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 $\infty$-categories with a model bisimplicial sets
Directed Type Theory

- Riehl-Shulman defines a type theory for $\infty$-categories with a model bisimplicial sets
  1. Begin with HoTT
Directed Type Theory

- Riehl-Shulman defines a type theory for $\infty$-categories with a model bisimplicial sets
  1. Begin with HoTT
  2. Add Hom-types
Directed Type Theory

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

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

• Riehl-Shulman defines a type theory for $\infty$-categories with a model bisimplicial sets
  1. Begin with HoTT
  2. Add Hom-types
  3. $\infty$-categories (Segal types) and univalent $\infty$-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 $\infty$-category of spaces and continuous functions) and shown

  \[ \text{Directed Univalence: } \text{Hom}_{UCov} A B \simeq A \to B \]
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)*

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

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>(in the style of Orton-Pitts)</em></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>in the style of Orton-Pitts</em></td>
<td></td>
</tr>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: $\mathbb{I}$</td>
<td>2. Add an interval: $\mathbb{2}$</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for $\mathbb{I}$</td>
<td>3. Specify gen. cofibrations for $\mathbb{2}$</td>
</tr>
<tr>
<td>4. Define filling problem for Kan fibrations</td>
<td></td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td></td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td></td>
</tr>
</tbody>
</table>
## 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
# Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Begin with MLTT</td>
<td><strong>1.</strong> Begin with Cubical Type Theory</td>
</tr>
<tr>
<td><strong>2.</strong> Add an interval: (\mathbb{I})</td>
<td><strong>2.</strong> Add an interval: (\mathbb{I})</td>
</tr>
<tr>
<td><strong>3.</strong> Specify gen. cofibrations for (\mathbb{I})</td>
<td><strong>3.</strong> Specify gen. cofibrations for (\mathbb{I})</td>
</tr>
<tr>
<td><strong>4.</strong> Define filling problem for Kan fibrations</td>
<td><strong>4.</strong> Define filling problem for covariant fibrations</td>
</tr>
<tr>
<td><strong>5.</strong> Define universe of Kan fibrations</td>
<td><strong>5.</strong> Define universe of covariant fibrations</td>
</tr>
<tr>
<td><strong>6.</strong> Construct univalence</td>
<td></td>
</tr>
<tr>
<td>Cubical Type Theory</td>
<td>Directed Type Theory</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: (\mathbb{I})</td>
<td>2. Add an interval: (\mathcal{2})</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for (\mathbb{I})</td>
<td>3. Specify gen. cofibrations for (\mathcal{2})</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>
### Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Begin with MLTT</td>
<td><strong>1.</strong> Begin with Cubical Type Theory</td>
</tr>
<tr>
<td><strong>2.</strong> Add an interval: $\mathbb{I}$</td>
<td><strong>2.</strong> Add an interval: $\mathbb{2}$</td>
</tr>
<tr>
<td><strong>3.</strong> Specify gen. cofibrations for $\mathbb{I}$</td>
<td><strong>3.</strong> Specify gen. cofibrations for $\mathbb{2}$</td>
</tr>
<tr>
<td><strong>4.</strong> Define filling problem for Kan fibrations</td>
<td><strong>4.</strong> Define filling problem for covariant fibrations</td>
</tr>
<tr>
<td><strong>5.</strong> Define universe of Kan fibrations</td>
<td><strong>5.</strong> Define universe of covariant fibrations</td>
</tr>
<tr>
<td><strong>6.</strong> Construct univalence</td>
<td><strong>6.</strong> Construct directed univalence</td>
</tr>
</tbody>
</table>
Defining Bicubical Directed Type Theory

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

2. Add an interval: \( \mathbb{I} \)

Directed Type Theory

2. Add an interval: \( \mathbb{I} \)
Defining Bicubical Directed Type Theory

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

2. Add an interval: \( \mathbb{I} \)

\[ \mathbb{I} : \text{Type} \]

Directed Type Theory

2. Add an interval: \( \mathbb{D} \)
Defining Bicubical Directed Type Theory

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

2. Add an interval: \( \mathbb{I} \)

\[
\begin{array}{c}
\mathbb{I} : \text{Type} \\
0_{\mathbb{I}} : \mathbb{I} \\
1_{\mathbb{I}} : \mathbb{I}
\end{array}
\]

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: \( \mathbb{I} \)

\[
\begin{align*}
\mathbb{I} : \text{Type} \\
0 : \mathbb{I} \\
1 : \mathbb{I}
\end{align*}
\]

i.e. generators for the Cartesian cubes

**Directed Type Theory**

2. Add an interval: \( \mathcal{D} \)
Defining Bicubical Directed Type Theory

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

2. Add an interval: \( \mathbb{I} \)

\[
\begin{align*}
\mathbb{I} &: \text{Type} \\
0_\mathbb{I} &: \mathbb{I} \\
1_\mathbb{I} &: \mathbb{I}
\end{align*}
\]

i.e. generators for the Cartesian cubes

Directed Type Theory

2. Add an interval: \( \mathbb{2} \)

\[
\begin{align*}
\mathbb{2} &: \text{Type}
\end{align*}
\]
Defining Bicubical Directed Type Theory

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

2. Add an interval: \( \mathbb{I} \)

\[
\begin{align*}
\mathbb{I} &: \text{Type} \\
0_\mathbb{I} &: \mathbb{I} \\
1_\mathbb{I} &: \mathbb{I}
\end{align*}
\]

i.e. generators for the Cartesian cubes

Directed Type Theory

2. Add an interval: \( \mathbb{2} \)

\[
\begin{align*}
\mathbb{2} &: \text{Type} \\
0_\mathbb{2} &: \mathbb{2} \\
1_\mathbb{2} &: \mathbb{2}
\end{align*}
\]

\[
\begin{align*}
x &: \mathbb{2} \\
y &: \mathbb{2} \\
x \wedge y &: \mathbb{2} \\
x \vee y &: \mathbb{2}
\end{align*}
\]
Defining Bicubical Directed Type Theory

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

2. Add an interval: \( \mathbb{I} \)

\[
\begin{align*}
\mathbb{I} &: \text{Type} \\
0_\mathbb{I} &: \mathbb{I} \\
1_\mathbb{I} &: \mathbb{I}
\end{align*}
\]

i.e. generators for the Cartesian cubes

**Directed Type Theory**

2. Add an interval: \( \mathbb{2} \)

\[
\begin{align*}
\mathbb{2} &: \text{Type} \\
0_\mathbb{2} &: \mathbb{2} \\
1_\mathbb{2} &: \mathbb{2}
\end{align*}
\]

\[
\begin{align*}
x &: \mathbb{2} \quad y &: \mathbb{2} \\
x \land y &: \mathbb{2} \\
x \lor y &: \mathbb{2}
\end{align*}
\]

and equations...
Defining Bicubical Directed Type Theory

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

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

\[ \begin{align*}
\mathbb{I} &: \text{Type} \\
0_\mathbb{I} &: \mathbb{I} \\
1_\mathbb{I} &: \mathbb{I}
\end{align*} \]

i.e. generators for the Cartesian cubes

Directed Type Theory

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

\[ \begin{align*}
\mathbb{2} &: \text{Type} \\
0_\mathbb{2} &: \mathbb{2} \\
1_\mathbb{2} &: \mathbb{2}
\end{align*} \]

\[ \begin{align*}
x &: \mathbb{2} \\
y &: \mathbb{2}
\end{align*} \]

\[ \begin{align*}
x \wedge y &: \mathbb{2} \\
x \vee y &: \mathbb{2}
\end{align*} \]

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?
  \[ x \leq y := x = x \land y \]

- We also add the following axioms:
The Directed Interval

• Why Dedekind cubes instead of Cartesian?
  \( x \leq y := x = x \land y \)

• We also add the following axioms:
  • \( p : \mathbb{I} \to \mathbb{2} \) is constant (\( \prod x y : \mathbb{I}, p x = p y \))
The Directed Interval

• Why Dedekind cubes instead of Cartesian?
  \( x \leq y := x = x \land y \)

• We also add the following axioms:
  
  • \( p : \mathbb{I} \rightarrow \mathbb{2} \) is constant (\( \prod x y : \mathbb{I}, p x = p y \))
  
  • \( p : \mathbb{2} \rightarrow \mathbb{2} \) is monotone (\( \prod x y : \mathbb{2}, \text{if } x \leq y \text{ then } p x \leq p y \))
## Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: ( \mathbb{I} )</td>
<td>2. Add an interval: ( \mathbb{2} )</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for ( \mathbb{I} )</td>
<td>3. Specify gen. cofibrations for ( \mathbb{2} )</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>
# Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: $\mathbb{I}$</td>
<td>2. Add an interval: $\mathbb{I}$</td>
</tr>
<tr>
<td><strong>3. Specify gen. cofibrations for $\mathbb{I}$</strong></td>
<td><strong>3. Specify gen. cofibrations for $\mathbb{2}$</strong></td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>

(in the style of Orton-Pitts)
<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(in the style of Orton-Pitts)</td>
<td></td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for (\mathbb{I})</td>
<td>3. Specify gen. cofibrations for (\mathbb{2})</td>
</tr>
</tbody>
</table>
3. Specify gen. cofibrations for \( \mathbb{I} \)

\[
isCof : \Omega \to \Omega
\]

\[
Cof := \Sigma \phi : \Omega . \text{isCof } \phi
\]
Defining Bicubical Directed Type Theory

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

3. Specify gen. cofibrations for \( \mathbb{I} \)

\[
isCof : \Omega \to \Omega
\]

\[
\text{Cof} := \Sigma \phi : \Omega . \text{isCof} \phi
\]

Cof closed under \( \wedge \), \( \vee \), \( \bot \), \( \top \)

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 \( \mathbb{I} \)

\[
\text{isCof} : \Omega \to \Omega
\]

\[\text{Cof} := \Sigma \phi : \Omega . \text{isCof} \phi\]

Cof closed under \( \wedge, \vee, \bot, \top \)

\[
\frac{x : \mathbb{I} \quad y : \mathbb{I}}{_ : \text{isCof} (x = y)}
\]

\[
\frac{\phi : \mathbb{I} \to \text{Cof}}{_ : \text{isCof} (\Pi x : \mathbb{I} . \phi x)}
\]
Defining Bicubical Directed Type Theory

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

3. Specify gen. cofibrations for \( \mathbb{I} \)

\[
\text{isCof : } \Omega \rightarrow \Omega
\]

\[
\text{Cof := } \Sigma \phi : \Omega . \text{isCof } \phi
\]

Cof closed under \( \land, \lor, \bot, \top \)

\[
\frac{x : \mathbb{I} \quad y : \mathbb{I}}{_ : \text{isCof } (x = y)}
\]

\[
\phi : \mathbb{I} \rightarrow \text{Cof}
\]

\[
_ : \text{isCof } (\Pi x : \mathbb{I} . \phi x)
\]

Directed Type Theory

3. Specify gen. cofibrations for \( \mathbb{2} \)

\[
\frac{x : \mathbb{2} \quad y : \mathbb{2}}{_ : \text{isCof } (x = y)}
\]

\[
\phi : \mathbb{2} \rightarrow \text{Cof}
\]

\[
_ : \text{isCof } (\Pi x : \mathbb{2} . \phi x)
\]
## Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(in the style of Orton-Pitts)</td>
<td></td>
</tr>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: (\mathbb{I})</td>
<td>2. Add an interval: (\mathcal{2})</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for (\mathbb{I})</td>
<td>3. Specify gen. cofibrations for (\mathcal{2})</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>
## Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: $\mathbb{I}$</td>
<td>2. Add an interval: $\mathbb{2}$</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for $\mathbb{I}$</td>
<td>3. Specify gen. cofibrations for $\mathbb{2}$</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>
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 : (𝕀 → U) → U
hasCom A = Π i j : ℍ .
  Π α : Cof .
  Π t : (Π x : ℍ . 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 : ℍ → 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

\[
\text{hasCom} : (\mathbb{I} \to U) \to U \\
\text{hasCom} A = \prod i j : \mathbb{I} . \\
\quad \prod \alpha : \text{Cof} . \\
\quad \prod t : (\prod x : \mathbb{I} . \alpha \to A x) \\
\quad \prod b : (A i)[\alpha \mapsto t i] . \\
\quad (A j)[\alpha \mapsto t j; i = j \mapsto b] \\
\]

\[
\text{relCom} : (A : U) \to (A \to U) \to U \\
\text{relCom} A B = \prod p : \mathbb{I} \to A . \\
\quad \text{hasCom} (B \circ p) \\
\]

## Directed Type Theory

4. Define filling problem for covariant fibrations

\[
\text{hasCov} : (2 \to U) \to U \\
\text{hasCov} A = \prod \alpha : \text{Cof} . \\
\quad \prod t : (\prod x : 2 . \alpha \to A x) \\
\quad \prod b : (A \emptyset)[\alpha \mapsto t \emptyset] . \\
\quad (A 1)[\alpha \mapsto t 1] \\
\]

\[
\text{relCov} : (A : U) \to (A \to U) \to U \\
\text{relCov} A B = \prod p : 2 \to A . \\
\quad \text{hasCov} (B \circ p) \\
\]
## Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th><strong>Cubical Type Theory</strong></th>
<th><strong>Directed Type Theory</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><em>(in the style of Orton-Pitts)</em></td>
<td></td>
</tr>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: ( \mathbb{I} )</td>
<td>2. Add an interval: ( \mathbb{2} )</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for ( \mathbb{I} )</td>
<td>3. Specify gen. cofibrations for ( \mathbb{2} )</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>
## Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: $\mathbb{I}$</td>
<td>2. Add an interval: $\mathbb{2}$</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for $\mathbb{I}$</td>
<td>3. Specify gen. cofibrations for $\mathbb{2}$</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>
Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(in the style of Orton-Pitts)</td>
<td></td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
</tbody>
</table>
Defining Bicubical Directed Type Theory

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

5. Define universe of Kan fibrations

• \( U_{\text{Kan}} \) given by LOPS construction for relCom

Directed Type Theory

5. Define universe of covariant fibrations
5. Define universe of Kan fibrations

• $U_{\text{Kan}}$ given by LOPS construction for relCom

5. Define universe of covariant fibrations

• $U_{\text{Cov}}$ given by LOPS construction for relCov.

Lemma: relCov is in $U_{\text{Kan}}$
<table>
<thead>
<tr>
<th>Cubical Type Theory (in the style of Orton-Pitts)</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: II</td>
<td>2. Add an interval: 𝟙</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
<td>6. Construct directed univalence</td>
</tr>
</tbody>
</table>
# Defining Bicubical Directed Type Theory

<table>
<thead>
<tr>
<th>Cubical Type Theory</th>
<th>Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>in the style of Orton-Pitts</em></td>
<td></td>
</tr>
<tr>
<td>1. Begin with MLTT</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: ( \mathbb{I} )</td>
<td>2. Add an interval: ( \mathbb{2} )</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for ( \mathbb{I} )</td>
<td>3. Specify gen. cofibrations for ( \mathbb{2} )</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td><strong>6. Construct univalence</strong></td>
<td><strong>6. Construct directed univalence</strong></td>
</tr>
</tbody>
</table>
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)*

6. Construct univalence

Directed Type Theory

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 \textit{functions} to morphism structure
Glue Types
Glue \[ \alpha \mapsto (T, f) \] B :=

\[ \alpha \vdash T \quad \alpha \vdash f \]

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

\[
\text{Glue}[\alpha \mapsto (T, f)] B := \alpha \vdash T \quad \alpha \vdash \text{Glue}[\alpha \mapsto (T, f)] B \equiv T
\]
Glue Types

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

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

\[ \alpha \vdash \text{glue} t b := \]
\[ \alpha \vdash t : T \]
\[ \alpha \vdash f \]
\[ \alpha \vdash b : B \]
Glue Types

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

\[ \alpha \vdash T \]

\[ \alpha \vdash f \]

\[ \text{glue } t \ b := \]

\[ \alpha \vdash t : T \]

\[ \alpha \vdash f \]

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

\[ \text{unglue } g : B \]

\[ b : B \]
Glue Types

Glue \[ α \mapsto (T, f) \] B :=

\[
\begin{align*}
α & \vdash T \\
α & \vdash f \\
\text{glue } t b & := \\
α & \vdash t : T \\
α & \vdash f \\
\text{g : Glue } α \mapsto (T, f) \ B & \equiv T \\
\text{unglue } g : B & \\
α & \vdash \text{glue } t b \equiv t \\
b & : B
\end{align*}
\]
Glue Types

\[
\text{Glue } \alpha \mapsto (T, f) \text{ B := } \\
\begin{array}{c}
\alpha \vdash T \\
\alpha \vdash f \\
\alpha \vdash t : T \\
\alpha \vdash f \\
\end{array} \\
\begin{array}{c}
\text{g : Glue } \alpha \mapsto (T, f) \text{ B} \\
\text{unglue g : B} \\
\end{array} \\
\begin{array}{c}
\alpha \vdash \text{unglue (glue t b)} \equiv f t \\
\end{array}
\]
Glue Types

\[
\text{Glue } [\alpha \mapsto (T, f)] B := \\
\text{glue } t b :=
\]

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

\[
\begin{align*}
\alpha \vdash T \\
\alpha \vdash f \\
\alpha \vdash t : T \\
\alpha \vdash f \\
\alpha \vdash b : B \\
\alpha \vdash \text{Glue } [\alpha \mapsto (T, f)] B \equiv T \\
\end{align*}
\]

\[
\begin{align*}
\alpha \vdash \text{glue } t b \equiv t \\
\alpha \vdash \text{unglue } (\text{glue } t b) \equiv f t \\
\end{align*}
\]
Defining Directed Univalence

dua i A B f := Glue [ \( i = 0 \mapsto (A, f : A \to B) \), \( i = 1 \mapsto (B, \text{id}) \) ] B : Hom_u A B
Naive Directed Univalence
Naive Directed Univalence

- dua is Kan + covariant, and thus lands in $U_{\text{Cov}}$
Naive Directed Univalence

- dua is Kan + covariant, and thus lands in $U_{\text{Cov}}$
- $U_{\text{Cov}}$ itself is Kan
Naive Directed Univalence

• dua is Kan + covariant, and thus lands in $U_{\text{Cov}}$

• $U_{\text{Cov}}$ itself is Kan

• Path univalence holds in $U_{\text{Cov}}$
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 $U_{Cov}$:
  - $dcoe : (\text{Hom } A B) \rightarrow (A \rightarrow B)$
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 $U_{Cov}$:
  • $\text{dcoe} : (\text{Hom } A B) \rightarrow (A \rightarrow B)$
  • $\text{dua} : (A \rightarrow B) \rightarrow \text{Hom } A B$
Naive Directed Univalence

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

- These allow us to define the following for \( U_{\text{Cov}} \):
  - \( dcoe : (\text{Hom } A \to B) \to (A \to B) \)
  - \( \text{dua} : (A \to B) \to \text{Hom } A \to B \)
  - \( \text{dua}_\beta : \prod f : A \to B . \text{Path } f (dcoe (\text{dua } f)) \)
Naive Directed Univalence

- \( \text{dua} \) is Kan + covariant, and thus lands in \( U_{\text{Cov}} \)

- \( U_{\text{Cov}} \) itself is Kan

- Path univalence holds in \( U_{\text{Cov}} \)

- These allow us to define the following for \( U_{\text{Cov}} \):
  - \( \text{dcoe} : (\text{Hom } A \to B) \to (A \to B) \)
  - \( \text{dua} : (A \to B) \to \text{Hom } A \to B \)
  - \( \text{dua}_\beta : \prod f : A \to B \cdot \text{Path } f (\text{dcoe } (\text{dua } f)) \)
  - \( \text{dua}_\eta_{\text{fun}} : \prod p : \text{Hom } A \to B \cdot \prod i : 2 \cdot p \, i \to (\text{dua } (\text{dcoe } p)) \, i \)
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 $\text{dua}_{\eta_{\text{fun}}}^{-1}$
Naive Directed Univalence

• We're thus left with the following picture:

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

• 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 $\delta : \text{ob } C \rightarrow \mathbb{N}$ 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

\[(x, y) \mapsto (x, y, y) \mapsto (x \land y, y)\]
The Dedekind Cubes

\[(x, y) \mapsto (x, y, y) \mapsto (x \land y, y)\]
The Dedekind Cubes

\[(x, y) \mapsto (x, y, y) \mapsto (x \land y, y)\]

up by a diagonal
The Dedekind Cubes

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

down by a connection
The Dedekind Cubes

\[(x, y) \mapsto (x \land y, y)\]
The Dedekind Cubes

\[(x, y) \mapsto (x \land y, y)\]
The Dedekind Cubes

\[(x, y) \mapsto (x \land y, y)\]
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 $\text{Im}(C)$ is the full subcategory of $[C^{\text{op}}, \text{Set}]$ 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* $\text{Im}(C)$ is the full subcategory of $[C^{\text{op}}, \text{Set}]$ containing, for each morphism $f$ in $C$, the coimage of $f$.

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

• The Idea: formally add image objects for every morphism

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

• Useful Lemma: We can build a topology $J_{\text{im}}$ (the image covering) on $\text{Im}(C)$ such that $[C^{\text{op}}, \text{Set}] \cong \text{Sh}(\text{Im}(C), J_{\text{im}})$.
  • 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 \( \text{Im}(C) \) is the full subcategory of \([C^{\text{op}}, \text{Set}]\) containing, for each morphism \( f \) in \( C \), the coimage of \( f \).

• Useful Lemma: We can build a topology \( J_{\text{im}} \) (the image covering) on \( \text{Im}(C) \) such that \([C^{\text{op}}, \text{Set}] \cong \text{Sh}(\text{Im}(C), J_{\text{im}})\).
  • 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 \((\Gamma, \phi)\) generated by the following formulae:
  • \(\top\) : true
  • \(x \leq y\) : the equalizer of the degeneracy map \(x\) and connection \(x \wedge y\)
  • \(\phi \land \psi\) : the pullback of the subobjects \((\Gamma, \phi)\) and \((\Gamma, \psi)\)
  • \(\phi \lor \psi\) : the pushout of the pullback for \((\Gamma, \phi \land \psi)\)
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 \text{Im}(C) Reedy?
Model Category One: Prismatic Cubical Sets
Model Category One: Prismatic Cubical Sets

- Reedy model structure on $[\text{Prism}^{\text{op}}, [\text{Cube}^{\text{op}}, \text{Set}]]$, starting with model structure on Cartesian cubes [Sattler, Awodey]
Model Category One: Prismatic Cubical Sets

- Reedy model structure on $[\text{Prism}^{\text{op}}, [\text{Cube}^{\text{op}}, \text{Set}]]$, 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 \([\text{Prism}^{\text{op}}, [\text{Cube}^{\text{op}}, \text{Set}]]\), 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 \([\text{Prism}^{\text{op}}, [\text{Cube}^{\text{op}}, \text{Set}]]\), 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