A Type Theory for Parameterised Spectra

Mitchell Riley
Dan Licata
jww. Eric Finster

Wesleyan University

8th March 2020

Spectra and Parameterised Spectra

Spectra

Definition

A *prespectrum* E is a sequence of pointed types $E : \mathbb{N} \to \mathcal{U}_\star$ together with pointed maps $\alpha_n : E_n \to_\star \Omega E_{n+1}$.

Spectra

Definition

A *prespectrum* E is a sequence of pointed types $E : \mathbb{N} \to \mathcal{U}_\star$ together with pointed maps $\alpha_n : E_n \to_\star \Omega E_{n+1}$.

A *spectrum* is a prespectrum such that the α_n are pointed equivalences.

Spectra

Definition

A *prespectrum* E is a sequence of pointed types $E : \mathbb{N} \to \mathcal{U}_\star$ together with pointed maps $\alpha_n : E_n \to_\star \Omega E_{n+1}$.

A *spectrum* is a prespectrum such that the α_n are pointed equivalences.

Example

Each abelian group G yields a spectrum with $E_n :\equiv K(G, n)$, the 'Eilenberg-MacLane spaces'.

Spectra

Definition

A *prespectrum* E is a sequence of pointed types $E : \mathbb{N} \to \mathcal{U}_\star$ together with pointed maps $\alpha_n : E_n \to_\star \Omega E_{n+1}$.

A *spectrum* is a prespectrum such that the α_n are pointed equivalences.

Example

Each abelian group G yields a spectrum with $E_n :\equiv K(G, n)$, the 'Eilenberg-MacLane spaces'.

Example

The *zero spectrum* with $E_n :\equiv 1$.

Spectra

Definition

A *prespectrum* E is a sequence of pointed types $E : \mathbb{N} \to \mathcal{U}_\star$ together with pointed maps $\alpha_n : E_n \to_\star \Omega E_{n+1}$.

A *spectrum* is a prespectrum such that the α_n are pointed equivalences.

Example

Each abelian group G yields a spectrum with $E_n :\equiv K(G, n)$, the 'Eilenberg-MacLane spaces'.

Example

The *zero spectrum* with $E_n :\equiv 1$.

Example

The *sphere prespectrum* has $E_n :\equiv S^n$, with α_n the transpose of $\Sigma S^n \to_\star S^{n+1}$

Cohomology and Homology

Definition

Given a spectrum E and a pointed type X,

- the *cohomology* of X with coefficients in E is

$$E^n(X) :\equiv \pi_0(X \to_\star E_n)$$

Cohomology and Homology

Definition

Given a spectrum E and a pointed type X,

- the *cohomology* of X with coefficients in E is

$$E^n(X) :\equiv \pi_0(X \to_\star E_n)$$

- the *homology* of X with coefficients in E is

$$E_n(X) :\equiv \operatorname*{colim}_{k \to \infty} \pi_{n+k}(X \wedge E_k)$$

where $A \wedge B := (A \times B)/(A \vee B)$ is the smash product.

Problem

Working with the smash product in HoTT is a serious endeavour!

Problem

Working with the smash product in HoTT is a serious endeavour!

There ought to be a smash product of two spectra.
(But how? Describe 'highly structured spectra' internally? Yow!)

Problem

Working with the smash product in HoTT is a serious endeavour!

There ought to be a smash product of two spectra.
(But how? Describe 'highly structured spectra' internally? Yow!)

Instead: Model type theory in a topos where spectra already exist.

Parameterised Spectra

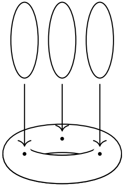
Definition

A *parameterised spectrum* is a space-indexed family of spectra.

Parameterised Spectra

Definition

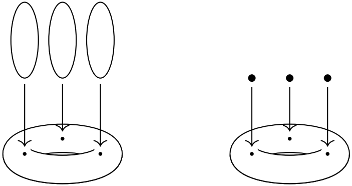
A *parameterised spectrum* is a space-indexed family of spectra.



Parameterised Spectra

Definition

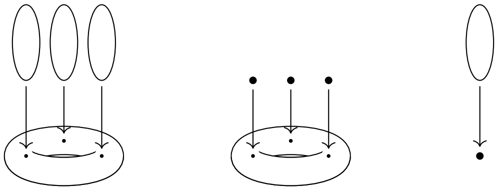
A *parameterised spectrum* is a space-indexed family of spectra.



Parameterised Spectra

Definition

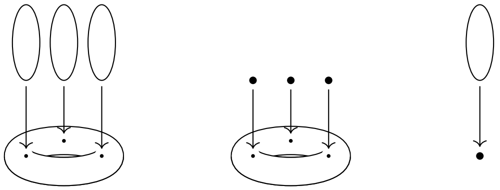
A *parameterised spectrum* is a space-indexed family of spectra.



Parameterised Spectra

Definition

A *parameterised spectrum* is a space-indexed family of spectra.



Theorem (Joyal 2008)

The ∞-category of parameterised spectra, $P\mathrm{Spec}$, is an ∞-topos.

So is a model of HoTT.

A Toy Model: Families of Pointed Types

Definition

A *context* Γ is a type Γ_B and a type family $\Gamma_E : \Gamma_B \to \mathcal{U}$ with a chosen basepoint $\gamma_0(\gamma) : \Gamma_E(\gamma)$ for each $\gamma : \Gamma_B$

A Toy Model: Families of Pointed Types

Definition

A *context* Γ is a type Γ_B and a type family $\Gamma_E : \Gamma_B \to \mathcal{U}$ with a chosen basepoint $\gamma_0(\gamma) : \Gamma_E(\gamma)$ for each $\gamma : \Gamma_B$

A *type A in context* Γ is a family $A_B : \Gamma_B \to \mathcal{U}$ and family

$$A_E : (\gamma : \Gamma_B) \to \Gamma_E(\gamma) \to A_B(\gamma) \to \mathcal{U}$$

with a chosen basepoint $a_0(\gamma, a) : A_E(\gamma, \gamma_0(\gamma), a)$ for each $\gamma : \Gamma_B$ and $a : A_B(\gamma)$.

A Toy Model: Families of Pointed Types

Definition

A *context* Γ is a type Γ_B and a type family $\Gamma_E : \Gamma_B \to \mathcal{U}$ with a chosen basepoint $\gamma_0(\gamma) : \Gamma_E(\gamma)$ for each $\gamma : \Gamma_B$

A *type A in context* Γ is a family $A_B : \Gamma_B \to \mathcal{U}$ and family

$$A_E : (\gamma : \Gamma_B) \to \Gamma_E(\gamma) \to A_B(\gamma) \to \mathcal{U}$$

with a chosen basepoint $a_0(\gamma, a) : A_E(\gamma, \gamma_0(\gamma), a)$ for each $\gamma : \Gamma_B$ and $a : A_B(\gamma)$.

$$\begin{array}{ccccc} \Gamma_E, & \leftarrow & A_E, & \leftarrow & B_E, \\ \downarrow & & \downarrow & & \downarrow \\ \Gamma_B, & \leftarrow & A_B, & \leftarrow & B_B, \end{array}$$

(This was one of Ulrik's 'toy models' of cohesion)

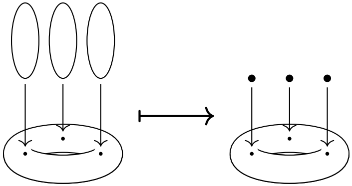
Goal:

Add type formers that capture some of the additional structure in these models.

The ‘Underlying Space’ Modality

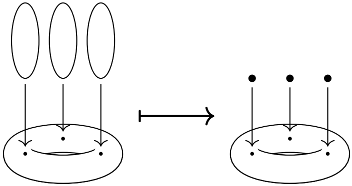
Underlying Space

For every type A there should be a type $\natural A$ that deletes the spectral information.



Underlying Space

For every type A there should be a type $\natural A$ that deletes the spectral information.



This $\natural$ is an idempotent monad and comonad that is adjoint to itself.

Like Mike's Spatial Type Theory, but with $\sharp \equiv \flat$.

Recall: Spatial Type Theory

$\flat$ is a lex idempotent comonad, $\sharp$ is an idempotent monad, and $\flat \dashv \sharp$.

We put in a judgemental version of $\flat$ and have the type formers interact with it.

$$\Delta \mid \Gamma \vdash a : A \qquad \text{corresponds to} \qquad a : \flat\Delta \times \Gamma \to A$$

Recall: Spatial Type Theory

$\flat$ is a lex idempotent comonad, $\sharp$ is an idempotent monad, and $\flat \dashv \sharp$.

We put in a judgemental version of $\flat$ and have the type formers interact with it.

$$\Delta \mid \Gamma \vdash a : A \qquad \text{corresponds to} \qquad a : \flat\Delta \times \Gamma \to A$$

$$\textsc{var-crisp}\ \frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

corresponds to

$$\flat(\Delta \times A \times \Delta') \times \Gamma \to \flat A \to A$$

Recall: Spatial Type Theory

$\flat$ is a lex idempotent comonad, $\sharp$ is an idempotent monad, and $\flat \dashv \sharp$.
We put in a judgemental version of $\flat$ and have the type formers interact with it.

$$\Delta \mid \Gamma \vdash a : A \qquad \text{corresponds to} \qquad a : \flat\Delta \times \Gamma \to A$$

$$\textsc{var-crisp}\ \frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A}$$

corresponds to

$$\flat(\Delta \times A \times \Delta') \times \Gamma \to \flat A \to A$$

$$\flat\textsc{-intro}\ \frac{\Delta \mid \cdot \vdash a : A}{\Delta \mid \Gamma \vdash a^\flat : \flat A}$$

corresponds to

$$\flat\Delta \times \Gamma \to \flat\Delta \to \flat\flat\Delta \to \flat A$$

The Unit?

In spatial type theory, the counit is invisible: there was an admissible rule

$$\textsc{counit}\ \frac{\Delta \mid x : A, \Gamma \vdash b : B}{\Delta, x : A \mid \Gamma \vdash b : B}$$

The Unit?

In spatial type theory, the counit is invisible: there was an admissible rule

$$\textsc{counit}\ \frac{\Delta \mid x : A, \Gamma \vdash b : B}{\Delta, x : A \mid \Gamma \vdash b : B}$$

With $\natural$ we have a dilemma: there is both a unit $A \to \natural A$ and a counit $\natural A \to A$, the round trip on A is not the identity.

$$\textsc{unit}?\ \frac{\Delta, x : A \mid \Gamma \vdash b : B}{\Delta \mid x : A, \Gamma \vdash b : B}$$

We choose to make the *counit* explicit.

Zones?

We can't just divide the context into two zones anymore.

$$x : A, y : B(x) \mid z : C \vdash d : D$$

Zones?

We can't just divide the context into two zones anymore.

$$x : A, y : B(x) \mid z : C \vdash d : D$$

What if we want to precompose with the unit on $x : A$ only?

$$y : B(x) \mid x : A, z : C \vdash d : D$$

Zeroed Variables

$$\frac{\Gamma\ \mathsf{ctx} \qquad \Gamma^0 \vdash A\ \mathsf{type}}{\Gamma, x^0 : A\ \mathsf{ctx}}$$

$$\frac{}{\Gamma, x^0 : A, \Gamma' \vdash x^0 : A} \qquad \frac{}{\Gamma, x : A, \Gamma' \vdash x^0 : A^0}$$

Γ^0 denotes an operation that zeroes all the variables in Γ.

Zeroed Variables

$$\frac{\Gamma\ \mathsf{ctx} \qquad \Gamma^0 \vdash A\ \mathsf{type}}{\Gamma, x^0 : A\ \mathsf{ctx}}$$

$$\frac{}{\Gamma, x^0 : A, \Gamma' \vdash x^0 : A} \qquad \frac{}{\Gamma, x : A, \Gamma' \vdash x^0 : A^0}$$

Γ^0 denotes an operation that zeroes all the variables in Γ.

$$\textsc{counit}\ \frac{\Gamma, x : A, \Gamma' \vdash b : B}{\Gamma, x^0 : A, \Gamma'[x^0/x] \vdash b[x^0/x] : B[x^0/x]}$$

$$\textsc{unit}\ \frac{\Gamma, x^0 : A, \Gamma' \vdash b : B}{\Gamma, x : A, \Gamma' \vdash b : B}$$

Rules for ♮

$$\natural\text{-}\textsc{form}\ \frac{\Gamma^0 \vdash A \text{ type}}{\Gamma \vdash \natural A \text{ type}}$$

$$\natural\text{-}\textsc{intro}\ \frac{\Gamma^0 \vdash a : A}{\Gamma \vdash a^\natural : \natural A} \qquad \natural\text{-}\textsc{elim}\ \frac{\Gamma \vdash a : \natural A}{\Gamma \vdash a_\natural : A}$$

$$a^\natural{}_\natural \equiv a \qquad n \equiv n^0{}_\natural{}^\natural$$

These are the ♯-style rules. The ♭-style rules are derivable!

$\natural$ and Dependency

A context

$$x : A, y^0 : B(x^0), z : C(x, y^0), w^0 : D(x^0, y^0, z^0)$$

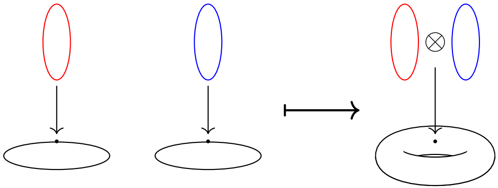
corresponds in the model to

$$\begin{array}{ccccccc}
x : A_E, & \dashleftarrow & & & z : C_E(x) & & \\
\downarrow & & & & \downarrow & & \\
x^0 : A_B, & \dashleftarrow & y^0 : B_B(x^0), & \dashleftarrow & z^0 : C_B(x^0, y^0), & \dashleftarrow & w^0 : D_B(x^0, y^0, z^0)
\end{array}$$

The Smash Product

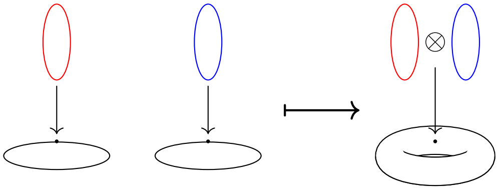
Smash Product

For two types A and B there should be a type $A \otimes B$ corresponding to the 'external smash product'.



Smash Product

For two types A and B there should be a type $A \otimes B$ corresponding to the 'external smash product'.



This is a symmetric monoidal product with no additional structural rules.

Bunched Contexts

We can take a cue from ‘bunched logics’, where there are two ways of combining contexts, an ordinary cartesian one and a linear one.

$$\frac{\Gamma_1 \text{ ctx} \qquad \Gamma_2 \text{ ctx}}{\Gamma_1, \Gamma_2 \text{ ctx}} \qquad\qquad \frac{\Gamma_1 \text{ ctx} \qquad \Gamma_2 \text{ ctx}}{(\Gamma_1)(\Gamma_2) \text{ ctx}}$$

For the comma *only*, we have weakening and contraction as normal.

Bunched Contexts

A typical context:

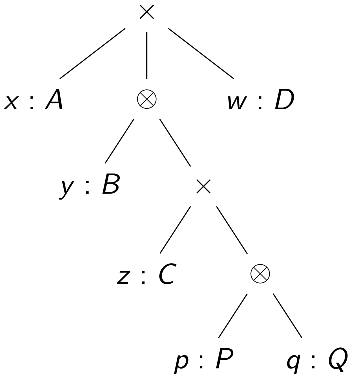
$$x : A, (y : B)(z : C, (p : P)(q : Q)), w : D$$

Bunched Contexts

A typical context:

$$x : A, (y : B)(z : C, (p : P)(q : Q)), w : D$$

Or as a tree:



Simple Smash

$$\otimes\text{-}\textsc{form}\ \frac{A\ \mathsf{type} \qquad B\ \mathsf{type}}{A \otimes B\ \mathsf{type}}$$

Simple Smash

$$\otimes\text{-FORM}\ \frac{A\ \mathsf{type} \qquad B\ \mathsf{type}}{A \otimes B\ \mathsf{type}}$$

$$\otimes\text{-INTRO}\ \frac{\Omega \vdash a : A \qquad \Omega' \vdash b : B}{(\Omega)(\Omega') \vdash a \otimes b : A \otimes B}$$

Simple Smash

$$\otimes\text{-FORM}\;\frac{A\ \text{type}\qquad B\ \text{type}}{A \otimes B\ \text{type}}$$

$$\otimes\text{-INTRO}\;\frac{\Omega \vdash a : A \qquad \Omega' \vdash b : B}{(\Omega)(\Omega') \vdash a \otimes b : A \otimes B}$$

$$\otimes\text{-ELIM}\;\frac{\begin{array}{c}\Gamma\{(x : A)(y : B)\} \vdash c : C \\ \Delta \vdash s : A \otimes B\end{array}}{\Gamma\{\Delta\} \vdash \text{let } x \otimes y = s \text{ in } c : C}$$

Smash and Dependency

When does a 'dependent external smash' $A \otimes B$ make sense?

Smash and Dependency

When does a 'dependent external smash' $A \otimes B$ make sense?

Recall: A *type A in context* Γ is a family $A_B : \Gamma_B \to \mathcal{U}$ and family

$$A_E : (\gamma : \Gamma_B) \to \Gamma_E(\gamma) \to A_B(\gamma) \to \mathcal{U}$$

with a chosen basepoint $a_0(\gamma, a) : A_E(\gamma, \gamma_0(\gamma), a)$ for each $\gamma : \Gamma_B$ and $a : A_B(\gamma)$.

Smash and Dependency

When does a 'dependent external smash' $A \otimes B$ make sense?

Recall: A *type A in context* Γ is a family $A_B : \Gamma_B \to \mathcal{U}$ and family

$$A_E : (\gamma : \Gamma_B) \to \Gamma_E(\gamma) \to A_B(\gamma) \to \mathcal{U}$$

with a chosen basepoint $a_0(\gamma, a) : A_E(\gamma, \gamma_0(\gamma), a)$ for each $\gamma : \Gamma_B$ and $a : A_B(\gamma)$.

We can only do it when the fibers of A and B don't depend on Γ_E.

Smash and Dependency

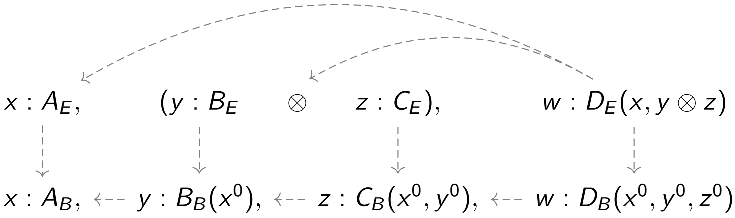
We will be allowed to form contexts like:

$$x : A, (y : B(x^0))(z : C(x^0, y^0)), w : D(x, y, z)$$

Smash and Dependency

We will be allowed to form contexts like:

$$x : A, (y : B(x^0))(z : C(x^0, y^0)), w : D(x, y, z)$$



Dependent Smash

The judgemental ‘context smash’.

$$\frac{\Gamma \vdash \Omega \ \mathsf{tele}}{\Gamma, \Omega \ \mathsf{ctx}} \qquad \frac{\Gamma \vdash \Omega \ \mathsf{tele} \quad \Gamma, \Omega \vdash \Omega' \ \mathsf{tele}}{\Gamma \vdash \Omega, \Omega' \ \mathsf{tele}} \qquad \frac{\Gamma^0 \vdash \Omega \ \mathsf{tele} \quad \Gamma^0, \Omega^0 \vdash \Omega' \ \mathsf{tele}}{\Gamma \vdash (\Omega)(\Omega') \ \mathsf{tele}}$$

Dependent Smash

The judgemental 'context smash'.

$$\frac{\Gamma \vdash \Omega \text{ tele}}{\Gamma, \Omega \text{ ctx}} \qquad \frac{\Gamma \vdash \Omega \text{ tele} \quad \Gamma, \Omega \vdash \Omega' \text{ tele}}{\Gamma \vdash \Omega, \Omega' \text{ tele}} \qquad \frac{\Gamma^0 \vdash \Omega \text{ tele} \quad \Gamma^0, \Omega^0 \vdash \Omega' \text{ tele}}{\Gamma \vdash (\Omega)(\Omega') \text{ tele}}$$

A type that internalises it:

$$\otimes\text{-FORM}\ \frac{\Gamma^0 \vdash A \text{ type} \qquad \Gamma^0, x^0 : A^0 \vdash B \text{ type}}{\Gamma \vdash \textstyle\bigcirc\!\!\!\!\!\sum_{(x^0 : A)} B \text{ type}}$$

Dependent Smash

The judgemental 'context smash'.

$$\frac{\Gamma \vdash \Omega \text{ tele}}{\Gamma, \Omega \text{ ctx}} \qquad \frac{\Gamma \vdash \Omega \text{ tele} \quad \Gamma, \Omega \vdash \Omega' \text{ tele}}{\Gamma \vdash \Omega, \Omega' \text{ tele}} \qquad \frac{\Gamma^0 \vdash \Omega \text{ tele} \quad \Gamma^0, \Omega^0 \vdash \Omega' \text{ tele}}{\Gamma \vdash (\Omega)(\Omega') \text{ tele}}$$

A type that internalises it:

$$\otimes\text{-FORM} \; \frac{\Gamma^0 \vdash A \text{ type} \qquad \Gamma^0, x^0 : A^0 \vdash B \text{ type}}{\Gamma \vdash \bigcirc\!\!\!\!\!\sum_{(x^0 : A)} B \text{ type}}$$

$$\otimes\text{-INTRO} \; \frac{\Gamma^0, \Omega, \Omega'^0, \Gamma'^0 \vdash a : A \qquad \Gamma^0, \Omega^0, \Omega', \Gamma'^0 \vdash b : B[a^0/x^0]}{\Gamma, (\Omega)(\Omega'), \Gamma' \vdash a \otimes b : \bigcirc\!\!\!\!\!\sum_{(x^0 : A)} B}$$

$$\ldots$$

Extras

Also rules for:

- The monoidal unit $\mathbb{S}$.
  (A bit of a pain! The unitor isomorphism has to be built into several of the other rules.)

Extras

Also rules for:

- The monoidal unit $\mathbb{S}$.
  (A bit of a pain! The unitor isomorphism has to be built into several of the other rules.)
- Dependent 'linear hom' types $A \multimap B$, right adjoint to $- \otimes A$.

Working Informally

Working Informally

This type theory would be unusable if we had to constantly keep track of the shape of the context.

Working Informally

This type theory would be unusable if we had to constantly keep track of the shape of the context.

A cute idea: use colours. Write

$$(x : A)(y : B(x^0)), z : C(x, y), w^0 : D(x^0, y^0, z^0)$$

as

$$\color{red}{x} : A, \color{blue}{y} : B(x), \color{purple}{z} : C(\color{red}{x}, \color{blue}{y}), w : D(x, y, z)$$

Zeroed terms are written in black, they do not contribute to the colour of a term.

Working Informally

This type theory would be unusable if we had to constantly keep track of the shape of the context.

A cute idea: use colours. Write

$$(x : A)(y : B(x^0)), z : C(x, y), w^0 : D(x^0, y^0, z^0)$$

as

$$x : A, y : B(x), z : C(x, y), w : D(x, y, z)$$

Zeroed terms are written in black, they do not contribute to the colour of a term.

In $\otimes$-intro, the two sides must have disjoint colours: $a \otimes b$.

Working Informally

This type theory would be unusable if we had to constantly keep track of the shape of the context.

A cute idea: use colours. Write

$$(x : A)(y : B(x^0)), z : C(x, y), w^0 : D(x^0, y^0, z^0)$$

as

$$x : A, y : B(x), z : C(x, y), w : D(x, y, z)$$

Zeroed terms are written in black, they do not contribute to the colour of a term.

In $\otimes$-intro, the two sides must have disjoint colours: $a \otimes b$.

In $\otimes$-elim, we create two new colours that sum to the colour of the target:

$$\text{let } x \otimes y = p \text{ in } c$$

Eg: Uniqueness principle for $\otimes$

Proposition

Suppose A and B are types. If $C : A \otimes B \to \mathcal{U}$ is a type family and $f : \prod_{(p:A\otimes B)} C(p)$, then for any $p : A \otimes B$ we have

$$(\mathsf{let}\ x \otimes y = p\ \mathsf{in}\ f(x \otimes y)) = f(p)$$

Eg: Uniqueness principle for $\otimes$

Proposition

Suppose A and B are types. If $C : A \otimes B \to \mathcal{U}$ is a type family and $f : \prod_{(p:A\otimes B)} C(p)$, then for any $p : A \otimes B$ we have

$$(\mathsf{let}\ x \otimes y = p\ \mathsf{in}\ f(x \otimes y)) = f(p)$$

Proof.

Let $P : A \otimes B \to \mathcal{U}$ denote the type family

$$P(p) :\equiv (\mathsf{let}\ x \otimes y = p\ \mathsf{in}\ f(x \otimes y)) = f(p)$$

We wish to find an element of $\prod_{(p:A\otimes B)} P(p)$.

Eg: Uniqueness principle for $\otimes$

Proposition

Suppose A and B are types. If $C : A \otimes B \to \mathcal{U}$ is a type family and $f : \prod_{(p:A\otimes B)} C(p)$, then for any $p : A \otimes B$ we have

$$(\text{let } x \otimes y = p \text{ in } f(x \otimes y)) = f(p)$$

Proof.

Let $P : A \otimes B \to \mathcal{U}$ denote the type family

$$P(p) :\equiv (\text{let } x \otimes y = p \text{ in } f(x \otimes y)) = f(p)$$

We wish to find an element of $\prod_{(p:A\otimes B)} P(p)$. By $\otimes$-induction we may assume $p \equiv x' \otimes y'$. Our goal is now

$$(\text{let } x \otimes y = x' \otimes y' \text{ in } f(x \otimes y)) = f(x' \otimes y')$$

Which by the β-rule reduces to $f(x' \otimes y') = f(x' \otimes y')$. $\square$

Non-Eg: A diagonal map for $\otimes$

We cannot define an interesting $\Delta : A \to A \otimes A$ in general.

Non-Eg: A diagonal map for $\otimes$

We cannot define an interesting $\Delta : A \to A \otimes A$ in general.

Given $a : A$, forming $a \otimes a : A \otimes A$ is not allowed: the two inputs to $\otimes$-intro do not have disjoint colours.

Non-Eg: A diagonal map for $\otimes$

We cannot define an interesting $\Delta : A \to A \otimes A$ in general.

Given $a : A$, forming $a \otimes a : A \otimes A$ is not allowed: the two inputs to $\otimes$-intro do not have disjoint colours.

But we *can* form $a \otimes a : A \otimes A$, the diagonal map on the base and constantly zero in the fibers.

Eg: Base of $\bigcirc\!\!\!\!\!\sum$ is $\sum$

Proposition

$\natural \bigcirc\!\!\!\!\!\!\sum_{(x:A)} B(x) \simeq \sum_{(u:\natural A)} \natural B(u_\natural)$

Eg: Base of $\bigcirc\!\!\!\!\!\sum$ is $\sum$

Proposition

$\natural \bigcirc\!\!\!\!\!\sum_{(x:A)} B(x) \simeq \sum_{(u:\natural A)} \natural B(u_\natural)$

Proof.

Given $w : \natural \bigcirc\!\!\!\!\!\sum_{(x:A)} B(x)$ we have a term $w_\natural : \bigcirc\!\!\!\!\!\sum_{(x:A)} B(x)$. Induction on this gives $x : A$ and $y : B(x)$, from which we can produce $(x^\natural, y^\natural) : \sum_{(u:\natural A)} \natural B(u_\natural)$.

Eg: Base of $\mathop{\bigcirc\kern-0.9em\sum}$ is $\sum$

Proposition

$\natural \mathop{\bigcirc\kern-0.9em\sum}_{(x:A)} B(x) \simeq \sum_{(u:\natural A)} \natural B(u_\natural)$

Proof.

Given $w : \natural \mathop{\bigcirc\kern-0.9em\sum}_{(x:A)} B(x)$ we have a term $w_\natural : \mathop{\bigcirc\kern-0.9em\sum}_{(x:A)} B(x)$. Induction on this gives $x : A$ and $y : B(x)$, from which we can produce $(x^\natural, y^\natural) : \sum_{(u:\natural A)} \natural B(u_\natural)$.

In the other direction, from $z : \sum_{(x:\natural A)} \natural B(x_\natural)$ we get $\mathrm{pr}_1(z)_\natural : A$ and $\mathrm{pr}_2(z)_\natural : B(\mathrm{pr}_1(z)_\natural)$. These terms are (vacuously) blue and red respectively so we can form

$$(\mathrm{pr}_1(z)_\natural \otimes \mathrm{pr}_2(z)_\natural) : \mathop{\bigcirc\kern-0.9em\sum}_{x:A} B(x)$$

and apply $(-)^\natural$. Now check round trips. □

Axioms?

How do we characterise parameterised spectra amongst the models? Possibilities:

- $\natural$ is $\mathbb{S}$-nullification
- $\Sigma^n\mathbb{S} \to \Omega\Sigma^{n+1}\mathbb{S}$ is an equivalence
- Relate $\mathbb{S}$ to the stable homotopy groups of ordinary spheres

References I

Joyal, André (2008). *Notes on logoi*. URL: http://www.math.uchicago.edu/~may/IMA/JOYAL/Joyal.pdf.

van Doorn, Floris (2018). "On the formalization of higher inductive types and synthetic homotopy theory". In: *arXiv preprint arXiv:1808.10690*.