# A Type Theory for Parameterised Spectra (A Dependent Bunched Logic)

Mitchell Riley[1]
jww. Dan Licata[1]
Eric Finster[2]

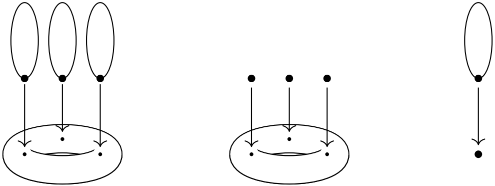
Wesleyan University[1]
University of Birmingham[2]

15th July 2020

# The Intended Semantics

### Definition

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



# Variables

VAR

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

VAR-ZERO

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

VAR-ROUNDTRIP

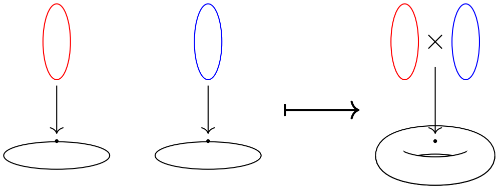
$$\frac{}{\Gamma, x : A, \Gamma' \vdash \underline{x} : \underline{A}}$$

Using a variable $x : A$ marked means only referring to the *base space* of $A$.

There is a modality $\natural$ internalising this marked property of contexts.

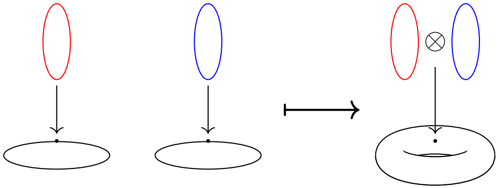
# The Ordinary Product

For two types $A$ and $B$ there is a type $A \times B$:



# The Monoidal 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. (i.e., no weakening or contraction)

# Simple 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.

$$\frac{\Gamma_1 \ \mathsf{ctx} \qquad \Gamma_2 \ \mathsf{ctx}}{\Gamma_1, \Gamma_2 \ \mathsf{ctx}} \qquad\qquad \frac{\Gamma_1 \ \mathsf{ctx} \qquad \Gamma_2 \ \mathsf{ctx}}{\{\Gamma_1\}\{\Gamma_2\} \ \mathsf{ctx}}$$

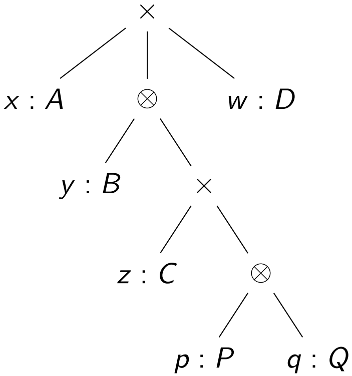
For the comma *only*, we have weakening and contraction as normal.

# Simple Bunched Contexts

A typical context:

$$x : A, \{y : B\}\{z : C, \{p : P\}\{q : Q\}\}, w : D$$

Or as a tree:



# Simple Smash

$$\otimes\text{-INTRO}\ \frac{\Gamma \equiv \Gamma', \{\Omega\}\{\Omega'\}, \Gamma'' \qquad \Omega \vdash a : A \qquad \Omega' \vdash b : B}{\Gamma \vdash a \otimes b : A \otimes B}$$

So to introduce a term of $A \otimes B$, use associativity/symmetry to rearrange the bunches into two pieces and use them separately.

# Smash and Dependency

- Within a $\times$-bunch (things 'comma'ed together), types can depend as normal.
- Within a $\otimes$-bunch, (things 'bracket'ed together), types can only use variables of other children *marked*.

$$x : A, \{y : B(\underline{x})\}\{z : C(\underline{x}, \underline{y}), w : D(\underline{x}, \underline{y}, z)\}$$

This doesn't degenerate to separate 'linear' and 'non-linear' worlds, as in other linear dependent theories.

$$x : A \otimes B, \{y : A\}\{z : B\}, p : \mathsf{Id}(x, y \otimes z)$$

And previous dependent bunched logics had no dependence between bunches.

# Smash and Dependency

Dependency has broken our intro rule, we cannot reorganise the context to get any 'split' we want. E.g., with simple types we have

$$\cfrac{\cfrac{\{x : A\}\{z : C\} \vdash e : E \qquad y : B \vdash f : F}{\{x : A\}\{z : C\}\{y : B\} \vdash e \otimes f : E \otimes F}}{\{x : A\}\{y : B\}\{z : C\} \vdash e \otimes f : E \otimes F}$$

But if $C$ depends on $y$ we are stuck.

$$\frac{\{x : A, \underline{y} :: B\}\{z : C\} \vdash e : E \qquad \underline{x} :: A, y : B, \underline{z} :: C \vdash f : F}{\{x : A\}\{y : B\}\{z : C\} \vdash e \otimes f : E \otimes F}$$

We can apply contraction to the underlying spaces.

# A Different Syntax for Bunches

Doing everything using this bunched structure gets complicated. And the context manipulation isn't annotated in the term. Idea: Have the structure of the bunches tracked separately.

$$\mathfrak{r} \otimes \mathfrak{b} \mid x^{\top} : A, y^{\mathfrak{r}} : B, z^{\mathfrak{b}} : C \text{ ctx}$$

- The 'palette' $\mathfrak{r} \otimes \mathfrak{b}$ describes the shape of the context tree,
- The context $x^{\top} : A, y^{\mathfrak{r}} : B, z^{\mathfrak{b}} : C$ describes what's at the leaves.

# Palettes

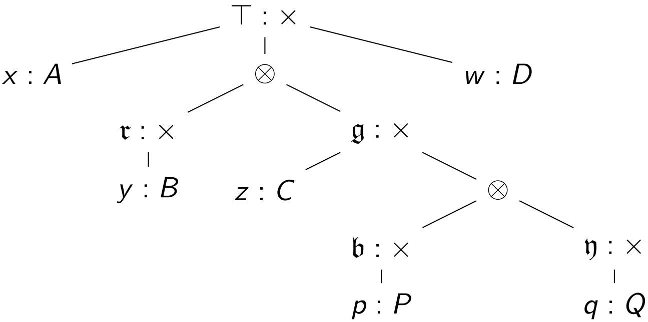
Our context from before

$$x : A, \{y : B\}\{z : C, \{p : P\}\{q : Q\}\}, w : D$$

would be written

$$\mathfrak{r} \otimes (\mathfrak{g} \equiv \mathfrak{b} \otimes \mathfrak{y}) \mid x^{\top} : A, y^{\mathfrak{r}} : B, z^{\mathfrak{g}} : C, p^{\mathfrak{b}} : P, q^{\mathfrak{y}} : Q, w^{\top} : D \text{ ctx}$$

The palette is describing the tree



# Dependent Smash

Specifying how to divide the context in $\otimes$-intro is now a little combinatorial problem on palettes. We write

$$\Phi \vdash \Phi_L \boxtimes \Phi_R \text{ split}$$

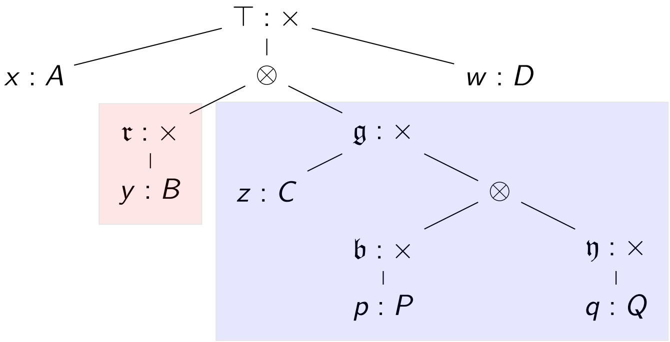
for this judgement.

$$\otimes\text{-FORM}\ \frac{\cdot \mid \underline{\Gamma} \vdash A \text{ type} \qquad \cdot \mid \underline{\Gamma}, \underline{x} :: \underline{A} \vdash B \text{ type}}{\Phi \mid \Gamma \vdash (\underline{x} : A) \otimes B \text{ type}}$$

$$\otimes\text{-INTRO}\ \frac{\begin{array}{c}\Phi \vdash \Phi_L \boxtimes \Phi_R \text{ split} \\ \Phi_L \mid \Gamma^{\Phi_L} \vdash a : A \qquad \Phi_R \mid \Gamma^{\Phi_R} \vdash b : B[\underline{a}/\underline{x}]\end{array}}{\Phi \mid \Gamma \vdash a \otimes b : (\underline{x} : A) \otimes B}$$

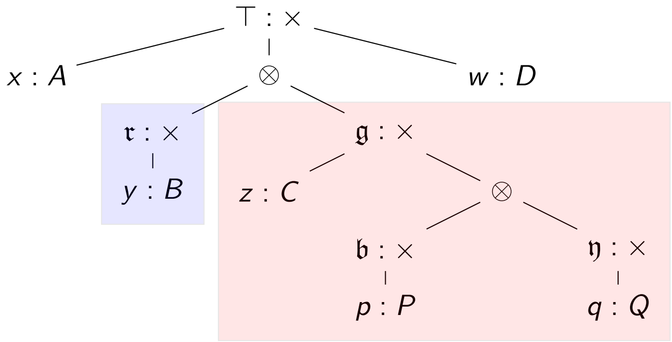
# Splits

$$\mathfrak{r} \otimes (\mathfrak{g} \equiv \mathfrak{b} \otimes \mathfrak{y}) \vdash (\mathfrak{r}) \boxtimes (\mathfrak{g} \equiv \mathfrak{b} \otimes \mathfrak{y}) \text{ split}$$



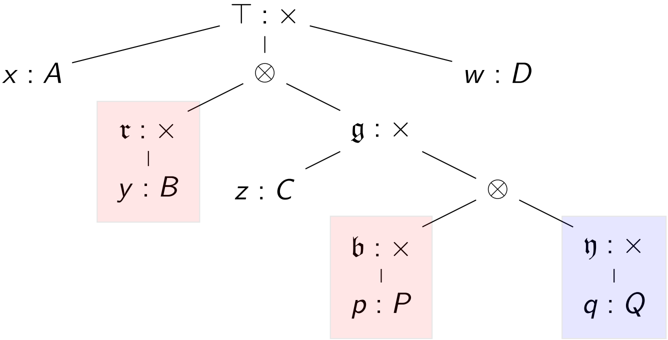
# Splits

$$\mathfrak{r} \otimes (\mathfrak{g} \equiv \mathfrak{b} \otimes \mathfrak{y}) \vdash (\mathfrak{g} \equiv \mathfrak{b} \otimes \mathfrak{y}) \boxtimes (\mathfrak{r}) \text{ split}$$



# Splits

$$\mathfrak{r} \otimes (\mathfrak{g} \equiv \mathfrak{b} \otimes \mathfrak{y}) \vdash (\mathfrak{p} \equiv \mathfrak{r} \otimes \mathfrak{b}) \boxtimes (\mathfrak{y}) \text{ split}$$



# Synthetic Stable Homotopy Theory

This type theory is intended to be actually used! Some simple test cases:

- For any map of base spaces $f : X \to Y$ one can internally define the $f_! \dashv f^* \dashv f_*$ and $\otimes, \mathrm{Hom}$ functors of a 'Wirthmüller context'.
- For any type $\underline{A}$, and point $\underline{x}$ in the base space, the spectrum over $\underline{x}$ is a module over $\Sigma^\infty \Omega_{\underline{x}}(\natural A)$.

# Implementation Challenges

- Multiple ways to use the same variable,
- The set of labels on variables is not fixed,
- In a proof assistant, one would prefer to not have to fix the context split in advance.