A Model of Type Theory with Directed Univalence in Bicubical Sets

Matthew Weaver and Dan Licata

Princeton University Wesleyan University

HoTT. August 14, 2019

Directed Type Theory

Directed Type Theory

- Riehl-Shulman defines a type theory for ∞-categories with a model bisimplicial sets

Directed Type Theory

- Riehl-Shulman defines a type theory for ∞-categories with a model bisimplicial sets
    1. Begin with HoTT

Directed Type Theory

- Riehl-Shulman defines a type theory for ∞-categories with a model bisimplicial sets
  1. Begin with HoTT
  2. Add Hom-types

Directed Type Theory

- Riehl-Shulman defines a type theory for ∞-categories with a model bisimplicial sets
    1. Begin with HoTT
    2. Add Hom-types
    3. ∞-categories (Segal types) and univalent ∞-category (Rezk types) given internally as predicates on types

Directed Type Theory

- Riehl-Shulman defines a type theory for ∞-categories with a model bisimplicial sets
  1. Begin with HoTT
  2. Add Hom-types
  3. ∞-categories (Segal types) and univalent ∞-category (Rezk types) given internally as predicates on types
  4. Predicate isCov(B : A → U) for covariant discrete fibrations

Directed Type Theory

- Riehl-Shulman defines a type theory for ∞-categories with a model bisimplicial sets
  1. Begin with HoTT
  2. Add Hom-types
  3. ∞-categories (Segal types) and univalent ∞-category (Rezk types) given internally as predicates on types
  4. Predicate isCov($B : A \to U$) for covariant discrete fibrations
  5. Cavallo, Riehl and Sattler have also (externally) defined the universe of covariant fibrations (the ∞-category of spaces and continuous functions) and shown *Directed Univalence:* $\mathrm{Hom}_{UCov}\, A\; B \simeq A \to B$

Constructive(?) Directed Type Theory

Constructive(?) Directed Type Theory

- Can we make this constructive?

Constructive(?) Directed Type Theory

- Can we make this constructive?
    1. Begin with Cubical Type Theory

Constructive(?) Directed Type Theory

- Can we make this constructive?
  1. Begin with Cubical Type Theory
  2. Use a second cubical interval to define Hom-types

Constructive(?) Directed Type Theory

- Can we make this constructive?
  1. Begin with Cubical Type Theory
  2. Use a second cubical interval to define Hom-types
  3. Use LOPS to define universe of covariant fibrations and construct directed univalence internally...

Constructive(?) Directed Type Theory

- Can we make this constructive?
  1. Begin with Cubical Type Theory
  2. Use a second cubical interval to define Hom-types
  3. Use LOPS to define universe of covariant fibrations and construct directed univalence internally...
     - ...unfortunately, directed univalence is a bit trickier than expected

Let's see how far the techniques from cubical type theory get us!

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

Directed Type Theory

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT

Directed Type Theory

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT

2. Add an interval: $\mathbb{I}$

Directed Type Theory

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT

2. Add an interval: $\mathbb{I}$

3. Specify gen. cofibrations for $\mathbb{I}$

Directed Type Theory

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

Directed Type Theory

1. Begin with MLTT

2. Add an interval: $\mathbb{I}$

3. Specify gen. cofibrations for $\mathbb{I}$

4. Define filling problem for Kan fibrations

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT

2. Add an interval: $\mathbb{I}$

3. Specify gen. cofibrations for $\mathbb{I}$

4. Define filling problem for Kan fibrations

5. Define universe of Kan fibrations

Directed Type Theory

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
1. Begin with MLTT	
2. Add an interval: $\mathbb{I}$	
3. Specify gen. cofibrations for $\mathbb{I}$	
4. Define filling problem for Kan fibrations	
5. Define universe of Kan fibrations	
6. Construct univalence	

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. Define filling problem for Kan fibrations
5. Define universe of Kan fibrations
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. Define filling problem for Kan fibrations
5. Define universe of Kan fibrations
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. Define filling problem for Kan fibrations
5. Define universe of Kan fibrations
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
1. Begin with MLTT	1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{I}$	2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{I}$	3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for Kan fibrations	4. Define filling problem for covariant fibrations
5. Define universe of Kan fibrations	
6. Construct univalence	

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. Define filling problem for Kan fibrations
5. Define universe of Kan fibrations
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for covariant fibrations
5. Define universe of covariant fibrations

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
1. Begin with MLTT	1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{I}$	2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{I}$	3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for Kan fibrations	4. Define filling problem for covariant fibrations
5. Define universe of Kan fibrations	5. Define universe of covariant fibrations
6. Construct univalence	6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
1. Begin with MLTT	1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{I}$	**2. Add an interval:** $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{I}$	3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for Kan fibrations	4. Define filling problem for covariant fibrations
5. Define universe of Kan fibrations	5. Define universe of covariant fibrations
6. Construct univalence	6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
2. Add an interval: $\mathbb{I}$	**2. Add an interval:** $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
2. Add an interval: $\mathbb{I}$	**2. Add an interval: $\mathbb{2}$**
$$\frac{}{\mathbb{I} : \text{Type}}$$	

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

2. Add an interval: $\mathbb{I}$

$$\frac{}{\mathbb{I} : \text{Type}}$$

$$\frac{}{\mathbb{0}_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{\mathbb{1}_{\mathbb{I}} : \mathbb{I}}$$

Directed Type Theory

2. Add an interval: $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

2. Add an interval: $\mathbb{I}$

$$\frac{}{\mathbb{I} : \text{Type}}$$

$$\frac{}{\mathbb{0}_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{\mathbb{1}_{\mathbb{I}} : \mathbb{I}}$$

i.e. generators for the Cartesian cubes

Directed Type Theory

2. Add an interval: $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

2. Add an interval: $\mathbb{I}$

$$\frac{}{\mathbb{I} : \text{Type}}$$

$$\frac{}{\mathbb{0}_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{\mathbb{1}_{\mathbb{I}} : \mathbb{I}}$$

i.e. generators for the Cartesian cubes

Directed Type Theory

2. Add an interval: $\mathbb{2}$

$$\frac{}{\mathbb{2} : \text{Type}}$$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

2. Add an interval: $\mathbb{I}$

$$\frac{}{\mathbb{I} : \mathrm{Type}}$$

$$\frac{}{\mathbb{0}_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{\mathbb{1}_{\mathbb{I}} : \mathbb{I}}$$

i.e. generators for the Cartesian cubes

Directed Type Theory

2. Add an interval: $\mathbb{2}$

$$\frac{}{\mathbb{2} : \mathrm{Type}}$$

$$\frac{}{\mathbb{0}_{\mathbb{2}} : \mathbb{2}} \qquad \frac{}{\mathbb{1}_{\mathbb{2}} : \mathbb{2}}$$

$$\frac{x : \mathbb{2} \quad y : \mathbb{2}}{x \wedge y : \mathbb{2}} \qquad \frac{x : \mathbb{2} \quad y : \mathbb{2}}{x \vee y : \mathbb{2}}$$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

2. Add an interval: $\mathbb{I}$

$$\frac{}{\mathbb{I} : \mathrm{Type}}$$

$$\frac{}{0_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{1_{\mathbb{I}} : \mathbb{I}}$$

i.e. generators for the Cartesian cubes

Directed Type Theory

2. Add an interval: $\mathbb{2}$

$$\frac{}{\mathbb{2} : \mathrm{Type}}$$

$$\frac{}{0_{\mathbb{2}} : \mathbb{2}} \qquad \frac{}{1_{\mathbb{2}} : \mathbb{2}}$$

$$\frac{x : \mathbb{2} \quad y : \mathbb{2}}{x \wedge y : \mathbb{2}} \qquad \frac{x : \mathbb{2} \quad y : \mathbb{2}}{x \vee y : \mathbb{2}}$$

and equations...

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

2. Add an interval: $\mathbb{I}$

$$\frac{}{\mathbb{I} : \mathrm{Type}}$$

$$\frac{}{\mathbb{0}_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{\mathbb{1}_{\mathbb{I}} : \mathbb{I}}$$

i.e. generators for the Cartesian cubes

Directed Type Theory

2. Add an interval: $\mathbb{2}$

$$\frac{}{\mathbb{2} : \mathrm{Type}}$$

$$\frac{}{\mathbb{0}_{\mathbb{2}} : \mathbb{2}} \qquad \frac{}{\mathbb{1}_{\mathbb{2}} : \mathbb{2}}$$

$$\frac{x : \mathbb{2} \quad y : \mathbb{2}}{x \wedge y : \mathbb{2}} \qquad \frac{x : \mathbb{2} \quad y : \mathbb{2}}{x \vee y : \mathbb{2}}$$

and equations...

i.e. generators for the Dedekind cubes

The Directed Interval

The Directed Interval

- Why Dedekind cubes instead of Cartesian?
  $x \leq y := x = x \wedge y$

The Directed Interval

- Why Dedekind cubes instead of Cartesian?
  $x \leq y := x = x \wedge y$

- We also add the following axioms:

The Directed Interval

- Why Dedekind cubes instead of Cartesian?
  $x \leq y := x = x \wedge y$

- We also add the following axioms:

  - $p : \mathbb{I} \to \mathbb{2}$ is constant ($\Pi\ x\ y : \mathbb{I},\ p\ x = p\ y$)

The Directed Interval

- Why Dedekind cubes instead of Cartesian?
  $x \leq y := x = x \wedge y$

- We also add the following axioms:

  - $p : \mathbb{I} \to \mathbb{2}$ is constant (Π x y : $\mathbb{I}$, p x = p y)

  - $p : \mathbb{2} \to \mathbb{2}$ is monotone (Π x y : $\mathbb{2}$, if $x \leq y$ then $p\ x \leq p\ y$)

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. Define filling problem for Kan fibrations
5. Define universe of Kan fibrations
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for covariant fibrations
5. Define universe of covariant fibrations
6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
1. Begin with MLTT	1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{I}$	2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{I}$	**3. Specify gen. cofibrations for $\mathbb{2}$**
4. Define filling problem for Kan fibrations	4. Define filling problem for covariant fibrations
5. Define universe of Kan fibrations	5. Define universe of covariant fibrations
6. Construct univalence	6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

3. Specify gen. cofibrations for $\mathbb{I}$

Directed Type Theory

3. Specify gen. cofibrations for $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

3. Specify gen. cofibrations for $\mathbb{I}$

$$\frac{}{\mathsf{isCof} : \Omega \to \Omega}$$

$\mathsf{Cof} := \Sigma\, \phi : \Omega \,.\, \mathsf{isCof}\ \phi$

Directed Type Theory

3. Specify gen. cofibrations for $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

3. Specify gen. cofibrations for $\mathbb{I}$

$$\frac{}{\mathrm{isCof} : \Omega \to \Omega}$$

Cof := Σ ϕ : Ω . isCof ϕ

Cof closed under _∧_, _∨_, ⊥, ⊤

Directed Type Theory

3. Specify gen. cofibrations for $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

3. Specify gen. cofibrations for $\mathbb{I}$

$$\frac{}{\mathrm{isCof} : \Omega \to \Omega}$$

$\mathrm{Cof} := \Sigma\, \phi : \Omega \,.\, \mathrm{isCof}\ \phi$

Cof closed under $\_\wedge\_$, $\_\vee\_$, $\bot$, $\top$

$$\frac{x : \mathbb{I} \quad y : \mathbb{I}}{\_ : \mathrm{isCof}\ (x = y)}$$

$$\frac{\phi : \mathbb{I} \to \mathrm{Cof}}{\_ : \mathrm{isCof}\ (\Pi\, x : \mathbb{I} \,.\, \phi\ x)}$$

Directed Type Theory

3. Specify gen. cofibrations for $\mathbb{2}$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

3. Specify gen. cofibrations for $\mathbb{I}$

$$\frac{}{\mathsf{isCof} : \Omega \to \Omega}$$

$\mathsf{Cof} := \Sigma\, \phi : \Omega \,.\, \mathsf{isCof}\ \phi$

Cof closed under $\_\wedge\_$, $\_\vee\_$, $\bot$, $\top$

$$\frac{x : \mathbb{I} \quad y : \mathbb{I}}{\_ : \mathsf{isCof}\ (x = y)}$$

$$\frac{\phi : \mathbb{I} \to \mathsf{Cof}}{\_ : \mathsf{isCof}\ (\Pi\, x : \mathbb{I} \,.\, \phi\ x)}$$

Directed Type Theory

3. Specify gen. cofibrations for $\mathbb{2}$

$$\frac{x : \mathbb{2} \quad y : \mathbb{2}}{\_ : \mathsf{isCof}\ (x = y)}$$

$$\frac{\phi : \mathbb{2} \to \mathsf{Cof}}{\_ : \mathsf{isCof}\ (\Pi\, x : \mathbb{2} \,.\, \phi\ x)}$$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. Define filling problem for Kan fibrations
5. Define universe of Kan fibrations
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for covariant fibrations
5. Define universe of covariant fibrations
6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. **Define filling problem for Kan fibrations**
5. Define universe of Kan fibrations
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{2}$
4. **Define filling problem for covariant fibrations**
5. Define universe of covariant fibrations
6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

4. Define filling problem for Kan fibrations

Directed Type Theory

4. Define filling problem for covariant fibrations

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

4. Define filling problem for Kan fibrations

hasCom : ($\mathbb{I}$ → U) → U
hasCom A = Π i j : $\mathbb{I}$.
Π α : Cof .
Π t : (Π x : $\mathbb{I}$. α → A x)
Π b : (A i)[α ↦ t i] .
(A j)[α ↦ t j; i = j ↦ b]

relCom : (A : U) → (A → U) → U
relCom A B = Π p : $\mathbb{I}$ → A .
hasCom (B ∘ p)

Directed Type Theory

4. Define filling problem for covariant fibrations

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

4. Define filling problem for Kan fibrations

$$
\begin{aligned}
&\mathrm{hasCom} : (\mathbb{I} \to U) \to U \\
&\mathrm{hasCom}\ A = \Pi\ i\ j : \mathbb{I}\ . \\
&\qquad \Pi\ \alpha : \mathrm{Cof}\ . \\
&\qquad \Pi\ t : (\Pi\ x : \mathbb{I}\ .\ \alpha \to A\ x) \\
&\qquad \Pi\ b : (A\ i)[\alpha \mapsto t\ i]\ . \\
&\qquad (A\ j)[\alpha \mapsto t\ j;\ i = j \mapsto b]
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{relCom} : (A : U) \to (A \to U) \to U \\
&\mathrm{relCom}\ A\ B = \Pi\ p : \mathbb{I} \to A\ . \\
&\qquad \mathrm{hasCom}\ (B \circ p)
\end{aligned}
$$

Directed Type Theory

4. Define filling problem for covariant fibrations

$$
\begin{aligned}
&\mathrm{hasCov} : (\mathbb{2} \to U) \to U \\
&\mathrm{hasCov}\ A = \Pi\ \alpha : \mathrm{Cof}\ . \\
&\qquad \Pi\ t : (\Pi\ x : \mathbb{2}\ .\ \alpha \to A\ x) \\
&\qquad \Pi\ b : (A\ \mathbb{0}_{\mathbb{2}})[\alpha \mapsto t\ \mathbb{0}_{\mathbb{2}}]\ . \\
&\qquad (A\ \mathbb{1}_{\mathbb{2}})[\alpha \mapsto t\ \mathbb{1}_{\mathbb{2}}]
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{relCov} : (A : U) \to (A \to U) \to U \\
&\mathrm{relCov}\ A\ B = \Pi\ p : \mathbb{2} \to A\ . \\
&\qquad \mathrm{hasCov}\ (B \circ p)
\end{aligned}
$$

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT

2. Add an interval: $\mathbb{I}$

3. Specify gen. cofibrations for $\mathbb{I}$

4. Define filling problem for Kan fibrations

5. Define universe of Kan fibrations

6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory

2. Add an interval: $\mathbb{2}$

3. Specify gen. cofibrations for $\mathbb{2}$

4. Define filling problem for covariant fibrations

5. Define universe of covariant fibrations

6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

1. Begin with MLTT
2. Add an interval: $\mathbb{I}$
3. Specify gen. cofibrations for $\mathbb{I}$
4. Define filling problem for Kan fibrations
5. **Define universe of Kan fibrations**
6. Construct univalence

Directed Type Theory

1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for covariant fibrations
5. **Define universe of covariant fibrations**
6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

5. Define universe of Kan fibrations

Directed Type Theory

5. Define universe of covariant fibrations

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

5. Define universe of Kan fibrations

- U_{Kan} given by LOPS construction for relCom

Directed Type Theory

5. Define universe of covariant fibrations

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

5. Define universe of Kan fibrations

- U_{Kan} given by LOPS construction for relCom

Directed Type Theory

5. Define universe of covariant fibrations

- U_{Cov} given by LOPS construction for relCov. **Lemma**: relCov is in UKan

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
1. Begin with MLTT	1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{I}$	2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{I}$	3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for Kan fibrations	4. Define filling problem for covariant fibrations
5. Define universe of Kan fibrations	5. Define universe of covariant fibrations
6. Construct univalence	6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
1. Begin with MLTT	1. Begin with Cubical Type Theory
2. Add an interval: $\mathbb{I}$	2. Add an interval: $\mathbb{2}$
3. Specify gen. cofibrations for $\mathbb{I}$	3. Specify gen. cofibrations for $\mathbb{2}$
4. Define filling problem for Kan fibrations	4. Define filling problem for covariant fibrations
5. Define universe of Kan fibrations	5. Define universe of covariant fibrations
6. Construct univalence	**6. Construct directed univalence**

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

6. Construct univalence

Directed Type Theory

6. Construct directed univalence

Defining Bicubical Directed Type Theory

Cubical Type Theory *(in the style of Orton-Pitts)*	**Directed Type Theory**
6. Construct univalence	**6. Construct directed univalence**
• Key Idea: Glue type to attach equivalences to path structure	

Defining Bicubical Directed Type Theory

Cubical Type Theory
(in the style of Orton-Pitts)

6. Construct univalence

- Key Idea: Glue type to attach equivalences to path structure

Directed Type Theory

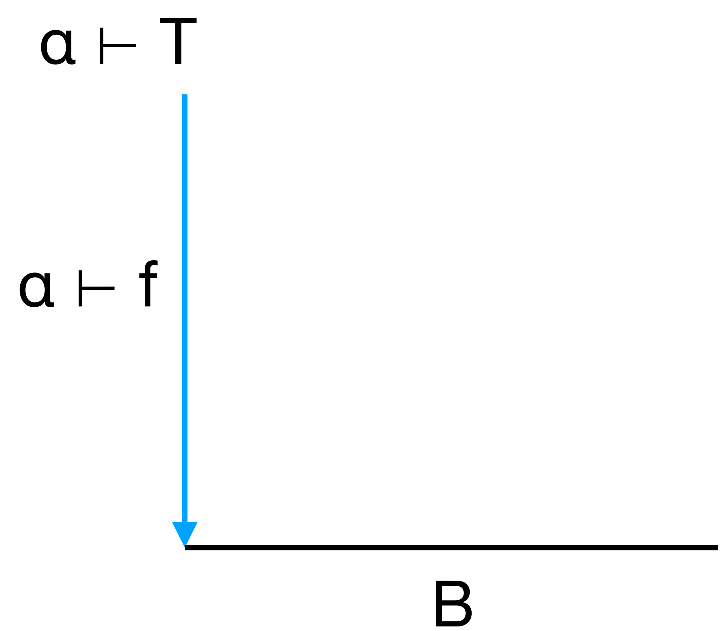
6. Construct directed univalence

- Key Idea: Glue type to attach ***functions*** to morphism structure

Glue Types

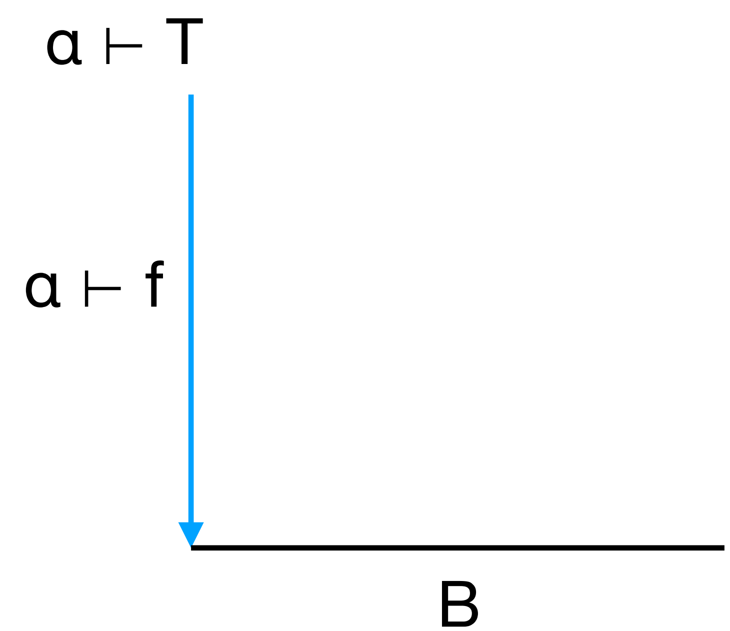
Glue Types

$\mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B :=$



Glue Types

Glue [$\alpha \mapsto (T , f)$] B :=



$\alpha \vdash$ Glue [$\alpha \mapsto (T , f)$] B $\equiv$ T

Glue Types

$\alpha \vdash T$

$\mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B :=$ $\quad \alpha \vdash f$

B

$\alpha \vdash \mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B \equiv T$

$\alpha \vdash t : T$

$\mathrm{glue}\ t\ b :=$ $\quad \alpha \vdash f$

$b : B$

Glue Types

$\alpha \vdash T$

Glue [$\alpha \mapsto (T , f)$] B :=

$\alpha \vdash f$

B

$\alpha \vdash \mathrm{Glue}\ [\ \alpha \mapsto (T , f)\ ]\ B \equiv T$

$\alpha \vdash t : T$

glue t b :=

$\alpha \vdash f$

b : B

$$\frac{g : \mathrm{Glue}\ [\ \alpha \mapsto (T , f)\ ]\ B}{\mathrm{unglue}\ g : B}$$

Glue Types

$\alpha \vdash T$

$\mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B :=$

$\alpha \vdash f$

B

$\alpha \vdash \mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B \equiv T$

$\alpha \vdash t : T$

$\mathrm{glue}\ t\ b :=$

$\alpha \vdash f$

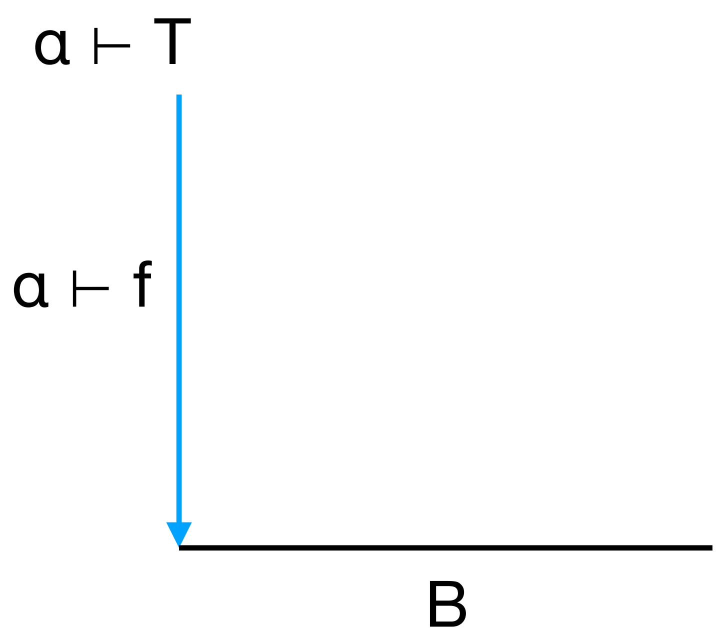
$b : B$

$$\frac{g : \mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B}{\mathrm{unglue}\ g : B}$$

$\alpha \vdash \mathrm{glue}\ t\ b \equiv t$

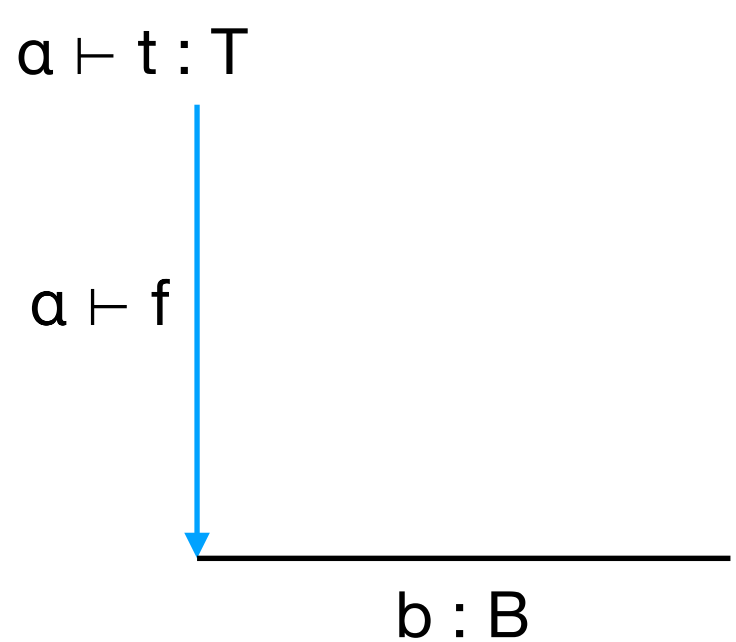
Glue Types

Glue [α ↦ (T , f)] B :=



$\alpha \vdash \text{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B \equiv T$

glue t b :=



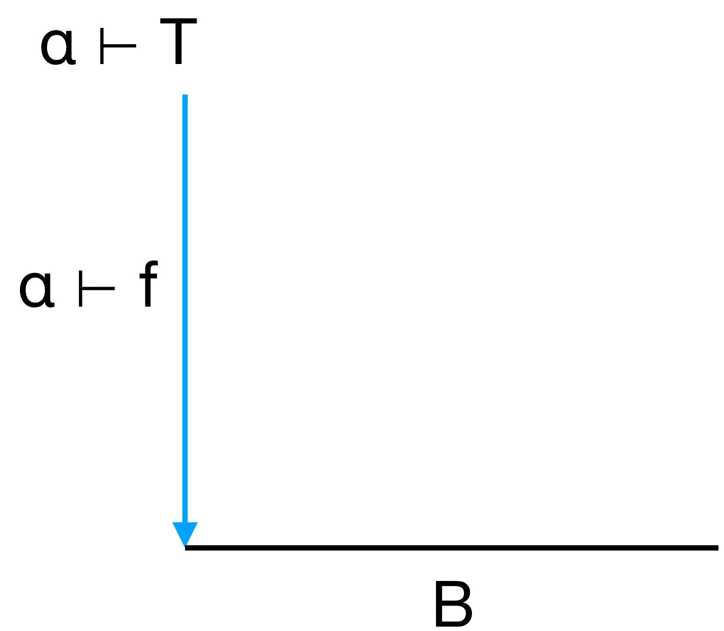
$$\frac{g : \text{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B}{\text{unglue}\ g : B}$$

$\alpha \vdash \text{glue}\ t\ b \equiv t$

$\alpha \vdash \text{unglue}\ (\text{glue}\ t\ b) \equiv f\ t$

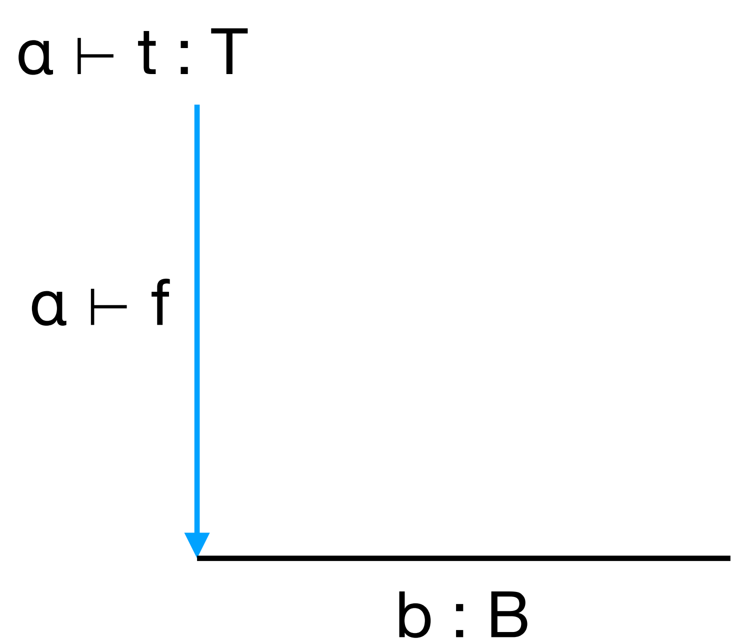
Glue Types

$\mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B :=$



$\alpha \vdash \mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B \equiv T$

$\mathrm{glue}\ t\ b :=$



$$\frac{g : \mathrm{Glue}\ [\ \alpha \mapsto (T\ ,\ f)\ ]\ B}{\mathrm{unglue}\ g : B}$$

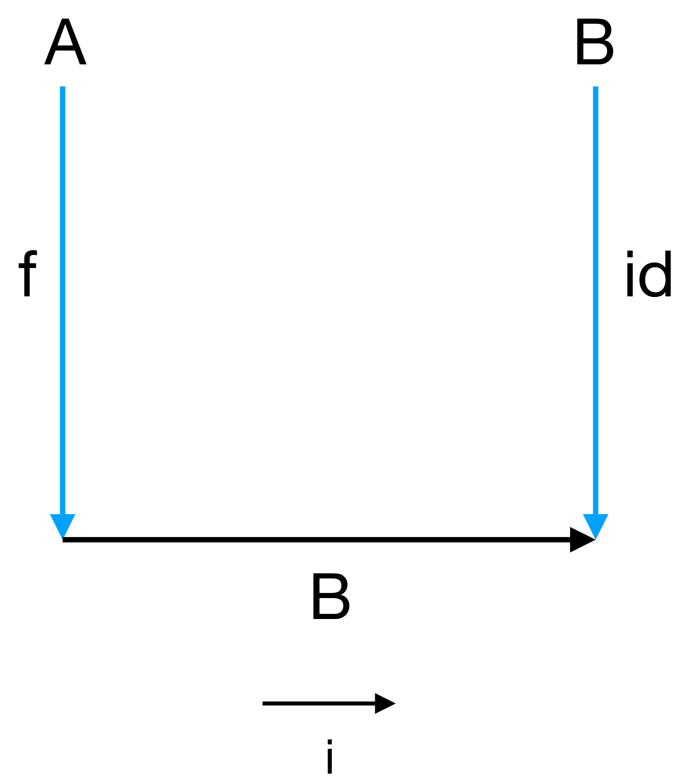
$\alpha \vdash \mathrm{glue}\ t\ b \equiv t$

$\alpha \vdash \mathrm{unglue}\ (\mathrm{glue}\ t\ b) \equiv f\ t$

$\mathrm{glue}\ g\ (\mathrm{unglue}\ g) \equiv g$

Defining
Directed Univalence

dua i A B f := Glue [i = 0 ↦ (A , f : A → B)
, i = 1 ↦ (B , id)] B : Hom_U A B



Naive Directed Univalence

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}
- U_{Cov} itself is Kan

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}
- U_{Cov} itself is Kan
- Path univalence holds in U_{Cov}

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}
- U_{Cov} itself is Kan
- Path univalence holds in U_{Cov}
- These allow us to define the following for U_{Cov}:

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}
- U_{Cov} itself is Kan
- Path univalence holds in U_{Cov}
- These allow us to define the following for U_{Cov}:
  - dcoe : (Hom A B) → (A → B)

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}
- U_{Cov} itself is Kan
- Path univalence holds in U_{Cov}
- These allow us to define the following for U_{Cov}:
  - dcoe : (Hom A B) → (A → B)
  - dua : (A → B) → Hom A B

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}
- U_{Cov} itself is Kan
- Path univalence holds in U_{Cov}
- These allow us to define the following for U_{Cov}:
  - dcoe : (Hom A B) → (A → B)
  - dua : (A → B) → Hom A B
  - dua_{β} : Π f : A → B . Path f (dcoe (dua f))

Naive Directed Univalence

- dua is Kan + covariant, and thus lands in U_{Cov}
- U_{Cov} itself is Kan
- Path univalence holds in U_{Cov}
- These allow us to define the following for U_{Cov}:
  - dcoe : (Hom A B) → (A → B)
  - dua : (A → B) → Hom A B
  - dua_{β} : Π f : A → B . Path f (dcoe (dua f))
  - $dua_{\eta fun}$: Π p : Hom A B . Π i : $\mathbb{2}$. p i → (dua (dcoe p)) i

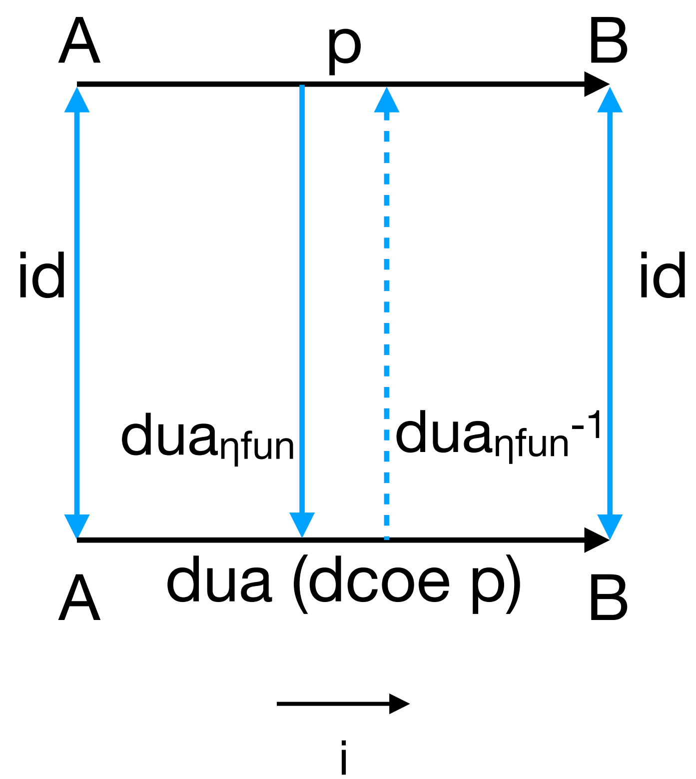
Naive Directed Univalence

Naive Directed Univalence

- We're thus left with the following picture:

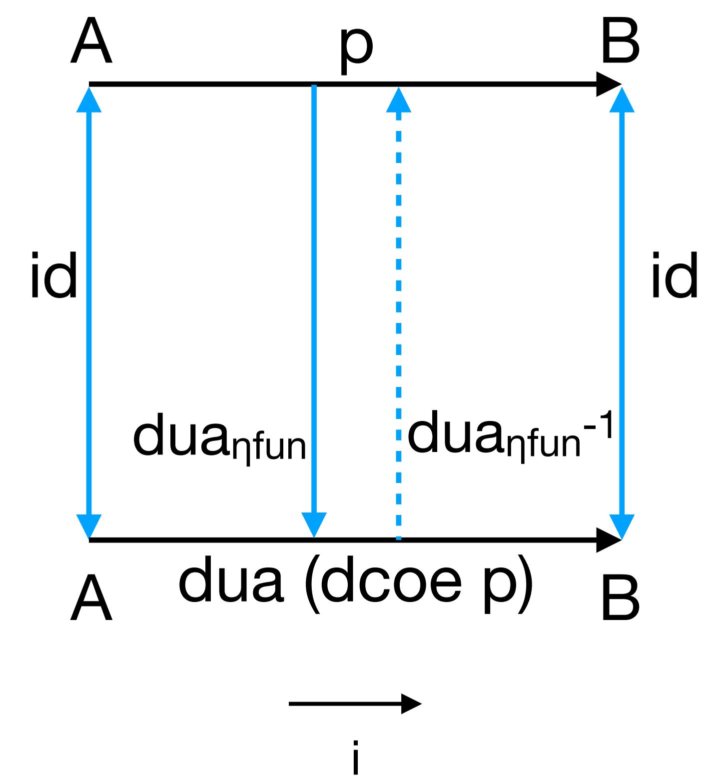
Naive Directed Univalence

- We're thus left with the following picture:



Naive Directed Univalence

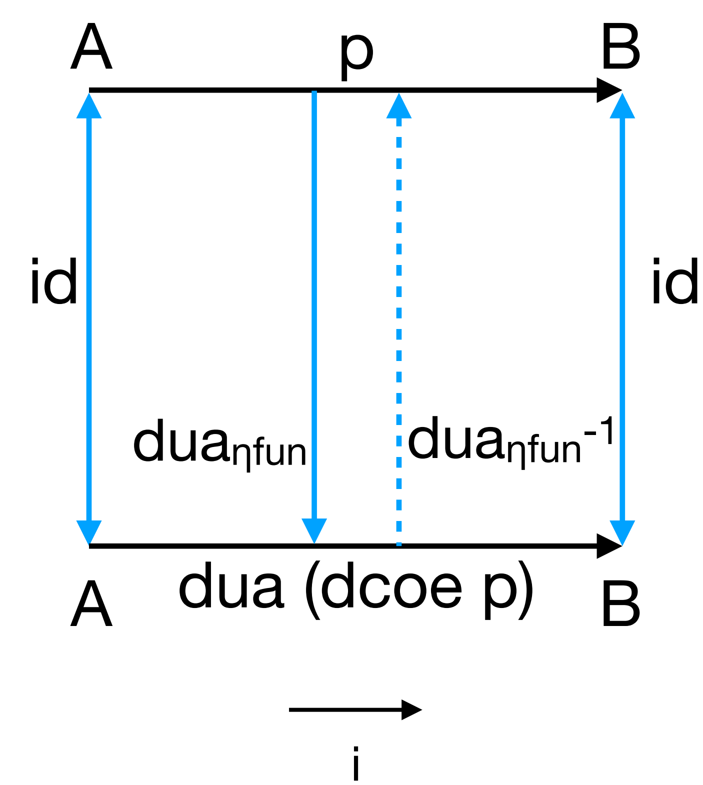
- We're thus left with the following picture:



- To complete directed univalence, we need dua$_{\eta fun}{}^{-1}$

Naive Directed Univalence

- We're thus left with the following picture:



- To complete directed univalence, we need $\mathrm{dua}_{\eta\mathrm{fun}}{}^{-1}$

- Agda: https://github.com/dlicata335/cart-cube

What next?

What next?

- Cavallo, Riehl and Sattler's proof of directed univalence contains the precise lemma we need to finish.

What next?

- Cavallo, Riehl and Sattler's proof of directed univalence contains the precise lemma we need to finish.
- New goal: use any techniques available to confirm directed univalence holds at all in a cubical setting.

What next?

- Cavallo, Riehl and Sattler's proof of directed univalence contains the precise lemma we need to finish.
- New goal: use any techniques available to confirm directed univalence holds at all in a cubical setting.
- Note: We would love any/all feedback on the math that follows.

What next?

What next?

- The proof in the bisimplicial model relies on simplices being a Reedy category

What next?

- The proof in the bisimplicial model relies on simplices being a Reedy category
  - specifically: weak equivalences in the model are level-wise weak equivalences of simplicial sets

What next?

- The proof in the bisimplicial model relies on simplices being a Reedy category
    - specifically: weak equivalences in the model are level-wise weak equivalences of simplicial sets
- Dedekind cubes are not Reedy...

Our New Goal

Our New Goal

- Find a setting that...

Our New Goal

- Find a setting that...

    1. is cubical set valued presheaves of a Reedy category

Our New Goal

- Find a setting that...

  1. is cubical set valued presheaves of a Reedy category

  2. interprets the axioms from our internal language

Our New Goal

- Find a setting that...

  1. is cubical set valued presheaves of a Reedy category

  2. interprets the axioms from our internal language

  3. allows for the LOPS construction of universes

Our New Goal

- Find a setting that...

  1. is cubical set valued presheaves of a Reedy category
  2. interprets the axioms from our internal language
  3. allows for the LOPS construction of universes
     - tiny interval

What are Reedy Categories?

What are Reedy Categories?

- The Idea: Categories permitting inductive constructions of presheaves and their morphisms (akin to cell complexes)

What are Reedy Categories?

- The Idea: Categories permitting inductive constructions of presheaves and their morphisms (akin to cell complexes)

- (informal/incomplete) Definition: A *generalized Reedy category* is a category C along with a degree function $\delta : \mathrm{ob}\, C \to \mathbb{N}$ such that every morphism (that isn't an iso) factors through an object of strictly smaller degree

The Dedekind Cubes

The Dedekind Cubes

- Free Cartesian category on an interval generated by:

The Dedekind Cubes

- Free Cartesian category on an interval generated by:
  - face maps (+)

The Dedekind Cubes

- Free Cartesian category on an interval generated by:
  - face maps (+)
  - diagonals (+)

The Dedekind Cubes

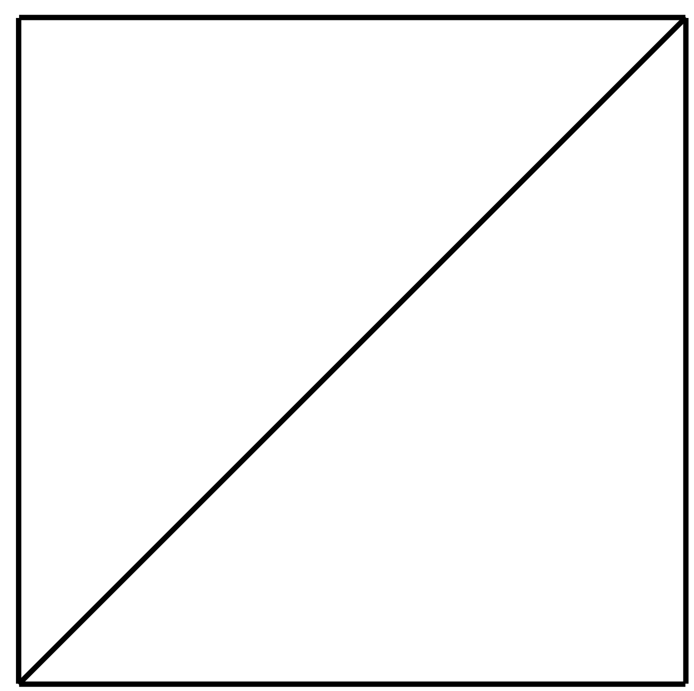
- Free Cartesian category on an interval generated by:
  - face maps (+)
  - diagonals (+)
  - degeneracies (-)

The Dedekind Cubes

- Free Cartesian category on an interval generated by:
  - face maps (+)
  - diagonals (+)
  - degeneracies (-)
  - connections (-)

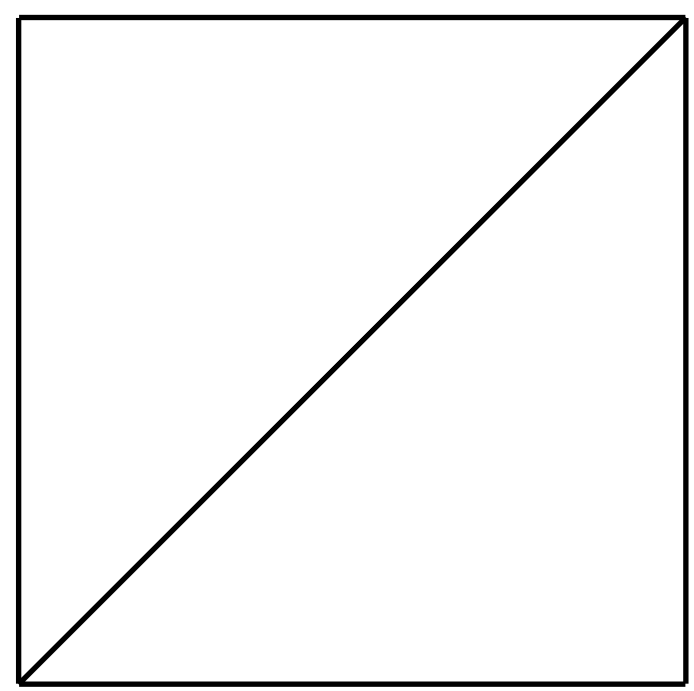
The Dedekind Cubes

$$(x, y) \mapsto (x, y, y) \mapsto (x \wedge y, y)$$



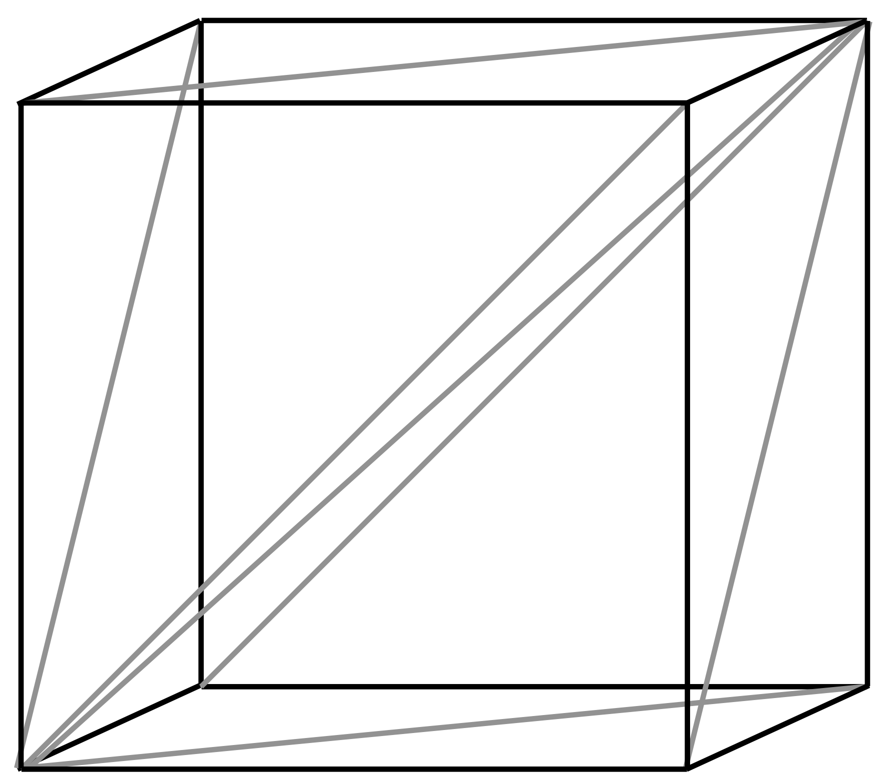
The Dedekind Cubes

$$\underline{(x, y)} \mapsto (x, y, y) \mapsto (x \wedge y, y)$$



The Dedekind Cubes

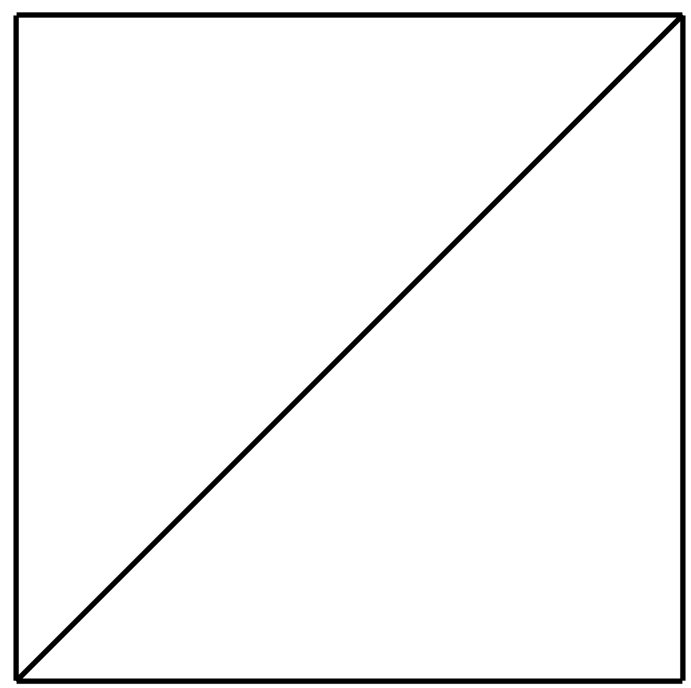
$(x, y) \mapsto \underline{(x, y, y)} \mapsto (x \wedge y, y)$



up by a diagonal

The Dedekind Cubes

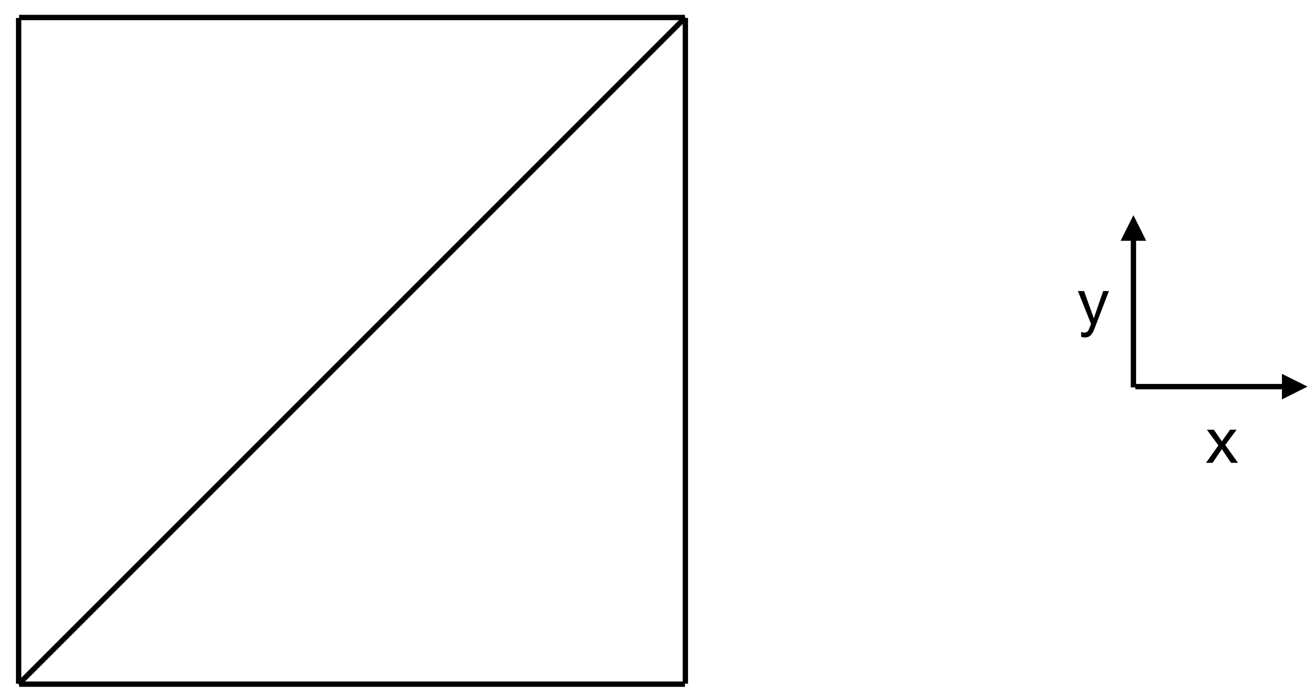
$$(x, y) \mapsto (x, y, y) \mapsto \underline{(x \wedge y, y)}$$



down by a connection

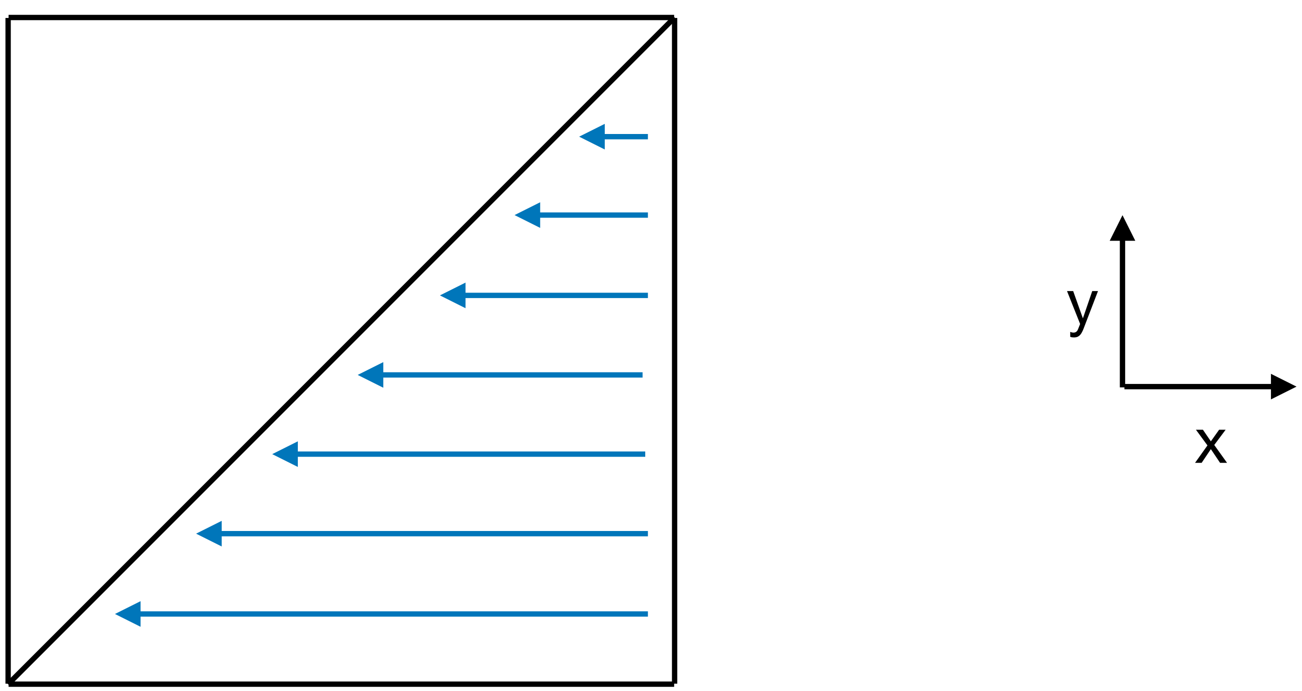
The Dedekind Cubes

$$(x, y) \mapsto (x \wedge y, y)$$



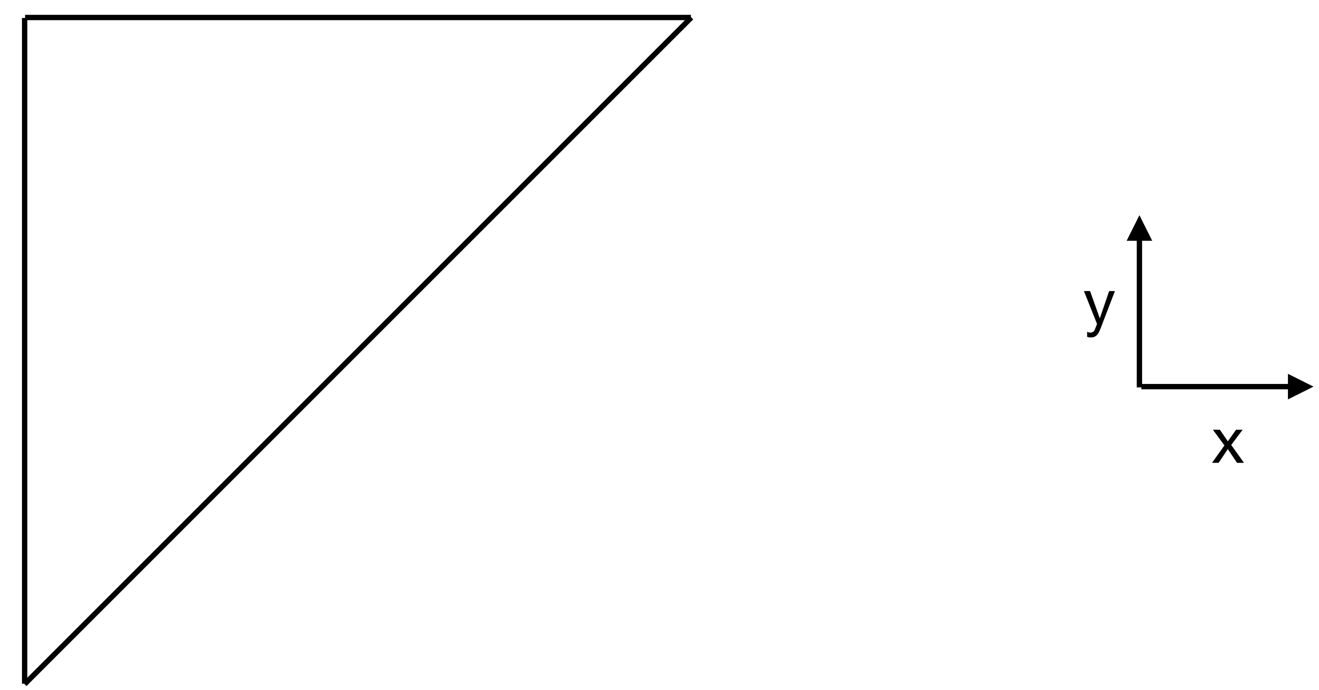
The Dedekind Cubes

$$(x, y) \mapsto (x \wedge y, y)$$



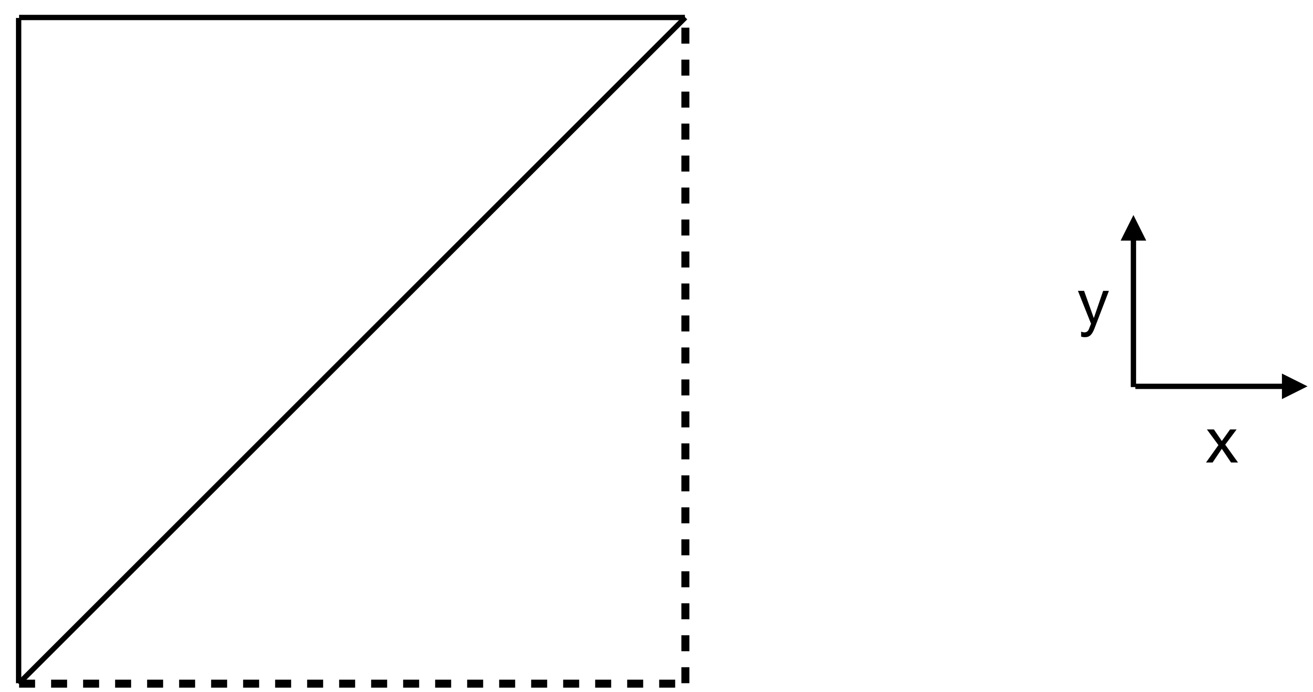
The Dedekind Cubes

$$(x, y) \mapsto (x \wedge y, y)$$



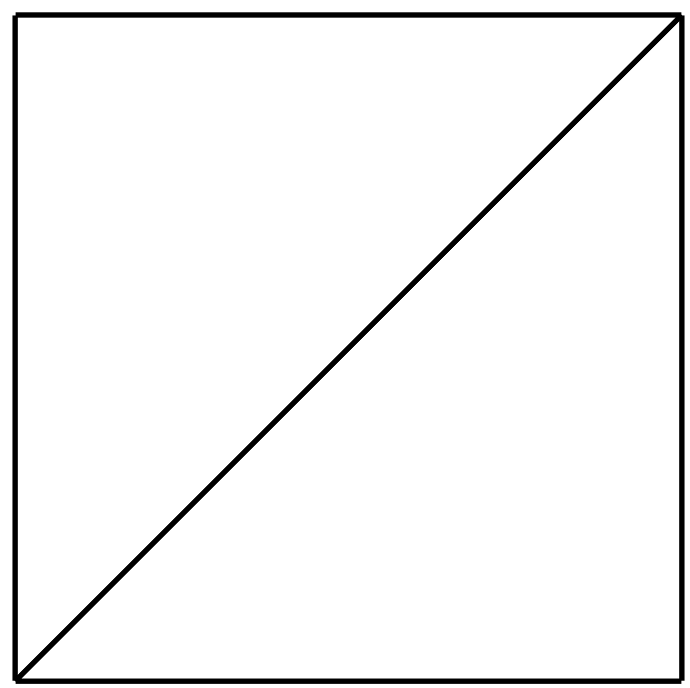
The Dedekind Cubes

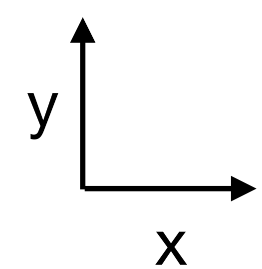
$$(x, y) \mapsto (x \wedge y, y)$$



The Dedekind Cubes

$$(x, y) \mapsto (x \wedge y, y)$$





The Image Closure

The Image Closure

- The Idea: formally add image objects for every morphism

The Image Closure

- The Idea: formally add image objects for every morphism

- The Construction: Given a small Category C, the *image closure* Im(C) is the full subcategory of [C^{op} , Set] containing, for each morphism f in C, the coimage of f.

The Image Closure

- The Idea: formally add image objects for every morphism

- The Construction: Given a small Category C, the *image closure* Im(C) is the full subcategory of $[C^{op}, Set]$ containing, for each morphism f in C, the coimage of f.

- Useful Lemma: We can build a topology J_{im} (the *image covering*) on Im(C) such that $[C^{op}, Set] \cong Sh(Im(C), J_{im})$.

The Image Closure

- The Idea: formally add image objects for every morphism

- The Construction: Given a small Category C, the *image closure* Im(C) is the full subcategory of $[C^{op}, Set]$ containing, for each morphism f in C, the coimage of f.

- Useful Lemma: We can build a topology J_{im} (the *image covering*) on Im(C) such that $[C^{op}, Set] \cong Sh(Im(C), J_{im})$.
  - Inspired by Kapulkin and Voevodsky

The Image Closure

- The Idea: formally add image objects for every morphism
- The Construction: Given a small Category C, the *image closure* Im(C) is the full subcategory of $[C^{op}, \mathrm{Set}]$ containing, for each morphism f in C, the coimage of f.
- Useful Lemma: We can build a topology J_{im} (the *image covering*) on Im(C) such that $[C^{op}, \mathrm{Set}] \cong \mathrm{Sh}(\mathrm{Im}(C), J_{im})$.
  - Inspired by Kapulkin and Voevodsky
  - The Comparison Lemma: [SGA 4, The Elephant]

The Prism Category

The Prism Category

- Definition: The *prism category* is the image closure of the Dedekind cube category.

The Prism Category

- Definition: The *prism category* is the image closure of the Dedekind cube category.

- Lemma (with Christian Sattler): The prism category is equivalent to the full subcategory of simplicial sets containing subobjects of the Dedekind cubes (Γ , ϕ) generated by the following formulae:
  - $\top$: true
  - $x \leq y$: the equalizer of the degeneracy map x and connection $x \wedge y$
  - $\phi \wedge \psi$: the pullback of the subobjects (Γ, ϕ) and (Γ, ψ)
  - $\phi \vee \psi$: the pushout of the pullback for $(\Gamma, \phi \wedge \psi)$

The Prism Category

The Prism Category

- The Prism category

The Prism Category

- The Prism category
    - is a finite product category...

The Prism Category

- The Prism category
  - is a finite product category...
  - ...and thus the Yoneda embedding of its interval is tiny...

Prisms are Reedy

Prisms are Reedy

- Theorem: The prism category is a generalized Reedy category.

Prisms are Reedy

- Theorem: The prism category is a generalized Reedy category.
  - The down maps are those that are regular epis in the presheaf category

Prisms are Reedy

- Theorem: The prism category is a generalized Reedy category.
  - The down maps are those that are regular epis in the presheaf category
  - The up maps are the monos

Prisms are Reedy

- Theorem: The prism category is a generalized Reedy category.
  - The down maps are those that are regular epis in the presheaf category
  - The up maps are the monos
  - The Reedy factorization is the image factorization

Prisms are Reedy

- Theorem: The prism category is a generalized Reedy category.
  - The down maps are those that are regular epis in the presheaf category
  - The up maps are the monos
  - The Reedy factorization is the image factorization

- Corollary: The opposite of the prism category is also generalized Reedy

Prisms are Reedy

- Theorem: The prism category is a generalized Reedy category.
  - The down maps are those that are regular epis in the presheaf category
  - The up maps are the monos
  - The Reedy factorization is the image factorization

- Corollary: The opposite of the prism category is also generalized Reedy

- Question: For which categories C is Im(C) Reedy?

Model Category One: Prismatic Cubical Sets

Model Category One: Prismatic Cubical Sets

- Reedy model structure on [Prism^{op}, [Cube^{op}, Set]], starting with model structure on Cartesian cubes [Sattler, Awodey]

Model Category One: Prismatic Cubical Sets

- Reedy model structure on [Prism^{op}, [Cube^{op}, Set]], starting with model structure on Cartesian cubes [Sattler, Awodey]

- The lemma missing from the bicubical internal language now is provable in the same way as in bisimplicial sets.

Model Category One: Prismatic Cubical Sets

- Reedy model structure on [Prism^{op}, [Cube^{op}, Set]], starting with model structure on Cartesian cubes [Sattler, Awodey]

- The lemma missing from the bicubical internal language now is provable in the same way as in bisimplicial sets.

- As our internal language axioms interpret into this model, we get a model with directed univalence!

Model Category One: Prismatic Cubical Sets

- Reedy model structure on [Prism^{op}, [Cube^{op}, Set]], starting with model structure on Cartesian cubes [Sattler, Awodey]
- The lemma missing from the bicubical internal language now is provable in the same way as in bisimplicial sets.
- As our internal language axioms interpret into this model, we get a model with directed univalence!
- Can we make this even more cubical?

Model Category Two:
Bicubical Sets

Model Category Two: Bicubical Sets

- Sheafification gives us an adjunction between prismatic cubical sets and bicubical sets

Model Category Two: Bicubical Sets

- Sheafification gives us an adjunction between prismatic cubical sets and bicubical sets
- We can transfer the model structure along the adjunction to bicubical sets

Model Category Two: Bicubical Sets

- Sheafification gives us an adjunction between prismatic cubical sets and bicubical sets
- We can transfer the model structure along the adjunction to bicubical sets
  - Left Induced Model Structure: [Hess-Kedziorek-Riehl-Shipley, Garner-Kedziorek-Riehl]

Model Category Two: Bicubical Sets

- Sheafification gives us an adjunction between prismatic cubical sets and bicubical sets
- We can transfer the model structure along the adjunction to bicubical sets
  - Left Induced Model Structure: [Hess-Kedziorek-Riehl-Shipley, Garner-Kedziorek-Riehl]
  - Path Object Argument: [Quillen]

Model Category Two: Bicubical Sets

- Sheafification gives us an adjunction between prismatic cubical sets and bicubical sets
- We can transfer the model structure along the adjunction to bicubical sets
  - Left Induced Model Structure: [Hess-Kedziorek-Riehl-Shipley, Garner-Kedziorek-Riehl]
  - Path Object Argument: [Quillen]
- Our internal language axioms still interpret after the transfer

Model Category Two: Bicubical Sets

- Sheafification gives us an adjunction between prismatic cubical sets and bicubical sets

- We can transfer the model structure along the adjunction to bicubical sets
  - Left Induced Model Structure: [Hess-Kedziorek-Riehl-Shipley, Garner-Kedziorek-Riehl]
  - Path Object Argument: [Quillen]

- Our internal language axioms still interpret after the transfer

- The lemma that finished directed univalence is still true after the transfer