A Constructive Model of Directed Univalence in Bicubical Sets

Matthew Weaver
Princeton University

HoTTTEST. April 16th, 2020

joint work with Dan Licata
Directed Type Theory

- Riehl-Shulman defines a type theory for $\infty$-categories with a model in 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

\[
\text{Directed Univalence: } \text{Hom}_{U_{\text{Cov}}} A B \simeq A \to B
\]
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
     • ...fortunately, we can still make it work!
Let's see how far the techniques from cubical type theory get us!
# Defining Bicubical Directed Type Theory

<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 a topos</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: I</td>
<td>2. Add an interval: 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>1. Begin with a topos</td>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: $\mathbb{I}$</td>
<td>2. Add an interval: $\aleph$</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for $\mathbb{I}$</td>
<td>3. Specify gen. cofibrations for $\aleph$</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

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

2. Add an interval: \( I \)

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

Directed Type Theory

2. Add an interval: \( 2 \)

\[
\begin{align*}
2 & : \text{Type} \\
0_2 & : 2 \\
1_2 & : 2
\end{align*}
\]

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

and equations...

i.e. generators for the Dedekind cubes

e.g. generators for the Cartesian cubes, although any cubical type theory works
The Directed Interval

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

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

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

1. Begin with a topos

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)

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

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

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

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

\[
\begin{array}{ll}
\text{x : } \mathbb{I} & \text{y : } \mathbb{I} \\
\_ : \text{isCof} (x = y)
\end{array}
\]

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

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

Directed Type Theory

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

\[
\begin{array}{ll}
x : \mathbb{2} & y : \mathbb{2}
\end{array}
\]

\[
\_ : \text{isCof} (x = y)
\]

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

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

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

1. Begin with a topos

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

\[
\begin{align*}
\text{hasCom} & : (\mathbb{I} \to U) \to U \\
\text{hasCom} A & = \Pi i j : \mathbb{I} . \\
& \quad \Pi \alpha : \text{Cof} . \\
& \quad \Pi t : (\Pi x : \mathbb{I} . \alpha \to A x) \\
& \quad \Pi b : (A i)[\alpha \mapsto t i] . \\
& \quad (A j)[\alpha \mapsto t j; i = j \mapsto b] \\
\end{align*}
\]

\[
\begin{align*}
\text{relCom} & : (A : U) \to (A \to U) \to U \\
\text{relCom} A B & = \Pi p : \mathbb{I} \to A . \\
& \quad \text{hasCom} (B \circ p)
\end{align*}
\]

Directed Type Theory

4. Define filling problem for covariant fibrations

\[
\begin{align*}
\text{hasCov} & : (\mathbb{2} \to U) \to U \\
\text{hasCov} A & = \Pi \alpha : \text{Cof} . \\
& \quad \Pi t : (\Pi x : \mathbb{2} . \alpha \to A x) \\
& \quad \Pi b : (A \mathbb{0}_2)[\alpha \mapsto t \mathbb{0}_2] . \\
& \quad (A \mathbb{1}_2)[\alpha \mapsto t \mathbb{1}_2] \\
\end{align*}
\]

\[
\begin{align*}
\text{relCov} & : (A : U) \to (A \to U) \to U \\
\text{relCov} A B & = \Pi p : \mathbb{2} \to A . \\
& \quad \text{hasCov} (B \circ p)
\end{align*}
\]
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
<table>
<thead>
<tr>
<th>Defining Bicubical Directed Type Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cubical Type Theory</strong></td>
</tr>
<tr>
<td><em>(in the style of Orton-Pitts)</em></td>
</tr>
<tr>
<td>1. Begin with a topos</td>
</tr>
<tr>
<td>2. Add an interval: (𝕀)</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for (𝕀)</td>
</tr>
<tr>
<td>4. Define filling problem for Kan fibrations</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
</tr>
<tr>
<td>6. Construct univalence</td>
</tr>
<tr>
<td><strong>Directed Type Theory</strong></td>
</tr>
<tr>
<td>1. Begin with Cubical Type Theory</td>
</tr>
<tr>
<td>2. Add an interval: (𝟐)</td>
</tr>
<tr>
<td>3. Specify gen. cofibrations for (𝟐)</td>
</tr>
<tr>
<td>4. Define filling problem for covariant fibrations</td>
</tr>
<tr>
<td>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<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>5. Define universe of covariant fibrations</td>
</tr>
<tr>
<td>5. Define universe of Kan fibrations</td>
<td></td>
</tr>
<tr>
<td>• $U_{\text{Kan}}$ given by LOPS construction for relCom</td>
<td>• $U_{\text{Cov}}$ given by LOPS construction for relCov</td>
</tr>
<tr>
<td>• Lemma: relCov is in $U_{\text{Kan}}$, so $\text{El}<em>{\text{Cov}} : U</em>{\text{Cov}} \to U_{\text{Kan}}$</td>
<td></td>
</tr>
</tbody>
</table>
# Defining Bicubical Directed Type Theory

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

1. Begin with a topos
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)

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
Defining Directed Univalence

dua \ i \ A \ B \ f \ := \ \lambda \ i . \ \text{Glue} \left[ i = 0 \rightarrow (A, f : A \rightarrow B), \ i = 1 \rightarrow (B, \text{id}) \right] \ B \ : \ \text{Hom}_U 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 $U_{\text{Cov}}$:
  • $d\text{coe} : (\text{Hom } A \to B) \to (A \to B)$
  • $\text{dua} : (A \to B) \to \text{Hom } A \to B$
  • $\text{dua}_\beta : \prod f : A \to B . \text{Path } f (d\text{coe} (\text{dua } f))$
  • $\text{dua}_{\eta\text{fun}} : \prod p : \text{Hom } A \to B . \prod i : \mathbb{2} . p i \to (\text{dua} (d\text{coe } p)) i$
Naive Directed Univalence

• We're thus left with the following picture:

• To complete directed univalence, we need $\text{dua}_{\eta_{\text{fun}}}^{-1}$
What next?

- The proof in the bisimplicial model relies on the fact weak equivalences in the model are level-wise weak equivalences of simplicial sets.

- Three potential model structures with level-wise weak equivalences...with three separate challenges.

<table>
<thead>
<tr>
<th></th>
<th>Dedekind cubes aren’t Reedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reedy</td>
<td></td>
</tr>
<tr>
<td>Projective</td>
<td>not all types we need are fibrant</td>
</tr>
<tr>
<td>Injective</td>
<td>not easily defined as cofibrantly generated</td>
</tr>
</tbody>
</table>
What next?

- The proof in the bisimplicial model relies on the fact weak equivalences in the model are level-wise weak equivalences of simplicial sets.

- Three potential model structures with level-wise weak equivalences...with three separate challenges.

<table>
<thead>
<tr>
<th>Model</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reedy</td>
<td>Run into constructivity issue: degenerate cells not always decidable</td>
</tr>
<tr>
<td>Projective</td>
<td>not all types we need are fibrant</td>
</tr>
<tr>
<td>Injective</td>
<td>not easily defined as cofibrantly generated</td>
</tr>
</tbody>
</table>
Cobar and the Injective Model Structure

• Shulman classifies an injective fibrant object $A : C \to M$ as:
  • For every $c$ in $C$, $A_c$ is fibrant in the underlying model structure $M$ (i.e. $A$ is object-wise fibrant)
  • $A$ is equivalent to $\coobar(A)$

• Coquand and Ruch internalize the cobar construction in a syntactic setting,
  • ...and constructively show weak equivalences are object-wise!

• Idea: use the internal cobar and prove all types $A$ we care about are equivalent to $\coobar(A)$
  • Spoiler: This works to finish the construction of directed univalence!
  • Fine Print: The formal connection between the internal version and Shulman's work has not yet been worked out.
Lex Operators and Stack Models of Type Theory

• Coquand and Ruch define a general framework for internalizing lex operators and defining models of type theory localized at them:

• $D : \text{Type} \to \text{Type}$ is a strict lex endofunctor on types

• $\eta$ is a strict natural transformation $\text{Id} \to D$

• We can restrict the model to types $A$ that are stacks, i.e. $\eta_A$ is an equivalence from $A$ to $D A$

• Note: Cobar is a lex operator
Lex Axioms

$D$ is an endofunctor on $U_{\text{Kan}}$

$\eta$ is a natural transformation $\text{Id}_{U_{\text{Kan}}} \to D$

\[
D : U_{\text{Kan}} \to U_{\text{Kan}}
\]

\[
f : A \to B
\]

\[
D \ f : D \ A \to D \ B
\]

\[
f : A \to B \quad g : B \to C
\]

\[
D \ (g \circ f) = D \ g \circ D \ f
\]

\[
D \ (\lambda x : A . \ x) = \lambda x : D \ A . \ x
\]

\[
A : U_{\text{Kan}}
\]

\[
\eta_A : A \to D \ A
\]

\[
f : A \to B
\]

\[
D \ f \cdot \eta_A = \eta_B \circ f
\]
Lex Axioms

Additionally...

\[ L : D_{\mathsf{U_{Kan}}} \to U_{\mathsf{Kan}} \]

\[ \eta\text{-Path}_A : \text{Path}_{D^2 A} \to D A (D \eta_A) (\eta_{D A}) \]

\[ dD : (A \to U_{\mathsf{Kan}}) \to (D A \to U_{\mathsf{Kan}}) \]

\[ dD \, B := L \circ D \, B \]

D is Lex

\[ A : U_{\mathsf{Kan}} \quad B : A \to U_{\mathsf{Kan}} \]

\[ \Sigma\text{-snd}_B : (x : D \Sigma A.B) \to dD \, B (D \text{fst} \, x) \]

\[ \Sigma\text{-iso}_B : \text{isIso} (\lambda x \to D \text{fst} \, x, \Sigma\text{-snd}_B \, x) \]

(i.e. \( D \Sigma A.B \cong \Sigma D A. dD \, B \))
Closure Properties

• For an arbitrary lex endofunctor $D$...

  • ...if $B : A \to U$ is a family of stacks, then $\prod A . B$ is a stack.

  • ...if $A$ is a stack and $B : A \to U$ is a family of stacks, then $\Sigma A . B$ is a stack.

• For the other type formers we care about (i.e. Path, Hom, and Glue), we need specific information $D$ and $\eta$. 
Internalizing Cobar

• Main Idea: A natural transformation $A \to \text{cobar}(B)$ corresponds to a homotopy coherent transformation $A \rightsquigarrow B$

• We define our internal cobar operator $D$ by first defining a helper operator $E$.

• Intuition for $E$: For a type $A$ and every $X$ in $\text{Ded}$, an element of $E A(X)$ is an element in $a$ in $A(X)$ along with a choice for the action of every substitution $Y \to X$ as an element in $A(Y)$.

• Intuition for $D$: For every type $A$ and $X$ in $\text{Ded}$, an element of $D A(X)$ is a choice of $n$ elements of $A$ for every chain of $n$ composable morphisms into $X$ that are weakly coherent with respect to the substitution action given by $A$. 
Definition of $E$

- Given a bicubical set $A$, we define the bicubical set $E A$:
  - For $X$ in $\square_{\text{Ded}}$, \[
  E A(X) := \prod f : \text{Hom}(Y, X). A(Y)
  \]
  - For $f : \text{Hom}(Y, X)$, \[
  E A(f) := u : E A(X) \mapsto \lambda g : \text{Hom}(Z, Y). u(f \circ g)
  \]
  - We also define a natural transformation $\alpha : \text{Id} \to E$:
    \[
    \alpha_A (X) := a : A(X) \mapsto \lambda f : \text{Hom}(Y, X). af
    \]
Definition of D

• Given a bicubical set A, we define the bicubical set D A:
  • For X in \( \square \text{Ded} \),
    \[
    D A(X) := \prod n : \mathbb{N} . \mathbb{I}^n \to E^{n+1} A(X)
    \]
  • The family u must additionally satisfy some conditions
So...what is D?

<table>
<thead>
<tr>
<th>n</th>
<th>$u : D A(X)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 0$</td>
<td>$\text{id}_X \mapsto$ $a_0 : A(X)$</td>
</tr>
<tr>
<td>$n = 1$</td>
<td>$\text{id}_X, f : \text{Hom}(Y, X) \mapsto$ $a_0f : A(Y) = u(0, \text{id}_X)f = u(0, \text{id}_X \circ f)$, $a_1 : A(Y) = u(0, \text{id}_X \circ f)$</td>
</tr>
<tr>
<td>$n = 2$</td>
<td>$\text{id}_X, f : \text{Hom}(Y, X), g : \text{Hom}(Z, Y) \mapsto$ $a_1g : A(Z) = u(0, \text{id}_X \circ f)g$</td>
</tr>
</tbody>
</table>

\[ u(1, \text{id}_X, f)g = a_0fg : A(Z) = u(0, \text{id}_X)fg \]
\[ u(1, \text{id}_X \circ f, g) = a_2 : A(Z) = u(0, \text{id}_X \circ f \circ g) \]
Definition of D

• \( \text{diag} : \mathbb{I}^n[0 = i_1 \lor i_1 = i_2 \lor ... \lor i_{n-1} = i_n \lor i_n = 1] \rightarrow \mathbb{I}^{n-1} \) by forgetting \( i_1 \) on \( 0 = i_1 \), \( i_k \) on \( i_k = i_{k+1} \) and \( i_n \) on \( i_n = 1 \).

• Given a bicubical set \( A \), we define the bicubical set \( D A \):
  • For \( X \) in \( \mathbb{I}_{\text{Ded}} \),
    \[
    D A(X) := \prod n : \mathbb{N} \cdot \mathbb{I}^n \rightarrow E^{n+1} A(X)
    \]
  • The family \( u \) must additionally satisfy the following:
    • \( u = \alpha \circ u \circ \text{diag} \) when \( 0 = i_1 \)
    • \( u = E^k(\alpha) \circ u \circ \text{diag} \) when \( i_k = i_{k+1} \)
    • \( u = E^n(\alpha) \circ u \circ \text{diag} \) when \( i_n = 1 \)
  • For \( f : \text{Hom}(Y, X) \),
    \[
    D A(f) := u : D A(X) \mapsto \lambda n, i_1, ..., i_n. E^{n+1} A(f)(u(n, i_1, ..., i_n))
    \]
• We also define a natural transformation \( \eta : \text{Id} \rightarrow D \):
  \[
  \eta_A (X) := a : A(X) \mapsto \lambda n, i_1, ..., i_n. \alpha^{n+1}(a)
  \]
Additional Internal Axioms

DPath-iso : Iso (D (Path_\text{A}(a_0, a_1))) \text{Path}_{\text{D} A}(\eta_\text{A} a_0, \eta_\text{A} a_1)

DHom-iso : Iso (D (Hom_\text{A}(a_0, a_1))) \text{Hom}_{\text{D} A}(\eta_\text{A} a_0, \eta_\text{A} a_1)

Ddua-iso : Iso (D (\text{dua}_i A B f)) (\text{dua}_i (\text{D} A) (\text{D} B) (\text{D} f))

(actual axioms specify how these isomorphisms compute in relation to $\eta$)
More Closure Properties

• For this specific lex endofunctor D...

  • If a type A is a stack, then for any terms \(a_0, a_1\) in A, both \(\text{Path}_A(a_0, a_1)\) and \(\text{Hom}_A(a_0, a_1)\) are stacks.

  • If types A and B are stacks, then for any \(i : 2\) and function \(f : A \rightarrow B\), \(\text{dua}_i A B f\) is a stack.
Completing Directed Univalence

- The construction of directed univalence follows in two steps:
  
  1. Given a function $f : A \rightarrow B$ between stacks, if $f$ is an object-wise equivalence of cubical sets then it is an equivalence of bicubical sets (Coquand and Ruch).
  
  2. The function $\text{dua}_{\eta \text{fun}}$ is an object-wise equivalence of cubical sets (modified from bisimplicial proof of Cavallo, Riehl and Sattler).
Lastly, we define the universe that supports directed univalence:

\[ U_{\text{CovStack}} := \Sigma A : U_{\text{Cov}} . \text{isStack } A \]
Our Results

- **Main Theorem:** There exists a constructive model of type theory in bicubical sets with a universe of fibrant types ($U_{Kan}$) and a universe of covariant fibrations ($U_{CovStack}$) such that:
  - $U_{CovStack}$ has a decode function into $U_{Kan}$;
  - $U_{Kan}$ is closed under $\Pi$, $\Sigma$, $DPath$, $DHom$ and contains codes for smaller $U_{CovStack}$ and $U_{Kan}$;
  - $U_{CovStack}$ is closed under $\Pi$ (with fixed closed domain), $\Sigma$, $DPath$ and $DHom$;
  - $U_{Kan}$ and $U_{CovStack}$ are both path univalent;
  - $U_{CovStack}$ is morphism (directed path) univalent.

- Formalized in Agda!