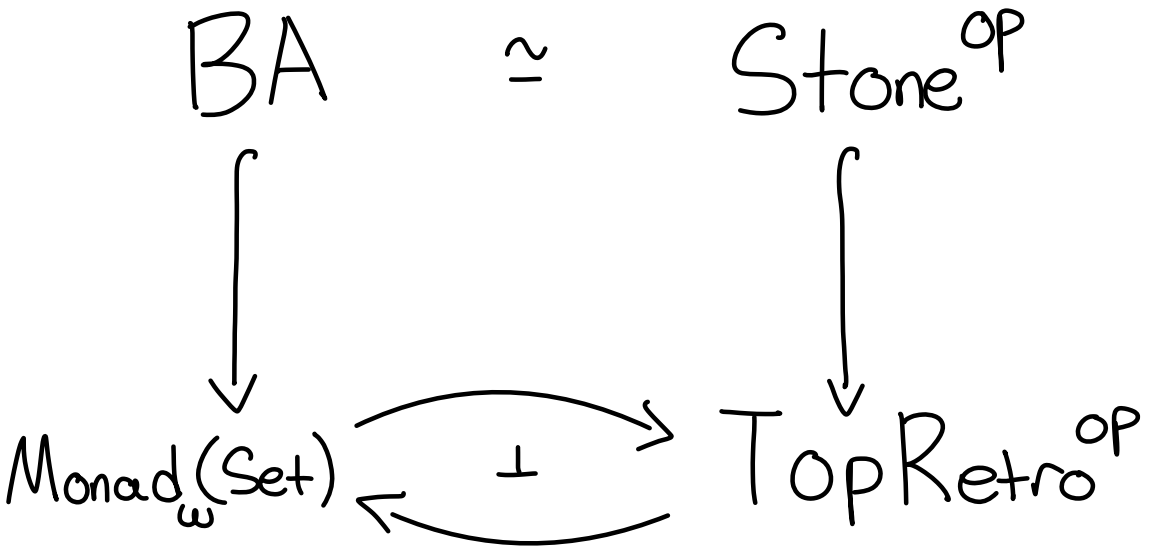
Stone Duality (Classically)



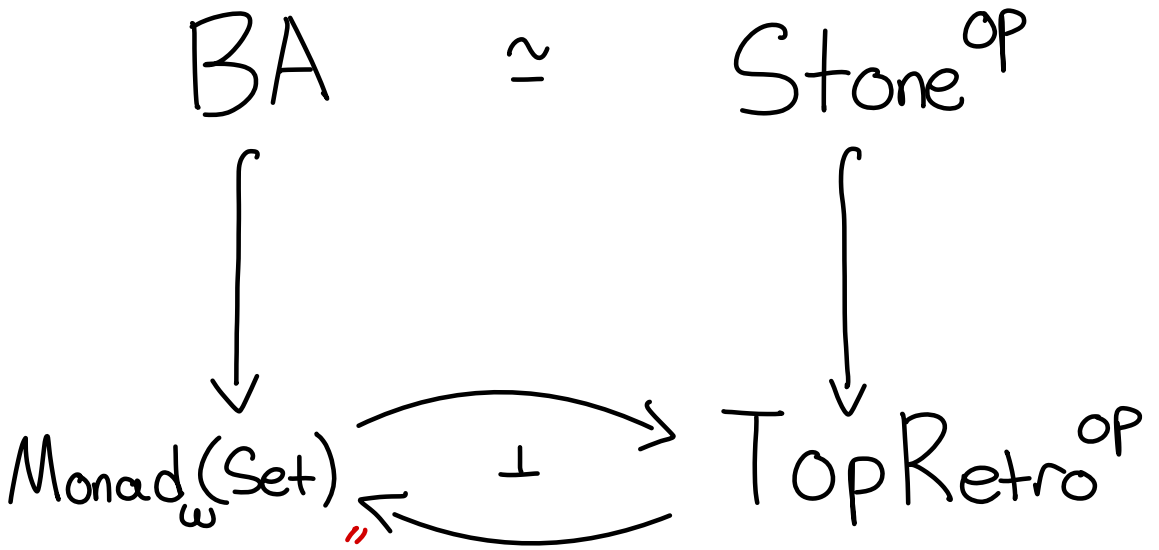
Good science...

What about bad science?

This Talk in One Slide



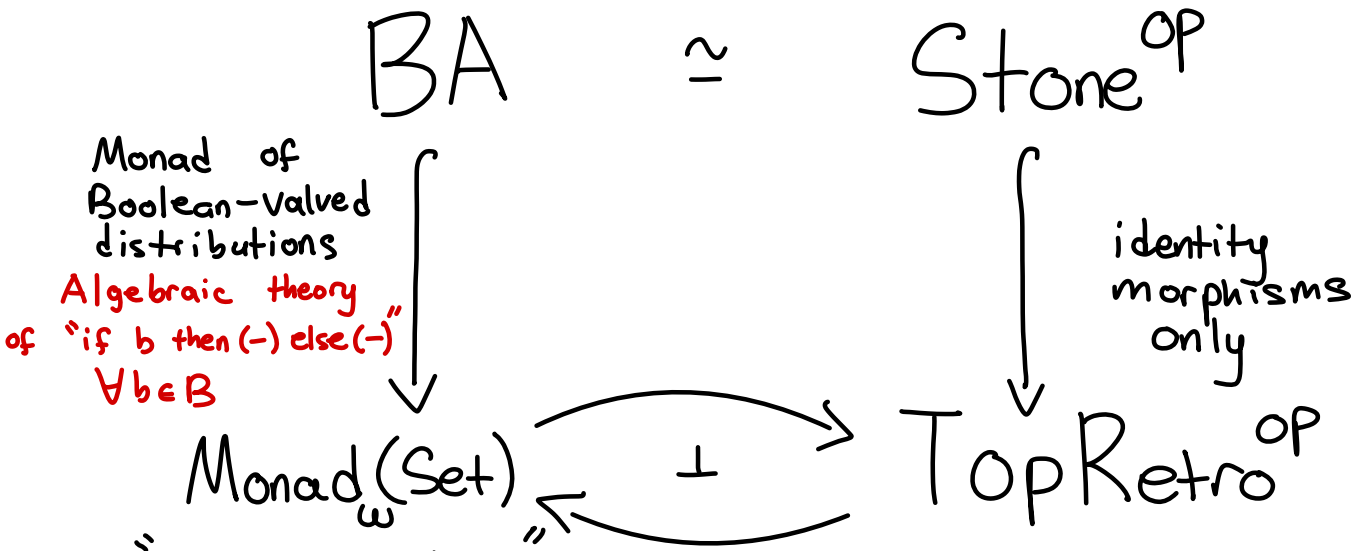
This Talk in One Slide



- "Computations"
- "Experiments"

- objects = categories internal in Top
  - ↳ objects = "state space"
  - ↳ morphisms = "transitions"
- morphisms = retrofunctors
  "functional back-simulation"

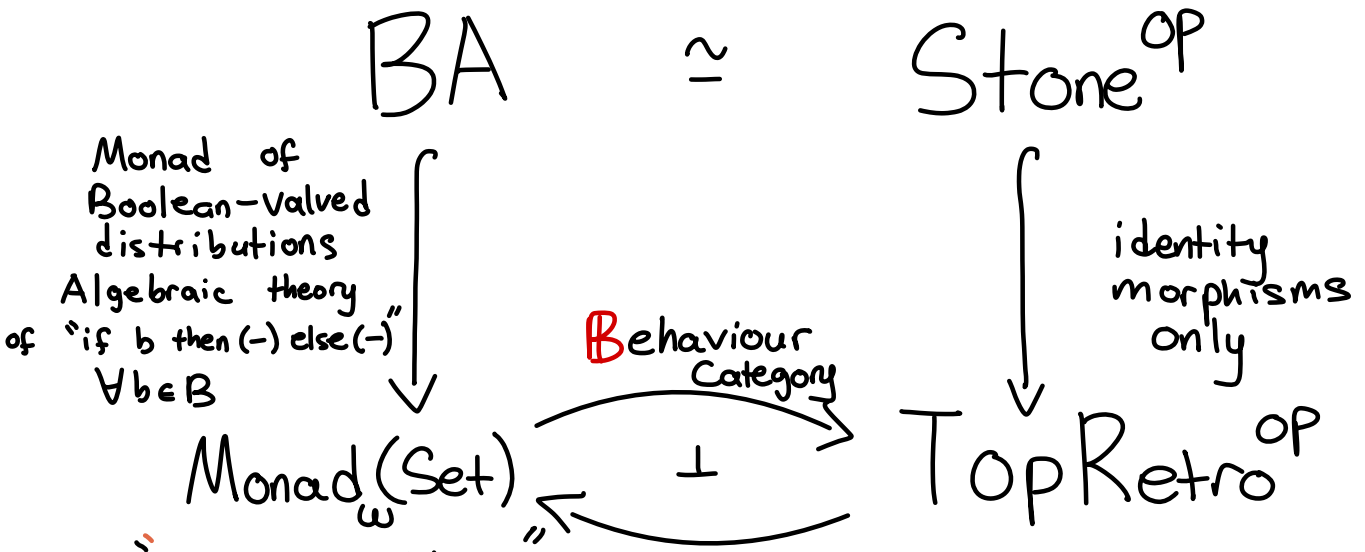
This Talk in One Slide



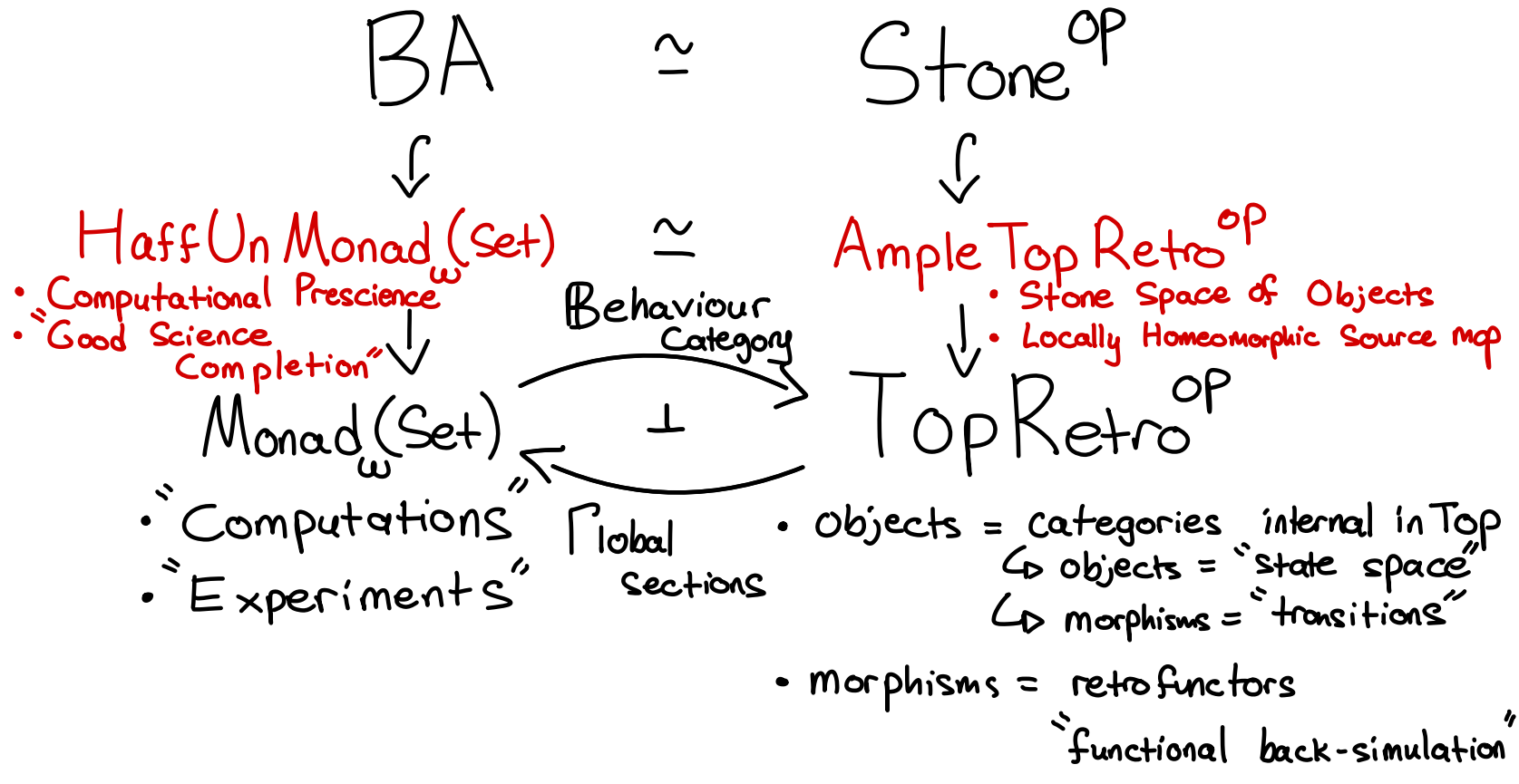
- "Computations"
- "Experiments"

- objects = categories internal in Top
  - ↳ objects = "state space"
  - ↳ morphisms = "transitions"
- morphisms = retrofunctors
  "functional back-simulation"

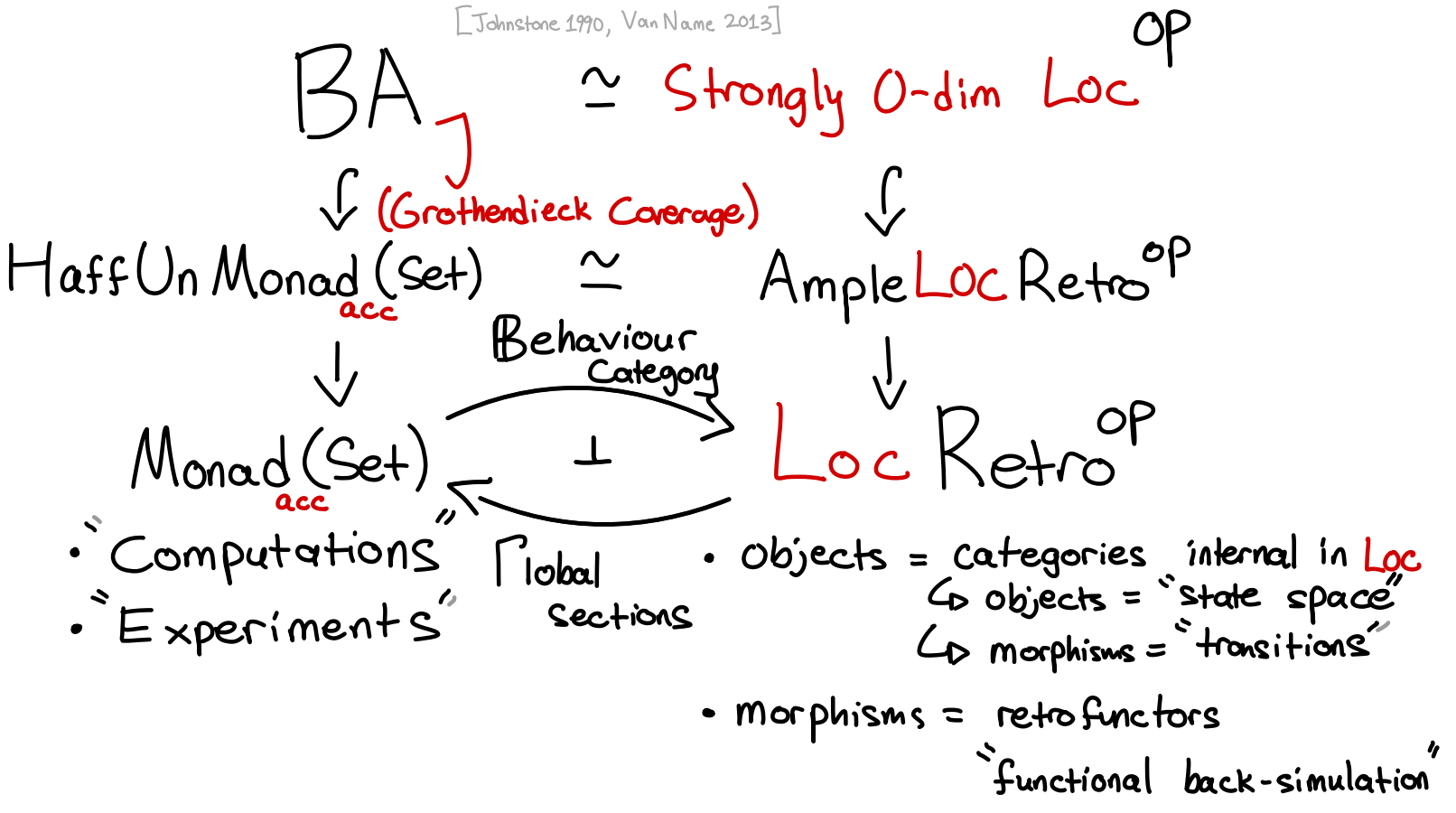
This Talk in One Slide



This Talk in One Slide



This Talk in One Slide



Why?

In TACL '22, Richard said:

The bigger picture

An obvious question: which topological catys are behaviour catys?

We can in fact over-answer this question. There's an adjunction

finitary for simplicity

$$\mathsf{Alg}\,\mathsf{Thy}^{\omega} \underset{\mathbb{T}_{\mathbb{C}} \longleftarrow \mathbb{C}}{\overset{\mathbb{T} \longmapsto \mathbb{B}_{\mathbb{T}}}{\rightleftarrows}} (\mathsf{Top}\mathsf{Cat})^{\mathrm{op}}$$

morphisms are cofunctors

where $\mathbb{T}_{\mathbb{C}}$ extends $\mathbb{T}_{\mathsf{clopen}(C_0)}$ with unary ops m for each $m : C_1 \to C_0$ + axioms.

This is a Galois (= idempotent) adjunction whose restriction to fixpoints is:

$$\mathsf{Cart}\,\mathsf{Closed}\,\mathsf{Thy}^{\omega} \overset{\simeq}{\rightleftarrows} (\mathsf{Ample}\,\mathsf{Top}\,\mathsf{Cat})^{\mathrm{op}}$$

induces cart closed variety

source map is étale
space of obs is Stone space

This extends the non-commutative Stone duality of Kudryavtseva & Lawson

Why?

In TACL '22, Richard said:

The topological behaviour category

In our examples, the topological behaviour categories give known objects from the world of non-commutative geometry:

- In the case of $\mathbb{T}_{RI}$, we get the Cuntz topological groupoid, whose associated C*-algebra is the Cuntz C*-algebra and whose associated R-algebra is the Leavitt algebra.
- In the case of $\mathbb{T}_\delta$ for $G \times I \xrightarrow{\delta} I \times G$ a self-similar group action, we get the Nekrashevych-Röver groupoid.
- ... just the start of a bigger story!

Why?

- Baby Step in a Non-commutative Algebraic Geometry

  replacing rings by Monads in the defn of schemes

Why?

- Baby Step in a Non-commutative Algebraic Geometry

  replacing rings by Monads in the defn of schemes

- For use in program Semantics! monad encodes program syntax,

  so Monadic Scheme encodes Syntax varying continuously over Semantics

Why?

- Baby Step in a Non-commutative Algebraic Geometry

  replacing rings by Monads in the defn of schemes

- For use in program semantics! monad encodes program syntax,

  so monadic scheme encodes syntax varying continuously over semantics

Geometric techniques, e.g. cohomology for local-to-global program reasoning ?!

Why?

- Baby Step in a Non-commutative Algebraic Geometry
  - replacing rings by Monads in the defn of schemes
  - For use in program semantics! monad encodes program syntax,

so Monadic Scheme encodes syntax varying continuously over Semantics

Geometric techniques, e.g. cohomology for local-to-global program reasoning ?!

example

loop_kitty.py

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

Why?

- Baby Step in a Non-commutative Algebraic Geometry
  - replacing rings by Monads in the defn of schemes
  - For use in program semantics! monad encodes program syntax,

so Monadic Scheme encodes syntax varying continuously over Semantics

Geometric techniques, e.g. cohomology for local-to-global program reasoning ?!

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?

- Baby Step in a Non-commutative Algebraic Geometry
  - replacing rings by Monads in the defn of schemes
  - For use in program semantics! monad encodes program syntax,

so Monadic Scheme encodes syntax varying continuously over Semantics

Geometric techniques, e.g. cohomology for local-to-global program reasoning ?!

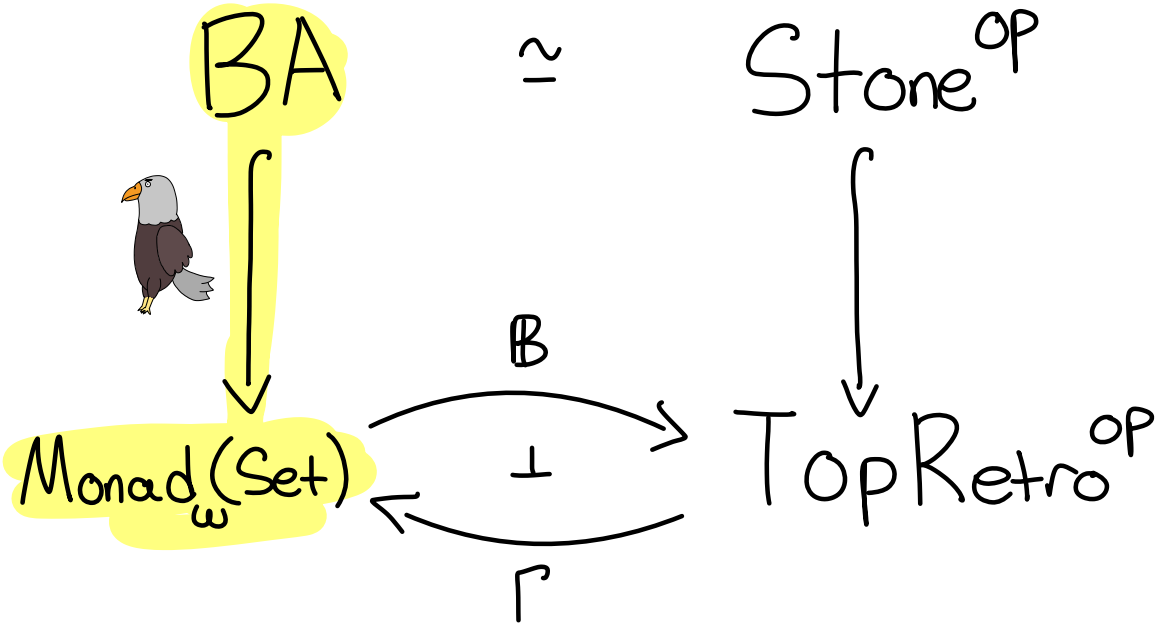
example

loop_kitty.py

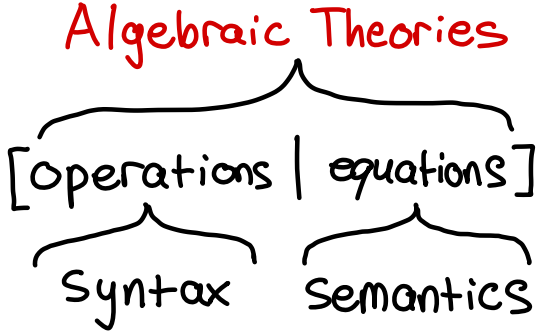
```
with open("cute_kitty.gif") as kitty:
    contents = kitty.read()
    kitty.write(loop(contents,50))
#kitty not in scope anymore
```

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

WORLD MAP



Monads are Algebraic Theories are Notions of Computation



Monads are Algebraic Theories are Notions of Computation

Algebraic Theories ⇝ (induce) Monads on Set

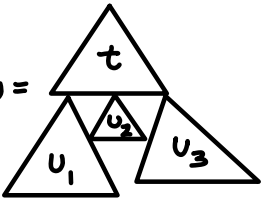
[operations | equations]

Syntax Semantics

$T : Set \to Set,$

$\gg= : TA \times TB^{A} \to TB,$ return : $A \to TA$

$t \gg= u =$



return $a = \overset{!}{a}$

Monads are Algebraic Theories are Notions of Computation

Algebraic Theories ⇝ (induce) Monads on Set

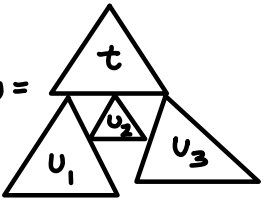
[operations | equations]

Syntax Semantics

$T : Set \to Set,$

$\gg= : TA \times TB^A \to TB,$ return : $A \to TA$

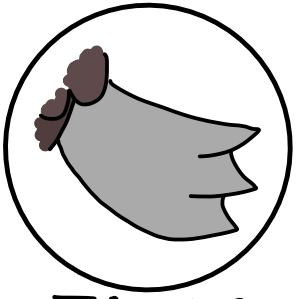
$t \gg= u =$



return $a = \overset{!}{a}$

Example Let $2 = \{$  ,  $\}$

HEADS



TAILS

Monads are Algebraic Theories are Notions of Computation

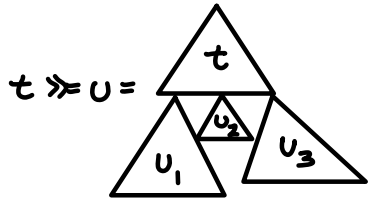
Algebraic Theories $\overset{\text{induce}}{\rightsquigarrow}$ Monads on Set

[operations | equations]

Syntax Semantics

$T : \mathrm{Set} \to \mathrm{Set}$,

$\gg\!= : TA \times TB^{A} \to TB$, return : $A \to TA$



return $a = \overset{!}{a}$

Example Let $2 = \{$  ,  $\}$

$T_{\mathrm{flip}} = [\,\mathrm{flip}/2 \mid \emptyset\,] \rightsquigarrow T_{\mathrm{flip}} : A \longmapsto \{\text{binary trees with A-leaves}\}$

Monads are Algebraic Theories are Notions of Computation

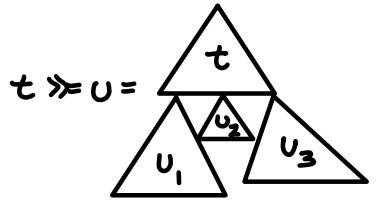
Algebraic Theories ⇝ (induce) Monads on Set

[operations | equations]

Syntax Semantics

$T : \mathrm{Set} \to \mathrm{Set}$,

$\gg\!= : TA \times TB^A \to TB$, return : $A \to TA$



$t \gg\!= u =$

return a = a

Example Let $2 = \{$, $\}$

$\mathbb{T}_{flip} = [\ flip/2 \mid \emptyset\ ] \rightsquigarrow T_{flip} : A \mapsto \{\text{binary trees with A-leaves}\}$

$\mathbb{T}_{(un)flip} = \left[\begin{array}{l|l} flip/2, & \text{return } * = flip \gg\!= \{\ \mapsto unflip,\ \mapsto unflip\ \} \\ unflip/1, & unflip \gg flip = \text{return} \\ unflip/1 & unflip \gg flip = \text{return} \end{array}\right]$

$\gg : TA \times TB \to TB$

$t \gg t' := t \gg\!= \lambda\_.t'$

Monads are Algebraic Theories are Notions of Computation

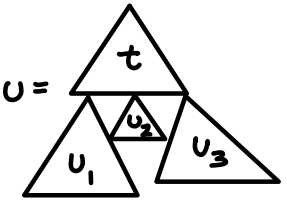
Algebraic Theories ⇝ (induce) Monads on Set

[operations | equations]

Syntax Semantics

$T : Set \to Set$,

$\gg= : TA \times TB^A \to TB$, return : $A \to TA$

$t \gg= u =$ 

return $a = \overset{!}{a}$

Example Let $2 = \{$  ,  $\}$

BA B

$Mnd_\omega(Set)$ $T_B A = \{ d : A \to B \mid |supp(d)| < \omega,$ for $a \neq a'$, $d(a) \wedge d(a') = \bot, \bigvee_{a \in A} d(a) = \top \}$

"finitely supported A-indexed partitions"

Monads are Algebraic Theories are Notions of Computation

Algebraic Theories ⇝ (induce) Monads on Set

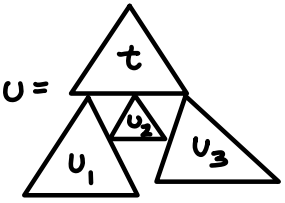
[operations | equations]

Syntax Semantics

$T : \mathrm{Set} \to \mathrm{Set}$,

$\gg= : TA \times TB^A \to TB$, return : $A \to TA$

$t \gg= u =$



return $a = \overset{!}{a}$

Example Let $2 = \{$  ,  $\}$

BA ↓ B ↓

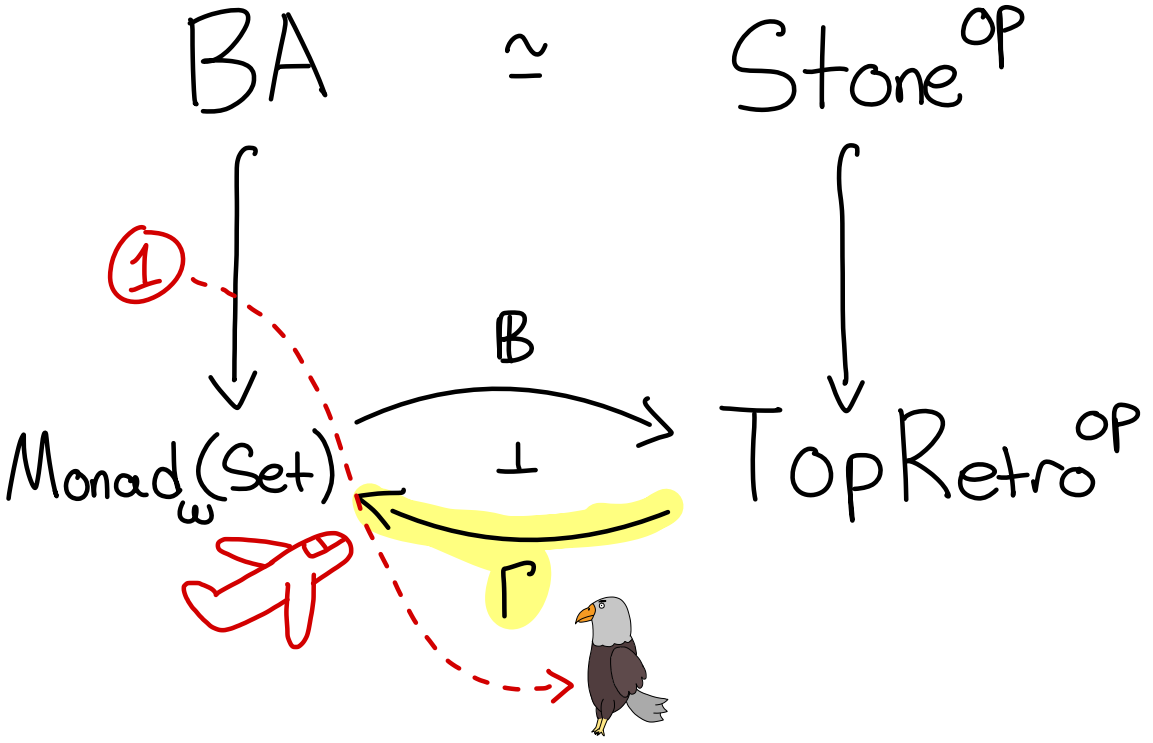
$$\left[\forall\, b \in B.\ b/2 \;\middle|\; \begin{array}{l} T(x,y) = x \\ (b \wedge c)(x,y) = b(c(x,y), y) \end{array} \quad \text{...etc.} \right]$$

$\Vert$ for $a \neq a'$,

$\mathrm{Mnd}_\omega(\mathrm{Set})$ $T_B A = \{ d : A \to B \mid |\mathrm{supp}(d)| < \omega,\ d(a) \wedge d(a') = \bot,\ \bigvee_{a \in A} d(a) = \top \}$

"finitely supported A-indexed partitions"

WORLD MAP



The Global Sections Monad

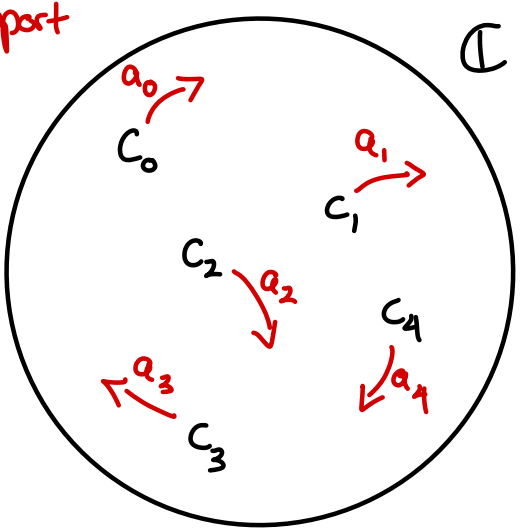
Let $\mathbb{C} \in \mathrm{TopRetro}$. Define monad $\Gamma_{\mathbb{C}}$

The Global Sections Monad

Let $\mathbb{C} \in \mathrm{TopRetro}$. Define monad $\Gamma\mathbb{C}$:

$$\Gamma\mathbb{C}(A) = \left\{ s : \mathbb{C}_0 \to A' \times \mathbb{C}_1 \;\middle|\; A' \subseteq_\omega A \text{ and } \mathrm{dom}(s(c)) = c \right\}$$

"finite support"



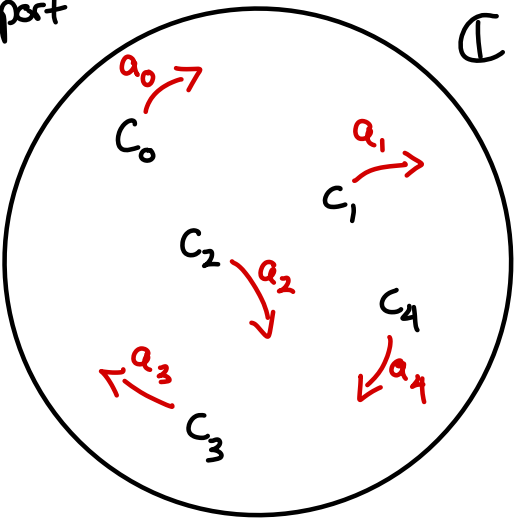
"Monad of Brownian Motions"

The Global Sections Monad

Let $\mathbb{C} \in \mathrm{Top}\mathrm{Retro}$. Define monad $\Gamma\mathbb{C}$:

$$\Gamma\,\mathbb{C}(A) = \left\{ s : \mathbb{C}_0 \to A' \times \mathbb{C}_1 \;\middle|\; A' \subseteq_\omega A \text{ and } \mathrm{dom}(s(c)) = c \right\}$$

"finite support"



"Monad of Brownian Motions"

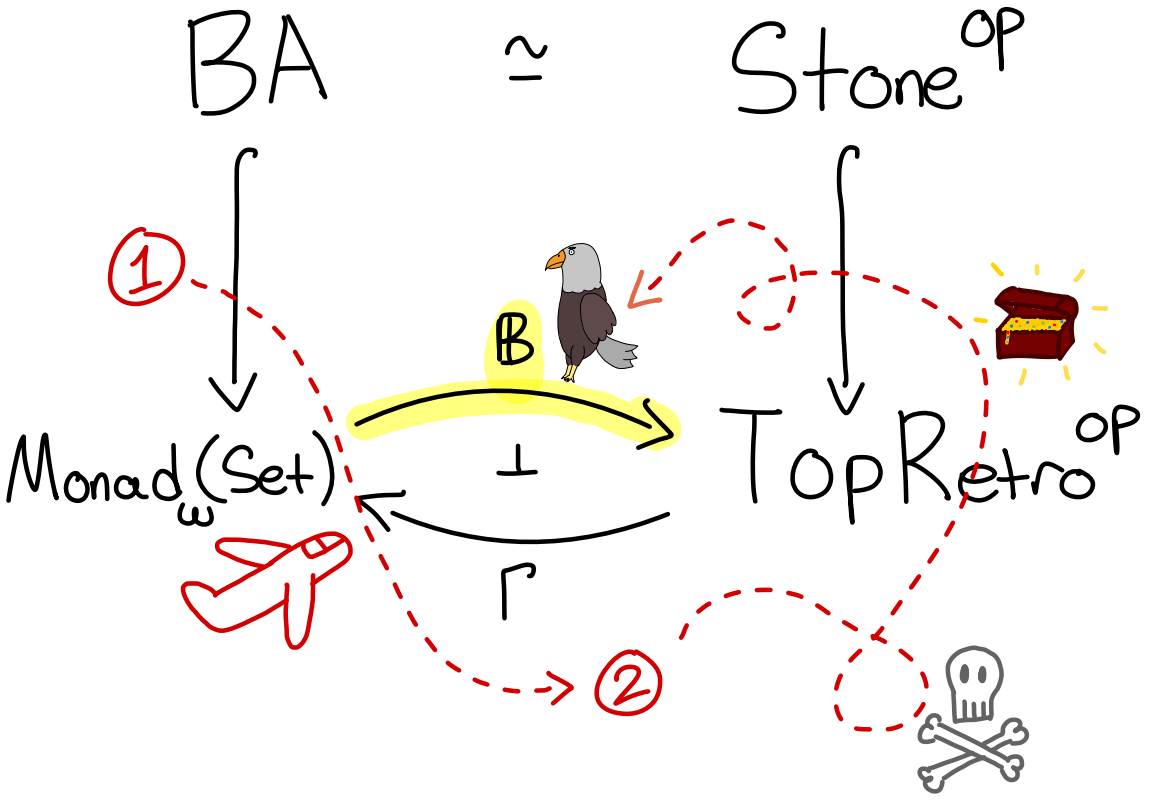
Notice: if $\mathbb{C}_0 = \mathbb{C}_1$ (identity maps only)

then $\Gamma\,\mathbb{C}(2) \cong \mathbb{C}_0 \to 2$

$\cong \mathrm{Clop}(\mathbb{C}_0)$

so this subsumes Stone Duality.

WORLD MAP



The Topological Behaviour Category

From T's perspective,

① $\mathbb{B}_{\circ}T = \{\beta : T \to Id \mid \quad \}$

The Topological Behaviour Category

From T's perspective,

① $\mathbb{B}_{\circ}T = \{\beta : T \to Id \mid \beta(t \gg\!= u) = \beta(t \gg u(\beta(t)))$
$\forall t \in TA,\ u : A \to TB\}$

The Topological Behaviour Category

From T's perspective,

naturality amounts to $\beta(t \gg \text{return } a) = a$

(1) $\mathbb{B}_0 T = \{\beta : T \to \mathrm{Id} \mid \beta(t \gg\!= u) = \beta(t \gg u(\beta(t)))$
$\forall t \in TA,\ u : A \to TB\}$

The Topological Behaviour Category

From T's perspective, naturality amounts to $\beta(t \gg \text{return } a) = a$

(1) $\mathbb{B}_0 T = \{\beta : T \to Id \mid \beta(t \gg\!= u) = \beta(t \gg u(\beta(t)))\}$
$\forall t \in TA,\ u : A \to TB$

topologized by $[t \mapsto a] = \{\beta \mid \beta(t) = a\}$ $\forall t \in TA, a \in A$

notice each $[t \mapsto a]$ is clopen.

The Topological Behaviour Category

From T's perspective, naturality amounts to $\beta(t \gg \text{return } a) = a$

① $\mathbb{B}_0 T = \{\beta : T \to Id \mid \beta(t \gg= u) = \beta(t \gg u(\beta(t)))\}$
$\forall t \in TA,\ u : A \to TB$

topologized by $[t \mapsto a] = \{\beta \mid \beta(t) = a\}$ $\forall t \in TA, a \in A$

notice each $[t \mapsto a]$ is clopen.

② For $\mathbb{B}_1 T$, from T's perspective, the only transitions that can exist come from T1 ⇝ Whatever $\mathbb{B}_1 T$ is, ideally $T1 \cong \{\text{global sections } \mathbb{B}_0 T \to \mathbb{B}_1 T\}$

The Topological Behaviour Category

From T's perspective, naturality amounts to $\beta(t \gg\!= \text{return } a) = a$

① $\mathbb{B}_0 T = \{\beta : T \to Id \mid \beta(t \gg\!= u) = \beta(t \gg u(\beta(t))) \quad \forall t \in TA,\ u : A \to TB\}$

topologized by $[t \mapsto a] = \{\beta \mid \beta(t) = a\}$ $\forall t \in TA, a \in A$

notice each $[t \mapsto a]$ is clopen.

② For $\mathbb{B}_1 T$, from T's perspective, the only transitions that can exist come from T1, but subject to

$\beta \xrightarrow{t \gg\!= u} \beta'' \sim_\beta \beta \xrightarrow{t} \beta' \xrightarrow{u(\beta'(t))} \beta''$

$\rightsquigarrow$ so $\mathbb{B}_1 T = \sum_{\beta \in \mathbb{B}_0 T} T1/\sim_\beta$

Topologized as local homeomorphism corresponding to sheaf generated by global sections T1.

The Topological Behaviour Category

From T's perspective, naturality amounts to $\beta(t \gg\!= \text{return } a) = a$

This makes $\mathbb{B}T$ ample

① $\mathbb{B}_0 T = \{\beta : T \to Id \mid \beta(t \gg\!= u) = \beta(t \gg u(\beta(t))) \quad \forall t \in TA,\ u : A \to TB\}$

topologized by $[t \mapsto a] = \{\beta \mid \beta(t) = a\} \quad \forall t \in TA, a \in A$

notice each $[t \mapsto a]$ is clopen.

② For $\mathbb{B}_1 T$, from T's perspective, the only transitions that can exist come from T1, but subject to

$$\beta \xrightarrow{t \gg\!= u} \beta'' \sim_\beta \beta \xrightarrow{t} \beta' \xrightarrow{u(\beta'(t))} \beta''$$

so $\mathbb{B}_1 T = \sum_{\beta \in \mathbb{B}_0 T} T1/\sim_\beta$

$\sigma = \pi_1$: $\mathbb{B}_1 T \to \mathbb{B}_0 T$

τ : $[t]_\beta \mapsto \beta(t \gg\!= -)$: $\mathbb{B}_1 T \to \mathbb{B}_0 T$

Topologized as local homeomorphism corresponding to sheaf generated by global sections T1.

The Topological Behaviour Category (Example)

Let $\mathbb{B}$ Boolean Algebra. Then:

- $\beta : T \to Id \in \mathcal{B}_0 T_{\mathbb{B}}$ determined by $\beta_2 : T2 \longrightarrow 2$ $\quad \mathbb{B} \cong 2^{2}$ (ultrafilter)

  so $\mathcal{B}_0 T \cong Spec(\mathbb{B})$

The Topological Behaviour Category (Example)

Let $\mathbb{B}$ Boolean Algebra. Then:

- $\beta : T \to Id \in \mathbb{B}_0 T_{\mathbb{B}}$ determined by $\beta_2 : T2 \to 2$ ($2 = \mathbb{B}$, ultrafilter)

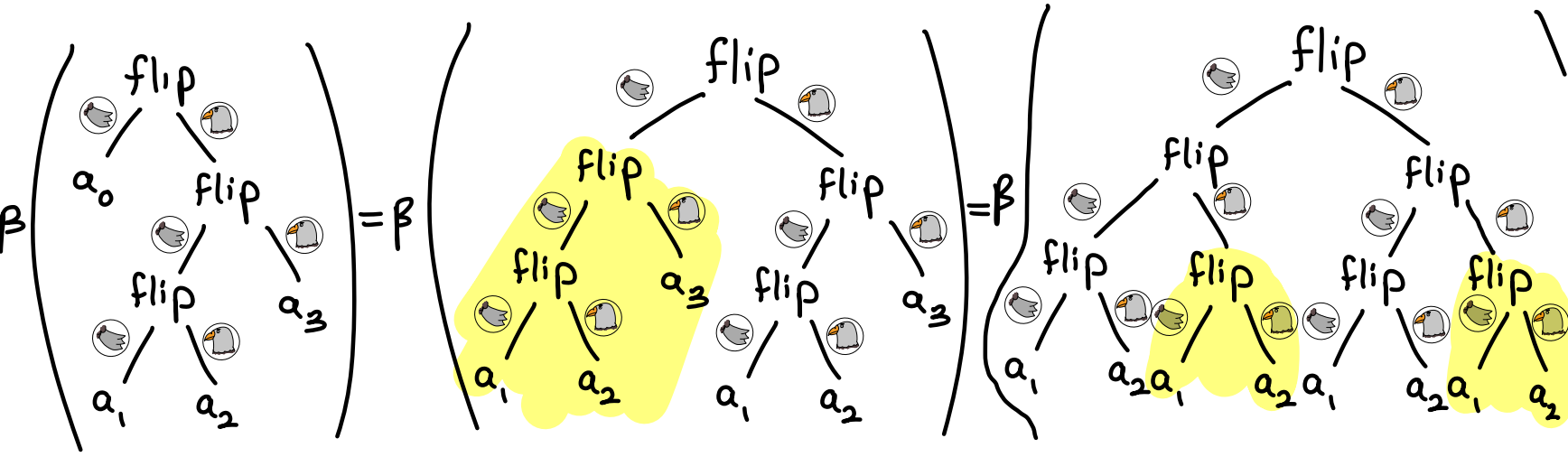
  so $\mathbb{B}_0 T \cong Spec(\mathbb{B})$

- $T1 \cong 1$ so $\mathbb{B}_1 T \cong \mathbb{B}_0 T$.

  (identity transitions only)

The Topological Behaviour Category (Example)

For T_{flip}, $\mathcal{B}_0 T_{flip} \cong 2^{\omega}$ since $\beta(t) = \beta(\underbrace{flip \gg \ldots \gg flip}_{n \text{ times}}) \in 2$, with Cantor space topology.



So $\beta = \bigcirc\bigcirc\bigcirc\ldots$

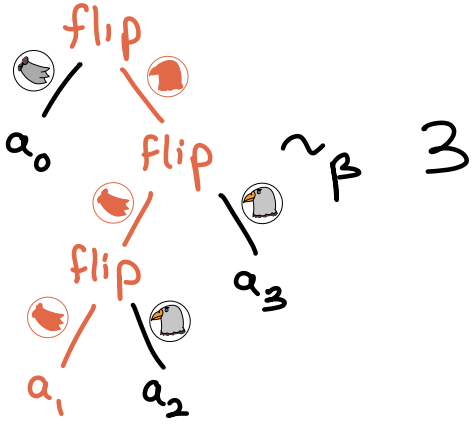
$= \beta(flip \gg flip \gg flip(a_1, a_2))$

$= a_1$

The Topological Behaviour Category (Example)

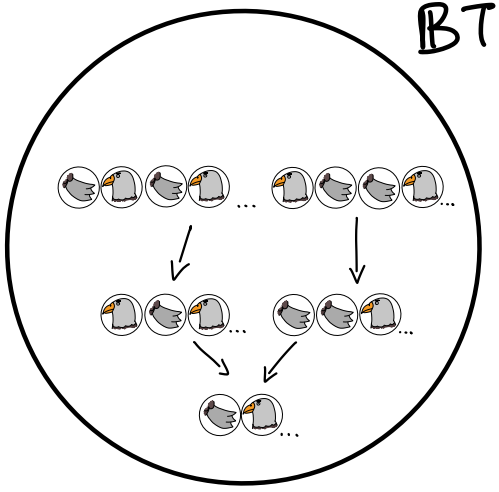
For T_{flip}, $\mathbb{B}_0 T_{flip} \cong 2^\omega$ since $\beta(t) = \beta(\underbrace{flip \gg \ldots \gg flip}_{n \text{ times}}) \in 2$, with cantor space topology.

$\mathbb{B}_1 T_{flip} \cong 2^\omega \times \mathbb{N}$ since $\sim_\beta$ counts number of flips along β-path.



$\sim_\beta$ 3

for $\beta =$ 



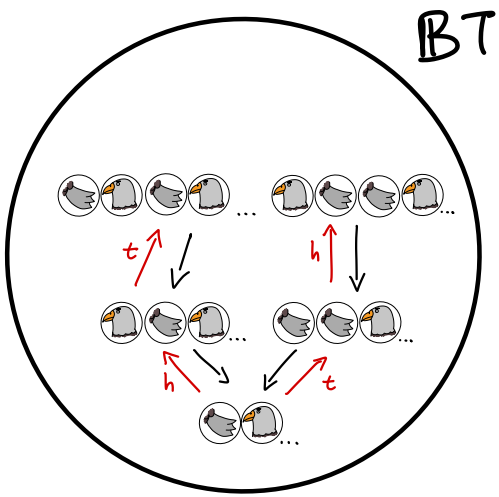
The Topological Behaviour Category (Example)

For T_{flip}, $\mathbb{B}_0 T_{flip} \cong 2^\omega$ since $\beta(t) = \beta(\underbrace{flip \gg \dots \gg flip}_{n \text{ times}}) \in 2$, with cantor space topology.

$\mathbb{B}_1 T_{flip} \cong 2^\omega \times \mathbb{N}$ since $\sim_\beta$ counts number of flips along β-path.

For $T_{(un)flip}$, $\mathbb{B}_0 T_{(un)flip} \cong 2^\omega$ (stays the same)

$$\mathbb{B}_1 T_{(un)flip} \cong 2^\omega \times (\mathbb{N} + \{t, h\}^+)$$



The Topological Behaviour Category (Example)

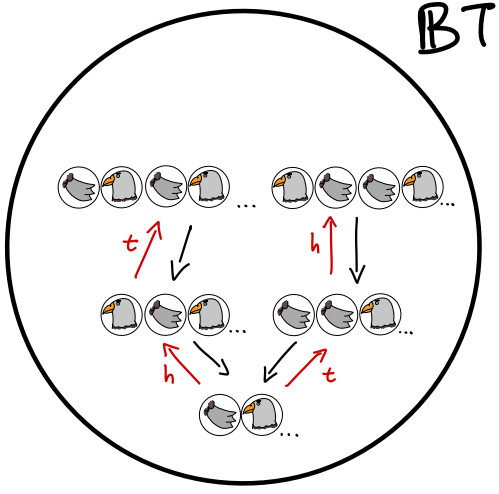
For T_{flip}, $\mathbb{B}_0 T_{flip} \cong 2^\omega$ since $\beta(t) = \beta(\underbrace{flip \gg \dots \gg flip}_{n \text{ times}}) \in 2$, with cantor space topology.

$\mathbb{B}_1 T_{flip} \cong 2^\omega \times \mathbb{N}$ since $\sim_\beta$ counts number of flips along β-path.

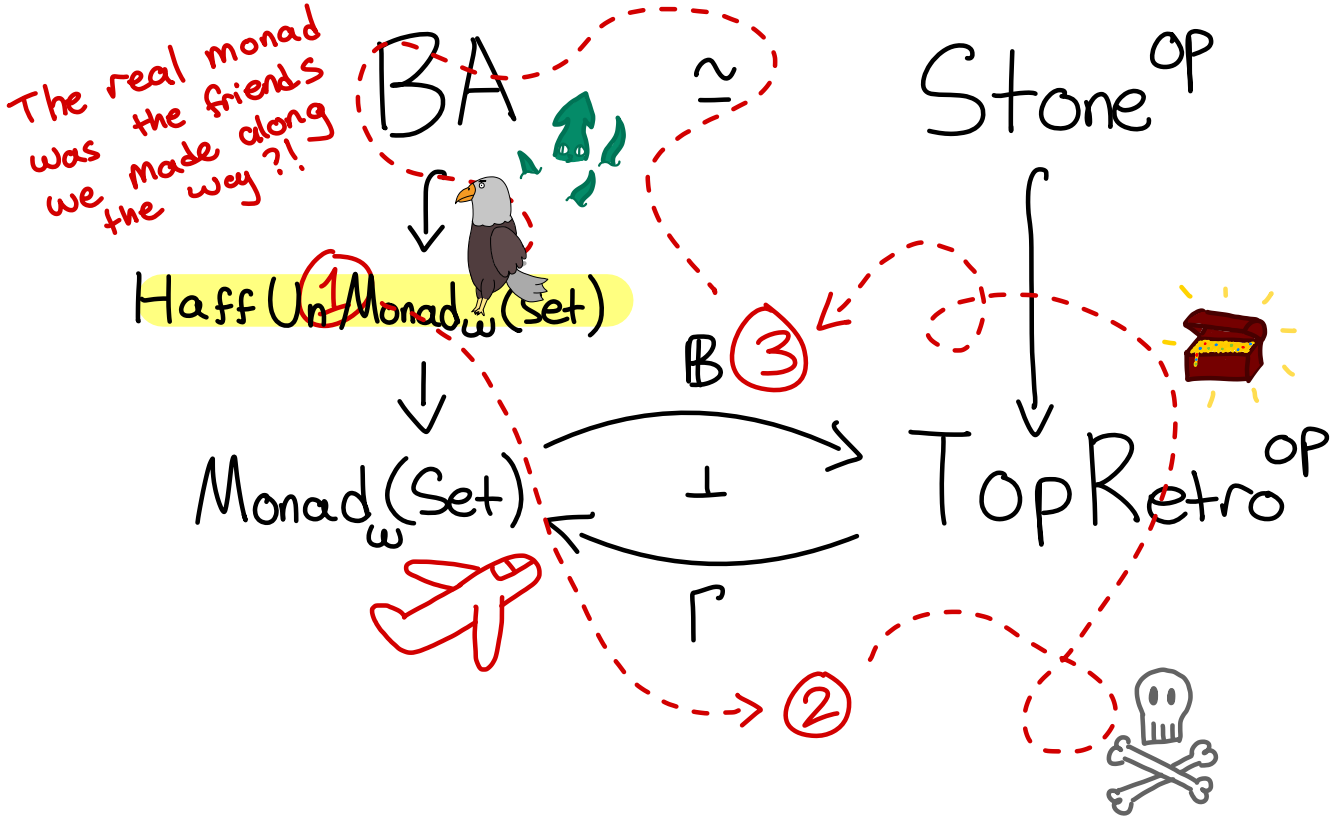
For $T_{(un)flip}$, $\mathbb{B}_0 T_{(un)flip} \cong 2^\omega$ (Stays the same)

$$\mathbb{B}_1 T_{(un)flip} \cong 2^\omega \times (\mathbb{N} + \{t, h\}^+)$$

Now is a groupoid!

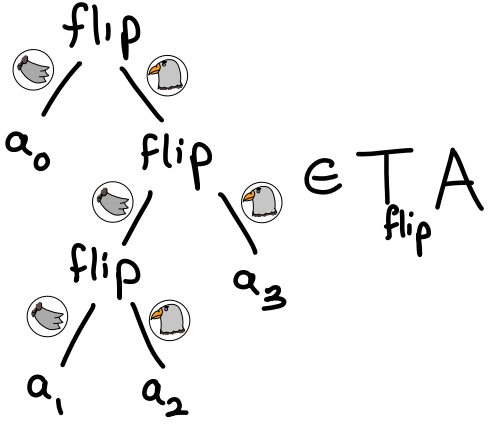


WORLD MAP

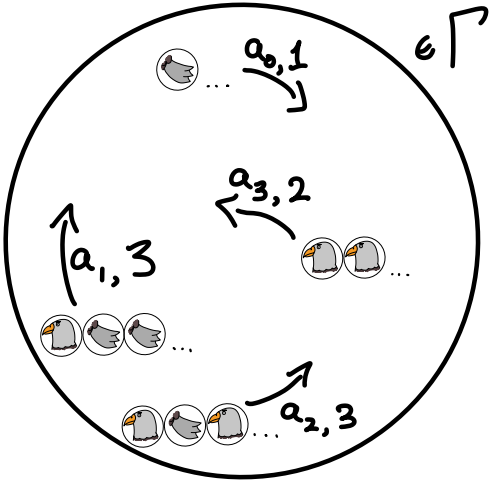

The real monad was the friends we made along the way?!
BA
≃
Stone^op
Haff(Un)
Monad_ω(Set)
1
B
3
Monad_ω(Set)
⊥
⊢
TopRetro^op
2

Prescient Operations

The unit map:



$\in T_{\text{flip}} A$ $\quad \eta \mapsto$

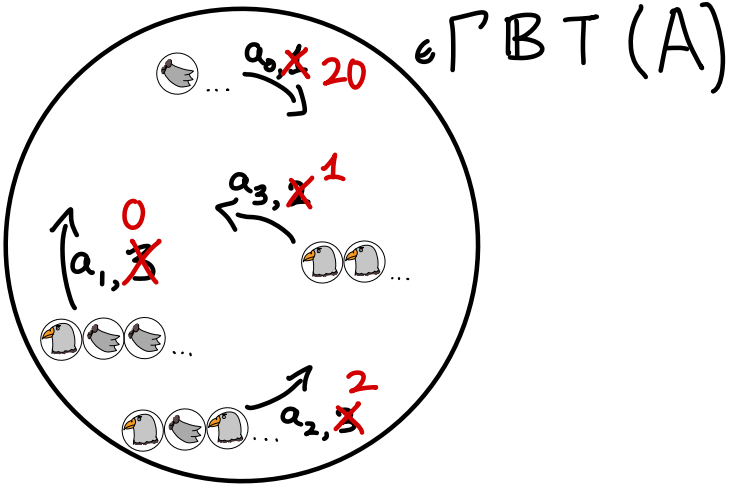


$\in \Gamma \mathbb{B} T(A)$

Prescient Operations

The unit map:

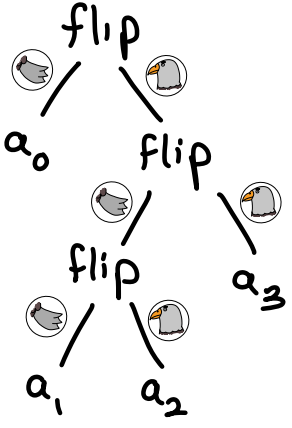
$\nexists\, t \in T_{\text{flip}} A \quad \not\vdash\!\!\rightarrow$



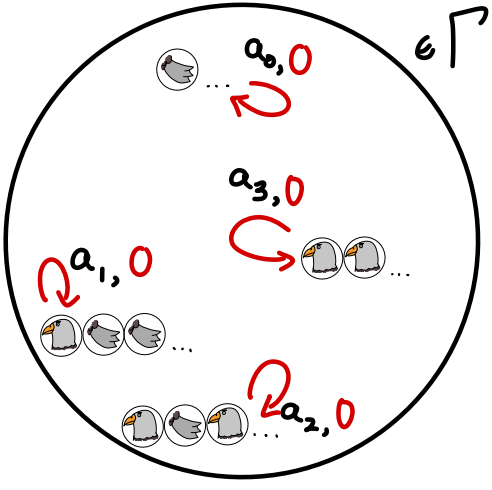
Topology enforces finite information, but not consumption.

Prescient Operations

The unit map :



$=: t \rightsquigarrow \overline{t} =$



$\in \Gamma \mathbb{B} T(A)$

Topology enforces finite information, but not consumption.

Idea: to each $t \in TA$, add $\overline{t}$ which predicts t without executing it.


LISAN AL GAIB

Prescient Operations

For $T_{(un)flip}$ each $\overline{t}$ already exists in T, so $T \tilde{=}> \Gamma \mathbb{B} T$.

Prescient Operations

For $T_{(un)flip}$ each $\bar{t}$ already exists in T, so $T \tilde{\Rightarrow} \Gamma \mathbb{B} T$.

Defn $h \in TA$ is **hyperaffine** if

$h \gg \text{return } a = \text{return } a$ and $h \gg\!= \lambda a.\, h \gg\!= \lambda b.\, \text{return } (a, b)$

"h does nothing if you ignore its output"

$= h \gg\!= \lambda a.\, \text{return } (a, a)$

"h is idempotent"

Prescient Operations

For $T_{(un)flip}$ each $\bar{t}$ already exists in T, so $T \tilde{\Rightarrow} \Gamma \mathbb{B} T$.

Defn $h \in TA$ is **hyperaffine** if

$h \gg \text{return } a = \text{return } a$ and $h \gg\!= \lambda a.\, h \gg\!= \lambda b.\, \text{return } (a, b)$

"h does nothing if you ignore its output"

$\quad = h \gg\!= \lambda a.\, \text{return } (a, a)$

"h is idempotent"

T is **hyperaffine-unary** if

$\forall t \in TA.\ \exists!\ \bar{t}$ hyperaffine. $t = \bar{t} \gg\!= \lambda a.\, t \gg \text{return } a.$

Prescient Operations

For $T_{(un)flip}$ each $\bar{t}$ already exists in T, so $T \tilde{\Rightarrow} \Gamma \mathbb{B} T$.

Defn $h \in TA$ is **hyperaffine** if

$h \gg \text{return } a = \text{return } a$ and $h \gg\!= \lambda a.\, h \gg\!= \lambda b.\, \text{return } (a, b)$

"h does nothing if you ignore its output"

$\overset{\shortparallel}{h \gg\!= \lambda a.\, \text{return } (a, a)}$

"h is idempotent"

T is **hyperaffine-unary** if

$$\forall t \in TA.\ \exists!\, \bar{t} \text{ hyperaffine}.\ t = \bar{t} \gg\!= \lambda a.\, t \gg \text{return } a.$$

prop $T \overset{\sim}{\Rightarrow} \Gamma \mathbb{B} T$ iso iff T hyperaffine-unary.

theorem $\mathbf{HaffUnMnd}_\omega \simeq \mathbf{AmpleTopRetro}^{op}$.

The Topological Behaviour Category (Bad!)

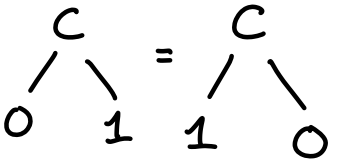
Let T monad.

- If $\mathrm{fail} \in T\,0$ then $\mathbb{B}_0 T = \mathbb{B}_1 T = \emptyset$.

The Topological Behaviour Category (Bad!)

Let T monad.

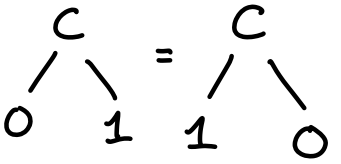
- If $\mathrm{fail} \in T0$ then $\mathbb{B}_0 T = \mathbb{B}_1 T = \emptyset$.
- If $c \in T2$ commutative, i.e.



then $\mathbb{B}_0 T = \mathbb{B}_1 T = \emptyset$.

The Topological Behaviour Category (Bad!)

Let T monad.

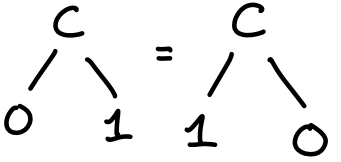
- If $\mathrm{fail} \in T0$ then $\mathbb{B}_0T = \mathbb{B}_1T = \emptyset$.
- If $c \in T2$ commutative, i.e. 

then $\mathbb{B}_0T = \mathbb{B}_1T = \emptyset$.

The Topological Behaviour Category (Bad!)

Let T monad.

- If $\text{fail} \in T0$ then $\mathbb{B}_0T = \mathbb{B}_1T = \emptyset$.
- If $c \in T2$ commutative, i.e.



then $\mathbb{B}_0T = \mathbb{B}_1T = \emptyset$.

- Also if T induced by

$$\left[\begin{array}{l|l} \forall x \in \mathbb{R}, & \mathsf{get}_x \gg\!= \lambda n.\mathsf{get}_x \lambda m.\ \mathsf{return}(n,m) = \mathsf{get}_x \lambda n.\ \mathsf{return}(n,n) \qquad \mathsf{get}_x \gg \mathsf{return}\ a = \mathsf{return}\ a \\ \mathsf{get}_x/\mathbb{N} & \mathsf{get}_x \gg\!= \lambda n.\mathsf{get}_y \gg\!= \lambda m.\ \mathsf{return}(n,m) = \mathsf{get}_y \gg\!= \lambda m.\mathsf{get}_x \gg\!= \lambda n.(n,m) \\ & \mathsf{get}_x \gg\!= \lambda n.\mathsf{get}_y \gg\!= \lambda m.\ \mathsf{return}(n,m) = \mathsf{get}_x \gg\!= \lambda n.\mathsf{get}_y \gg\!= \lambda m.f(n,m) \\ & \text{for all } x \neq y \in \mathbb{R} \text{ and } f\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}\times\mathbb{N} \text{ satisfying } \forall n \neq m \in \mathbb{N}.f(n,m) = (n,m) \end{array}\right]$$

Future Work

- BT is "Kripke Semantics" for a PDL on T

Future Work

- $\mathbb{B}T$ is "Kripke Semantics" for a PDL on T
  - interpret φ as clopen sets on $\mathbb{B}_0 T$

Future Work

- $\mathbb{BT}$ is "Kripke Semantics" for a PDL on T
  - interpret φ as clopen sets on $\mathbb{B}_0 T$
  - $[t]\varphi = \{\beta \mid \beta \xrightarrow{t} \beta' \text{ and } \beta' \in \varphi\}$

Future Work

- $\mathbb{BT}$ is "Kripke Semantics" for a PDL on T
  - interpret φ as clopen sets on $\mathbb{B}_0 T$
  - $[t]\varphi = \{\beta \mid \beta \xrightarrow{t} \beta' \text{ and } \beta' \in \varphi\}$

TODO. axiomatise and prove meta-theorems.

Good Bsc or MSc project?

Future Work

- BT is "Kripke Semantics" for a PDL on T
  - interpret φ as clopen sets on $\mathbb{B}_0 T$
  - $[t]\varphi = \{\beta \mid \beta \xrightarrow{t} \beta' \text{ and } \beta' \in \varphi\}$

TODO. axiomatise and prove meta-theorems.

Q: What about the programs $t_1 \cup t_2$, t^*, $\varphi?$

Future Work

- BT is "Kripke Semantics" for a PDL on T
  - interpret φ as clopen sets on $\mathbb{B}_0 T$
  - $[t]\varphi = \{\beta \mid \beta \xrightarrow{t} \beta' \text{ and } \beta' \in \varphi\}$

  TODO. axiomatise and prove meta-theorems.

  Q: What about the programs $t_1 \cup t_2$, t^*, φ?

  } Relatedly

- Hyperaffine-unary T is "like" a Kleene Algebra with Test (KAT) (without $+$, $*$)

  Can we tweak to get a duality subsuming KAT?

Future Work

About $t_1 \cup t_2$ and φ?

Future Work

About $t_1 \cup t_2$ and φ?

In fact $\mathrm{Open}(\mathbb{B}, T)$ is a quantal frame with

$$U ; V = \{ f ; g \mid f \in U , \; g \in V, \; \mathrm{cod}(f) = \mathrm{dom}(g) \}$$

The composition of $\mathbb{B}T$ is a shadow of this.
A more general relationship with quantales?

Future Work

About $t_1 \cup t_2$ and φ?

In fact $\mathrm{Open}(\mathbb{B}, T)$ is a quantal frame with

$$U ; V = \{ f ; g \mid f \in U , \; g \in V, \; \mathrm{cod}(f) = \mathrm{dom}(g) \}$$

The composition of $\mathbb{B}T$ is a shadow of this.

A more general relationship with quantales?

theorem $\quad \mathrm{Monad}_{T0 \neq \emptyset, \mathrm{acc}}(\mathrm{Set}) \underset{\perp}{\rightleftarrows} \mathrm{Quantale} =$ "property-free localic categories" (Nomadic Systems, c.f. Deleuze on Nomadic Science)

Future Work

About $t_1 \cup t_2$ and φ?

In fact $\mathrm{Open}(\mathcal{B}, T)$ is a quantal frame with

$$U ; V = \{ f ; g \mid f \in U , \; g \in V, \; \mathrm{cod}(f) = \mathrm{dom}(g) \}$$

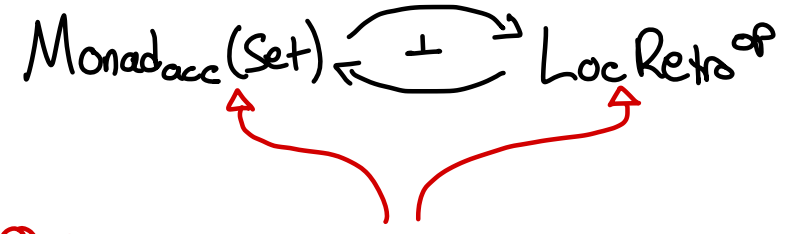
The composition of $\mathcal{BT}$ is a shadow of this.

A more general relationship with quantales?

theorem $\quad \mathrm{Monad}_{T 0 \neq \emptyset, \mathrm{acc}}(\mathrm{Set}) \underset{\perp}{\rightleftarrows} \mathrm{Quantale} =$ "property-free localic categories" (Nomadic Systems, c.f. Deleuze on Nomadic Science)

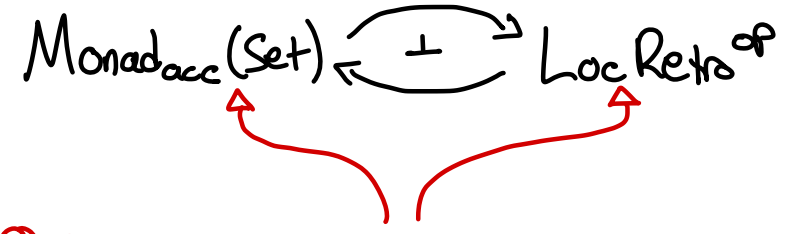
Q: Relationship with Stone Duality?

Future Work



Q Both of these ought to be 2-dimensional?

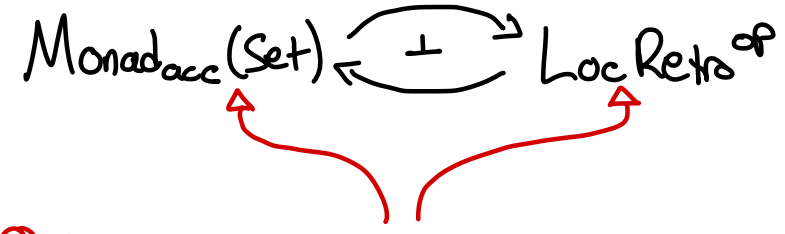
Future Work



Q Both of these ought to be 2-dimensional?
A Yes, in fact double categorical!

Can we get a double-adjunction version of this?

Future Work



Q Both of these ought to be 2-dimensional?
A Yes, in fact double categorical!

Can we get a double-adjunction version of this?

Motivation The point of scheme theory is gluing.

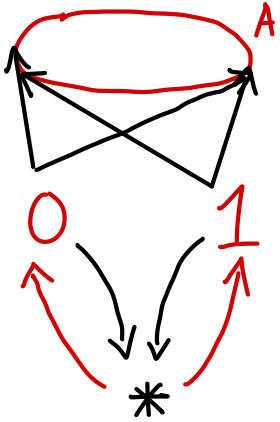
But it seems the correct notion of gluing we want is **lax colimits** (the computer science mandates this...)

One More Thing...

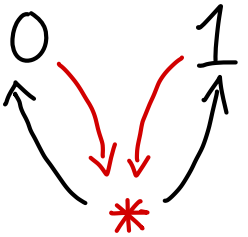
Stone Spaces from Games

= opponent

= player



Dualize ⇝



$T_{flip}\, A$

$\cong$

wf winning strategies

2^{ω}

$\cong$

winning strategies

Beginnings of a Girardian Monist Duality?


THANK
YOU!

fin.