A Type Theory for Parameterised Spectra

Mitchell Riley

12$^{\text{th}}$ February 2020

Some toposes:

- Set
- Sheaves on a space – algebraic geometry
- The effective topos – computability

Some toposes:

- Set
- Sheaves on a space – algebraic geometry
- The effective topos – computability

Type Theory

M is a finitely generated module

Type Theory for the Working Mathematician

Some toposes:

- Set
- Sheaves on a space – algebraic geometry
- The effective topos – computability

Type Theory
M is a finitely generated module

Set
M is an ordinary finitely generated module

Type Theory for the Working Mathematician

Some toposes:

- Set
- Sheaves on a space – algebraic geometry
- The effective topos – computability

$$\begin{tikzcd}[column sep=tiny]
& \begin{array}{c}\text{Type Theory}\\ \textit{M is a finitely}\\ \textit{generated module}\end{array} \arrow[dl] \arrow[dr] & \\
\begin{array}{c}\text{Set}\\ \textit{M is an ordinary}\\ \textit{finitely generated module}\end{array} & & \begin{array}{c}\text{Sheaves on } X\\ \textit{M is a sheaf of modules}\\ \textit{of finite type}\end{array}
\end{tikzcd}$$

Type Theory for the Working Mathematician

Some ∞-toposes:

- Spaces – homotopy theory
- ∞-sheaves – derived algebraic geometry
- Smooth spaces – synthetic differential geometry

Some ∞-toposes:

- Spaces – homotopy theory
- ∞-sheaves – derived algebraic geometry
- Smooth spaces – synthetic differential geometry

Type Theory
Freudenthal
Suspension Theorem

Type Theory for the Working Mathematician

Some ∞-toposes:

- Spaces – homotopy theory
- ∞-sheaves – derived algebraic geometry
- Smooth spaces – synthetic differential geometry

Type Theory
Freudenthal
Suspension Theorem

Spaces
Freudenthal
Suspension Theorem

Type Theory for the Working Mathematician

Some ∞-toposes:

- Spaces – homotopy theory
- ∞-sheaves – derived algebraic geometry
- Smooth spaces – synthetic differential geometry

Type Theory
Freudenthal
Suspension Theorem

Spaces
Freudenthal
Suspension Theorem

∞-Sheaves on X
Freudenthal
Suspension Theorem

Type Theory for the Working Mathematician

Some ∞-toposes:

- Spaces – homotopy theory
- ∞-sheaves – derived algebraic geometry
- Smooth spaces – synthetic differential geometry
- Parameterised spectra – stable homotopy theory

Type Theory
Freudenthal
Suspension Theorem

Spaces
Freudenthal
Suspension Theorem

∞-Sheaves on X
Freudenthal
Suspension Theorem

Overview

- Doing mathematics in type theory
- Spectra and the ∞-topos of parameterised spectra
- A type theory that interprets into parameterised spectra

Type Theories

Products

Suppose we have two arbitrary sets A and B. We can build a function, say

$$f : A \times B \to A \times (B \times A)$$

Products

Suppose we have two arbitrary sets A and B. We can build a function, say

$$f : A \times B \to A \times (B \times A)$$

by writing

$$f(x, y) := (x, (y, x))$$

Products

Suppose we have two arbitrary sets A and B. We can build a function, say

$$f : A \times B \to A \times (B \times A)$$

by writing

$$f(x, y) := (x, (y, x))$$

Or:

$$A \times B \xrightarrow{\Delta \times B} (A \times A) \times B \xrightarrow{\alpha} A \times (A \times B) \xrightarrow{A \times s} A \times (B \times A)$$

Products

Suppose we have two arbitrary sets A and B. We can build a function, say

$$f : A \times B \to A \times (B \times A)$$

by writing

$$f(x, y) := (x, (y, x))$$

Or:

$$A \times B \xrightarrow{\Delta \times B} (A \times A) \times B \xrightarrow{\alpha} A \times (A \times B) \xrightarrow{A \times s} A \times (B \times A)$$

Type theory lets us mechanically convert from the former version to the latter.

Judgements and Rules

- If 'A type', then A is an object of the category
- If '$\Gamma \vdash a : A$', where Γ is a list of assumptions

  $$x_1 : X_1, x_2 : X_2, \ldots, x_n : X_n$$

  then there is a map

  $$a : X_1 \times X_2 \times \cdots \times X_n \to A$$

The rules look like:

$$\textsc{rule-name}\ \frac{\mathcal{J}_1 \qquad \ldots \qquad \mathcal{J}_n \qquad \text{(premises)}}{\mathcal{J} \qquad \text{(conclusion)}}$$

Products

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

Products

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

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

Products

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

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

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

Products

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

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

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

$$\times\text{-}\textsc{elim}\ \frac{\Gamma \vdash p : A \times B \qquad \Gamma, x : A, y : B \vdash c : C}{\Gamma \vdash \mathsf{let}\,(x, y) = p \,\mathsf{in}\, c : C}$$

and some equations.

Products

The function from before:

$$
\dfrac{\dfrac{x : A, y : B \vdash x : A \qquad \dfrac{x : A, y : B \vdash y : B \qquad x : A, y : B \vdash x : A}{x : A, y : B \vdash (y, x) : B \times A}\ \times\text{-INTRO}}{p : A \times B, x : A, y : B \vdash (x, (y, x)) : A \times (B \times A)}\ \times\text{-INTRO}}{p : A \times B \vdash \mathsf{let}\,(x, y) = p\,\mathsf{in}\,(x, (y, x)) : A \times (B \times A)}\ \times\text{-ELIM}
$$

Products

The function from before:

$$\dfrac{x : A, y : B \vdash x : A \qquad \dfrac{x : A, y : B \vdash y : B \qquad x : A, y : B \vdash x : A}{x : A, y : B \vdash (y, x) : B \times A}\ {\times\text{-INTRO}}}{\dfrac{p : A \times B, x : A, y : B \vdash (x, (y, x)) : A \times (B \times A)}{p : A \times B \vdash \mathsf{let}\,(x, y) = p\,\mathsf{in}\,(x, (y, x)) : A \times (B \times A)}\ {\times\text{-ELIM}}}\ {\times\text{-INTRO}}$$

Theorem

The rules on the previous slide present the free category-with-products on a set of objects.

Functions

For any two sets A, B, there is a set of functions $A \to B$.

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

$$\to\text{-INTRO}\ \frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda x.b : A \to B} \qquad \to\text{-ELIM}\ \frac{\Gamma \vdash f : A \to B \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B}$$

(and again some equations)

This kind of *exponential object* exists in any *cartesian closed category* (sets, nice spaces, sheaves, . . .).

Dependent Type Theory

Type theory can be made more powerful by allowing types to *depend* on terms.

Dependent Type Theory

Type theory can be made more powerful by allowing types to *depend* on terms.

Example

The set of days in a month depends on which month we are talking about:

$$x : \mathsf{Month} \vdash \mathsf{DayOf}(x)\ \mathsf{type}$$

Dependent Type Theory

Type theory can be made more powerful by allowing types to *depend* on terms.

Example

The set of days in a month depends on which month we are talking about:

$$x : \mathsf{Month} \vdash \mathsf{DayOf}(x) \text{ type}$$

Example

Each point of a differentiable manifold has a tangent space:

$$x : M \vdash T_x M \text{ type}$$

Dependent Type Theory

The product type can be generalised to *dependent* pairs:

$$\Sigma\text{-}\textsc{form}\ \frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash (x : A) \times B(x) \text{ type}}$$

$$\Sigma\text{-}\textsc{intro}\ \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B(a)}{\Gamma \vdash (a, b) : (x : A) \times B(x)}$$

$$\ldots$$

Dependent Type Theory

The product type can be generalised to *dependent* pairs:

$$\Sigma\text{-}\textsc{form}\ \frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash (x : A) \times B(x) \text{ type}}$$

$$\Sigma\text{-}\textsc{intro}\ \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B(a)}{\Gamma \vdash (a, b) : (x : A) \times B(x)}$$

$$\ldots$$

Example

The dependent pair type $(x : \mathsf{Month}) \times \mathsf{DayOf}(x)$ is type of all days in the year.

Dependent Type Theory

The product type can be generalised to *dependent* pairs:

$$\Sigma\text{-FORM}\ \frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash (x : A) \times B(x) \text{ type}}$$

$$\Sigma\text{-INTRO}\ \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B(a)}{\Gamma \vdash (a, b) : (x : A) \times B(x)}$$

$$\ldots$$

Example

The dependent pair type $(x : \mathrm{Month}) \times \mathrm{DayOf}(x)$ is type of all days in the year.

The dependent pair type $(x : M) \times T_x M$ is the tangent bundle TM.

Dependent Type Theory

Similarly for *dependent* functions:

$$\Pi\text{-}\textsc{form}\ \frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash (x : A) \to B(x) \text{ type}}$$

$$\Pi\text{-}\textsc{elim}\ \frac{\Gamma \vdash f : (x : A) \to B(x) \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B(a)}$$

$$\ldots$$

Dependent Type Theory

Similarly for *dependent* functions:

$$\Pi\text{-}\textsc{form}\ \frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash (x : A) \to B(x) \text{ type}}$$

$$\Pi\text{-}\textsc{elim}\ \frac{\Gamma \vdash f : (x : A) \to B(x) \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B(a)}$$

$$\dots$$

Example

The dependent function type $(x : \text{Month}) \to \text{DayOf}(x)$ is a choice of one day from each month.

Dependent Type Theory

Similarly for *dependent* functions:

$$\Pi\text{-}\textsc{form}\ \frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash (x : A) \to B(x) \text{ type}}$$

$$\Pi\text{-}\textsc{elim}\ \frac{\Gamma \vdash f : (x : A) \to B(x) \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B(a)}$$

$$\cdots$$

Example

The dependent function type $(x : \text{Month}) \to \text{DayOf}(x)$ is a choice of one day from each month.

The dependent function type $(x : M) \to T_x M$ is a vector field.
(sort of, one would need to think carefully about continuity)

Homotopy Type Theory

The category of simplicial sets can handle all of the above structure, with all constructions automatically continuous.

The category of simplicial sets can handle all of the above structure, with all constructions automatically continuous. We can add new types that let us talk about the spacial information:

$$\text{Path-}\textsc{form}\ \frac{\Gamma \vdash a : A \qquad \Gamma \vdash a' : A}{\Gamma \vdash \mathsf{Path}_A(a, a')\ \mathsf{type}}$$

Homotopy Type Theory

The category of simplicial sets can handle all of the above structure, with all constructions automatically continuous. We can add new types that let us talk about the spacial information:

$$\text{Path-}\textsc{form}\ \frac{\Gamma \vdash a : A \qquad \Gamma \vdash a' : A}{\Gamma \vdash \mathsf{Path}_A(a, a')\ \mathsf{type}}$$

$$\text{Path-}\textsc{intro}\ \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}_a : \mathsf{Path}_A(a, a)}$$

The category of simplicial sets can handle all of the above structure, with all constructions automatically continuous. We can add new types that let us talk about the spacial information:

$$\text{Path-}\textsc{form}\ \frac{\Gamma \vdash a : A \qquad \Gamma \vdash a' : A}{\Gamma \vdash \mathsf{Path}_A(a, a')\ \mathsf{type}}$$

$$\text{Path-}\textsc{intro}\ \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}_a : \mathsf{Path}_A(a, a)}$$

$$\text{Path-}\textsc{elim}\ \frac{\begin{array}{c}\Gamma, x : A, x' : A, z : \mathsf{Path}_A(x, x') \vdash C\ \mathsf{type} \\ \Gamma, x : A \vdash c : C[x/x', \mathsf{refl}_x/p] \\ \Gamma \vdash p : \mathsf{Path}_A(a, a')\end{array}}{\Gamma \vdash \mathsf{ind}(z.c, a, a', p) : C[a/x, a'/x', p/z]}$$

Homotopy Type Theory

Definition

A type A is *contractible* if there is a term of the type

$$\mathrm{isContr}(A) := (c : A) \times ((x : A) \to \mathsf{Path}_A(c, x))$$

(Don't worry, this doesn't mean just path-connected!)

Homotopy Type Theory

Definition

A type A is *contractible* if there is a term of the type

$$\mathrm{isContr}(A) := (c : A) \times ((x : A) \to \mathsf{Path}_A(c, x))$$

(Don't worry, this doesn't mean just path-connected!)

Definition

The *homotopy fiber* of a function $f : A \to B$ over a point $b : B$ is

$$\mathrm{hfib}_f(b) := (x : A) \times Path_B(f(x), b)$$

Homotopy Type Theory

Definition

A type A is *contractible* if there is a term of the type

$$\mathrm{isContr}(A) := (c : A) \times ((x : A) \to \mathsf{Path}_A(c, x))$$

(Don't worry, this doesn't mean just path-connected!)

Definition

The *homotopy fiber* of a function $f : A \to B$ over a point $b : B$ is

$$\mathrm{hfib}_f(b) := (x : A) \times \mathit{Path}_B(f(x), b)$$

Definition

A function is an *equivalence* if the homotopy fiber over every point is contractible:

$$\mathrm{isEquiv}(f) := (b : B) \to \mathrm{isContr}(\mathrm{hfib}_f(b))$$

Homotopy Type Theory

With a few more type formers (some higher inductive types, univalent universes) the system is called Homotopy Type Theory.

Homotopy Type Theory

With a few more type formers (some higher inductive types, univalent universes) the system is called Homotopy Type Theory.

Some synthetic results:

- Some homotopy groups of spheres (Shulman, Brunerie, Licata)
- Freudenthal Suspension Theorem (Lumsdaine, Licata)
- Localisation (Chrsitensen, Opie, Rijke, Scoccola)
- Blakers–Massey Theorem (Anel, Biedermann, Finster, Joyal)
- Serre Spectral Sequence (Avigad, Awodey, Buchholtz, Rijke, Shulman, van Doorn)

Homotopy Type Theory

Theorem (Kapulkin and Lumsdaine 2012, after Voevodsky)

The Kan model structure on the category of simplicial sets admits a model of HoTT.

Homotopy Type Theory

Theorem (Kapulkin and Lumsdaine 2012, after Voevodsky)

The Kan model structure on the category of simplicial sets admits a model of HoTT.

Theorem (Shulman 2019)

Every ∞-topos can be presented by a model category that admits a model of HoTT. (modulo universes being closed under HITs, expected to be true)

Homotopy Type Theory

Theorem (Kapulkin and Lumsdaine 2012, after Voevodsky)

The Kan model structure on the category of simplicial sets admits a model of HoTT.

Theorem (Shulman 2019)

Every ∞-topos can be presented by a model category that admits a model of HoTT. (modulo universes being closed under HITs, expected to be true)

Types in the theory become categorical constructions:

Γ ctx	Objects Γ
$\Gamma \vdash A$ type	Fibrations $A \twoheadrightarrow \Gamma$
Σ and Π types	Adjoints to pullback functors $\mathcal{C}/\Gamma \to \mathcal{C}/A$
Path types	Path space fibration
...	...

Spectra and Parameterised Spectra

Motivating Spectra

Theorem

Singular cohomology is representable: for any abelian group G and pointed CW-complex X,

$$\tilde{H}^n(X; G) \cong [X, K(G, n)]_{\mathrm{pt}}$$

where $K(G, n)$ is an Eilenberg-MacLane space.

Motivating Spectra

Definition (Eilenberg–Steenrod axioms)

A *reduced cohomology theory* is a sequence of functors

$$\tilde{E}^n : \begin{pmatrix} \text{pointed connected CW-complexes} \\ \text{up to homotopy} \end{pmatrix}^{\mathrm{op}} \to (\text{abelian groups})$$

such that

1. Wedge sums are taken to products; and,
2. For each CW-pair (X, A), the sequence
$$\tilde{E}^n(X/A) \to \tilde{E}^n(X) \to \tilde{E}^n(A)$$
is exact.
3. There is a natural isomorphism $\tilde{E}^n(X) \cong \tilde{E}^{n+1}(\Sigma X)$.

Motivating Spectra

Theorem (Brown Representability)

For any reduced cohomology theory $\tilde{E}^*$, there is a sequence of pointed connected CW-complexes K_n so that

$$\tilde{E}^n(X) \cong [X, K_n]_{\mathrm{pt}}$$

naturally in X.

Motivating Spectra

Theorem (Brown Representability)

For any reduced cohomology theory $\tilde{E}^*$, there is a sequence of pointed connected CW-complexes K_n so that

$$\tilde{E}^n(X) \cong [X, K_n]_{\mathrm{pt}}$$

naturally in X.

A sequence $\{K_n\}$ does not quite determine a cohomology theory by itself: we are missing the suspension isomorphisms.

Motivating Spectra

For any X we have a natural isomorphism:

$$[X, K_n]_{\mathrm{pt}} \cong \tilde{E}^n(X) \cong \tilde{E}^{n+1}(\Sigma X) \cong [\Sigma X, K_{n+1}]_{\mathrm{pt}} \cong [X, \Omega K_{n+1}]_{\mathrm{pt}}$$

The image of the identity map on K_n is a map $\alpha_n : K_n \to \Omega K_{n+1}$.

Motivating Spectra

For any X we have a natural isomorphism:

$$[X, K_n]_{\mathrm{pt}} \cong \tilde{E}^n(X) \cong \tilde{E}^{n+1}(\Sigma X) \cong [\Sigma X, K_{n+1}]_{\mathrm{pt}} \cong [X, \Omega K_{n+1}]_{\mathrm{pt}}$$

The image of the identity map on K_n is a map $\alpha_n : K_n \to \Omega K_{n+1}$.

Letting X vary over the spheres S^k, we see in fact α_n is a weak equivalence.

Motivating Spectra

Definition

A *spectrum* is a sequence of pointed connected spaces $\{K_n\}_{n\in\mathbb{N}}$ together with weak equivalences $\alpha_n : K_n \to \Omega K_{n+1}$.

Motivating Spectra

Definition

A *spectrum* is a sequence of pointed connected spaces $\{K_n\}_{n\in\mathbb{N}}$ together with weak equivalences $\alpha_n : K_n \to \Omega K_{n+1}$.

Example

Each abelian group yields a spectrum with $K_n = K(G, n)$

Motivating Spectra

Definition

A *spectrum* is a sequence of pointed connected spaces $\{K_n\}_{n\in\mathbb{N}}$ together with weak equivalences $\alpha_n : K_n \to \Omega K_{n+1}$.

Example

Each abelian group yields a spectrum with $K_n = K(G, n)$

Example

The *zero spectrum* with $K_n = \{\star\}$.

Motivating Spectra

Definition

A *spectrum* is a sequence of pointed connected spaces $\{K_n\}_{n\in\mathbb{N}}$ together with weak equivalences $\alpha_n : K_n \to \Omega K_{n+1}$.

Example

Each abelian group yields a spectrum with $K_n = K(G, n)$

Example

The *zero spectrum* with $K_n = \{\star\}$.

Example

The *sphere spectrum*, whose homotopy groups are the stable homotopy groups of spheres

Option 1: Spectra in Type Theory

Spectra can be defined almost verbatim in HoTT:

$$\begin{aligned}\mathrm{Spectrum} :&= (K : (\mathbb{N} \to \mathrm{PtdType})) \\ &\times ((n : \mathbb{N}) \to \mathrm{Equiv}(K(n), \Omega K(n+1)))\end{aligned}$$

Option 1: Spectra in Type Theory

Spectra can be defined almost verbatim in HoTT:

$$\begin{aligned}\mathrm{Spectrum} := &(K : (\mathbb{N} \to \mathrm{PtdType})) \\ &\times ((n : \mathbb{N}) \to \mathrm{Equiv}(K(n), \Omega K(n+1)))\end{aligned}$$

An important operation on spectra is the *smash product*. Recall the smash product of pointed spaces:

$$A \wedge B := (A \times B)/(A \vee B)$$

Even showing this is associative in HoTT is a task! (van Doorn 2018)

Option 1: Spectra in Type Theory

Idea: Instead of rebuilding spectra inside type theory, model type theory in a category where they already exist.

Option 1: Spectra in Type Theory

Idea: Instead of rebuilding spectra inside type theory, model type theory in a category where they already exist.

The category of spectra is lousy for modelling type theory.

- Yes: dependent pair type, path type
- No: everything else

Option 2: A Model in Parameterised Spectra

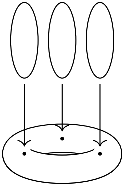
“Definition”

A *parameterised spectrum* is a bundle of spectra over a space.

Option 2: A Model in Parameterised Spectra

"Definition"

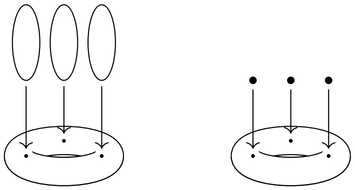
A *parameterised spectrum* is a bundle of spectra over a space.



Option 2: A Model in Parameterised Spectra

"Definition"

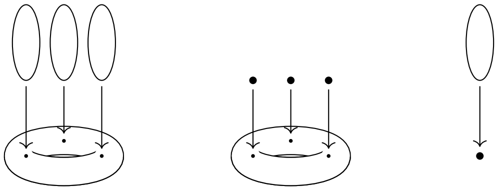
A *parameterised spectrum* is a bundle of spectra over a space.



Option 2: A Model in Parameterised Spectra

"Definition"

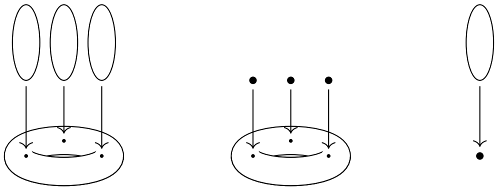
A *parameterised spectrum* is a bundle of spectra over a space.



Option 2: A Model in Parameterised Spectra

"Definition"

A *parameterised spectrum* is a bundle of spectra over a space.



Theorem (Joyal 2008)

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

So is a model of HoTT.

A Type Theory for Parameterised Spectra

Stable Homotopy Type Theory?

We now think of our types as spaces + extra spectral information over every point.

Stable Homotopy Type Theory?

We now think of our types as spaces + extra spectral information over every point.

$$
\begin{array}{ccc}
\text{Homotopy Type Theory} & & \\
& \searrow & \\
& & P\mathsf{Spec}
\end{array}
$$

Stable Homotopy Type Theory?

We now think of our types as spaces + extra spectral information over every point.

```tikzcd
& \text{Homotopy Type Theory} \arrow[dl] \arrow[dr] & \\
\text{Spaces} & & P\text{Spec}
\end{tikzcd}
```

Stable Homotopy Type Theory?

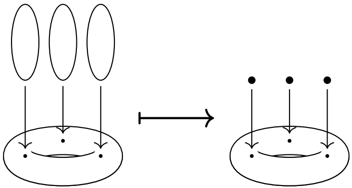
We now think of our types as spaces + extra spectral information over every point.

$$\begin{tikzcd} & \text{Homotopy Type Theory} \arrow[dl] \arrow[dr] & \\ \mathsf{Spaces} & & P\mathsf{Spec} \end{tikzcd}$$

We need to figure out how to add new type formers that give access to that structure.

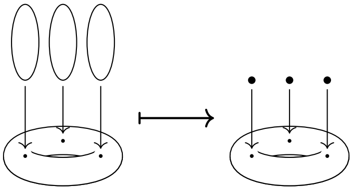
Underlying Space

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



Underlying Space

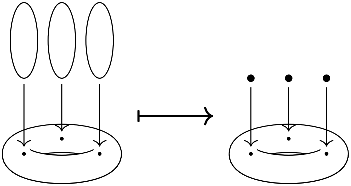
For every type A there should be a type $\natural A$ that deletes the spectral information.



$$\textsc{var-zero}\ \frac{}{\Gamma, x : A, \Gamma' \vdash x^0 : A^0}$$

Underlying Space

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



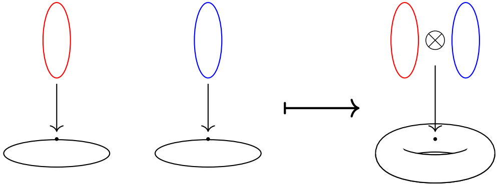
$$\textsc{var-zero}\ \frac{}{\Gamma, x : A, \Gamma' \vdash x^0 : A^0}$$

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

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

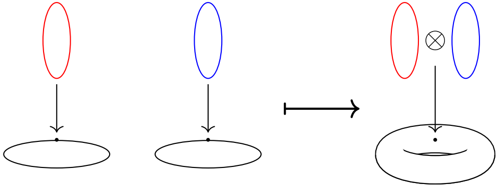
Smash Product

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



Smash Product

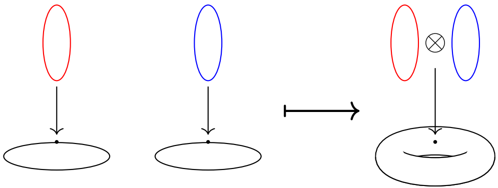
For two types A and B, there should be a type $A \otimes B$ that corresponding to the 'external smash product'.



$$\otimes\text{-INTRO}\ \frac{\Gamma^0, \Omega, \Omega'^0, \Gamma'^0 \vdash a : A \qquad \Gamma^0, \Omega^0, \Omega', \Gamma'^0 \vdash b : B}{\Gamma, (\Omega)(\Omega'), \Gamma' \vdash a \otimes b : A \otimes B}$$

Smash Product

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



$$\otimes\text{-INTRO}\ \frac{\Gamma^0, \Omega, \Omega'^0, \Gamma'^0 \vdash a : A \qquad \Gamma^0, \Omega^0, \Omega', \Gamma'^0 \vdash b : B}{\Gamma, (\Omega)(\Omega'), \Gamma' \vdash a \otimes b : A \otimes B}$$

$$\otimes\text{-ELIM}\ \frac{\begin{array}{c}\Gamma, z : A \otimes B \vdash C \text{ type} \\ \Gamma, (x : A)(y : B) \vdash c : C[x \otimes y/z] \\ \Gamma \vdash s : A \otimes B\end{array}}{\Gamma \vdash \text{let } x \otimes y = s \text{ in } c : C[s/z]}$$

More Formers

- The sphere spectrum $\mathbb{S}$: the monoidal unit for $\otimes$

More Formers

- The sphere spectrum $\mathbb{S}$: the monoidal unit for $\otimes$
- Hom types $A \multimap B$: right adjoint to $- \otimes A$

Progress

What's done:

- Judgemental structure
- Type formers and their interactions with the context

Combining dependent types and 'linear' features is difficult! And interesting!

Progress

What's done:

- Judgemental structure
- Type formers and their interactions with the context

Combining dependent types and 'linear' features is difficult! And interesting!

What's left:

- Check all the admissible rules work
- Actually use it!
- Describe intended semantics more precisely
- Code up a type-checker?

References I

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

Kapulkin, Chris and Peter LeFanu Lumsdaine (2012). "The simplicial model of univalent foundations (after Voevodsky)". In: *arXiv preprint arXiv:1211.2851*.

Shulman, Michael (2019). "All $(\infty, 1)$-toposes have strict univalent universes". In: *arXiv preprint arXiv:1904.07004*.

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