A Fibrational Framework for Substructural and Modal Dependent Type Theories

Dan Licata
Wesleyan University

joint work with Mitchell Riley and Mike Shulman

Modalities

Axiomatic cohesion [Lawvere]

$$
\begin{tikzcd}
\mathrm{Spaces} \arrow[d, bend right=60, "\Pi_0"'] \arrow[d, "\Gamma"] \\
\mathrm{Sets} \arrow[u, bend left=40, "\Delta"] \arrow[u, bend right=60, "\nabla"']
\end{tikzcd}
\qquad \Pi_0 \dashv \Delta \dashv \Gamma \dashv \nabla
$$

∞-categorical Cohesion

[Schreiber,Shulman]

$$\begin{tikzcd}[row sep=huge]
\text{Topological } \infty\text{-groupoids}
\arrow[d, shift right=8ex, bend right=40, "\Pi"']
\arrow[d, phantom, shift right=6ex, "\dashv"]
\arrow[d, <-, shift right=3ex, bend right=30, "\Delta"']
\arrow[d, phantom, shift right=1ex, "\dashv"]
\arrow[d, "\Gamma"]
\arrow[d, phantom, shift left=3ex, "\dashv"]
\arrow[d, <-, shift left=5ex, bend left=40, "\nabla"]
\\
\infty\text{-groupoids}
\end{tikzcd}$$

∞-categorical Cohesion

[Schreiber,Shulman]

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\text{Topological } \infty\text{-groupoids}
  \arrow[d, bend right=70, "\Pi"']
  \arrow[d, shift right=3, phantom, "\dashv"]
  \arrow[d, shift right=8, phantom, "\dashv"]
  \arrow[d, "\Gamma"]
  \arrow[d, shift left=8, phantom, "\dashv"]
  \\
\infty\text{-groupoids}
  \arrow[u, bend left=40, "\Delta"]
  \arrow[u, bend right=70, "\nabla"']
\end{tikzcd}
$$

$\Pi \dashv \Delta \dashv \Gamma \dashv \nabla$

fundamental ∞-groupoid!

e.g. $\Delta\Pi$(topological $\mathbb{S}^1$) = HIT S^1 in real-cohesive HoTT

∞-categorical Cohesion

[Schreiber,Shulman]

$$\begin{tikzcd}[column sep=large]
\text{Topological } \infty\text{-groupoids} \arrow[d, bend right=60, "\Pi"'] \arrow[d, "\Gamma"] \\
\infty\text{-groupoids} \arrow[u, bend left=40, "\Delta"'] \arrow[u, bend right=60, "\nabla"']
\end{tikzcd}$$

$$\Pi \dashv \Delta \dashv \Gamma \dashv \nabla$$

fundamental ∞-groupoid!
e.g. $\Delta\Pi$(topological $\mathbb{S}^1$) = HIT S^1 in real-cohesive HoTT

Δ and ∇ full and faithful…

∞-categorical cohesion

$$\begin{tikzcd}[column sep=large, row sep=huge]
\text{Topological } \infty\text{-groupoids} \arrow[d, "\flat"] \\
\text{Topological } \infty\text{-groupoids} \arrow[u, bend left=60, "\int"] \arrow[u, bend right=60, "\#"']
\end{tikzcd}$$

$$\int \dashv \flat \dashv \#$$

$$\int = \Delta\Pi$$

$$\flat = \Delta\Gamma$$

$$\# = \nabla\Gamma$$

∞-categorical cohesion

$$\begin{tikzcd}[row sep=huge]
\text{Topological } \infty\text{-groupoids} \arrow[d, "\flat"] \arrow[d, phantom, shift right=6ex, "\dashv"] \arrow[d, phantom, shift left=6ex, "\dashv"] \\
\text{Topological } \infty\text{-groupoids} \arrow[u, bend left=50, "\int"] \arrow[u, bend right=50, "\#"']
\end{tikzcd}$$

$$\int = \Delta\Pi$$

$$\flat = \Delta\Gamma$$ **comonad**

$$\# = \nabla\Gamma$$

∞-categorical cohesion

$$\begin{tikzcd}[row sep=huge]
\text{Topological } \infty\text{-groupoids} \arrow[d, "\flat"] \\
\text{Topological } \infty\text{-groupoids} \arrow[u, bend left=50, "\int"] \arrow[u, bend right=50, "\sharp"']
\end{tikzcd}$$

$\int = \Delta\Pi$

$\flat = \Delta\Gamma$ **comonad**

$\# = \nabla\Gamma$ **monad**

∞-categorical cohesion

$$\begin{tikzcd}[row sep=huge]
\text{Topological } \infty\text{-groupoids} \arrow[d, "\flat" description] \\
\text{Topological } \infty\text{-groupoids} \arrow[u, bend left=60, "\int"] \arrow[u, bend right=60, "\sharp"']
\end{tikzcd}$$

$\int = \Delta\Pi$

$\flat = \Delta\Gamma$ **comonad**

$\sharp = \nabla\Gamma$ **monad**

idempotent

∞-categorical cohesion

$$
\begin{tikzcd}
\text{Topological } \infty\text{-groupoids} \\
\text{Topological } \infty\text{-groupoids}
\arrow[u, bend left=40, "\int"]
\arrow[u, phantom, shift right=8, "\dashv"]
\arrow[from=1-1, to=2-1, "\flat"]
\arrow[u, phantom, shift left=8, "\dashv"]
\arrow[u, bend right=40, "\#"']
\end{tikzcd}
$$

$\int = \Delta\Pi$ **monad**

$\flat = \Delta\Gamma$ **comonad**

$\# = \nabla\Gamma$ **monad**

idempotent

∞-categorical cohesion

$$
\begin{tikzcd}[row sep=huge]
\text{Topological } \infty\text{-groupoids} \arrow[d, phantom, "\int \;\; \dashv \;\; \flat \;\; \dashv \;\; \#"] \\
\text{Topological } \infty\text{-groupoids}
\end{tikzcd}
$$

$\int = \Delta\Pi$ **monad**

$\flat = \Delta\Gamma$ **comonad**

$\# = \nabla\Gamma$ **monad**

idempotent

***Modality*: historically endofunctor on types/propositions**

$\Box A \quad \Diamond A \quad !A \quad ?A$

Differential cohesion

[Scheiber; Wellen; Gross,L.,New,Paykin,Riley,Shulman,Wellen]

$$
\begin{array}{ccccccccc}
\Re & \dashv & \Im & \dashv & \& & & \\
 & & \cup & & \cup & & \\
 & & \int & \dashv & \flat & \dashv & \sharp
\end{array}
$$

Differential cohesion

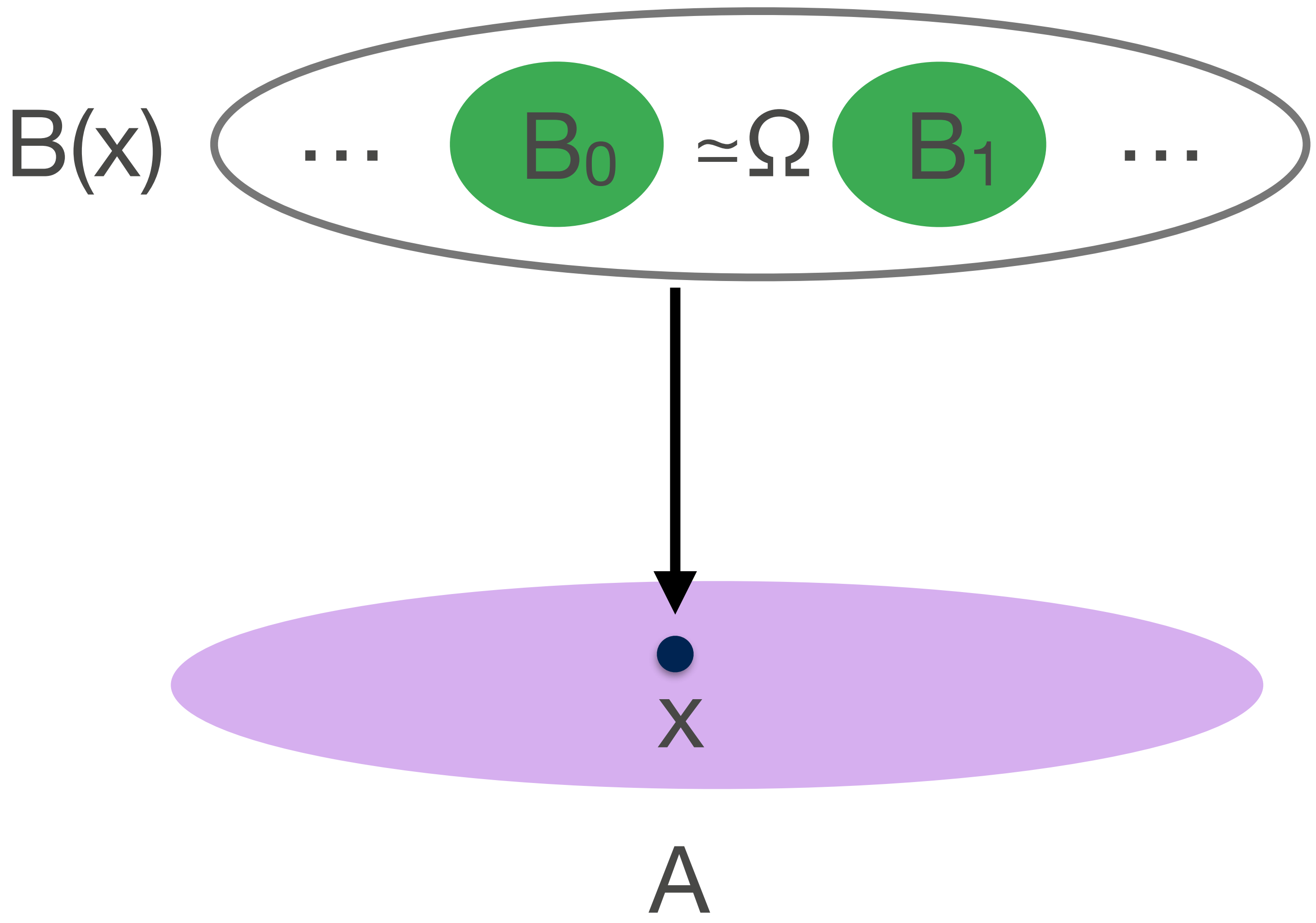
[Scheiber; Wellen; Gross,L.,New,Paykin,Riley,Shulman,Wellen]

$$\begin{array}{ccccccccc} \Re & \dashv & \Im & \dashv & \& & & \\ & & \cup & & \cup & & \\ & & \int & \dashv & \flat & \dashv & \sharp \end{array}$$

Next level: super homotopy theory [Schreiber]

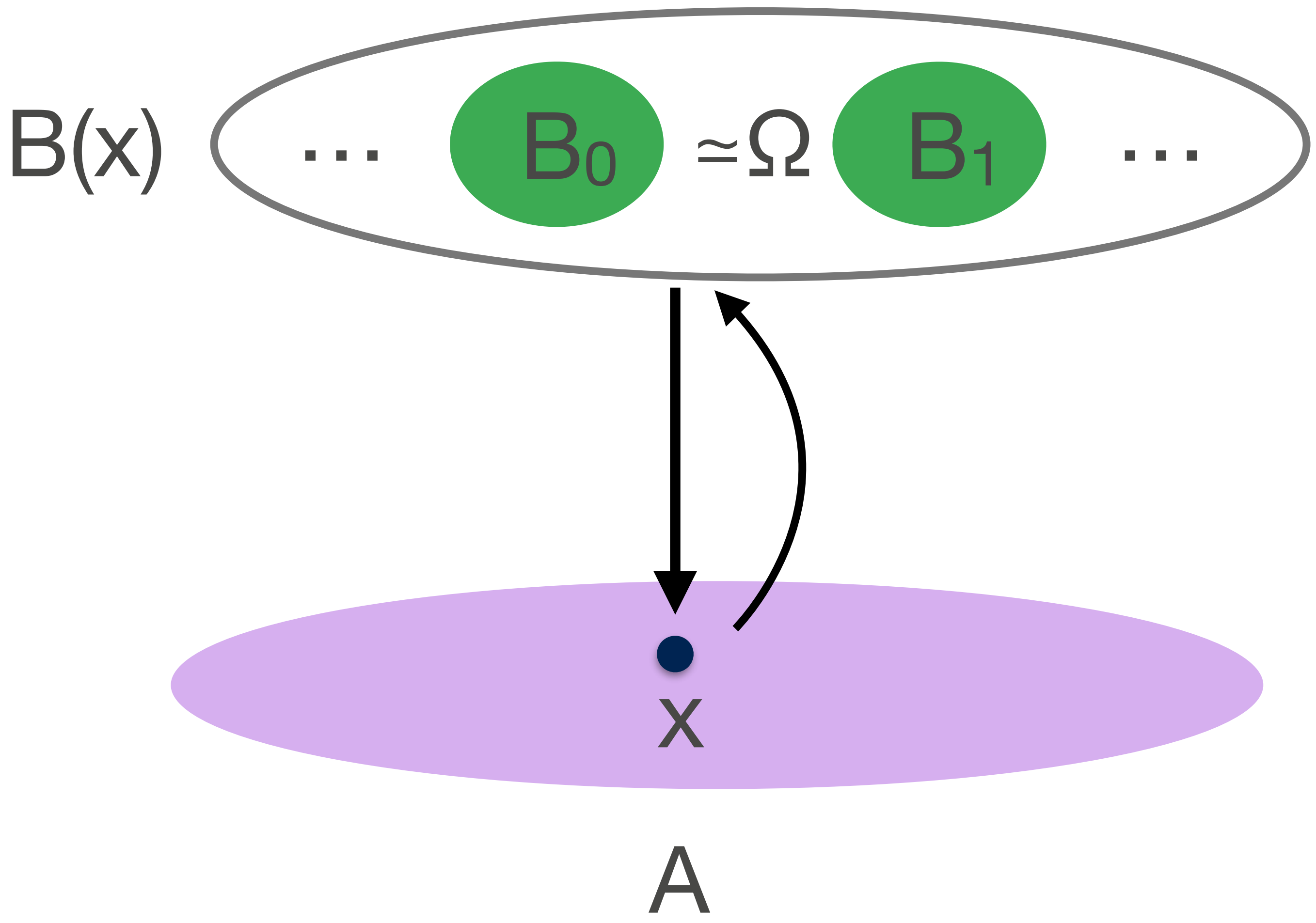
Parametrized spectra

[Finster,L.,Morehouse,Riley]



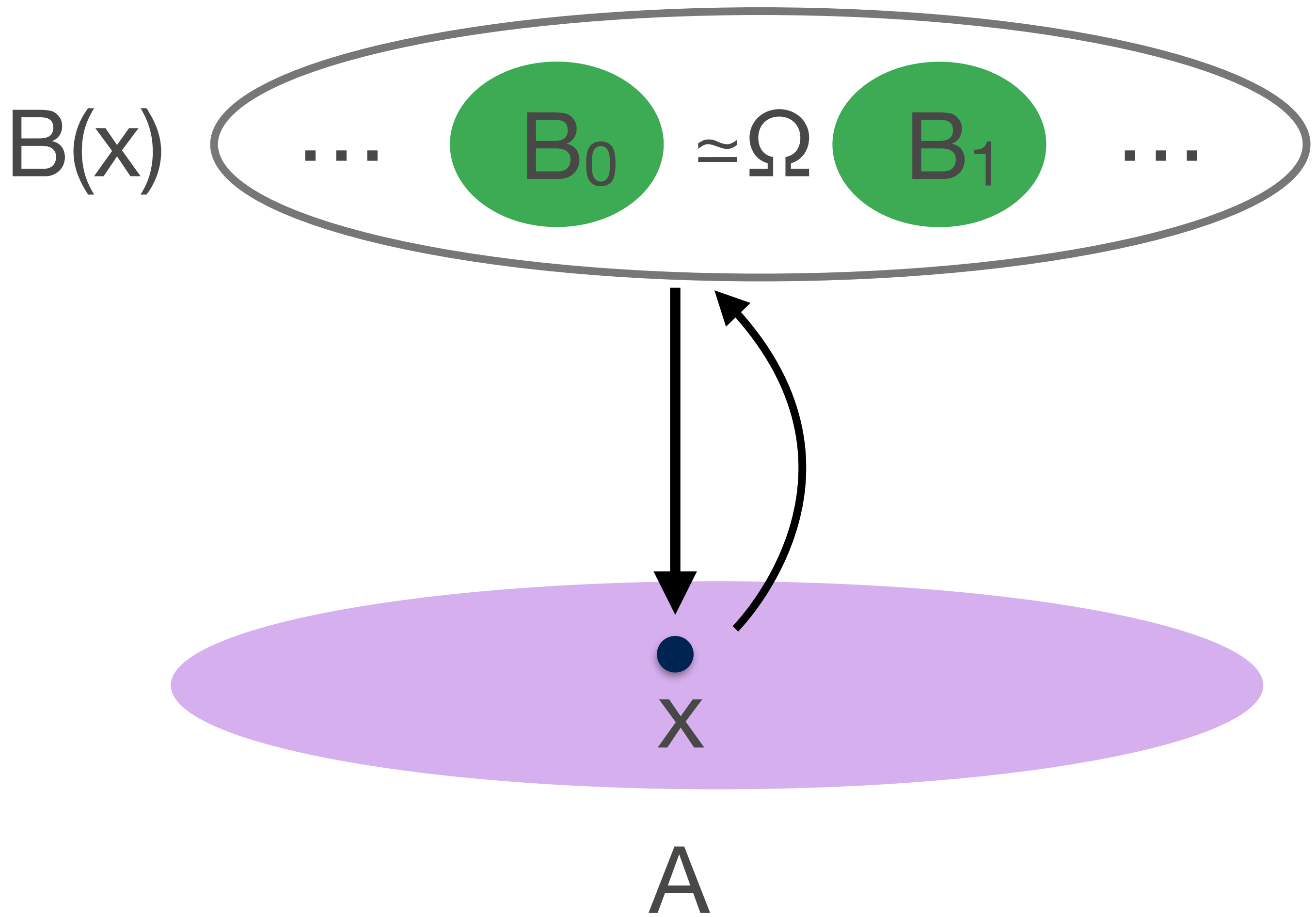
Parametrized spectra

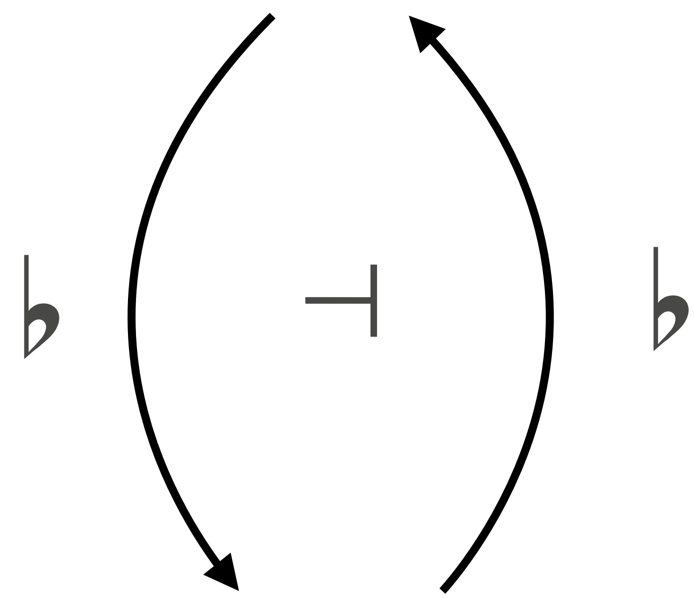
[Finster,L.,Morehouse,Riley]



Parametrized spectra

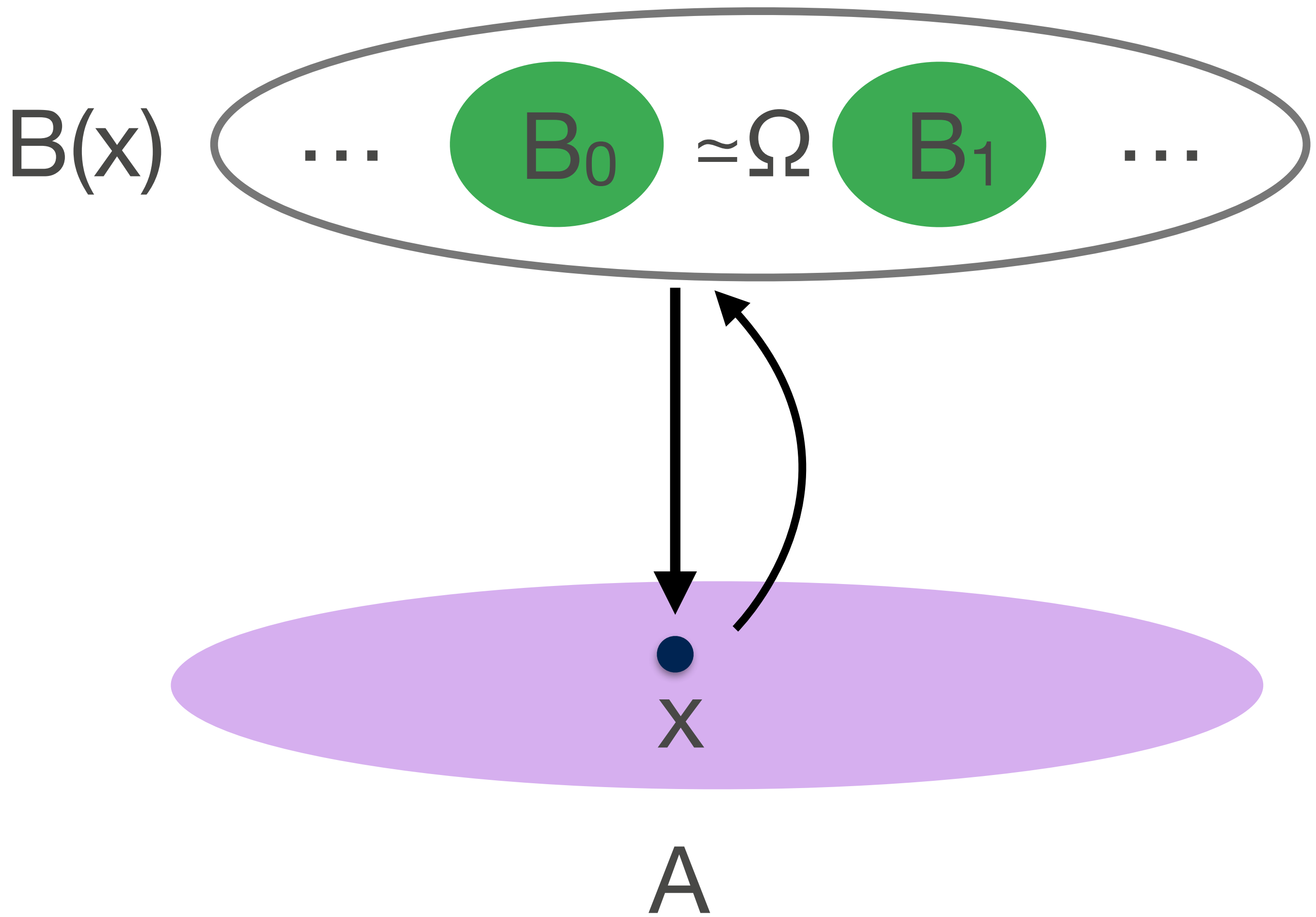
[Finster,L.,Morehouse,Riley]

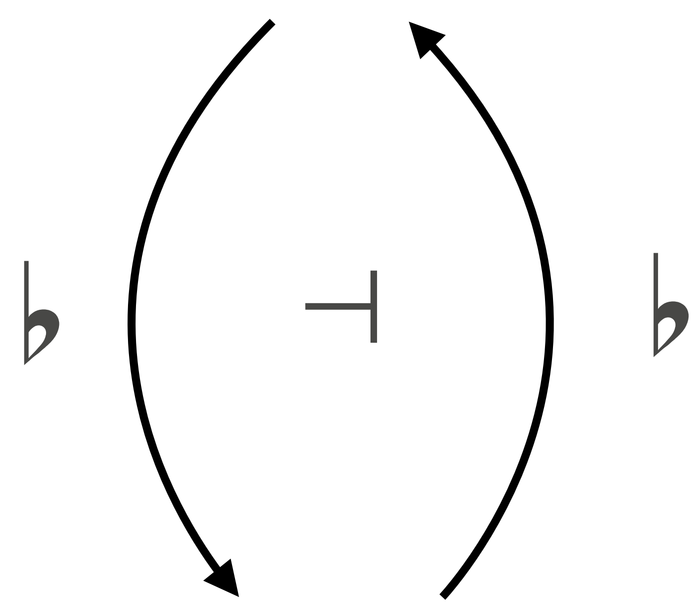




Parametrized spectra

[Finster,L.,Morehouse,Riley]

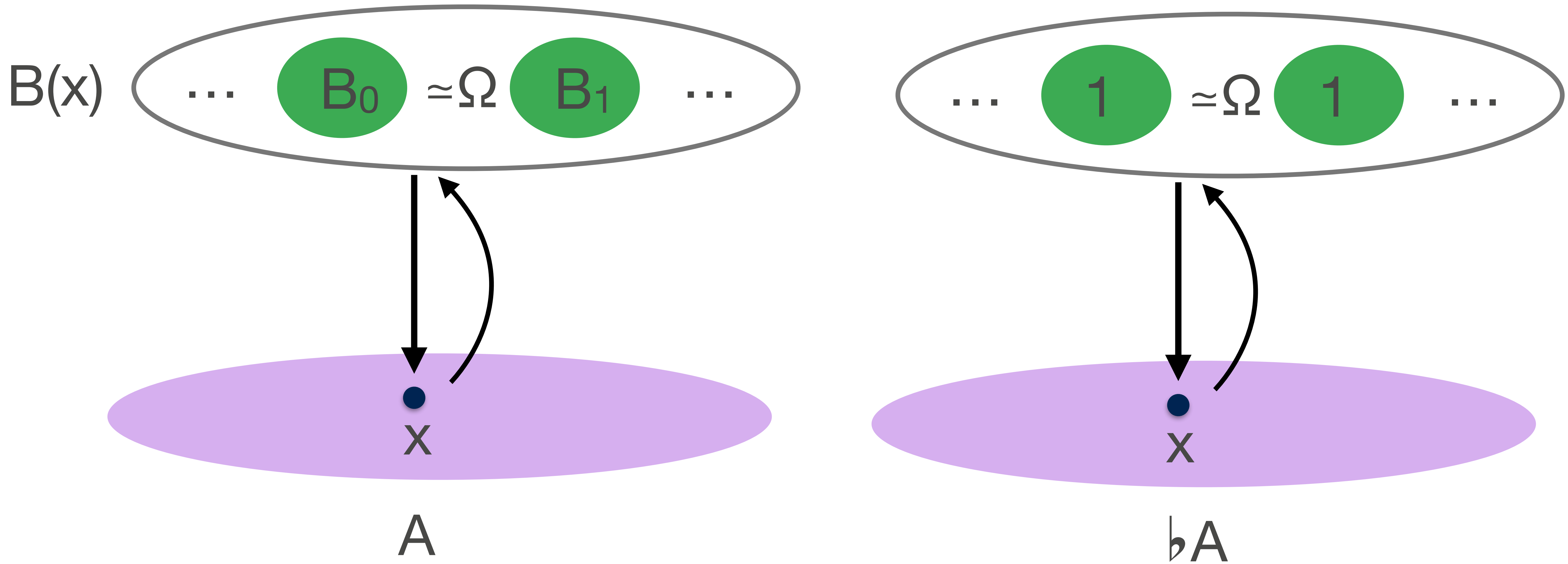




self-adjoint, idempotent
monad and comonad

Parametrized spectra

[Finster,L.,Morehouse,Riley]



Other places with cohesion

- universes in cubical models (presheaves/sets) [L.,Orton,Pitts,Spitters]
- parametricity (bridge-path cubical sets, bicubical sets) [Nuyts,Vezzosi,Devriese; Cavallo,Harper]
- bisimplicial/bicubical directed type theories [Riehl,Shulman; Riehl,Sattler; L.-Weaver; Nuyts]
- information flow security (classified sets) [Kavvos]

Other modalities

- whole area of proof theory and programming langs, mostly simply typed
- linear logic ! comonad [Girard] in dependently indexed linear logic [Vákár; Benton, Pradic, Krishnaswami]
- Squash types [Constable+], bracket types [Awodey, Bauer], contextual modal type theory [Nanevski, Pientka, Pfenning]
- Dependent right adjoints (generalizing #) [Birkedal, Clouston, Mannaa, Møgelberg, Pitts,Spitters]
- “Later” in guarded recursion [Nakano,Birkedal+]

∞-categorical Cohesion

[Schreiber,Shulman]

“Topological ∞-groupoids”

$$\Pi \dashv \Delta \dashv \Gamma \dashv \nabla$$

∞-groupoids

fundamental ∞-groupoid! e.g. $\Delta\Pi(\text{topological } \mathbb{S}^1) = \text{HIT } S^1$

Δ and ∇ full and faithful…

∞-categorical Cohesion

“Topological ∞-groupoids”

$$\int \dashv \flat \dashv \#$$

“Topological ∞-groupoids”

$\int = \Delta\Pi$

$\flat = \Delta\Gamma$ comonad

$\# = \nabla\Gamma$ monad

idempotent

Modality: historically endofunctor on types/propositions

$\square A \quad \diamond A \quad !A \quad ?A$

Cohesion in cubical models

Presheaves on C with terminal object 1

$$\Pi \dashv \Delta \dashv \Gamma \dashv \nabla$$

Sets

$\Gamma(A)$ = set of objects (A_1)

$\Delta(X)$ = constant presheaf on X

Parametricity [Nuyts,Vezzosi,Devriese]

$\mathrm{Sets}^{\mathrm{BPCube}}$

two kinds of intervals, paths and “bridges”/relations

$$\Pi \dashv \Delta \dashv \Gamma \dashv \nabla \dashv \boxminus$$

$\mathrm{Sets}^{\mathrm{Cube}}$

Bi-{simplicial,cubical} TT

[Riehl,Shulman; Riehl,Sattler; L.-Weaver; Cavallo,Harper]

$\mathrm{Sets}^{\mathrm{Cop\ x\ Dop}}$

$$\Delta \dashv \Gamma \dashv \nabla$$

forget morphisms/relations

$\mathrm{Sets}^{\mathrm{Cop}}$

$$\Delta \dashv \Gamma \dashv \nabla$$

forget paths

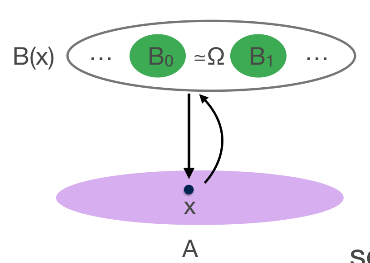
Sets

also core, opposites (self-adjoint)? [Nuyts’15]

12

Parametrized Spectra

[Finster,L.,Morehouse,Riley]



$$\flat \dashv \flat$$

self-adjoint,
monad and comonad

Differential Cohesion [Friday!]

[Scheiber; W.; Gross,L.,New,Paykin,Riley,Shulman,W.]

$$\begin{array}{ccccc} \Re & \dashv & \Im & \dashv & \& \\ & & \cup & & \cup \\ & & \int & \dashv & \flat & \dashv & \sharp \end{array}$$

Type theories with modalities

Monadic modality in Book HoTT

[Rijke,Shulman,Spitters]

Definition 7.7.5. A **modality** is an operation $\bigcirc : \mathcal{U} \to \mathcal{U}$ for which there are

(i) functions $\eta^{\bigcirc}_A : A \to \bigcirc(A)$ for every type A.

(ii) for every $A : \mathcal{U}$ and every type family $B : \bigcirc(A) \to \mathcal{U}$, a function

$$\mathsf{ind}_{\bigcirc} : \Big(\prod_{a:A} \bigcirc(B(\eta^{\bigcirc}_A(a)))\Big) \to \prod_{z:\bigcirc(A)} \bigcirc(B(z)).$$

(iii) A path $\mathsf{ind}_{\bigcirc}(f)(\eta^{\bigcirc}_A(a)) = f(a)$ for each $f : \prod_{(a:A)} \bigcirc(B(\eta^{\bigcirc}_A(a)))$.

(iv) For any $z, z' : \bigcirc(A)$, the function $\eta^{\bigcirc}_{z=z'} : (z = z') \to \bigcirc(z = z')$ is an equivalence.

Monadic modality in Book HoTT

[Rijke,Shulman,Spitters]

(idempotent, monadic)

Definition 7.7.5. A **modality** is an operation $\bigcirc : \mathcal{U} \to \mathcal{U}$ for which there are

(i) functions $\eta^{\bigcirc}_A : A \to \bigcirc(A)$ for every type A.

(ii) for every $A : \mathcal{U}$ and every type family $B : \bigcirc(A) \to \mathcal{U}$, a function

$$\mathsf{ind}_{\bigcirc} : \Big(\prod_{a:A} \bigcirc(B(\eta^{\bigcirc}_A(a)))\Big) \to \prod_{z:\bigcirc(A)} \bigcirc(B(z)).$$

(iii) A path $\mathsf{ind}_{\bigcirc}(f)(\eta^{\bigcirc}_A(a)) = f(a)$ for each $f : \prod_{(a:A)} \bigcirc(B(\eta^{\bigcirc}_A(a)))$.

(iv) For any $z, z' : \bigcirc(A)$, the function $\eta^{\bigcirc}_{z=z'} : (z = z') \to \bigcirc(z = z')$ is an equivalence.

Monadic modality in Book HoTT

[Rijke,Shulman,Spitters]

(idempotent, monadic)

Definition 7.7.5. A **modality** is an operation $\bigcirc : \mathcal{U} \to \mathcal{U}$ for which there are

(i) functions $\eta^{\bigcirc}_A : A \to \bigcirc(A)$ for every type A.

(ii) for every $A : \mathcal{U}$ and every type family $B : \bigcirc(A) \to \mathcal{U}$, a function

$$\mathsf{ind}_{\bigcirc} : \Big(\prod_{a:A} \bigcirc(B(\eta^{\bigcirc}_A(a)))\Big) \to \prod_{z:\bigcirc(A)} \bigcirc(B(z)).$$

(iii) A path $\mathsf{ind}_{\bigcirc}(f)(\eta^{\bigcirc}_A(a)) = f(a)$ for each $f : \prod_{(a:A)} \bigcirc(B(\eta^{\bigcirc}_A(a)))$.

(iv) For any $z, z' : \bigcirc(A)$, the function $\eta^{\bigcirc}_{z=z'} : (z = z') \to \bigcirc(z = z')$ is an equivalence.

works because terms $\Gamma \vdash a : A$ have many variables but one conclusion A — easy to control

Comonadic modalities

Internal definitions don't work:
need new *rules* to control use of context

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A} \qquad \frac{\Delta \mid \diamond \vdash a : A \qquad \Delta, x :: A, \Delta' \mid \Gamma \vdash b : B}{\Delta, \Delta'[a/x] \mid \Gamma[a/x] \vdash b[a/x] : B[a/x]}$$

$$\frac{\Delta \mid \cdot \vdash A : \mathsf{Type}}{\Delta \mid \Gamma \vdash \flat A : \mathsf{Type}} \qquad \frac{\Delta \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash M^{\flat} : \flat A}$$

[Pfenning-Davies; dependent/idempotent version by Shulman]

Monadic modalities via new rules

[Shulman]

Define $\flat \dashv \sharp$, $\flat\sharp A \simeq \flat A$
then can prove it satisfies modality axioms

$$\frac{\Delta, \Gamma \mid \cdot \vdash A : \mathsf{Type}}{\Delta \mid \Gamma \vdash \sharp A : \mathsf{Type}}$$

$$\frac{\Delta, \Gamma \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash M^{\sharp} : \sharp A} \qquad \frac{\Delta \mid \cdot \vdash M : \sharp A}{\Delta \mid \Gamma \vdash M_{\sharp} : A}$$

Monadic modalities not via rules

[Shulman]

In real-cohesive HoTT, shape $\int A$ is nullification:
monadic modality for $\mathbb{R}$-null types $A \simeq (\mathbb{R} \to A)$

defined modalities: $\int$, truncation, …
judgemental modalities: $\flat$, $\sharp$

Substructural/Modal Logics

- Multiple kinds of assumptions/multi-zoned contexts: Andreoli'92; Wadler'93; Plotkin'93;Barber'96; Benton'94; Pfenning,Davies'01
- Tree-structured contexts:
  Display logic: Belnap
  Bunched contexts: O'Hearn,Pym'99,
  Resource separation: Atkey,'04
- Multiple modes: Benton'94; Benton,Wadler'96, Reed'09
- Fibrational perspective: Melliès,Zeilberger'15

Substructural/Modal T.T.

1. Add a new form of judgement for left adjoints
2. Left adjoint types have a left universal property relative to that judgement
3. Right adjoint types have a right universal property relative to that judgement
4. Structural rules are equations, natural isomorphisms, or natural transformations between contexts
5. Optimize placement of structural rules

Monoidal Product

1. Add a new form of judgement for left adjoints
2. Left adjoint types have a left universal property relative to that judgement
3. Right adjoint types have a right universal property relative to that judgement
4. Structural rules are equations, natural isomorphisms, or natural transformations between contexts
5. Optimize placement of structural rules

Monoidal Product

new judgement: the context Γ,Γ’

left adjoint:

right adjoint:

Monoidal Product

new judgement: the context Γ,Γ’

left adjoint:

$$\frac{\Gamma[A,B] \vdash C}{\Gamma[A \otimes B] \vdash C}$$

right adjoint:

Monoidal Product

new judgement: the context Γ,Γ'

left adjoint:

$$\frac{\frac{\Gamma[A,B] \vdash C}{}}{\Gamma[A \otimes B] \vdash C}$$

right adjoint:

$$\frac{\frac{\Gamma, A \vdash B}{}}{\Gamma \vdash A \multimap B}$$

Structural rules

If , is associative then ⊗ is

$$\frac{}{A \otimes (B \otimes C) \vdash (A \otimes B) \otimes C}$$

Structural rules

If , is associative then ⊗ is

$$\frac{\overline{A,(B\otimes C) \vdash (A\otimes B)\otimes C}}{A\otimes(B\otimes C) \vdash (A\otimes B)\otimes C}$$

Structural rules

If , is associative then ⊗ is

$$\dfrac{\dfrac{}{A,(B,C) \vdash (A\otimes B)\otimes C}}{\dfrac{A,(B\otimes C) \vdash (A\otimes B)\otimes C}{A\otimes(B\otimes C) \vdash (A\otimes B)\otimes C}}$$

Structural rules

If , is associative then $\otimes$ is

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(A,B),C \vdash (A\otimes B)\otimes C}}{A,(B,C) \vdash (A\otimes B)\otimes C}}{A,(B\otimes C) \vdash (A\otimes B)\otimes C}}{A\otimes(B\otimes C) \vdash (A\otimes B)\otimes C}$$

Structural rules

If , is associative then ⊗ is

$$\dfrac{\dfrac{\dfrac{\dfrac{A,B \vdash A \otimes B \qquad C \vdash C}{(A,B),C \vdash (A\otimes B)\otimes C}}{A,(B,C) \vdash (A\otimes B)\otimes C}}{A,(B\otimes C) \vdash (A\otimes B)\otimes C}}{A\otimes(B\otimes C) \vdash (A\otimes B)\otimes C}$$

Structural rules

If , is associative then ⊗ is

$$\dfrac{\dfrac{\dfrac{\dfrac{A,B \vdash A \otimes B \qquad C \vdash C}{(A,B),C \vdash (A\otimes B)\otimes C}}{A,(B,C) \vdash (A\otimes B)\otimes C}}{A,(B\otimes C) \vdash (A\otimes B)\otimes C}}{A\otimes(B\otimes C) \vdash (A\otimes B)\otimes C}$$

equality or isomorphism

Optimization

Pick a canonical associativity, and build re-associating into the other rules

basic

$$\frac{\Gamma,(A,B) \vdash C}{\Gamma,A\otimes B \vdash C}$$

optimized

$$\frac{\dfrac{(\Gamma,A),B \vdash C}{\Gamma,(A,B) \vdash C}}{\Gamma,A\otimes B \vdash C}$$

Cartesian Product (Positive)

1. Add a new form of judgement for left adjoints
2. Left adjoint types have a left universal property relative to that judgement
3. Right adjoint types have a right universal property relative to that judgement
4. **Structural rules are equations, natural isos, or natural transformations between contexts**
5. **Optimize placement of structural rules**

Cartesian Product

$$\frac{}{x : A \vdash x : A}$$

$$\frac{\Gamma \vdash a : A \qquad \Delta \vdash b : B}{\Gamma, \Delta \vdash (a,b) : A \times B}$$

$$\frac{}{\Gamma \vdash w : \varnothing} \qquad \frac{}{\Gamma \vdash c : \Gamma, \Gamma}$$

“Optimized” Rules

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a,b) : A \times B}$$

“Optimized” Rules

$$\frac{}{\Gamma, x : A \vdash x : A}$$

can weaken at the leaves

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a,b) : A \times B}$$

“Optimized” Rules

$$\frac{}{\Gamma, x : A \vdash x : A}$$

can weaken at the leaves

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a,b) : A \times B}$$

so might as well always contract

$\flat$ in spatial type theory

1. Add a new form of judgement for left adjoints
2. Left adjoint types have a left universal property relative to that judgement
3. Right adjoint types have a right universal property relative to that judgement
4. Structural rules are equations, natural isos, or natural transformations between contexts
5. Optimize placement of structural rules

Simply-typed ♭

new judgement: the context **f**(Γ)

left adjoint:

right adjoint:

Simply-typed ♭

new judgement: the context $\mathbf{f}(\Gamma)$

left adjoint:

$$\frac{\mathbf{f}\,\Gamma \vdash A}{\flat\Gamma \vdash A}$$

right adjoint:

Simply-typed ♭

new judgement: the context $\mathbf{f}(\Gamma)$

left adjoint:

$$\frac{\frac{\mathbf{f}\,\Gamma \vdash A}{}}{\flat\Gamma \vdash A}$$

right adjoint:

$$\frac{\frac{\mathbf{f}\,\Gamma \vdash A}{}}{\Gamma \vdash \sharp A}$$

Simply-typed ♭

structural rules for idempotent comonad:

counit: $\flat\,\Gamma \vdash \Gamma$

comult: $\flat\,\Gamma \simeq \flat\,\flat\,\Gamma$

Optimization

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

Optimization

pick canonical “associativity” of contexts: placement of f

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

Optimization

pick canonical "associativity" of contexts: placement of f

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

$\mathbf{f}(\Delta, A, \Delta'), \Gamma \vdash \mathbf{f}(A) \vdash A$

Optimization

pick canonical "associativity" of contexts: placement of f

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

f(projection)

$$\mathbf{f}(\Delta, A, \Delta'), \Gamma \vdash \mathbf{f}(A) \vdash A$$

Optimization

pick canonical “associativity” of contexts: placement of f

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

f(projection) counit

f(Δ,A,Δ’),Γ ⊢ **f**(A) ⊢ A

Optimization

pick canonical "associativity" of contexts: placement of f

$$\overline{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

f(projection) counit

f(Δ,A,Δ'),Γ ⊢ **f**(A) ⊢ A

$$\frac{\Delta \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash M^{\flat} : \flat A}$$

might as well use comult, because you can counit Δ later if you need to

Optimization

pick canonical "associativity" of contexts: placement of f

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

f(projection)   counit

f(Δ,A,Δ'),Γ ⊢ **f**(A) ⊢ A

$$\frac{\Delta \mid \cdot \vdash M : A}{\Delta \mid \Gamma \vdash M^{\flat} : \flat A}$$

might as well use comult, because you can counit Δ later if you need to

[same placement as comonoid for ×]

Pattern

1. Add a new form of judgement for left adjoints
2. Left adjoint types have a left universal property relative to that judgement
3. Right adjoint types have a right universal property relative to that judgement
4. Structural rules are equations, natural isos, or natural transformations between contexts
5. Optimize placement of structural rules

Only part of the story…

- Structural rules for interaction of modalities (e.g. **f**(Δ,Δ') vs. **f**(Δ),**f**(Δ'))
- Rules for dependency

$$\frac{\Delta \mid \cdot \vdash A : \mathsf{Type}}{\Delta \mid \Gamma \vdash \flat A : \mathsf{Type}} \qquad \frac{\Delta, \Gamma \mid \cdot \vdash A : \mathsf{Type}}{\Delta \mid \Gamma \vdash \sharp A : \mathsf{Type}}$$

- Interaction with identity types, inductive types, HITs
- Universes
- Stability under substitution
- Fibrancy

Pattern to Framework

Fibrational Framework

- A Judgemental Deconstruction of Modal Logic [Reed'09]
- Adjoint Logic with a 2-Category of Modes [L.Shulman'16]
- A Fibrational Framework for Substructural and Modal Logics [L.,Shulman,Riley,'17]
- A Fibrational Framework for Substructural and Modal Dependent Type Theories [L.,Riley,Shulman, in progress]

Logical Framework

[Martin-Löf; Harper,Honsell,Plotkin]

a type theory where other type theories
are specified by **signatures**

Logical Framework

[Martin-Löf; Harper,Honsell,Plotkin]

a type theory where other type theories are specified by **signatures**

- implement one proof assistant for a number of type theories

Logical Framework

[Martin-Löf; Harper,Honsell,Plotkin]

a type theory where other type theories
are specified by **signatures**

- implement one proof assistant for
  a number of type theories
- semantics: prove initiality for
  a class of type theories at once

Goals for Modal Framework

Goals for Modal Framework

- covers lots of examples

Goals for Modal Framework

- covers lots of examples
- easy to go from intended semantics to a signature

Goals for Modal Framework

- covers lots of examples
- easy to go from intended semantics to a signature
- automatically get type theoretic rules
  (but with explicit structural rules)

Goals for Modal Framework

- covers lots of examples
- easy to go from intended semantics to a signature
- automatically get type theoretic rules
  (but with explicit structural rules)
- can derive “optimized” rules (requires cleverness)

Goals for Modal Framework

- covers lots of examples
- easy to go from intended semantics to a signature
- automatically get type theoretic rules
  (but with explicit structural rules)
- can derive “optimized” rules (requires cleverness)
- categorical semantics for whole framework at once

Goals for Modal Framework

- covers lots of examples
- easy to go from intended semantics to a signature
- automatically get type theoretic rules
  (but with explicit structural rules)
- can derive “optimized” rules (requires cleverness)
- categorical semantics for whole framework at once
- expected structures are models of signatures

Goals for Modal Framework

- covers lots of examples
- easy to go from intended semantics to a signature
- automatically get type theoretic rules
  (but with explicit structural rules)
- can derive “optimized” rules (requires cleverness)
- categorical semantics for whole framework at once
- expected structures are models of signatures
- proof assistant with enough automation
  to make it convenient

Fibrational Framework

$$\begin{tikzcd} \mathcal{D} \arrow[d, "\pi"] \\ \mathcal{M} \end{tikzcd}$$

$\mathcal{D}$ top type theory — types and terms of a modal type theory

$\mathcal{M}$ base type theory — algebraic syntax for judgements

Fibrational Framework

$\mathcal{D}$ and $\mathcal{M}$ both dependent type theories

$$
\begin{tikzcd}
\mathcal{D} \arrow[d, "\pi"] & \vdash_{\gamma} \Gamma \ \mathrm{ctx} \arrow[d] & \Gamma \vdash_{p} A \ \mathrm{type} \arrow[d] & \Gamma \vdash_{f} a : A \arrow[d] \\
\mathcal{M} & \gamma \ \mathrm{ctx} & \gamma \vdash p \ \mathrm{mode} & \gamma \vdash f : p
\end{tikzcd}
$$

Fibrational Framework

top $x{:}A, y{:}B, z{:}C \vdash_f d : D$

base $x{:}p,\ y{:}q,\ z{:}r \vdash f : s$

Fibrational Framework

framework contexts are both standard:
not modal or substructural

top $\quad x{:}A, y{:}B, z{:}C \vdash_f d : D$

base $\quad x{:}p,\ y{:}q,\ z{:}r \vdash f : s$

Fibrational Framework

framework contexts are both standard:
not modal or substructural

top $\quad x{:}A, y{:}B, z{:}C \vdash_f d : D$

base $\quad x{:}p,\ y{:}q,\ z{:}r \vdash f : s$

base term f represents
the modal structure of the context

A semantic intuition (non-dependent)

$$\begin{tikzcd} \mathcal{D} \arrow[d, "\pi"] \\ \mathcal{M} \end{tikzcd}$$

$\vdash_\gamma \Gamma\ \mathrm{ctx}$	objects of γ
$\Gamma \vdash_f a : \Delta$	maps $f(\Gamma) \to_\delta \Delta$
$\gamma\ \mathrm{ctx}$	categories
$\gamma \vdash f : \delta$	functors
$\gamma \vdash s : f \Rightarrow_\delta g$	natural trans.

Pattern

1. **Judgement for left adjoint: modes and mode terms**
2. Left adjoint types have a left universal property relative to that judgement
3. Right adjoint types have a right universal property relative to that judgement
4. **Structural rules: 2-cells between mode terms**
5. Optimize placement of structural rules

SpatialTT

$$\begin{tikzcd} \mathbf{p} \arrow[d, bend left=60, "\sharp"] \arrow[d, phantom, "\dashv"] \\ \mathbf{p} \arrow[u, bend left=60, "\flat"] \end{tikzcd}$$

$\flat$ idem comonad

$\sharp$ idem monad

SpatialTT

$$\begin{tikzcd} \mathbf{p} \arrow[d, bend left, "\sharp"] \arrow[d, phantom, "\dashv"] \\ \mathbf{p} \arrow[u, bend left, "\flat"] \end{tikzcd}$$

$\flat$ idem comonad

$\sharp$ idem monad

Mode theory

p mode	category
x:**p** ⊢ **f**(x) : **p**	functor
counit : x:**p** ⊢ **f**(x) ⟹ x	nat. trans
comult : x:**p** ⊢ **f**(x) ⇒ **f f**(x)	
… + equations	

A semantic intuition (dependent)

$$\begin{tikzcd} \mathcal{D} \arrow[d, "\pi"] \\ \mathcal{M} \end{tikzcd}$$

$\Gamma \vdash_p A \text{ type}$	objects of $p(\Gamma)$
$\Gamma \vdash_f a : A$	maps $f(\Gamma) \to_{p(\Gamma)} A$

$\gamma \vdash p \text{ mode}$	functors $\gamma^{op} \to \mathrm{Cat}$
$\gamma \vdash f : p$	sections of $\int p \to \gamma$
$\gamma \vdash s : f \Rightarrow_{\mathbf{p}} g$	natural trans. over id

Dependent Contexts

mode $\mathbf{p}$ “contexts”

mode $a{:}\mathbf{p} \vdash \mathbf{T}(a)$ “types” in context a

2-cells $a \Rightarrow_{\mathbf{p}} b$ “substitutions”

Dependent Contexts

mode $\mathbf{p}$	“contexts”
mode $a{:}\mathbf{p} \vdash \mathbf{T}(a)$	“types” in context a
2-cells $a \Rightarrow_{\mathbf{p}} b$	“substitutions”

mode term $\quad a{:}\mathbf{p},\ x{:}\mathbf{T}(a) \vdash a\mathbf{.}x : \mathbf{p}$

Dependent Contexts

mode $\mathbf{p}$ "contexts"

mode $a{:}\mathbf{p} \vdash \mathbf{T}(a)$ "types" in context a

2-cells $a \Rightarrow_{\mathbf{p}} b$ "substitutions"

mode term $a{:}\mathbf{p}, x{:}\mathbf{T}(a) \vdash a.x : \mathbf{p}$

mode 2-cell $a{:}\mathbf{p}, x{:}\mathbf{T}(a) \vdash \boldsymbol{\pi} : a.x \Rightarrow_{\mathbf{p}} a$

Dependent Contexts

mode $\mathbf{p}$ "contexts"

mode $a{:}\mathbf{p} \vdash \mathbf{T}(a)$ "types" in context a

2-cells $a \Rightarrow_{\mathbf{p}} b$ "substitutions"

mode term $a{:}\mathbf{p}, x{:}\mathbf{T}(a) \vdash a.x : \mathbf{p}$

mode 2-cell $a{:}\mathbf{p}, x{:}\mathbf{T}(a) \vdash \boldsymbol{\pi} : a.x \Rightarrow_{\mathbf{p}} a$

…

Dependent Contexts

mode $\mathbf{p}$ "contexts"

mode $a{:}\mathbf{p} \vdash \mathbf{T}(a)$ "types" in context a

2-cell $a \Rightarrow_{\mathbf{p}} b$ "substitutions"

A **comprehension object** on $(\mathbf{p},\mathbf{T})$ has

mode term $a{:}\mathbf{p} \vdash \mathbf{1}_a : \mathbf{T}(a)$

such that $a{:}\mathbf{p} \vdash (a,\mathbf{1}_a) : (a : \mathbf{p}, \mathbf{T}(a))$

has a right adjoint

[Lawvere,Ehrhard]

Dependent Contexts

mode $\mathbf{p}$ "contexts"

mode $a{:}\mathbf{p} \vdash \mathbf{T}(a)$ "types" in context a

2-cell $a \Rightarrow_{\mathbf{p}} b$ "substitutions"

A **comprehension object** on $(\mathbf{p},\mathbf{T})$ has

mode term $a{:}\mathbf{p} \vdash \mathbf{1}_a : \mathbf{T}(a)$

such that $a{:}\mathbf{p} \vdash (a,\mathbf{1}_a) : (a : \mathbf{p}, \mathbf{T}(a))$

has a right adjoint

unit/counit mode term 2-cells

[Lawvere,Ehrhard]

Dependent Contexts

$a \Rightarrow_{\mathbf{p}} b.x \;\cong\; s : a \Rightarrow_{\mathbf{p}} b$ and $t : \mathbf{1}_a \Rightarrow_{\mathbf{T}(a)} s^+ x$

A **comprehension object** on (**p**,**T**) has

mode term $a{:}\mathbf{p} \vdash \mathbf{1}_a : \mathbf{T}(a)$

such that $a{:}\mathbf{p} \vdash (a, \mathbf{1}_a) : (a : \mathbf{p}, \mathbf{T}(a))$

has a right adjoint

unit/counit mode term 2-cells

[Lawvere,Ehrhard]

Dependent Contexts

framework level transport that represents "substitution"
(mode types are functors $\gamma^{op} \to$ Cat)

$$a \Rightarrow_{\mathbf{p}} b.x \;\cong\; s : a \Rightarrow_{\mathbf{p}} b \text{ and } t : \mathbf{1}_a \Rightarrow_{\mathbf{T}(a)} s^+ x$$

A **comprehension object** on (**p**,**T**) has

mode term $a{:}\mathbf{p} \vdash \mathbf{1}_a : \mathbf{T}(a)$

such that $a{:}\mathbf{p} \vdash (a, \mathbf{1}_a) : (a : \mathbf{p}, \mathbf{T}(a))$

has a right adjoint

unit/counit mode term 2-cells

[Lawvere,Ehrhard]

Dependent Contexts

$$
\begin{tikzcd}
\mathrm{T}{:=}(a{:}\mathbf{p},\, \mathbf{T(a)}) \arrow[rr] \arrow[dr, "\mathrm{fst}"'] & & \mathbf{p}^{\rightarrow} \arrow[dl, "\mathrm{cod}"] \\
& \mathbf{p} &
\end{tikzcd}
$$

comprehension induces
comprehension category [Jacobs]

full comprehension category with
1 terminal has comprehension

Dependent Contexts

$$\begin{tikzcd}
T{:=}(a{:}\mathbf{p},\, \mathbf{T(a)}) \arrow[rr] \arrow[dr, "\mathrm{fst}"'] & & \mathbf{p}^{\rightarrow} \arrow[dl, "\mathrm{cod}"] \\
& \mathbf{p} &
\end{tikzcd}$$

if full and **1** terminal, then
maps in the fiber $1_a \Rightarrow_{T(a)} x$

$\cong$ $a \Rightarrow_p a.x$ sections of $\pi : a.x \Rightarrow a$

Encodings

MLTT $\quad x{:}A, y{:}B, z{:}C \vdash d : D$

$\mathcal{D}$

$\mathcal{M}$

Encodings

MLTT $\qquad x{:}A, y{:}B, z{:}C \vdash d : D$

$\mathcal{D} \qquad x{:}A, \qquad y{:}B, \qquad z{:}C \qquad \vdash_{\mathbf{1}} \qquad d : D$

$\mathcal{M}$

Encodings

MLTT $\qquad x{:}A, y{:}B, z{:}C \vdash d : D$

$\mathcal{D} \qquad x{:}A, \quad y{:}B, \quad z{:}C \quad \vdash_{\mathbf{1}} \quad d : D$

$\mathcal{M} \qquad x{:}\mathbf{T}(\varnothing), y{:}\mathbf{T}(\varnothing.x), z{:}\mathbf{T}(\varnothing.x.y) \vdash \mathbf{1}_{\varnothing.x.y.z} : \mathbf{T}(\varnothing.x.y.z)$

Encodings

MLTT $x{:}A, y{:}B, z{:}C \vdash d : D$

$\mathcal{D}$ $x{:}A,\quad y{:}B,\quad z{:}C \quad \vdash_{\mathbf{1}} \quad d : D$

$\mathcal{M}$ $x{:}\mathbf{T}(\varnothing), y{:}\mathbf{T}(\varnothing.x), z{:}\mathbf{T}(\varnothing.x.y) \vdash \mathbf{1}_{\varnothing.x.y.z} : \mathbf{T}(\varnothing.x.y.z)$

- A depends on nothing ($\varnothing$ terminal in **p**)

Encodings

MLTT $x{:}A, y{:}B, z{:}C \vdash d : D$

$\mathcal{D}$ $x{:}A,\quad y{:}B,\quad z{:}C \quad \vdash_{\mathbf{1}} \quad d : D$

$\mathcal{M}$ $x{:}\mathbf{T}(\varnothing), y{:}\mathbf{T}(\varnothing.x), z{:}\mathbf{T}(\varnothing.x.y) \vdash \mathbf{1}_{\varnothing.x.y.z} : \mathbf{T(}\varnothing.x.y.z\mathbf{)}$

- A depends on nothing ($\varnothing$ terminal in **p**)
- B depends on x

Encodings

MLTT $\quad x{:}A, y{:}B, z{:}C \vdash d : D$

$\mathcal{D} \quad x{:}A, \quad y{:}B, \quad z{:}C \quad \vdash_{\mathbf{1}} \quad d : D$

$\mathcal{M} \quad x{:}\mathbf{T}(\varnothing), y{:}\mathbf{T}(\varnothing.x), z{:}\mathbf{T}(\varnothing.x.y) \vdash \mathbf{1}_{\varnothing.x.y.z} : \mathbf{T}(\varnothing.x.y.z)$

- A depends on nothing ($\varnothing$ terminal in **p**)
- B depends on x
- C depends on x and y

Encodings

MLTT $\quad x{:}A, y{:}B, z{:}C \vdash d : D$

$\mathcal{D} \quad x{:}A, \quad y{:}B, \quad z{:}C \quad \vdash_{\mathbf{1}} \quad d : D$

$\mathcal{M} \quad x{:}\mathbf{T}(\varnothing), y{:}\mathbf{T}(\varnothing.x), z{:}\mathbf{T}(\varnothing.x.y) \vdash \mathbf{1}_{\varnothing.x.y.z} : \mathbf{T}(\varnothing.x.y.z)$

- A depends on nothing ($\varnothing$ terminal in **p**)
- B depends on x
- C depends on x and y
- so does c (over **1** b/c $\Gamma \vdash a : A$ where $\Gamma \vdash A$ type)

Encodings

SpatialTT $\qquad x{:}A \mid y{:}B \vdash c : C$

$\mathcal{D}$

$\boldsymbol{\mathcal{M}}$

Encodings

SpatialTT $\quad x{:}A \mid y{:}B \vdash c : C$

$\mathcal{D} \quad x{:}A, \quad y{:}B \quad \vdash_1 c : C$

$\mathcal{M}$

Encodings

SpatialTT $\quad x{:}A \mid y{:}B \vdash c : C$

$\mathcal{D} \quad x{:}A, \quad y{:}B \quad \vdash_1 c : C$

$\mathcal{M} \quad x{:}\mathbf{T}(\mathbf{f}(\varnothing)), y{:}\mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x)) \vdash 1 \quad : \mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x)\ .\ y)$

Encodings

SpatialTT $\quad x{:}A \mid y{:}B \vdash c : C$

$\mathcal{D} \quad x{:}A, \quad y{:}B \quad \vdash_1 c : C$

$\mathcal{M} \quad x{:}\mathbf{T}(\mathbf{f}(\varnothing)), y{:}\mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x)) \vdash 1 \quad : \mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x) \,.\, y)$

- A depends on nothing ("crisply"/flatly)

Encodings

SpatialTT $\quad x{:}A \mid y{:}B \vdash c : C$

$\mathcal{D} \quad x{:}A, \quad y{:}B \quad \vdash_1 c : C$

$\mathcal{M} \quad x{:}\mathbf{T}(\mathbf{f}(\emptyset)), y{:}\mathbf{T}(\mathbf{f}(\mathbf{f}(\emptyset).x)) \vdash 1 \quad : \mathbf{T}(\mathbf{f}(\mathbf{f}(\emptyset).x) \,.\, y)$

- A depends on nothing (“crisply”/flatly)
- B depends on x crisply

Encodings

SpatialTT $\quad x{:}A \mid y{:}B \vdash c : C$

$\mathcal{D} \quad x{:}A, \quad y{:}B \quad \vdash_1 c : C$

$\mathcal{M} \quad x{:}\mathbf{T}(\mathbf{f}(\varnothing)), y{:}\mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x)) \vdash 1 \quad : \mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x)\ .\ y)$

- A depends on nothing ("crisply"/flatly)
- B depends on x crisply
- C depends on x crisply and y normally/cohesively

Encodings

SpatialTT $\quad x{:}A \mid y{:}B \vdash \; c : C$

$\mathcal{D} \quad x{:}A, \qquad y{:}B \qquad\qquad \vdash_1 c : C$

$\mathcal{M} \quad x{:}\mathbf{T}(\mathbf{f}(\varnothing)), y{:}\mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x)) \vdash 1 \quad : \mathbf{T}(\mathbf{f}(\mathbf{f}(\varnothing).x)\;.\;y)$

- A depends on nothing ("crisply"/flatly)
- B depends on x crisply
- C depends on x crisply and y normally/cohesively
- so does c

Pattern

1. **Judgement for left adjoint: modes and mode terms**
2. Left adjoint types have a left universal property relative to that judgement
3. Right adjoint types have a right universal property relative to that judgement
4. **Structural rules: 2-cells between mode terms**
5. Optimize placement of structural rules

Pattern

1. Judgement for left adjoint: modes and mode terms
2. **Left adjoint types have a left universal property relative to that judgement**
3. **Right adjoint types have a right universal property relative to that judgement**
4. Structural rules: 2-cells between mode terms
5. Optimize placement of structural rules

Cat semantics (non-dependent)

$$\begin{tikzcd} \mathcal{D} \arrow[d, "\pi"] \\ \mathcal{M} \end{tikzcd}$$

$$\begin{array}{ll}
\vdash_{\gamma} \Gamma\ \mathrm{ctx} & \text{objects of } \gamma \\
\Gamma \vdash_{f} a : \Delta & \text{maps } f(\Gamma) \to_{\delta} \Delta \\
\hline
\gamma\ \mathrm{ctx} & \text{categories} \\
\gamma \vdash f : \delta & \text{functors} \\
\gamma \vdash s : f \Rightarrow_{\delta} g & \text{natural trans.}
\end{array}$$

Fibrational Generalization

$$\begin{tikzcd} \mathcal{D} & \text{types and terms} \\ \mathcal{M} & \text{2-category} \arrow[from=1-1, to=2-1, "\pi"] \end{tikzcd}$$

$\mathcal{D}$ types and terms

π local discrete fib.

$\mathcal{M}$ 2-category

Fibrational Generalization

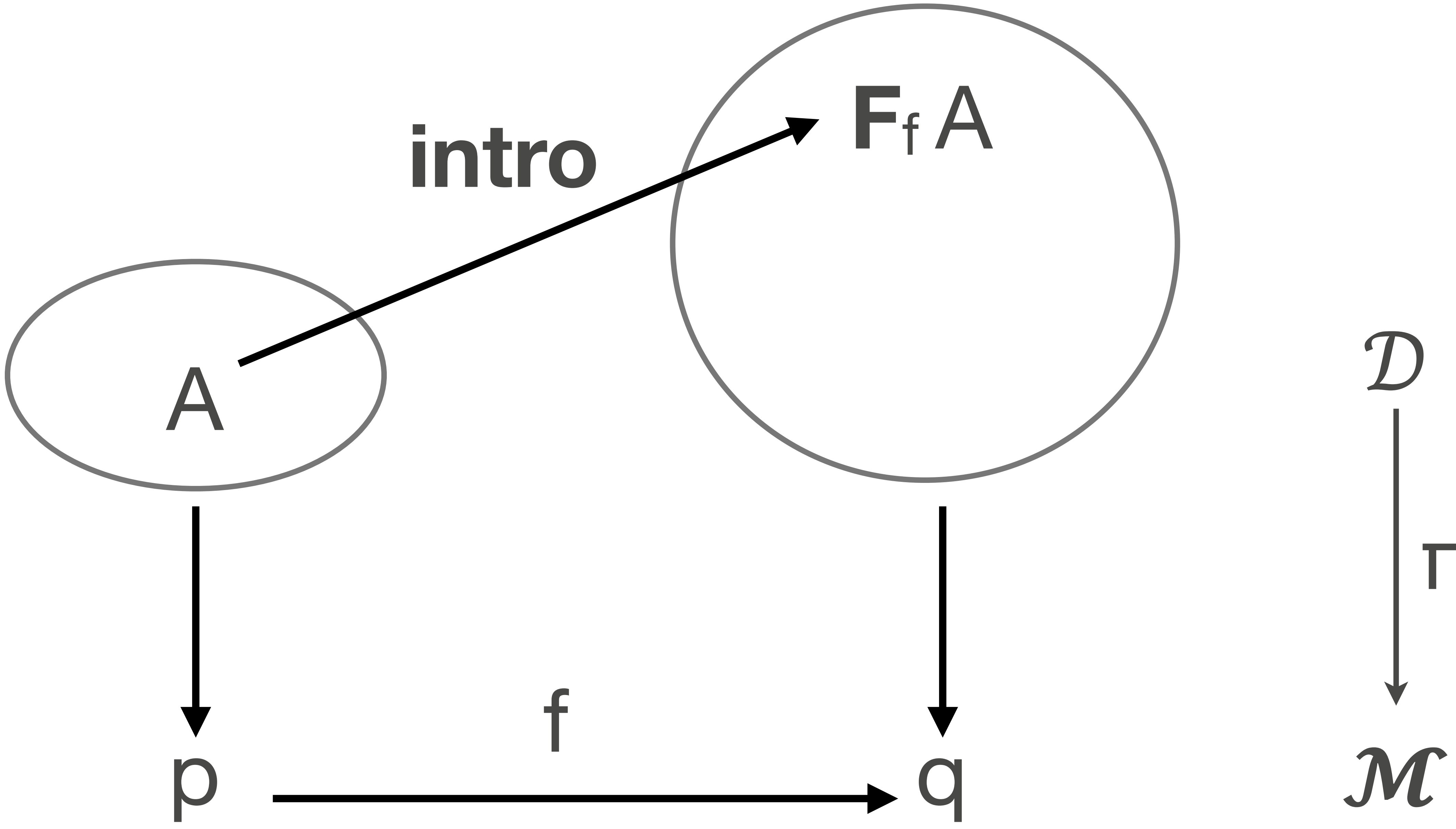
$\mathcal{D}$ types and terms

$\big\downarrow \pi$ local discrete fib.

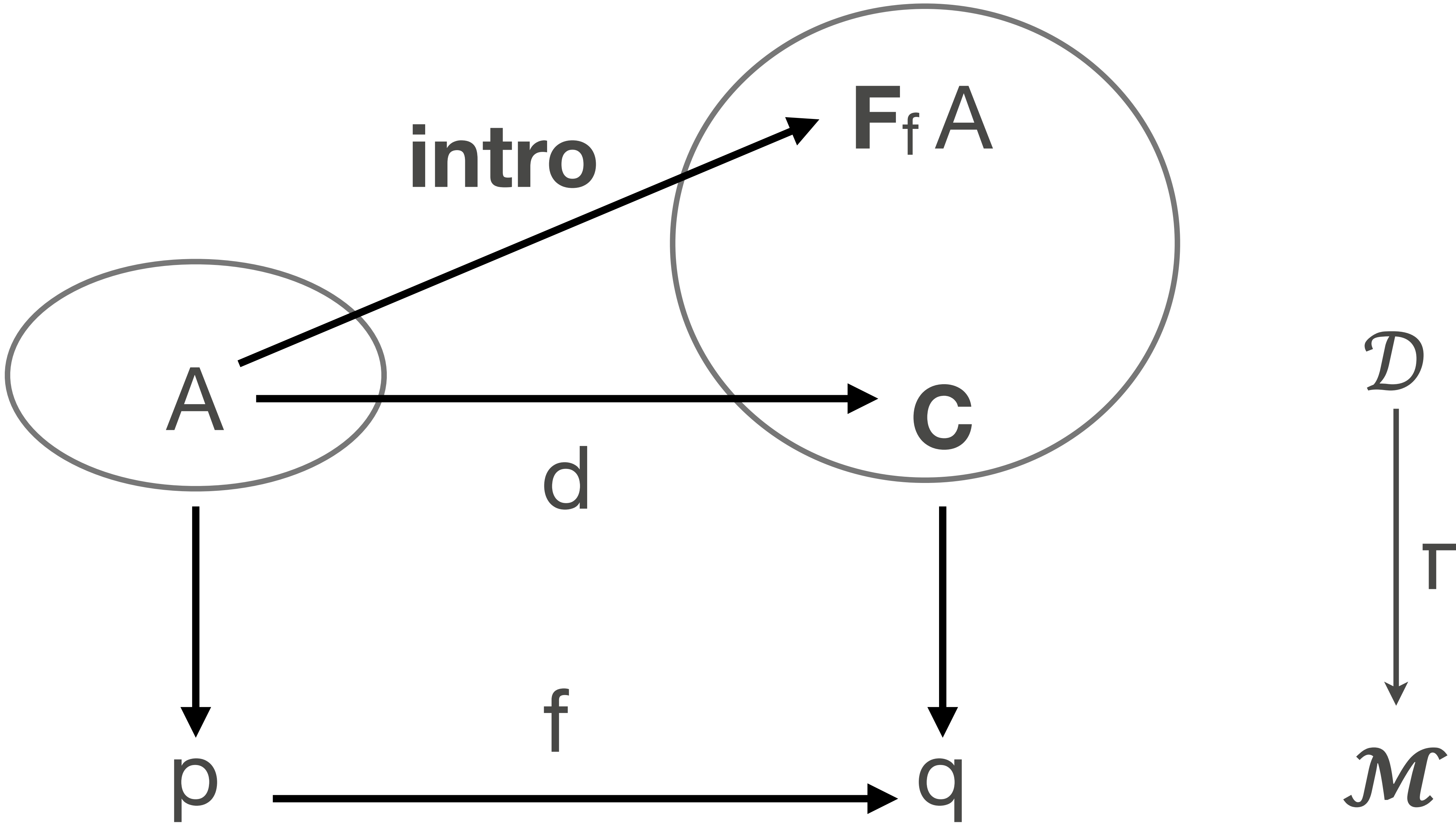
$\mathcal{M}$ 2-category

F and **U** types:
(op)cartesian lifts

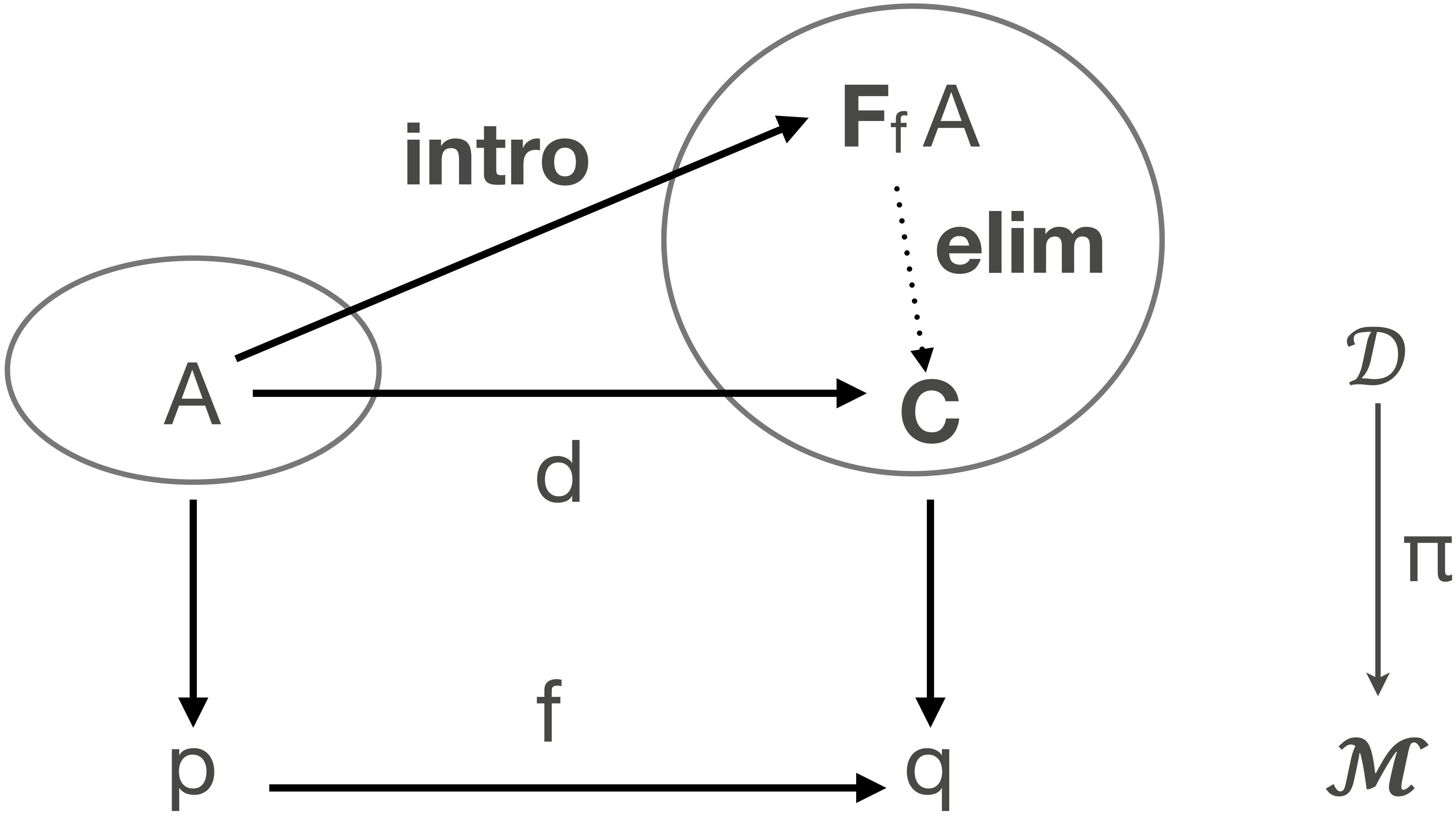
F types: opcartesian



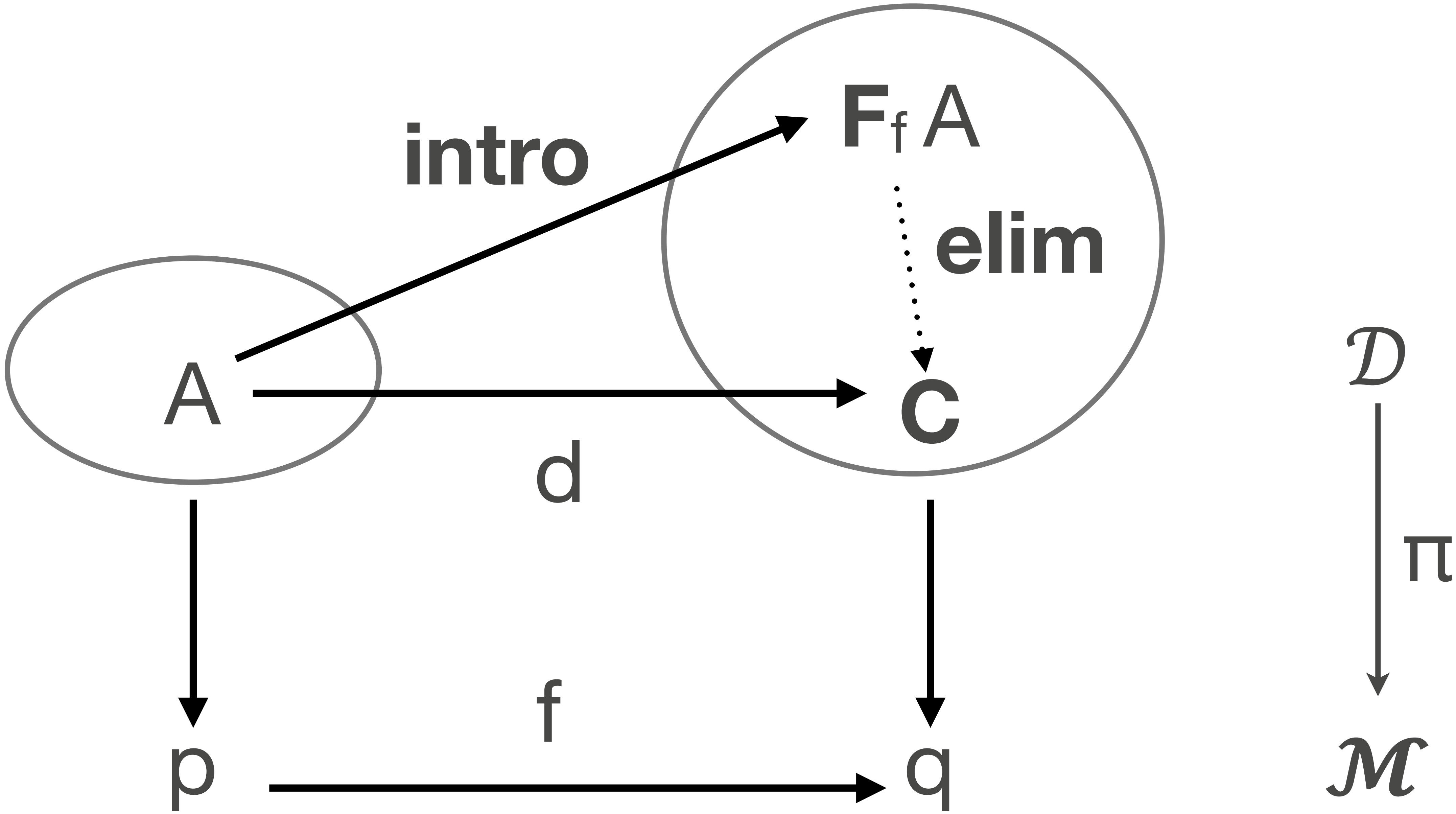
F types: opcartesian



F types: opcartesian



F types: opcartesian



[simplified]

F types: opcartesian

$$\textsc{F-form}\ \frac{\Gamma \vdash_p A\ \mathsf{Type} \quad (\text{over } \gamma \vdash p\ \mathsf{mode}) \qquad \gamma, x : p \vdash \mu : q}{\Gamma \vdash_q \mathsf{F}_{x.\mu}(A)\ \mathsf{Type} \quad (\text{over } \gamma \vdash q\ \mathsf{mode})}$$

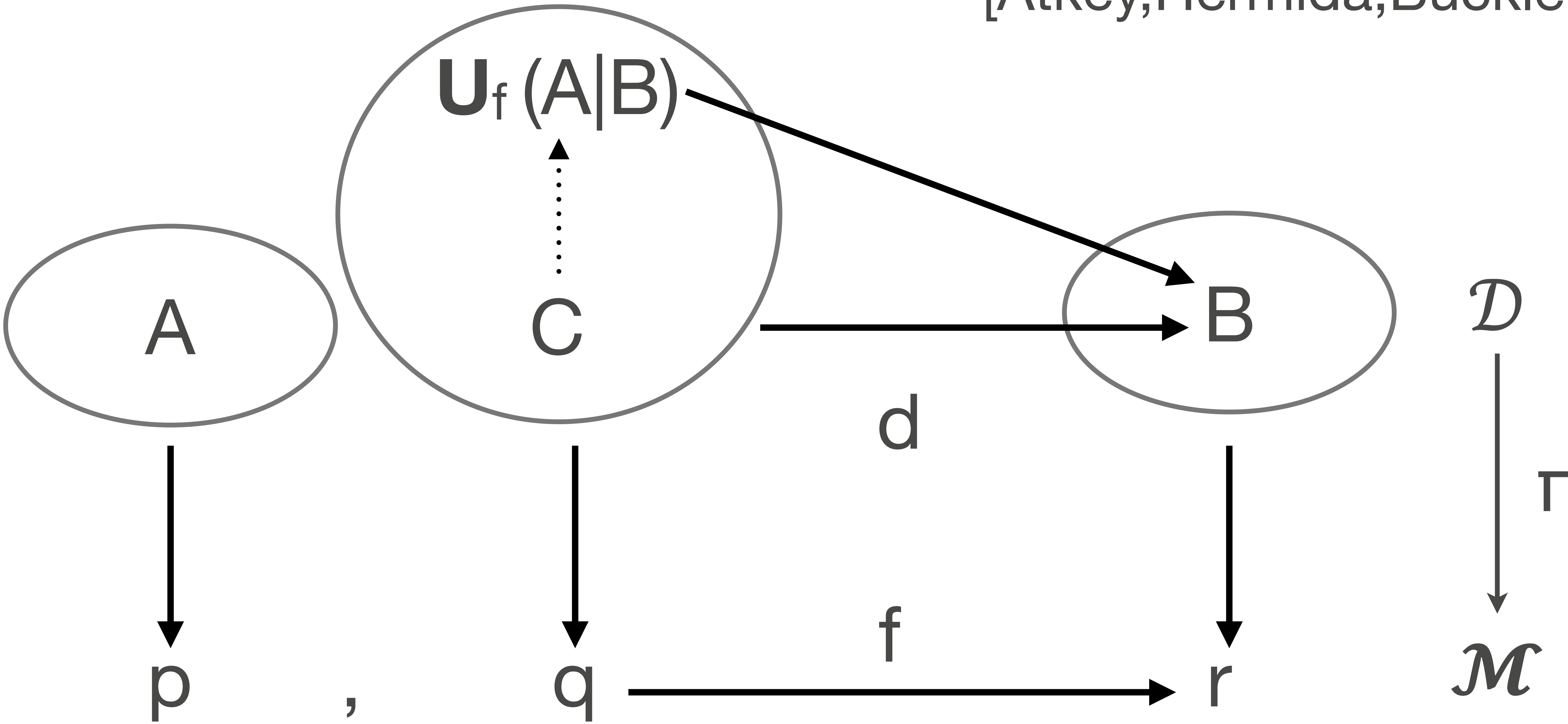
$$\textsc{F-intro}\ \frac{\Gamma \vdash_\nu M : A \quad (\text{over } \gamma \vdash \nu : p)}{\Gamma \vdash_{\mu[\nu/x]} \mathsf{F}(M) : \mathsf{F}_{x.\mu}(A) \quad (\text{over } \gamma \vdash \mu[\nu/x] : q)}$$

$$\textsc{F-elim}\ \frac{\begin{array}{c}\Gamma, y : \mathsf{F}_{x.\mu}(A) \vdash_r C\ \mathsf{Type} \quad (\text{over } \gamma, y : q \vdash r\ \mathsf{mode}) \\ \Gamma \vdash_\nu M : \mathsf{F}_{x.\mu}(A) \quad (\text{over } \gamma \vdash \nu : q) \\ \Gamma, x : A \vdash_{\nu'[\mu/y]} N : C[\mathsf{F}(x)/y] \quad (\text{over } \gamma, x : p \vdash \nu'[\mu/y] : r[\mu/y])\end{array}}{\Gamma \vdash_{\nu'[\nu/y]} \mathsf{let}\, \mathsf{F}(x) = M\ \mathsf{in}\ N : C[M/y] \quad (\text{over } \gamma \vdash \nu'[\nu/y] : r[\nu/y])}$$

$$\mathsf{let}\, \mathsf{F}(x) = \mathsf{F}(M)\ \mathsf{in}\ N \equiv N[M/x]$$

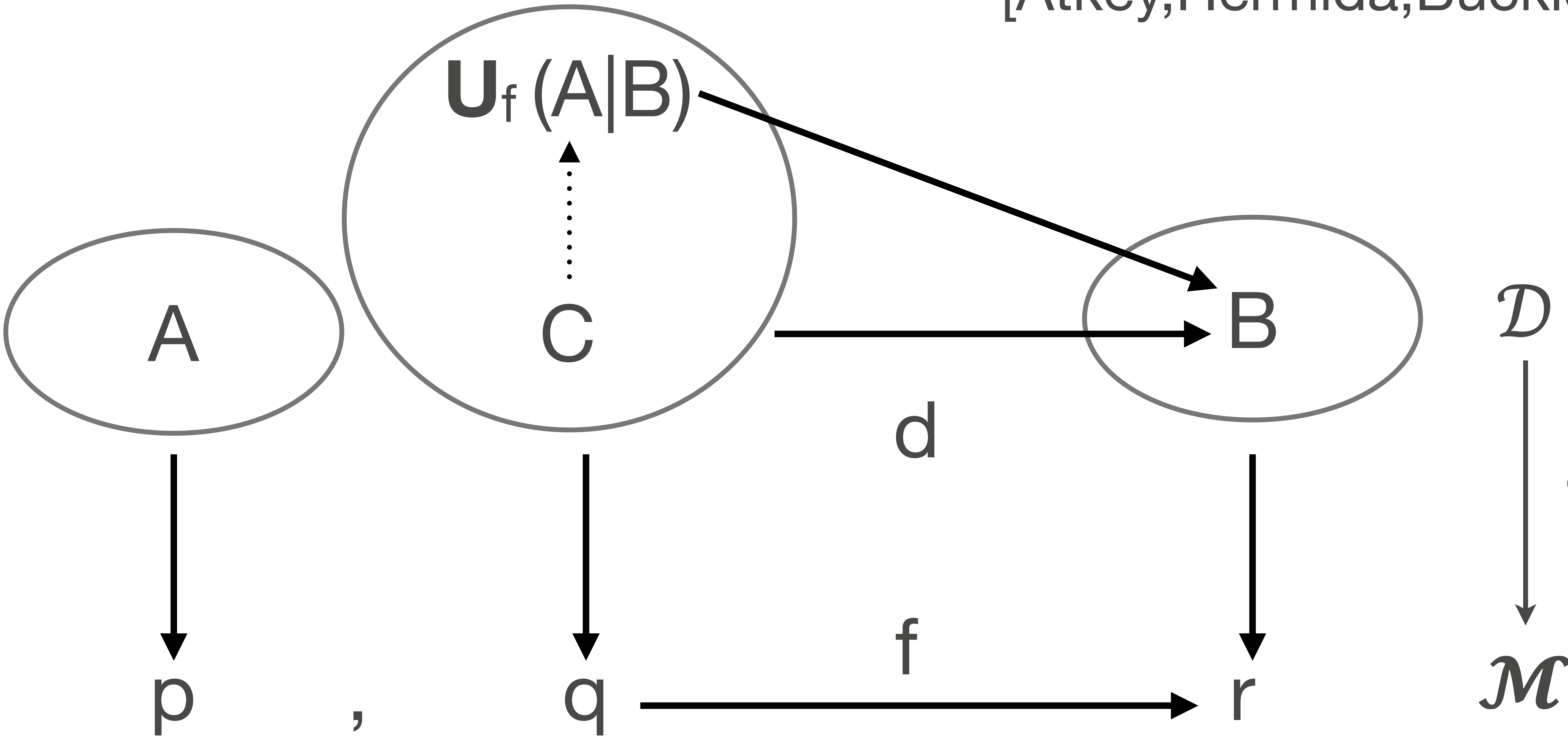
U types: cartesian w/contra.

[Atkey,Hermida,Buckley]



U types: cartesian w/contra.

[Atkey,Hermida,Buckley]



[simplified]

U types: cartesian

$$\text{U-FORM}\ \frac{\begin{array}{c}\Gamma \vdash_p A \ \mathsf{Type} \quad (\text{over } \gamma \vdash p \ \mathsf{mode}) \\ \Gamma, x : A \vdash_q B \ \mathsf{Type} \quad (\text{over } \gamma, x : p \vdash q \ \mathsf{mode}) \\ \gamma, x : p, c : r \vdash \mu : q\end{array}}{\Gamma \vdash_r \mathsf{U}_{c.\mu}(x : A \mid B) \ \mathsf{Type} \quad (\text{over } \gamma \vdash r \ \mathsf{mode})}$$

$$\text{U-INTRO}\ \frac{\Gamma, x : A \vdash_{\mu[\nu/c]} M : B \quad (\text{over } \gamma, x : p \vdash \mu[\nu/c] : q)}{\Gamma \vdash_\nu \lambda x.M : \mathsf{U}_{c.\mu}(x : A \mid B) \quad (\text{over } \gamma \vdash \nu : r)}$$

$$\text{U-ELIM}\ \frac{\begin{array}{c}\Gamma \vdash_{\nu_1} N_1 : \mathsf{U}_{c.\mu}(x : A \mid B) \quad (\text{over } \gamma \vdash \nu_1 : r) \\ \Gamma \vdash_{\nu_2} N_2 : A \quad (\text{over } \gamma \vdash \nu_2 : p)\end{array}}{\Gamma \vdash_{\mu[\nu_2/x, \nu_1/c]} N_1(N_2) : B[N_2/x] \quad (\text{over } \gamma \vdash \mu[\nu_2/x, \nu_1/c] : q)}$$

$$(\lambda x.M)(N) \equiv M[N/x] \qquad\qquad \lambda x.N(x) \equiv N$$

Σ types

Σ types

A comprehension object **supports Σ types** if

$a{:}\mathbf{p},\ x{:}\mathbf{T}(a),\ y{:}\mathbf{T}(a.x) \vdash \boldsymbol{\Sigma}_1(a,x,y) : \mathbf{T}(a)$ **type constructor**

$\mathbf{contract} : 1_{\mathbf{a}} \Rightarrow \boldsymbol{\Sigma}_1(a, \mathbf{1}_{\mathbf{a}}, \mathbf{1}_{a.\mathbf{1}})$

induced $a.x.y \Rightarrow a.\boldsymbol{\Sigma}_1(a,x,y)$ is an $\cong$ **characterize comprehension**

Σ types

A comprehension object **supports Σ types** if

$a{:}\mathbf{p},\ x{:}\mathbf{T}(a),\ y{:}\mathbf{T}(a.x) \vdash \boldsymbol{\Sigma}_{\mathbf{1}}(a,x,y) : \mathbf{T}(a)$ **type constructor**

contract : $1_{\mathbf{a}} \Rightarrow \boldsymbol{\Sigma}_{\mathbf{1}}(a, \mathbf{1}_{\mathbf{a}}, \mathbf{1}_{a.\mathbf{1}})$

induced $a.x.y \Rightarrow a.\boldsymbol{\Sigma}_{\mathbf{1}}(a,x,y)$ is an $\cong$ **characterize comprehension**

$\boxed{\text{Represent } \Sigma x{:}A.B := \mathbf{F}_{\boldsymbol{\Sigma}\mathbf{1}(x,y)}\,(x{:}A,\ y{:}B)}$

Π types

Π types

A comprehension object **supports Π types** if unit types are stable under weakening:

$\mathbf{1}_{a.x} \Rightarrow \boldsymbol{\pi}^{+}(\mathbf{1}_{a})$ is an isomorphism

Π types

A comprehension object **supports Π types** if unit types are stable under weakening:

$\mathbf{1}_{a.x} \Rightarrow \boldsymbol{\pi}^{+}(\mathbf{1}_{a})$ is an isomorphism

$\boxed{\text{Represent } \Pi x{:}A.B := \mathbf{U}_{y.\boldsymbol{\pi}(y)}\ (x{:}A \mid B)}$

Morphism of comp. objects

Morphism of comp. objects

A **morphism of comprehension objects** (**p**,**T**) **to** (**q**,**S**) has

mode term $a{:}\mathbf{p} \vdash \mathbf{f}(a) : \mathbf{q}$

mode term $a{:}\mathbf{p}, x{:}\mathbf{T}(a) \vdash \mathbf{f_1}(a,x) : \mathbf{S}(\mathbf{f}(a))$

2-cell $a{:}\mathbf{p} \vdash \mathbf{1}_{\mathbf{f}(a)} \Rightarrow \mathbf{f_1}(a,\mathbf{1}_a)$

L and R types

L and R types

A morphism of comprehension objects **supports left adjoint types** if induced map $\mathbf{f}(a.x) \Rightarrow \mathbf{f}(a).\mathbf{f}_1(x)$ is an isomorphism

L and R types

A morphism of comprehension objects **supports left adjoint types** if induced map $\mathbf{f}(a\mathbf{.}x) \Rightarrow \mathbf{f}(a).\mathbf{f}_1(x)$ is an isomorphism

A morphism of comprehension objects **supports right adjoint types** if $\mathbf{1}_{\mathbf{f}(a)} \Rightarrow \mathbf{f}_1(a,\mathbf{1}_a)$ is an isomorphism

L and R types

A morphism of comprehension objects **supports left adjoint types** if induced map $\mathbf{f}(a.x) \Rightarrow \mathbf{f}(a).\mathbf{f}_1(x)$ is an isomorphism

A morphism of comprehension objects **supports right adjoint types** if $\mathbf{1}_{\mathbf{f}(a)} \Rightarrow \mathbf{f}_1(a, \mathbf{1}_a)$ is an isomorphism

$\boxed{\text{Represent } \mathbf{R}\, A := \mathbf{U}_{y.\mathbf{f1}(y)}\, (A)}$

L and R types

A morphism of comprehension objects **supports left adjoint types** if induced map $\mathbf{f}(a.x) \Rightarrow \mathbf{f}(a).\mathbf{f_1}(x)$ is an isomorphism

$$\boxed{\text{Define } \mathbf{L}\ A := \mathbf{F}_{y.\mathbf{f1}(y)}\ (A)}$$

A morphism of comprehension objects **supports right adjoint types** if $\mathbf{1}_{\mathbf{f}(a)} \Rightarrow \mathbf{f_1}(a,\mathbf{1}_a)$ is an isomorphism

$$\boxed{\text{Represent } \mathbf{R}\ A := \mathbf{U}_{y.\mathbf{f1}(y)}\ (A)}$$

Spatial type theory

Spatial type theory

A endomorphism of comprehension objects **supports spatial type theory** if it supports **L** and **R** types and $x{:}\mathbf{p} \vdash \mathbf{f}(x) : \mathbf{p}$ is an idempotent comonad

Spatial type theory

A endomorphism of comprehension objects **supports spatial type theory** if it supports **L** and **R** types and
$x{:}\mathbf{p} \vdash \mathbf{f}(x) : \mathbf{p}$ is an idempotent comonad

$$\boxed{\begin{array}{l} \text{Represent } \flat A := \mathrm{comult}^*(\mathbf{L}(A)) \\ \qquad\qquad\quad \sharp A := \mathbf{R}\, A \end{array}}$$

R types [Birkedal+ dependent right adjoints]

$$\textsc{ctx-}🔓\ \frac{\Vdash_p \Gamma\ \mathsf{Ctx}}{\Vdash_q \Gamma, 🔓\ \mathsf{Ctx}} \qquad \textsc{sub-}🔓\ \frac{\Gamma \Vdash_p \Theta : \Delta}{\Gamma, 🔓 \Vdash_q \Theta, 🔓 : \Delta, 🔓}$$

$$\textsc{R-form}\ \frac{\Gamma, 🔓 \Vdash_q A\ \mathsf{Type}}{\Gamma \Vdash_p \mathsf{R}A\ \mathsf{Type}}$$

$$\textsc{R-intro}\ \frac{\Gamma, 🔓 \Vdash_q a : A}{\Gamma \Vdash_p \mathsf{shut}(a) : \mathsf{R}A} \qquad \textsc{R-elim}\ \frac{\Gamma \Vdash_p b : \mathsf{R}B}{\Gamma, 🔓 \Vdash_q \mathsf{open}(b) : B}$$

L and R types

$$\text{CTX-🔓}\ \frac{\Vdash_p \Gamma\ \mathsf{Ctx}}{\Vdash_q \Gamma, \text{🔓}\ \mathsf{Ctx}} \qquad \text{SUB-🔓}\ \frac{\Gamma \Vdash_p \Theta : \Delta}{\Gamma, \text{🔓} \Vdash_q \Theta, \text{🔓} : \Delta, \text{🔓}}$$

$$\text{R-FORM}\ \frac{\Gamma, \text{🔓} \Vdash_q A\ \mathsf{Type}}{\Gamma \Vdash_p \mathsf{R}A\ \mathsf{Type}}$$

$$\text{R-INTRO}\ \frac{\Gamma, \text{🔓} \Vdash_q a : A}{\Gamma \Vdash_p \mathsf{shut}(a) : \mathsf{R}A} \qquad \text{R-ELIM}\ \frac{\Gamma \Vdash_p b : \mathsf{R}B}{\Gamma, \text{🔓} \Vdash_q \mathsf{open}(b) : B}$$

$$\text{L-FORM}\ \frac{\Gamma \Vdash_p A\ \mathsf{Type}}{\Gamma, \text{🔓} \Vdash_q \mathsf{L}A\ \mathsf{Type}}$$

$$\text{L-INTRO}\ \frac{}{\Gamma, A, \text{🔓} \Vdash_q \mathsf{left}_A : \mathsf{L}A[\mathsf{proj}_{\Gamma,A}, \text{🔓}]} \qquad \text{L-ELIM}\ \frac{\Gamma, A, \text{🔓} \Vdash_q c : C[\mathsf{proj}_{\Gamma,A}, \text{🔓}, \mathsf{left}_A]}{\Gamma, \text{🔓}, \mathsf{L}A \Vdash_q \mathsf{letleft}(c) : C}$$

Spatial L and R types

$$\textsc{ctx-}🔓\ \frac{\Vdash_p \Gamma\ \mathsf{Ctx}}{\Vdash_q \Gamma, 🔓\ \mathsf{Ctx}} \qquad \textsc{sub-}🔓\ \frac{\Gamma \Vdash_p \Theta : \Delta}{\Gamma, 🔓 \Vdash_q \Theta, 🔓 : \Delta, 🔓}$$

$$\textsc{R-form}\ \frac{\Gamma, 🔓 \Vdash_q A\ \mathsf{Type}}{\Gamma \Vdash_p \mathsf{R}A\ \mathsf{Type}}$$

$$\textsc{R-intro}\ \frac{\Gamma, 🔓 \Vdash_q a : A}{\Gamma \Vdash_p \mathsf{shut}(a) : \mathsf{R}A} \qquad \textsc{R-elim}\ \frac{\Gamma \Vdash_p b : \mathsf{R}B}{\Gamma, 🔓 \Vdash_q \mathsf{open}(b) : B}$$

$$\textsc{L-form}\ \frac{\Gamma \Vdash_p A\ \mathsf{Type}}{\Gamma, 🔓 \Vdash_q \mathsf{L}A\ \mathsf{Type}}$$

$$\textsc{L-intro}\ \frac{}{\Gamma, A, 🔓 \Vdash_q \mathsf{left}_A : \mathsf{L}A[\mathsf{proj}_{\Gamma,A}, 🔓]} \qquad \textsc{L-elim}\ \frac{\Gamma, A, 🔓 \Vdash_q c : C[\mathsf{proj}_{\Gamma,A}, 🔓, \mathsf{left}_A]}{\Gamma, 🔓, \mathsf{L}A \Vdash_q \mathsf{letleft}(c) : C}$$

$$\Gamma, 🔓 \Vdash \mathsf{counit}_\Gamma : \Gamma$$
$$\Gamma, 🔓 \Vdash \mathsf{comult}_\Gamma : \Gamma, 🔓, 🔓$$

Spatial L and R types

[can translate Shulman's optimized rules into this]

$$\textsc{ctx-}\text{🔓}\ \frac{\Vdash_p \Gamma\ \mathsf{Ctx}}{\Vdash_q \Gamma, \text{🔓}\ \mathsf{Ctx}} \qquad \textsc{sub-}\text{🔓}\ \frac{\Gamma \Vdash_p \Theta : \Delta}{\Gamma, \text{🔓} \Vdash_q \Theta, \text{🔓} : \Delta, \text{🔓}}$$

$$\textsc{R-form}\ \frac{\Gamma, \text{🔓} \Vdash_q A\ \mathsf{Type}}{\Gamma \Vdash_p \mathsf{R}A\ \mathsf{Type}}$$

$$\textsc{R-intro}\ \frac{\Gamma, \text{🔓} \Vdash_q a : A}{\Gamma \Vdash_p \mathsf{shut}(a) : \mathsf{R}A} \qquad \textsc{R-elim}\ \frac{\Gamma \Vdash_p b : \mathsf{R}B}{\Gamma, \text{🔓} \Vdash_q \mathsf{open}(b) : B}$$

$$\textsc{L-form}\ \frac{\Gamma \Vdash_p A\ \mathsf{Type}}{\Gamma, \text{🔓} \Vdash_q \mathsf{L}A\ \mathsf{Type}}$$

$$\textsc{L-intro}\ \frac{}{\Gamma, A, \text{🔓} \Vdash_q \mathsf{left}_A : \mathsf{L}A[\mathsf{proj}_{\Gamma,A}, \text{🔓}]} \qquad \textsc{L-elim}\ \frac{\Gamma, A, \text{🔓} \Vdash_q c : C[\mathsf{proj}_{\Gamma,A}, \text{🔓}, \mathsf{left}_A]}{\Gamma, \text{🔓}, \mathsf{L}A \Vdash_q \mathsf{letleft}(c) : C}$$

$$\Gamma, \text{🔓} \Vdash \mathsf{counit}_\Gamma : \Gamma$$
$$\Gamma, \text{🔓} \Vdash \mathsf{comult}_\Gamma : \Gamma, \text{🔓}, \text{🔓}$$

Semantics (dependent)

$$\begin{tikzcd} \mathcal{D} \arrow[d, "\pi"] \\ \mathcal{M} \end{tikzcd}$$

"local discrete fibration
of 2-categories with families"

WIP

- Current translations of object-language substitutions use some stricter **F** types, or modified mode theory; trying to reconcile with the semantics
- Top **F** and **U** types are *strictly* stable under substitution — move to mode theory? strictification?
- Semantics with fibrancy for homotopy models

Pattern

1. New judgements for left adjoints: **mode types/terms**
2. Left adjoint types: **F types**
3. Right adjoint types: **U types**
4. Structural rules: **2-cells between mode terms**
5. Optimize placement of structural rules: **derived rules**

Goals for Modal Framework

- covers lots of examples
- easy to go from intended semantics to a signature
- automatically get type theoretic rules
  (but with explicit structural rules)
- can derive “optimized” rules (requires cleverness)
- categorical semantics for whole framework at once
- expected structures are models of signatures
- proof assistant with enough automation
  to make it convenient