A Constructive Model of Directed Univalence in Bicubical Sets

Matthew Z. Weaver
Department of Computer Science
Princeton University
mzw@cs.princeton.edu

Daniel R. Licata
Department of Mathematics and Computer Science
Wesleyan University
dlicata@wesleyan.edu

Abstract

Directed type theory is an analogue of homotopy type theory where types represent categories, generalizing groupoids. A bisimplicial approach to directed type theory, developed by Riehl and Shulman, is based on equipping each type with both a notion of path and a separate notion of directed morphism. In this setting, a directed analogue of the univalence axiom asserts that there is a universe of covariant discrete fibrations whose directed morphisms correspond to functions—a higher-categorical analogue of the category of sets and functions. In this paper, we give a constructive model of a directed type theory with directed univalence in bicubical, rather than bisimplicial, sets. We formalize much of this model using Agda as the internal language of a 1-topos, following Orton and Pitts. First, building on the cubical techniques used to give computational models of homotopy type theory, we show that there is a universe of covariant discrete fibrations, with a partial directed univalence principle asserting that functions are a retract of morphisms in this universe. To complete this retraction into an equivalence, we refine the universe of covariant fibrations using the constructive sheaf models by Coquand and Ruch.

CCS Concepts: • **Theory of computation → Type theory**; *Constructive mathematics.*

Keywords: homotopy type theory, directed type theory, directed univalence

ACM Reference Format:
Matthew Z. Weaver and Daniel R. Licata. 2020. A Constructive Model of Directed Univalence in Bicubical Sets. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20), July 8–11, 2020, Saarbrücken, Germany.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3373718.3394794

LICS '20, July 8–11, 2020, Saarbrücken, Germany

ACM ISBN 978-1-4503-7104-9/20/07.
https://doi.org/10.1145/3373718.3394794

1 Introduction

In homotopy type theory [6, 38, 39], each type is equipped with an abstract notion of path, corresponding to the morphisms of a higher groupoid. Voevodsky's univalence axiom states that the type of paths between types is equivalent (a generalization of isomorphic) to the type of equivalences between types. This can be seen as extending type theory with a generic program for lifting equivalences between types along any type family: given an equivalence $A \simeq B$ and any family of types $C : U \to U$, one can automatically construct an equivalence $C(A) \simeq C(B)$. Defining this lifting for all dependent types is quite subtle, but univalence has recently been given a computational interpretation via cubical type theories [2, 4, 7, 11].

A natural question is whether there are any *directed* analogues of homotopy type theory, where types are equipped with a notion of directed morphism and correspond to higher categories, generalizing groupoids. In such a setting, one possible directed analogue of univalence would circumscribe a class of type constructors that represent covariant functors, and provide a means to automatically lift a *function* $A \to B$ to a function $C(A) \to C(B)$, generalizing e.g. the Functor type class of Haskell to dependent types. By analogy with univalence, such a lifting would result from a universe classifying covariant type families that satisfies a *directed univalence* principle stating that the directed morphisms in this universe are equivalent to functions. On the mathematical side, directed type theory provides a more general language for synthetic mathematics. On the computational side, one potential application is to defining abstract syntax [16, 20, 22]; others might arise from the connections between directed homotopy theory and concurrency [15, 18, 24].

In directed type theory, a universe of covariant type families will not contain all of the usual types of type theory—e.g., Π is contravariant, not covariant, in its domain. One approach that has been taken by directed type theory is to employ some kind of modal typing discipline, tracking variances or preventing certain uses of variables [22, 24, 25], so that being a covariant family is part of the typing of a term. Another approach, recently developed by Riehl and Shulman [32], is based on instead equipping each type with *both* a notion of path and a separate notion of directed morphism (representing the arrows in a category). In this approach, all existing homotopy type theory can be interpreted using the

path structure on each type, providing a rich language of possible constructions. Being a covariant family is an *internally definable property* of a type, represented by another type that can be inhabited for a variety of type constructors—i.e. covariance is a type class. While ordinary homotopy type theory has models in *simplicial sets*, the Riehl-Shulman type theory is based on a model in *bisimplicial sets*, $\mathbf{Set}^{\Delta^{\mathrm{op}}\times\Delta^{\mathrm{op}}}$, where the two copies of the simplex category Δ represent these two notions of path and directed morphism. While this suffices for formalizing mathematics, for applications to computer science we would like a computational interpretation of the type theory, and a constructive account of even ordinary homotopy type theory in simplicial sets has not fully been worked out (see [17] for some recent progress).

In this paper, in place of a bisimplicial model of directed type theory, we consider a *bicubical* one, so that we can exploit the techniques that have been used to give constructive interpretations of ordinary homotopy type theory. We give a model of directed type theory with directed univalence in bicubical sets. This model is constructive in the sense of being defined in a constructive metatheory; we leave a more explicit operational semantics to future work. We have formalized[1] much of the model in Agda using the internal language approach to presheaf models of homotopy type theory [8, 23, 27, 28]. In this approach, an extensional type theory (or Agda with some axioms as a substitute [19]) is used to describe cubical-set level constructions, and the main task is to program the definitions/proofs of the *Kan operations*, which describe how to transport in each type family. Bicubical sets is also a presheaf topos, and in Section 2 we describe the axioms we add to the Agda proof assistant to work in its internal language.

In Section 3, we use this internal language to define covariant discrete fibrations (covariant functors valued in types that themselves have trivial morphism structure) and a universe classifying them. In Section 4, we show that functions are a *reflection* of morphisms in this universe: every morphism determines a function, and vice versa; converting a function to a morphism and back is the identity; but for the other round-trip there seems to be only a morphism in one direction, not a path. This reflection is most of the directed univalence *equivalence* between functions and morphisms, but to complete the equivalence, it is necessary to invert this morphism, yielding a path. To do this, Riehl and Sattler's proof of directed univalence in the bisimplicial model [31, 35] uses a fact about the Reedy model structure on bisimplicial sets, that weak equivalences (and therefore homotopy equivalences between bifibrant objects) are *objectwise*—this means that if a morphism is an equivalence for each object of the directed category separately, then it is an equivalence. For our constructive model, we use a cubical, not simplicial, directed category (specifically, the "Dedekind" cube category with faces, degeneracies, symmetries, diagonals, and connections, but no reversals) so that we can build universes of covariant fibrations constructively [23]. This cube category is not Reedy [34], so the proof for bisimplicial sets does not immediately apply to bicubical sets. However, for bisimplicial sets, the Reedy model structure coincides with the injective model structure (see e.g. [33, Example 7.8]), and the injective perspective applies to bicubical sets, allowing us to port this aspect of Sattler and Riehl's proof. Recently, Shulman [37] gave a new characterization of the fibrations in the injective model structure as objectwise fibrations equipped with an algebra for the *cobar construction* (see e.g. [29, 36]). Coquand and Ruch [12, 13] give a constructive analogue of this definition as a special case of sheaf models (though the precise connection with the injective model structure remains to be worked out). In Section 5, we describe an Agda formalization of this approach, and check that the axioms describing this extension are true in bicubical sets. We then add an axiom covEquivAx that is true for this refined notion of fibration, and use it to complete the directed univalence equivalence for a refined covariant universe.

Related work. Our proof of the directed univalence reflection does not overlap very much with Riehl and Sattler's directed univalence for bisimplicial sets [31, 35]—for example, we use the "glue" types used to define univalence in constructive cubical models [11], while their proof uses a dual type (called "weld" types in [26]), and we use [23] to build universes constructively. However, for completing the reflection to an equivalence, we do use a lemma from their proof, which follows from objectwise homotopy equivalences inducing homotopy equivalences.

Buchholtz and Weinberger [9] give some extensions of the Riehl-Shulman directed type theory, such as adding opposite categories, defining cartesian fibrations, and defining a universe of simplicial spaces inside of cubical spaces via a sheaf condition. In future work, we plan to investigate whether the techniques we develop here can be used to give a constructive account of directed univalence for a universe of cartesian fibrations (categories and functors, instead of groupoids and functors) as well.

Bicubical sets have also been used for type theory with parametricity: in Nuyts's approach to parametricity for (non-homotopy) type theory [26], both cube categories are cartesian, and there is an inclusion from one interval the other. Here, we do not take an inclusion from the directed interval to the interval to be part of the cube category (but paths can be turned into morphisms in Kan types). In Cavallo and Harper's approach to parametricity for homotopy type theory [10], the cube category used for paths is cartesian, while the cube category used for relations is semicartesian (affine logic). In contrast, our directed cube category also supports

[1] See the `agda/directed` directory of https://github.com/dlicata335/cart-cube, starting with `LICS.agda` and `Summary.agda`

diagonals and connections to encode the simplicial shapes used by Riehl and Shulman [32].

Motivating Example. Before the technical details, we sketch a motivating example of defining abstract syntax in a directed type theory, building on [16, 20, 22]. This sketch will use a directed higher inductive type, which we do not formally study in this paper. We write $\mathsf{Hom}_A(x, y)$ for the directed morphism type, representing a morphism from x to y—this should be thought of like the $\mathsf{Path}_A(x, y)$ or identity $\mathsf{Id}_A(x, y)$ type of homotopy type theory, except in general the morphisms have identity, but not necessarily composition or inverses. Those types A whose morphisms have (homotopy) unique composites (and so correspond to categories) are called *Segal* [32].

The first idea is to represent object-language contexts by a type whose elements represent contexts, and whose morphisms represent object-language substitutions or renamings. As a concrete example, we use the untyped lambda calculus, indexed by contexts denoting how many free variables are present, and take the substitutions to be weakenings only. The type of contexts should be defined as a Segal directed higher inductive type:

```
data Ctx : SegalType where
  empt : Ctx
  ext  : Ctx → Ctx
  wk   : (G : Ctx) → Hom Ctx G (ext G)
```

This declaration is intended to mean that the elements of Ctx are in bijection with the natural numbers, with directed morphisms generated by $\mathsf{wk}_n : \mathsf{Hom}_{\mathsf{Ctx}}(n, n+1)$ corresponding to weakening. We declare Ctx to be Segal, which means that it should have freely generated composites; e.g. $\mathsf{wk}_0; \mathsf{wk}_1 : \mathsf{Hom}_{\mathsf{Ctx}}(0, 2)$ represents weakening the empty context with two variables. Because functions such as ext act on morphisms, we also have morphisms like $\mathsf{ext}(\mathsf{wk}_0) : \mathsf{Hom}_{\mathsf{Ctx}}(1, 2)$ that weakens in the middle of a context.

We then define a type family $\mathsf{Var} : \mathsf{Ctx} \to \mathsf{U}_{\mathsf{cov}}$ representing the variables in each context. The type of Var expresses that, in addition to providing a type of "variables in context G for each context G," it takes morphisms in Ctx (i.e. weakenings) to morphisms in $\mathsf{U}_{\mathsf{cov}}$, which by directed univalence will be functions. A particular transport function from n to $n+1$ is given for the weakening axiom—the inl injection, which is turned into a morphism in UCov using directed univalence. The elimination form for a Segal higher inductive type would then send compositions of weakenings to compositions of these injections. Thus, the type family Var includes the code for shifting a de Bruijn index.

```
Var : Ctx → UCov
Var empt    = 0
Var (ext G) = (Var G) + 1
Var (wk G)  = dua inl
```

```
data Tm (G : Ctx) : UCov where
  var : Var G → Tm G
  abs : Tm (ext G) → Tm G
  app : Tm G → Tm G → Tm G
```

Finally, we define a type family of lambda terms $\mathsf{Tm} : \mathsf{Ctx} \to \mathsf{U}_{\mathsf{cov}}$ that is constructed from a base case Var that maps into $\mathsf{U}_{\mathsf{cov}}$, and otherwise is a sum of products of inductive occurrences. Because $\mathsf{U}_{\mathsf{cov}}$ is closed under products, sums, and certain inductive types, Tm will *automatically* land in $\mathsf{U}_{\mathsf{cov}}$. This means that, while the definition of Tm looks identical to what one would write in standard type theory, the usual code for lifting weakening to terms is written automatically by the transport functions for these types. We use this by

```
dtransport Tm (wk G) : Tm G → Tm (ext G)
```

to, for example, weaken a term with one extra variable, where dtransport is an operation defined for any type family that lands in $\mathsf{U}_{\mathsf{cov}}$. General facts about directed transport will imply that these substitution actions commute with composition, which is needed in metatheory and in defining typing judgements, program logics, or dependent types.

For this example, we need a universe $\mathsf{U}_{\mathsf{cov}}$ with directed univalence, closed under products, sums, and inductive types, which is itself a category (*Segal* [32]); and we also need directed higher inductive types. Of these, we leave the closure of $\mathsf{U}_{\mathsf{cov}}$ under inductive types and the mechanism of directed higher inductive types to future work, and focus on defining a universe $\mathsf{U}_{\mathsf{cov}}$ supporting directed univalence.

2 Directed Type Theory in a Topos

2.1 Axioms for Cubical Type Theory

Cubical type theory [2, 4, 7, 11] extends standard dependent type theory with interval variables, which we write as $x : \mathbb{I}$. A term $x : \mathbb{I} \vdash a : A$ is a "line" in A, a term $x : \mathbb{I}, y : \mathbb{I} \vdash a : A$ is a "square", and so on. The interval type has terms $\mathbb{0}_{\mathbb{I}}$ and $\mathbb{1}_{\mathbb{I}}$, substitution by which gives the endpoints of a line or the sides of a square. Dimension formulas represent subshapes of a cube—for example, the formula $x : \mathbb{I}, y : \mathbb{I} \vdash x = \mathbb{0}_{\mathbb{I}} \vee y = \mathbb{1}_{\mathbb{I}}$ represents the left and top sides of a square. The main two constructive models of homotopy type theory use the Cartesian cube category $\square_{\mathrm{Cart}}$ [2, 4] (the free cartesian category on an interval object $\mathbb{I}$ with disjoint endpoints $\mathbb{0}_{\mathbb{I}}$ and $\mathbb{1}_{\mathbb{I}}$) and the de Morgan cube category [11] (which adds connection maps, giving the interval the structure of a distributive lattice, as well as reversals). A variation on the de Morgan cube category is the Dedekind cube category $\square_{\mathrm{Ded}}$, which adds the connections but not reversals, and was shown to also give a constructive model in [28]. For the path portion of our directed homotopy type theory, we will use the axiomatic description of Cartesian cubical type theory [2], which can be interpreted in presheaves on any cartesian category with an interval object, including both the Cartesian cube category and the Dedekind cube category. To abstract the specific

choice between these, we write $\square_{\text{path}}$ to represent some cartesian cube category with an interval.

In the internal language approach [8, 23, 27, 28], models of cubical type theories are described by axioms in an extensional type theory that serves as the internal language of a presheaf 1-topos. This means that we use type theoretic syntax (formalized in Agda, with postulates for function extensionality and uniqueness of identity proofs [19]) to describe constructions in cubical sets, or, in our case, bicubical sets. In homotopy type theory based on Martin-Löf intensional type theory [38], all types are *fibrant*, which roughly means that all types have a transport function for paths. In the internal language approach, types instead denote "raw" cubical sets, which do not necessarily have such transport functions. Being fibrant is an internally definable structure on a type, in the style of type classes: for any type (family) A, there is another type classifying fibration structures on A.

We start with the axioms of Cartesian cubical type theory in [2, Section 3.1.1]. These describe the free Cartesian category generated by an interval object $\mathbb{I}$ with two distinct terms $\mathbb{0}_{\mathbb{I}}$ and $\mathbb{1}_{\mathbb{I}}$. A function $f : \mathbb{I} \to A$ represents a path in A, with source $f(\mathbb{0}_{\mathbb{I}})$ and target $f(\mathbb{1}_{\mathbb{I}})$. In cubical type theory, the

$$\begin{array}{rl} \mathsf{transport} : & \Pi(A : \mathsf{Type}), (a_0, a_1 : A), (C : A \to \mathsf{Type}). \\ & \mathsf{Path}_A\,(a_0, a_1) \to C(a_0) \to C(a_1) \end{array}$$

operation of dependent type theory is generalized to a notion of *Kan filling*, which permits the induction to go through in a recursive definition of transport for each type family C. For example, Kan filling includes a primitive notion of "filling an open box", meaning that given three sides of a square, one can produce the missing side and an inside *filler* witnessing that the square commutes, which is used to define transport in path types. The shapes of the filling problems are determined by the *cofibrations*, which are a subset of propositions. In the internal language approach [27, 28], an Agda type iscof identifies those propositions which are cofibrations; the axioms about cofibrations needed in [2] are:

$$\begin{array}{rcl} \mathsf{iscof} & : & \Omega \to \Omega \\ \\ \Omega_{\text{cof}} & : & \mathsf{Type} \\ \Omega_{\text{cof}} & := & \Sigma\alpha : \Omega.\mathsf{iscof}\ \alpha \\ \\ [\![\_]\!]_{\text{cof}} & : & \Omega_{\text{cof}} \to \Omega \\ [\![\alpha]\!]_{\text{cof}} & := & \mathsf{fst}\ \alpha \\ \\ \mathsf{iscof}\bot & : & \mathsf{iscof}\ \bot \\ \mathsf{iscof}\wedge & : & \Pi\alpha\ \beta : \Omega_{\text{cof}}.\mathsf{iscof}\ [\![\alpha]\!]_{\text{cof}} \wedge [\![\beta]\!]_{\text{cof}} \\ \mathsf{iscof}\vee & : & \Pi\alpha\ \beta : \Omega_{\text{cof}}.\mathsf{iscof}\ [\![\alpha]\!]_{\text{cof}} \vee [\![\beta]\!]_{\text{cof}} \\ \mathsf{iscof}\mathbb{I}\mathsf{eq} & : & \Pi i\ j : \mathbb{I}.\mathsf{iscof}(i = j) \\ \mathsf{iscof}\forall_{\mathbb{I}} & : & \Pi\alpha : \mathbb{I} \to \Omega_{\text{cof}}.\mathsf{iscof}(\Pi i : \mathbb{I}.[\![\alpha\ i]\!]_{\text{cof}}) \end{array}$$

This hypothesizes a predicate iscof, which determines when a proposition is a cofibration, and says that cofibrations are closed under conjunction, disjunction, equality on the interval, and universal quantification over the interval. We write Ω_{cof} for the type of cofibrations, and $[\![\alpha]\!]_{\text{cof}}$ for the underlying proposition of a cofibration. Here, Ω is the type of strict propositions in Type_0, i.e. types such that $\Pi x, y : \alpha, x = y$. Conjunction $\wedge$ and universal quantification Π are the Agda types, while a "squashed" disjunction $\vee$ (where any two proofs are equal) is postulated.

Given a type A and a cofibration α, a *partial term* of type A over the cofibration α is an element of $[\![\alpha]\!]_{\text{cof}} \to A$. Given a partial term $a : [\![\alpha]\!]_{\text{cof}} \to A$, the type of *extensions*[2] of a is defined by $A[\alpha \mapsto a] := \Sigma x : A.\Pi p : [\![\alpha]\!]_{\text{cof}}.a\ p = x$. This represents an element of A, that when α is true, restricts to a. For disjunctions, we use the notation $A[\alpha_1 \mapsto a_1, \alpha_2 \mapsto a_2]$ to denote the extension type $A[\alpha_1 \vee \alpha_2 \mapsto a]$ where a_x is the restriction of a to α_x. The disjunction $\alpha_1 \vee \alpha_2$ represents the pushout of the pullback of α_1 and α_2, which requires that a_1 and a_2 be strictly equal along the intersection $\alpha_1 \wedge \alpha_2$. When using an extension $x : A[\alpha \mapsto a]$, we often will omit the projection from $A[\alpha \mapsto a]$ to A and write $x : A$ as well.

Using cofibrations and extensions, the *Kan operation* that fibrant types satisfy is defined as follows [2, Section 3.1.2]: Given a type Γ, a type family $A : \Gamma \to \mathsf{Type}$ is a *Kan fibration* if one can define a relCom_Γ structure for it:[3]

$$\begin{array}{rl} \mathsf{hasCom}\ (A : \mathbb{I} \to \mathsf{Type}) := & \Pi r\ r' : \mathbb{I}.\Pi\alpha : \Omega_{\text{cof}}. \\ & \Pi t : (\Pi z : \mathbb{I}.[\![\alpha]\!]_{\text{cof}} \to A\ z). \\ & \Pi b : (A\ r)[\alpha \mapsto t\ r]. \\ & (A\ r')[\alpha \mapsto t\ r', r = r' \mapsto b] \\ \mathsf{relCom}_\Gamma\ (A : \Gamma \to \mathsf{Type}) := & \Pi p : (\mathbb{I} \to \Gamma).\mathsf{hasCom}(A \circ p) \end{array}$$

This says that a type family $A : \Gamma \to \mathsf{Type}$ is Kan when, for any path p in Γ, any partial path t over p in A, and any total point b over the path p in A at r, we can get a new point in A over p at any location r' along the path, which restricts to t and b where they are defined.

These Kan fibrations form a model of Martin-Löf type theory with univalence and (some) higher inductive types: relCom can be inhabited for Π, Σ, path, identity, natural number, boolean, glue and some higher inductive types, and the universe itself is fibrant and univalent [2, Section 3]. We write $\mathsf{U}_{\mathsf{Kan}} : \mathsf{Type}$ for the universe of Kan fibrations [23]. We can define all standard concepts of homotopy type theory, and we write $\mathsf{isEquiv}\ f$ and $\mathsf{Equiv}\ A\ B$ for the standard definition of half-adjoint equivalence [38].

2.2 Axioms for Bicubical Type Theory

In Figure 1, we describe the additional axioms needed to describe a directed type theory based on bicubical sets. Riehl-Shulman [32] extend homotopy type theory with a directed interval $\mathbb{2}$, which is used to describe simplices (point, line, triangle, tetrahedron, etc.) by inequality formulas—e.g. the triangle is the top half of the square $x : \mathbb{2}, y : \mathbb{2} \mid y \leq x$. Here, we extend cubical type theory, which already has an interval

[2] This is slightly different than the extension types in [32], which combine extension in this sense with a quantifier, and are restricted in such a way that they are always fibrant.

[3] Formally the term after the $\mapsto$ should be a function that takes a proof of the boundary formula as input, but since we're eliding the names for these variables, we will often write what would be in the body of that function.

$$
\begin{array}{rl}
2 : & \mathsf{Type} \\
\mathbb{0}_2 : & 2 \\
\mathbb{1}_2 : & 2 \\
2\mathsf{neq} : & \mathbb{0}_2 = \mathbb{1}_2 \to \bot \\
\_ \sqcap \_ : & 2 \to 2 \to 2 \\
\sqcap \mathbb{0}_2 : & \Pi x : 2.x \sqcap \mathbb{0}_2 = \mathbb{0}_2 \\
\sqcap \mathbb{1}_2 : & \Pi x : 2.x \sqcap \mathbb{1}_2 = x \\
\sqcap\mathsf{comm} : & \Pi x\ y : 2.x \sqcap y = y \sqcap x \\
\sqcap\mathsf{idem} : & \Pi x : 2.x \sqcap x = x \\
\sqcap\mathsf{assoc} : & \Pi(x, y, z : 2). \\
& ((x \sqcap y) \sqcap z) = (x \sqcap (y \sqcap z)) \\
\_ \sqcup \_ : & 2 \to 2 \to 2 \\
\sqcup \mathbb{0}_2 : & \Pi x : 2.x \sqcup \mathbb{0}_2 = x \\
\sqcup \mathbb{1}_2 : & \Pi x : 2.x \sqcup \mathbb{1}_2 = \mathbb{1}_2 \\
\sqcup\mathsf{comm} : & \Pi x\ y : 2.x \sqcup y = y \sqcup x \\
\sqcup\mathsf{idem} : & \Pi x : 2.x \sqcup x = x \\
\sqcup\mathsf{assoc} : & \Pi(x, y, z : 2). \\
& ((x \sqcup y) \sqcup z) = (x \sqcup (y \sqcup z)) \\
\mathsf{abs}_0 : & \Pi x\ y : 2.x \sqcap (x \sqcup y) = x \\
\mathsf{abs}_1 : & \Pi x\ y : 2.x \sqcup (x \sqcap y) = x \\
\\
\mathsf{iscof}2\mathsf{eq} : & \Pi i\ j : 2.\mathsf{iscof}\ (i = j) \\
\mathsf{iscof}\forall_2 : & \Pi\alpha : 2 \to \Omega_{\mathsf{cof}}. \\
& \mathsf{iscof}(\ \Pi i : 2.[\![\alpha\ i]\!]_{\mathsf{cof}}) \\
2 & \text{ is tiny} \\
\\
\mathbb{I}\mathsf{to}2\mathsf{triv} : & \Pi p : \mathbb{I} \to 2.\Pi x : \mathbb{I}. \\
& p\ x = p\ \mathbb{0}_{\mathbb{I}} \\
2\mathsf{mono} : & \Pi p : 2 \to 2.\Pi x\ y : 2. \\
& x = x \sqcap y \to p\ x = p(x) \sqcap p(y)
\end{array}
$$

Figure 1. Axioms for Bicubical Directed Type Theory

for paths $\mathbb{I}$, with a second *directed interval* 2. On the one hand, we would like the directed interval to support a constructive definition of a universe of covariant fibrations using [23], which relies on the interval being tiny (exponentiation by the interval has a right adjoint), which is not true for the interval in simplical sets. On the other hand, we would like this interval to support the body of definitions that are made in [32], where triangular shapes are sometimes important—e.g. a type is a category (roughly) if, given any two composable directed morphisms, there is a unique composite morphism and a triangle relating them. We can accomplish both of these using the Dedekind cube category $\square_{\mathrm{Ded}}$, in which the interval is defined to be a distributive lattice, where the lattice meet and join are referred to as *connections*. Using the lattice structure, we can define $x \le y$ as $x = x \sqcap y$ (x is the min of x and y). In Figure 1, the first group of axioms describe Dedekind cubes: the interval has two disjoint points; there are meet and join operations satisfying the distributive lattice laws. Relative to the simplex category axioms [32], this omits the requirement that $\le$ is a *total* order ($x \le y \vee y \le x$ for all x and y, which allows for defining a square in a type A by pasting together two triangles in A on the diagonal). This omission does not seem to be a problem: first, we have checked that some of the definitions that use totality in [32] in fact only use it via the connections; second, many internal constructions involve Segal or discrete types, which satisfy totality weakly (with a uniqueness principle up to paths not strict equality); third, one can internally carve out the simplices via a sheaf condition [9] to state theorems about types that do satisfy totality.

The next two axioms in Figure 1, $\mathsf{iscof}2\mathsf{eq}$ and $\mathsf{iscof}\forall_2$, say that equality on the directed interval, and Π over the directed interval, are cofibrations, extending the class of shapes that can be filled in Kan types. For example, $\mathsf{iscof}2\mathsf{eq}$ allows converting a path to a morphism in a Kan type: given $p : \mathbb{I} \to A$, we make $2 \to A$ by the Kan composition

$$
\begin{array}{l}
\mathsf{path\_to\_hom}(p) := \\
\quad \lambda(x : 2).\mathsf{com}_A^{z:\mathbb{0}_{\mathbb{I}} \to \mathbb{1}_{\mathbb{I}}}\ [x = \mathbb{0}_2 \mapsto p(\mathbb{0}_{\mathbb{I}}) \\
\qquad\qquad\qquad\qquad , x = \mathbb{1}_2 \mapsto p(z)](p(\mathbb{0}_{\mathbb{I}}))
\end{array}
$$

which has endpoints $p(\mathbb{0}_{\mathbb{I}})$ at $\mathbb{0}_2$ and $p(\mathbb{1}_{\mathbb{I}})$ at $\mathbb{1}_2$. $\mathsf{iscof}\forall_2$ is used to show that glue types [11] (where the function is an equivalence) are covariant families, which is the key lemma for showing that $\mathsf{U}_{\mathsf{cov}}$ is path-univalent. Tininess is for building $\mathsf{U}_{\mathsf{cov}}$ [23].

The final two axioms, $\mathbb{I}\mathsf{to}2\mathsf{triv}$ and $2\mathsf{mono}$, are used below in the construction of directed univalence. $\mathbb{I}\mathsf{to}2\mathsf{triv}$ says that any map from the undirected interval to the directed interval is constant. The second axiom says that any function $2 \to 2$ is monotonic—if $x \le y$ then $p(x) \le p(y)$—which, for example, rules out a reversal $1 - x$.

2.3 Soundness in Bicubical Sets

Extensional type theory with the axioms in Figure 1 can be interpreted in bicubical sets, the presheaf topos $\mathbf{Set}^{\square_{\mathrm{path}}^{\mathrm{op}} \times \square_{\mathrm{Ded}}^{\mathrm{op}}}$:

Theorem 2.1. *Let $\square_{\mathrm{path}}$ be a cartesian category with an interval object $\mathbb{I}$ with two distinct endpoints $\mathbb{0}_{\mathbb{I}}$ and $\mathbb{1}_{\mathbb{I}}$ in $1 \to \square_{\mathrm{path}}\ \mathbb{I}$, such that for any object C, the hom-set $C \to \square_{\mathrm{path}}\ \mathbb{I}$ has decidable equality. Then the axioms for cartesian cubical type theory in [2, Section 3.1.1], and the axioms for bicubical directed type theory in Figure 1 are true in $\mathbf{Set}^{\square_{\mathrm{path}}^{\mathrm{op}} \times \square_{\mathrm{Ded}}^{\mathrm{op}}}$.*

The proof is in an extended version of this article [40].

3 Universe of Covariant Discrete Fibrations

Intuitively, a type $B : A \to \mathsf{Type}$ is a covariant discrete fibration [32] when it is covariant, in the sense that given a directed morphism $\mathsf{Hom}_A(a_0, a_1)$ there is a transport function $B(a_0) \to B(a_1)$ (satisfying a certain universal property), and each fiber $B(a)$ is discrete, in the sense that it is a groupoidal type with trivial directed morphisms. In this section, we define a universe $\mathsf{U}_{\mathsf{cov}}$, where a type $B : A \to \mathsf{U}_{\mathsf{cov}}$ will be a covariant discrete fibration over A. Because of the directed univalence principle we construct below, $\mathsf{U}_{\mathsf{cov}}$ itself is a higher-categorical analogue of the category of sets and functions, but where "set" means ∞-groupoid and function

means functor. We could similarly define a universe of contravariant discrete fibrations, but we focus (arbitrarily) on the covariant case for this paper.

Because the definition of covariant uses the directed morphism type, we first show that it follows from our axioms.

3.1 Directed Morphism Type

Given a type A and two terms a_0 and a_1, the type of directed morphisms from a_0 to a_1 is

$$\mathsf{Hom}_A\,(a_0, a_1) := \Pi i : \mathbb{2}.A[i = \mathbb{0}_{\mathbb{2}} \mapsto a_0, i = \mathbb{1}_{\mathbb{2}} \mapsto a_1]$$

Compare with the cubical path type $\mathsf{Path}_A\,(a_0, a_1) := \Pi i : \mathbb{I}.A[i = \mathbb{0}_{\mathbb{I}} \mapsto a_0, i = \mathbb{1}_{\mathbb{I}} \mapsto a_1]$. We can also define dependent morphism types where, instead of a fixed type, we consider a family of types indexed by the corresponding interval type: Given a type family $A : \mathbb{2} \to \mathsf{Type}$ and two terms $a_0 : A\ \mathbb{0}_{\mathbb{2}}$ and $a_1 : A\ \mathbb{1}_{\mathbb{2}}$, the type of directed paths from a_0 to a_1 is

$$\mathsf{DHom}_A\,(a_0, a_1) := \Pi i : \mathbb{2}.(A\ i)[i = \mathbb{0}_{\mathbb{2}} \mapsto a_0, i = \mathbb{1}_{\mathbb{2}} \mapsto a_1]$$

In [32], morphism types are defined as extension types, which combine quantification over interval variables and extension constraints, restricted in such a way that an extension type is always fibrant. Here, we must show that the morphism type is Kan:

Theorem 3.1 (`universe.Hom.Hom-code-universal`). *There is a code*

$$\mathsf{dhom} : \Pi(A : \mathbb{I} \to \mathsf{U}_{\mathsf{Kan}}).A(\mathbb{0}_{\mathbb{I}}) \to A(\mathbb{1}_{\mathbb{I}}) \to \mathsf{U}_{\mathsf{Kan}}$$

such that $\mathsf{El}(\mathsf{dhom}(A, a_0, a_1)) = \mathsf{DHom}_{\mathsf{El}\circ A}\,(a_0, a_1)$

The proof uses the fact that, for a directed interval variable x, $x = 0$ and $x = 1$ are cofibrations; the argument is then the same as the argument for Kan path types [2, 11].

3.2 Covariant Discrete Fibrations

For a type family $A : \mathbb{2} \to \mathsf{Type}$ to be covariant should mean that there is a "coercion" function $A(\mathbb{0}_{\mathbb{2}}) \to A(\mathbb{1}_{\mathbb{2}})$. For this to be well-behaved, it is necessary to moreover insist that this function has a universal property relative to A: for each input, its result is related to the input by the relation DHom_A, and in fact the unique such element of $A(\mathbb{1}_{\mathbb{2}})$. Uniqueness here is up to homotopy (as opposed to strict equality), using the standard notion of a type A being contractible ($\mathsf{iscontr}\ A := \Sigma a : A.\Pi x : A.\mathsf{Path}_A\,(a, x)$)—this style of definition is one reason why the separate notions of morphisms and paths in bisimplicial setting are helpful. For a general type family $A : \Gamma \to \mathsf{Type}$, one asks that any precomposition with a directed path in Γ has such a coercion function:

Definition 3.2 ([32, Definition 8.2]). Consider a type Γ and type family $A : \Gamma \to \mathsf{Type}$. A is *covariant* if for every morphism $p : \mathsf{Hom}_\Gamma\,(x, y)$ and point $a_0 : A\ x$, there is a unique element $a_1 : A(p\mathbb{1}_{\mathbb{2}})$ with a dependent path over $A\circ p$ starting at a_0:

$$\begin{aligned}\mathsf{iscov}\ A := \quad & \Pi p{:}\mathbb{2} \to \Gamma.\Pi a_0{:}A\,(p\ \mathbb{0}_{\mathbb{2}}).\\ & \mathsf{iscontr}\,(\sigma a_1 : A\,(p\ \mathbb{1}_{\mathbb{2}}).\mathsf{DHom}_{A\circ p}\,(a_0, a_1))\end{aligned}$$

While we could use this definition unchanged, in our setting, we can give an equivalent definition that will be more convenient. First, an equivalent (for Kan types) definition of being contractible is that any partial element can be extended to a complete element [11, 27]:

$$\mathsf{iscontr}'\ A := \Pi\alpha{:}\Omega_{\mathsf{cof}}.\Pi p{:}([\![\alpha]\!]_{\mathsf{cof}} \to A).A[\alpha \mapsto p]$$

The idea is that taking α to be $\top$ gives the center of contraction, and taking α to be $(i = \mathbb{0}_{\mathbb{I}}) \vee (i = \mathbb{1}_{\mathbb{I}})$ can be used to create a path between any two elements.

Substituting the partial-extends-to-total definition of contractibility into the definition of covariance, and rearranging some quantifiers, we obtain a cubical filling operation for morphism variables:

Definition 3.3. A *covariant filling structure* on $A : \Gamma \to \mathsf{Type}$ is a term of type $\mathsf{relCov}_\Gamma\ A$:

$$\begin{aligned}\mathsf{hasCov}\ (A : \mathbb{2} \to \mathsf{Type}) := \ & \Pi\alpha{:}\Omega_{\mathsf{cof}}.\\ & \Pi t{:}(\Pi z{:}\mathbb{2}.[\![\alpha]\!]_{\mathsf{cof}} \to A\ z).\\ & \Pi b{:}(A\ \mathbb{0}_{\mathbb{2}})[\alpha \mapsto t\ \mathbb{0}_{\mathbb{2}}].\\ & \Pi i{:}\mathbb{2}.(A\ i)[\alpha \mapsto t\ i, (\mathbb{0}_{\mathbb{2}} = i) \mapsto b]\\ \mathsf{relCov}_\Gamma\ (A : \Gamma \to \mathsf{Type}) := \ & \Pi p{:}\mathbb{2} \to \Gamma.\mathsf{hasCov}(A \circ p)\end{aligned}$$

In the formalization, we show

Theorem 3.4 (`Covariant.rcov-hprop, Covariant.relCov-internal-equiv, Covariant-is-Fibrant.com-hasCov`). $\mathsf{relCov}_\Gamma A$ *is a homotopy proposition, Kan, and logically equivalent (and therefore equivalent) to being covariant (Definition 3.2).*

An advantage of this formulation is that many types can be shown to be covariant by a nearly identical construction to their Kan composition structure. For example, because $\mathbb{2}$ has connections, we can apply the filling-from-composition lemma [11], which states that to construct the above, it suffices to define

$$\begin{aligned}\mathsf{hasCov}_{\mathbb{1}_{\mathbb{2}}}\ (A) := \quad & \Pi\alpha{:}\Omega_{\mathsf{cof}}.\\ & \Pi t{:}(\Pi z{:}\mathbb{2}.[\![\alpha]\!]_{\mathsf{cof}} \to A\ z).\\ & \Pi b{:}(A\ \mathbb{0}_{\mathbb{2}})[\alpha \mapsto t\ \mathbb{0}_{\mathbb{2}}].\\ & (A\ \mathbb{1}_{\mathbb{2}})[\alpha \mapsto t\ \mathbb{1}_{\mathbb{2}}]\end{aligned}$$

This is exactly CCHM Kan composition [11] but stated for the directed interval, rather than the path interval—thus, we can prove that the covariant types are closed under certain type constructors by the same argument used to show that they are Kan in [11].

In the formalization (`Segal.agda, Discrete.agda`), we also give analogous reformulations for [32]'s definitions of discrete and Segal types.

3.3 The Universe $\mathsf{U}_{\mathsf{cov}}$

We construct the universe $\mathsf{U}_{\mathsf{cov}}$ following [23]: we work within Agda using an idempotent comonadic modality implemented by Andrea Vezzosi[4] to formalize constructions involving the universe. We write $\mathsf{U}_{\mathsf{Kan}}$ for the universe of Kan types constructed in [2]. We build the universe $\mathsf{U}_{\mathsf{cov}}$ of

[4] This modality has been integrated into Agda as of version 2.6.1.

those Kan types that are covariant discrete fibrations (relCov) using [23, Theorem 5.2]. The assumptions of this are (1) that the interval 2 used in the definition of hasCov is *tiny*, i.e. exponentiation by the interval has a right adjoint (which we assumed in Figure 1); (2) that $\bot$ is a cofibration (an assumption from [2]); and (3) that relCov is itself Kan (Theorem 3.4); it would be immediate if we had used the original [32] definition, in terms of Π, Σ, path and hom types, all of which are Kan). Thus, we can construct a type $\mathsf{U}_{\mathsf{cov}}$, which has decoding function $\mathsf{El} : \mathsf{U}_{\mathsf{cov}} \to \mathsf{U}_{\mathsf{Kan}}$, along with an introduction rule saying that a "crisp" [23] (closed) term of type $\Gamma \to \mathsf{U}_{\mathsf{Kan}}$ has a code in the universe for every covariant discrete fibration structure on it. Moreover, $\mathsf{El} \circ A$ is covariant for any $A : \Gamma \to \mathsf{U}_{\mathsf{cov}}$—any type family in $\mathsf{U}_{\mathsf{cov}}$ has a transport function lifting directed morphisms in its domain to morphisms in $\mathsf{U}_{\mathsf{cov}}$—with the definition of $\mathsf{relCov}(\mathsf{El} \circ A)$ determined by the data in a code. Note that the modal restrictions on universes concern its introduction rule, but the type $\mathsf{U}_{\mathsf{cov}}$ and its elimination forms are unrestricted, and the statement of directed univalence below has no modal restrictions.

We have checked that this definition of $\mathsf{U}_{\mathsf{cov}}$ is useful by proving it is closed under some example type constructors:

Theorem 3.5 (`universe.{Sigma,Pi,Path,Hom}.agda`)**.** *There are codes*

- $\Sigma_{cov} : \Pi(A : \mathsf{U}_{\mathsf{cov}}).\Pi(B : \mathsf{El}A \to \mathsf{U}_{\mathsf{cov}}).\mathsf{U}_{\mathsf{cov}}$
- $\Pi_{cov} : \Pi(A :^\flat \mathsf{U}_{\mathsf{Kan}}).\Pi(B : A \to \mathsf{U}_{\mathsf{cov}}).\mathsf{U}_{\mathsf{cov}}$
- $\mathsf{path}_{cov} : \Pi(A : \mathsf{U}_{\mathsf{cov}}).\mathsf{El}A \to \mathsf{El}A \to \mathsf{U}_{\mathsf{cov}}$
- $\mathsf{hom}_{cov} : \Pi(A : \mathsf{U}_{\mathsf{cov}}).\mathsf{El}A \to \mathsf{El}A \to \mathsf{U}_{\mathsf{cov}}$

that decode to Σ, Π, Path *and* Hom *types respectively.*

This shows that $\mathsf{U}_{\mathsf{cov}}$ is closed under Σ types, Π types *with a constant domain* (because the domain is a contra- not covariant position; this is enforced by the modal restriction to a crisp A), Path_A, and Hom_A types. While in general $\mathsf{Hom}_A(-, a)$ should be contravariant rather than covariant, if A is in $\mathsf{U}_{\mathsf{cov}}$, then it is discrete, so its morphisms are equivalent to paths and thus invertible. The full proof is in the above files in the formalization; because of our reformulation of covariance as analogous to CCHM Kan filling, the proofs for Σ, Path and Hom are analogous to the arguments that Σ and Path are Kan in [11], while the argument for Π with a constant domain is analogous to the argument that Π has homogeneous composition (which does not use Kan filling in the domain).

3.4 $\mathsf{U}_{\mathsf{cov}}$ is path-univalent

The universe $\mathsf{U}_{\mathsf{cov}}$ should enjoy two kinds of univalence: First, like universes in ordinary homotopy type theory, a path in $\mathsf{U}_{\mathsf{cov}}$ should be an equivalence. Second, a directed morphism should be a function. We write $[\![-]\!]_{\mathsf{cov}}$ for the composite of the two El functions $\mathsf{U}_{\mathsf{cov}} \to \mathsf{U}_{\mathsf{Kan}} \to \mathsf{Type}$. For both kinds of univalence, we use glue types [11], which are formed from a cofibration α, a partial type $T : \alpha \to \mathsf{Type}$, a type $B : \mathsf{Type}$, and a partial function $f : \Pi p : \alpha.T\ p \to B$. The type $\mathsf{Glue}[\alpha \mapsto (T, f)]B$ is isomorphic to $\Sigma t : ([\![\alpha]\!]_{\mathsf{cof}} \to T).B[\alpha \mapsto f\ t]$, but when α is satisfied, $\mathsf{Glue}[\alpha \mapsto (T, f)]B$ is strictly equal to T.

We consider path-univalence first, as its proof is straightforward from previous work. The main lemma is that $\mathsf{U}_{\mathsf{cov}}$ is closed under glue types:

Lemma 3.6 (`universe.Glue-Equiv-Covariant.GlueUCov`)**.** *Given any* $\alpha : \Omega_{cof}, T : [\![\alpha]\!]_{\mathsf{cof}} \to \mathsf{U}_{\mathsf{cov}}$, $B : \mathsf{U}_{\mathsf{cov}}$, *and* $f : [\![\alpha]\!]_{\mathsf{cof}} \to [\![T]\!]_{\mathsf{cov}} \to [\![B]\!]_{\mathsf{cov}}$ *where* f *is an equivalence, the type* $\mathsf{Glue}[\alpha \mapsto ([\![T]\!]_{\mathsf{cov}}, f)][\![B]\!]_{\mathsf{cov}}$ *is in* $\mathsf{U}_{\mathsf{cov}}$.

Because of the above reformulation (relCov) of being covariant as CCHM [11] Kan filling, but for the directed interval, we can use exactly the CCHM algorithm for composition in glue types to prove this. Glue types have univalence as the special case of gluing with an equivalence on one side: $\mathsf{Glue}[x = \mathbb{0}_{\mathbb{I}} \mapsto (A, e), (x = 1) \mapsto (B, \mathsf{id}_B)](b)$.

Theorem 3.7 (`DirUnivalence.ua`)**.** *For any two types* A *and* B *in* $\mathsf{U}_{\mathsf{cov}}$*, we have the following equivalence.*

$$\mathsf{ua} : \mathsf{Equiv}\ (\mathsf{Equiv}\ [\![A]\!]_{\mathsf{cov}}\ [\![B]\!]_{\mathsf{cov}})\ (\mathsf{Path}_{\mathsf{U}_{\mathsf{cov}}}\ (A, B))$$

Glue types also imply that $\mathsf{U}_{\mathsf{cov}}$ is itself Kan—given some paths to compose in $\mathsf{U}_{\mathsf{cov}}$, we can project paths in $\mathsf{U}_{\mathsf{Kan}}$, which determine equivalences, which can then be composed using the glue type, which is covariant by above.

Theorem 3.8 (`universe.UCov-Com.UCovU`)**.** $\mathsf{U}_{\mathsf{cov}}$ *is Kan.*

The proof is analogous to the proof that $\mathsf{U}_{\mathsf{Kan}}$ is Kan in [2, 11].

4 The Directed Univalence Retraction

Next, we define the directed univalence reflection of morphisms $\mathsf{Hom}_{\mathsf{U}_{\mathsf{cov}}}(A, B)$ onto functions $[\![A]\!]_{\mathsf{cov}} \to [\![B]\!]_{\mathsf{cov}}$.

4.1 Morphisms to Functions.

Given a morphism in $\mathsf{U}_{\mathsf{cov}}$, part of the definition of $\mathsf{U}_{\mathsf{cov}}$ is a covariant filling structure for $\mathsf{El}_{\mathsf{cov}} : \mathsf{U}_{\mathsf{cov}} \to \mathsf{Type}$, which yields a coercion function along from $\mathbb{0}_2$ to $\mathbb{1}_2$:

$$\begin{aligned}&\mathsf{dcoe} : \Pi A\ B : \mathsf{U}_{\mathsf{cov}}.\mathsf{Hom}_{\mathsf{U}_{\mathsf{cov}}}(A, B) \to ([\![A]\!]_{\mathsf{cov}} \to [\![B]\!]_{\mathsf{cov}})\\ &\mathsf{dcoe}\ A\ B\ p := \lambda x : \mathsf{cov}_{\mathsf{El}}\ p\ \bot\ \mathsf{exfalso}\ x\ \mathbb{1}_2\end{aligned}$$

4.2 Functions to Morphisms.

We turn a function into a morphism using the same glue type used for undirected univalence, but this time without requiring that the function f be an equivalence:

$$\begin{aligned}&\mathsf{duahom}' : \Pi A\ B : \mathsf{Type}.(A \to B) \to \mathsf{Hom}_{\mathsf{Type}}(A, B)\\ &\mathsf{duahom}'\ A\ B\ f := \lambda i : 2.\mathsf{Glue}[i = \mathbb{0}_2 \mapsto (A, f)\\ &\qquad\qquad , i = \mathbb{1}_2 \mapsto (B, \mathsf{id}_B)]B\end{aligned}$$

In presenting the definition of duahom′, we omit the boundary proofs for the sake of clarity.

For this to be a morphism in $\mathsf{U}_{\mathsf{cov}}$, we need to show that it is Kan and covariant. In general, glue types with an arbitrary function (as opposed to an equivalence) are not Kan, but

the fact that the cofibration restricts a *directed* interval i gives some additional leverage that is not present in ordinary cubical models.

Lemma 4.1 (`universe.FunGlue.FunGlueUKan`). *For every pair of types $A\ B : \mathsf{U}_{\mathsf{Kan}}$, function $f : [\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}$, and $i : \mathbb{2}$, the type* $\mathsf{duahom}'\ A\ B\ f\ i$ *determines a code in* $\mathsf{U}_{\mathsf{Kan}}$.

Proof. We need to show that duahom' is a Kan fibration. Thus, for any path p in $\mathbb{2} \to (\Sigma A : \mathsf{U}_{\mathsf{Kan}}.\Sigma B : \mathsf{U}_{\mathsf{Kan}}.A \to B)$, we construct a solution to the hasCom filling problem for the path in Type induced by the composition of p and duahom'. For the sake of clarity, we will only present the proof that this path has a coercion structure (i.e. hasCom from $\mathbb{0}_{\mathbb{I}}$ to $\mathbb{1}_{\mathbb{I}}$ over the false cofibration proposition $\bot$), but the full proof is available in the Agda formalization.

Let p_i be the path in $\mathbb{2}$ induced by p, p_A and p_B be the two paths in $\mathsf{U}_{\mathsf{Kan}}$ induced by p, and lastly p_f be the dependent path over $\lambda j.[\![p_A\ j]\!]_{\mathsf{Kan}} \to [\![p_B\ j]\!]_{\mathsf{Kan}}$ contained in p. Now, given any term

$$g_0 : \mathsf{duahom}'\ [\![p_A\ \mathbb{0}_{\mathbb{I}}]\!]_{\mathsf{Kan}}\ [\![p_B\ \mathbb{0}_{\mathbb{I}}]\!]_{\mathsf{Kan}}\ (p_f\ \mathbb{0}_{\mathbb{I}})\ (p_i\ \mathbb{0}_{\mathbb{I}})$$

we must construct a term in

$$\mathsf{duahom}'\ [\![p_A\ \mathbb{1}_{\mathbb{I}}]\!]_{\mathsf{Kan}}\ [\![p_B\ \mathbb{1}_{\mathbb{I}}]\!]_{\mathsf{Kan}}\ (p_f\ \mathbb{1}_{\mathbb{I}})\ (p_i\ \mathbb{1}_{\mathbb{I}})$$

Note that, for any A, B, f, and i, a term in $\mathsf{duahom}'\ A\ B\ f\ i$ is just the pair of a term $a : A$ under the assumption $i = \mathbb{0}_{\mathbb{2}}$, and a term $b : B$ where $f\ a$ is strictly equal to b when $i = \mathbb{0}_{\mathbb{2}}$. Thus, let a_0 and b_0 be the components defining g_0. Let a_1 and b_1 denote the corresponding components we must construct to define a term in $\mathsf{duahom}'\ [\![p_A\ \mathbb{1}_{\mathbb{I}}]\!]_{\mathsf{Kan}}\ [\![p_B\ \mathbb{1}_{\mathbb{I}}]\!]_{\mathsf{Kan}}\ (p_f\ \mathbb{1}_{\mathbb{I}})\ (p_i\ \mathbb{1}_{\mathbb{I}})$.

As p_A is a path in $\mathsf{U}_{\mathsf{Kan}}$ and thus is Kan fibration, we can define a_1 by coercing a_0 along p_A, which we will denote as

$$a_1 := \mathsf{com}^{\mathbb{0}_{\mathbb{I}} \mapsto \mathbb{1}_{\mathbb{I}}} p_A\ a_0$$

We must be able to define a_1 whenever $p_i\ \mathbb{1}_{\mathbb{I}} = \mathbb{0}_{\mathbb{2}}$, but it depends on a_0 which only exists when $p_i\ \mathbb{0}_{\mathbb{I}} = \mathbb{0}_{\mathbb{2}}$. Thus, for this to work, we require the triviality axiom ($\mathbb{I}$to$\mathbb{2}$triv) that states any function $\mathbb{I} \to \mathbb{2}$ is trivial. In our setting, this means that if $p_i\ \mathbb{1}_{\mathbb{I}} = \mathbb{0}_{\mathbb{2}}$ we know that for every $j : \mathbb{I}$ it must be the case that $p_i\ j = \mathbb{0}_{\mathbb{2}}$, and therefore $p_i\ \mathbb{0}_{\mathbb{I}} = \mathbb{0}_{\mathbb{2}}$ confirming a_0 exists whenever we need to define a_1.

Next, to define b_1, we analogously fill along p_B, but to ensure it is strictly equal to $f\ \mathbb{1}_{\mathbb{I}}\ a_1$ when $p_i\ \mathbb{1}_{\mathbb{I}} = \mathbb{0}_{\mathbb{2}}$, we must fill along the cofibration proposition $p_i\ \mathbb{1}_{\mathbb{I}} = \mathbb{0}_{\mathbb{2}}$.

$$b_1 := \mathsf{com}^{\mathbb{0}_{\mathbb{I}} \mapsto \mathbb{1}_{\mathbb{I}}} p_B [p_i\ \mathbb{1}_{\mathbb{I}} = \mathbb{0}_{\mathbb{2}} \mapsto \lambda j.p_f\ j\ (\mathsf{com}^{\mathbb{0}_{\mathbb{I}} \mapsto j} p_A\ a_0)]\ b_0$$

Again, b_1 also depends on $\mathbb{I}$to$\mathbb{2}$triv in order to define the boundary constraint.

The construction for a general composition problem in our formalization is nearly identical to that of the coercion described above. As duahom' is Kan, for every pair of types $A\ B : \mathsf{U}_{\mathsf{Kan}}$, function $f : [\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}$, and $i : \mathbb{2}$, the type $\mathsf{duahom}'\ A\ B\ f\ i$ determines a code in $\mathsf{U}_{\mathsf{Kan}}$. □

Lemma 4.2 (`universe.FunGlue.FunGlueUCov`). *For every pair of types $A\ B : \mathsf{U}_{\mathsf{cov}}$, function $f : [\![A]\!]_{\mathsf{cov}} \to [\![B]\!]_{\mathsf{cov}}$, and $i : \mathbb{2}$, the type* $\mathsf{duahom}'\ A\ B\ f\ i$ *determines a code in* $\mathsf{U}_{\mathsf{cov}}$.

Proof. As we already know that duahom' is Kan, this amounts to showing it is covariant. The construction of the solution of the covariant filling problem is nearly identical to that of the Kan filling problem. The sole difference is that we use covariant filling in A and B instead of Kan filling, and we have a different justification for why a_0 always exists whenever we need to be able to define a_1.

Let p_i be the morphism in $\mathbb{2}$ induced by the morphism we are considering in the filling problem. We must be able to define a_1 whenever $p_i\ \mathbb{1}_{\mathbb{2}} = \mathbb{0}_{\mathbb{2}}$ using a_0 which only exists when $p_i\ \mathbb{0}_{\mathbb{2}} = \mathbb{0}_{\mathbb{2}}$. In this case, we use the monotonicity axiom ($\mathbb{2}$mono): if $p_i\ \mathbb{1}_{\mathbb{2}} = \mathbb{0}_{\mathbb{2}}$, we know that for every $j : \mathbb{2}$ it also holds that $p_i\ j = \mathbb{0}_{\mathbb{2}}$, and thus we always have a_0 when it is needed to define a_1.

□

Since duahom' is both Kan and covariant, it constructs a morphism in $\mathsf{U}_{\mathsf{cov}}$:

$$\mathsf{duahom} : \Pi A\ B : \mathsf{U}_{\mathsf{cov}}.([\![A]\!]_{\mathsf{cov}} \to [\![B]\!]_{\mathsf{cov}}) \to \mathsf{Hom}_{\mathsf{U}_{\mathsf{cov}}}\ (A, B)$$

4.3 Reflection

A calculation with the definition of Lemma 4.2 shows that

Lemma 4.3 (`DirUnivalence.dua`β). *For every pair types $A\ B : \mathsf{U}_{\mathsf{cov}}$ and every function $f : [\![A]\!]_{\mathsf{cov}} \to [\![B]\!]_{\mathsf{cov}}$, we can construct a path*

$$\mathsf{dua}\beta : \mathsf{Path}_{[\![A]\!]_{\mathsf{cov}} \to [\![B]\!]_{\mathsf{cov}}}\ (f, \mathsf{dcoe}\ A\ B\ (\mathsf{duahom}\ A\ B\ f))$$

Proof. To define the path, we construct the following function of type $\mathbb{I} \to A \to B$:

$$\begin{aligned}&\lambda i\ a.\mathsf{cov}^{\mathbb{0}_{\mathbb{2}} \mapsto \mathbb{1}_{\mathbb{2}}}(\mathsf{duahom}\ A\ B\ f)\\ &\quad [i = \mathbb{0}_{\mathbb{I}} \mapsto \lambda j.\mathsf{glue}[j = \mathbb{0}_{\mathbb{2}} \mapsto a, j = \mathbb{1}_{\mathbb{2}} \mapsto f\ a] f\ a\\ &\quad , i = \mathbb{1}_{\mathbb{I}} \mapsto \lambda j.\mathsf{cov}^{\mathbb{0}_{\mathbb{2}} \mapsto j}(\mathsf{duahom}\ A\ B\ f)\ a]\ a\end{aligned}$$

As the function has the correct boundaries at $i = \mathbb{0}_{\mathbb{I}}$ and $i = \mathbb{1}_{\mathbb{I}}$, it is the desired path. □

However, for the other composite, we obtain only a morphism:

Lemma 4.4 (`DirUnivalence.dua`η`fun`). *For every $A\ B : \mathsf{U}_{\mathsf{cov}}$, $p : \mathsf{Hom}_{\mathsf{U}_{\mathsf{cov}}}\ (A, B)$, and $i : \mathbb{2}$, we can construct a function (and therefore a* $\mathsf{Hom}_{\mathsf{U}_{\mathsf{cov}}}$ *by Lemma 4.2)*

$$\mathsf{dua}\eta\mathsf{fun}\ i : (p\ i) \to (\mathsf{duahom}\ A\ B\ (\mathsf{dcoe}\ A\ B\ p)\ i)$$

Proof. The idea is that, as $\mathsf{duahom}\ A\ B\ (\mathsf{dcoe}\ A\ B\ p)\ i$ is a glue type with base B (i.e. $p\ \mathbb{1}_{\mathbb{2}}$), we can take any term in $p\ i$ and coerce it to a term in $p\ \mathbb{1}_{\mathbb{2}}$ to define our function. We construct the function as $\lambda i\ x.\mathsf{cov}^{i \mapsto \mathbb{1}_{\mathbb{2}}} p\ x$. Note that we used a different covariant filler that goes from i to $\mathbb{1}_{\mathbb{2}}$; this follows from our standard covariant filler that goes from $\mathbb{0}_{\mathbb{2}}$ to i using the connections on the directed interval. □

5 Cobar Construction and Bicubical Stacks

Next, we will restrict $\mathsf{U}_{\mathsf{cov}}$ to a subuniverse for which we can show that duaηfun is an equivalence (we do not have countermodel showing definitively that duaηfun is not an

$$
\begin{array}{rcl}
\mathsf{D} & : & \mathsf{U}_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}} \\
\eta & : & \{A : \mathsf{U}_{\mathsf{Kan}}\} \to [\![A]\!]_{\mathsf{Kan}} \to [\![\mathsf{D}A]\!]_{\mathsf{Kan}} \\
\mathsf{L} & : & [\![\mathsf{D}(\mathsf{U}_{\mathsf{Kan}})]\!]_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}} \\
\tilde{\mathsf{D}} & : & \{A : \mathsf{U}_{\mathsf{Kan}}\} \to ([\![A]\!]_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}}) \\
 & & \to [\![\mathsf{D}A]\!]_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}} \\
\tilde{\mathsf{D}}\ B & := & \mathsf{L} \circ \mathsf{Dfun}(B) \\
\mathsf{isStack} & : & \mathsf{U}_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}} \\
\mathsf{isStack}\ A & := & \mathsf{isEquiv}\ \eta_A \\
\mathsf{Dfun} & : & \{A\ B : \mathsf{U}_{\mathsf{Kan}}\} \to ([\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}) \\
 & & \to [\![\mathsf{D}A]\!]_{\mathsf{Kan}} \to [\![\mathsf{D}B]\!]_{\mathsf{Kan}} \\
\mathsf{Dcomp} & : & \{A\ B\ C : \mathsf{U}_{\mathsf{Kan}}\} \to (g : [\![B]\!]_{\mathsf{Kan}} \to [\![C]\!]_{\mathsf{Kan}}) \\
 & & \to (f : [\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}) \\
 & & \to \mathsf{Dfun}(g \circ f) = \mathsf{Dfun}(g) \circ \mathsf{Dfun}(f) \\
\mathsf{Did} & : & (A : \mathsf{U}_{\mathsf{Kan}}) \\
 & & \to \mathsf{Dfun}(\lambda a : [\![A]\!]_{\mathsf{Kan}}.a) = \lambda a : [\![\mathsf{D}A]\!]_{\mathsf{Kan}}.a \\
\eta\mathsf{nat} & : & \{A\ B : \mathsf{U}_{\mathsf{Kan}}\} \to (f : [\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}) \\
 & & \to \eta_B \circ f = \eta_A \circ \mathsf{Dfun}(f) \\
\eta\mathsf{path} & : & (A : \mathsf{U}_{\mathsf{Kan}}) \\
 & & \to \mathsf{Path}_{[\![\mathsf{D}A]\!]_{\mathsf{Kan}} \to [\![\mathsf{D}^2A]\!]_{\mathsf{Kan}}} (\mathsf{Dfun}(\eta_A),\ \eta_{\mathsf{D}A}) \\
\mathsf{Leq} & : & \mathsf{L} \circ \eta_{\mathsf{U}_{\mathsf{Kan}}} = \mathsf{D} \\
\mathsf{D}\Sigma\mathsf{snd} & : & \{A : \mathsf{U}_{\mathsf{Kan}}\} \to \{B : [\![A]\!]_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}}\} \\
 & & \to \Pi p : [\![\mathsf{D}(\Sigma a : A.B\ a)]\!]_{\mathsf{Kan}}.[\![\tilde{\mathsf{D}}B\ (\mathsf{Dfun}\ \mathsf{fst}\ a)]\!]_{\mathsf{Kan}} \\
\mathsf{D}\Sigma\mathsf{eq} & : & \{A : \mathsf{U}_{\mathsf{Kan}}\} \to (B : [\![A]\!]_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}}) \\
 & & \to \mathsf{isIso}\ (\lambda x.(\mathsf{Dfun}\ \mathsf{fst}\ x, \mathsf{D}\Sigma\mathsf{snd}\ x))
\end{array}
$$

Figure 2. Axioms and Definitions for Descent Operator

equivalence for $\mathsf{U}_{\mathsf{cov}}$, but we have not been able to prove it). We do this using a notion of stack (but for the trivial topology), which picks out those types that are equivalent to their *cobar construction*. The definitions and many of the theorems in this section follow Coquand and Ruch [13]; we also check that this work applies to our setting and prove a few extra lemmas needed for our use of theses ideas. Coquand and Ruch abstract many of the definitions and lemmas to the level of an arbitrary left-exact (lex) operator on types, and we have formalized this part of the construction in Agda. The parts that are cobar-specific are postulated as axioms.

We axiomatize the cobar operator as is shown in Figure 2. The main idea is that we add a new endofunctor D on fibrant types which comes equipped with a natural transformation $\eta : \mathrm{id}_{\mathsf{U}_{\mathsf{Kan}}} \to D$. Using this, we make a restricted version of our covariant universe containing those covariant types A for which $\eta_A : A \to \mathsf{D}A$ is an equivalence, and call these types stacks. We show that all of the basic type formers preserve the property of being a stack. With the added structure of this equivalence for every type, we are able to show that a function is an equivalence of bicubical stacks if at every object in $\square_{\mathsf{Ded}}$ it is an equivalence of cubical sets, from which we can complete the construction of directed univalence for the universe of covariant stacks.

5.1 Universes and Closure Properties

First, we show that the stacks are closed under the following:

Lemma 5.1 (`Descent.Pi.ΠisStack`)**.** *Given a Kan type A and a Kan fibration $B : [\![A]\!]_{\mathsf{Kan}} \to \mathsf{U}_{\mathsf{Kan}}$, if the fibers of B are stacks then $\Pi a : A.B\ a$ is a stack.*

Lemma 5.2 (`Descent.Sigma.ΣisStack`, Proposition 1.1 in [13])**.** *If a Kan type A is a stack and the fibers of a Kan fibration $B : A \to \mathsf{U}_{\mathsf{Kan}}$ are stacks, then $\Sigma a : A.B\ a$ is a stack.*

We define the universe of covariant stacks, $\mathsf{U}_{\mathsf{covstack}}$, as $\Sigma X : \mathsf{U}_{\mathsf{cov}}.\mathsf{isStack}\ X$ (`UCovStack.UCovStack`).

Proposition 5.3. $\mathsf{U}_{\mathsf{covstack}}$ *is logically non-degenerate, in the sense that it contains an empty type* Void.

Proof. First, Void is discrete, because being discrete means that for all $x, y :$ Void, $\mathsf{Hom}_{\mathsf{Void}}(x, y) \simeq \mathsf{Path}_{\mathsf{Void}}(x, y)$ (by a particular map), which is vacuously true. Thus, $\lambda\_$.Void is a constant family of discrete types, and any such family has a code in $\mathsf{U}_{\mathsf{cov}}$. Finally, Void is a stack as well: in the specific case of cobar, D(Void) ≅ Void, which implies the inverse to η_{Void}. Thus, Void has a code in $\mathsf{U}_{\mathsf{covstack}}$. □

5.2 The Cobar Construction

Next, we work externally (so this part has not been formalized) and check that axioms in Figure 2 are true. [13] defines the cobar operator by first defining a type operator E.

Definition 5.4. Given a Kan type A in a context Γ, the type $\mathsf{E}A$ in context Γ has terms as follows: For every X in $\square_{\mathsf{Ded}}$ and ρ in $\Gamma(X)$, and element u in $\mathsf{E}A\rho(X)$ is given by a family of elements $u(f)$ of $A(\rho f)(Y)$ indexed by morphisms $f \in \square_{\mathsf{Ded}}(Y, X)$. It is equipped with a natural transformation $\alpha_A : A \to \mathsf{E}A$ (fibered over each element ρ in Γ) sending a in $A\rho(X)$ to the family $f \mapsto af$.

In particular, notice that for X in $\square_{\mathsf{Ded}}$, an element of $\mathsf{E}^n A(X)$ is a function that takes as input an n-chain of composable morphisms with codomain X and returns an element of $A(Y)$ where Y is the domain of the chain.

In the definition of the cobar operator, we will need the following: Given a list $(i_1, \ldots, i_n) : \mathbb{I}^n$, the formula $\delta(i_1, \ldots, i_n)$ is the $(n+1)$-ary disjunction

$$(\mathbb{0}_{\mathbb{I}} = i_1) \vee (i_1 = i_2) \vee \ldots \vee (i_{n-1} = i_n) \vee (i_n = \mathbb{1}_{\mathbb{I}})$$

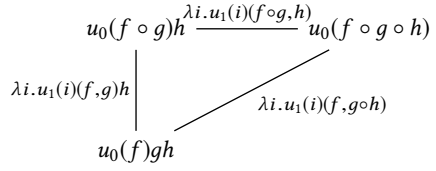
We then define a partial term $\mathrm{diag} : (\vec{i} : \mathbb{I}^n) \to \delta(\vec{i}) \to \mathbb{I}^{n-1}$ by omitting i_1 on $\mathbb{0}_{\mathbb{I}} = i_1$, i_k on $i_k = i_{k+1}$, and i_n on $i_n = \mathbb{1}_{\mathbb{I}}$.

Definition 5.5. Given a Kan type A in a context Γ, the type $\mathsf{D}A$ in context Γ has terms as follows: For every X in $\square_{\mathsf{Ded}}$ and ρ in $\Gamma(X)$, and element u in $\mathsf{D}A\rho(X)$ is given by a family of elements $u(i_1, \ldots, i_n)$ of $\mathsf{E}^{n+1}A$ indexed by lists of interval variables $(i_1, \ldots, i_n) : \mathbb{I}^n$ for every natural number n. These families must also satisfy the following conditions:

$$
\begin{array}{ll}
u = \alpha \circ u \circ \mathrm{diag} & \text{when } \mathbb{0}_{\mathbb{I}} = i_1 \\
u = \mathsf{E}^k(\alpha) \circ u \circ \mathrm{diag} & \text{when } i_k = i_{k+1} \\
u = \mathsf{E}^n(\alpha) \circ u \circ \mathrm{diag} & \text{when } i_n = \mathbb{1}_{\mathbb{I}}
\end{array}
$$

We define the natural transformation $\eta_A : A \to \mathrm{D}A$ such that $(\eta_A a)(i_1, \ldots, i_n) = \alpha^{n+1} a$.

While the details are rather technical, the main idea is that for every Dedekind cube X a term in $\mathrm{D}A(X)$ is a family u_n indexed by $n : \mathbb{N}$, and each u_n is a function that takes as input an $n+1$-length chain of morphisms in $\square_{\mathrm{Ded}}$ with codomain X, and returns an n-cube in $\mathrm{D}A(Y)$ where Y is the domain of the chain. These families are constrained so that some of the boundaries of the $n+1$-cubes given by u_{n+1} are the various n-cubes given by u_n evaluated at all of the $n+1$-chains given by replacing an adjacent pair in the $n+2$-chain with their composition, along with the n-cube given by using the first morphism in the chain to substitute the n-cube given by evaluating u_n at the tail of the original chain. As a simple example, consider $f : \square_{\mathrm{Ded}}(X, W)$, $g : \square_{\mathrm{Ded}}(Y, X)$, and $h : \square_{\mathrm{Ded}}(Z, Y)$. The square $\lambda i\, j.u_2(i, j)(f, g, h)$ will have $\lambda i.u_1(i)(f, g)h$ on one side followed by $\lambda i.u_1(i)(f \circ g, h)$ on another, and $\lambda i.u_1(i)(f, g \circ h)$ is along the diagonal. $\lambda i.u_1(i)(f, g)$ is a path starting at $u_0(f)g$ and ending at $u_0(f \circ g)$, $\lambda i.u_1(i)(f \circ g, h)$ is a path from $u_0(f \circ g)h$ to $u_0(f \circ g \circ h)$, and $\lambda i.u_1(i)(f, g \circ h)$ is a path from $u_0(f)gh$ to $u_0(f \circ g \circ h)$, as depicted below:



In particular, note that the conditions determine the boundaries of an n-simplex inside of every n-cube.

Proposition 5.6. *For any Kan type A, there is a path in $\mathrm{D}A \to \mathrm{D}^2 A$ between $\eta_{\mathrm{D}A}$ and $\mathrm{Dfun}\ \eta_A$.*

Proof. As is described in Proposition 4.3 of [13], an element of $\mathrm{D}^2 A$ is of the form $v_{m,n}(i_1, \ldots, i_m)(j_1, \ldots, j_n)$ satisfying a number of equations. For u in $\mathrm{D}A$, a homotopy from $\eta_{\mathrm{D}A}(u)_{m,n}$ to $(\mathrm{Dfun}\ \eta_A)(u)_{m,n}$, in the presence of connections, is given by $\lambda k : \mathbb{I}.v^k_{m,n}$ where $v^k_{m,n}(\vec{i})(\vec{j}) := u_{m+n+1}(\vec{i} \wedge k, k, k \vee \vec{j})$.

Given our work is agnostic to the choice of cube category $\square_{\mathrm{path}}$ for paths, we would like a proof that does not depend on the presence of connections; however, unlike in [13], we restrict our functor D to only be over fibrant types, and thus we can use path induction to replace the use of connections in their proof. Given some term $a : \mathbb{I} \to A$ for a Kan type A, we can construct a term $a_\wedge : \Pi(i, j) : \mathbb{I}^2.A[i = \mathbb{0}_\mathbb{I} \mapsto a\ \mathbb{0}_\mathbb{I}, i = \mathbb{1}_\mathbb{I} \mapsto a\ j, j = \mathbb{0}_\mathbb{I} \mapsto a\ \mathbb{0}_\mathbb{I}, j = \mathbb{1}_\mathbb{I} \mapsto a\ i, i = j \mapsto a\ i]$ by path induction on a, as the extension type is fibrant. The square for the other connection is constructed similarly. A reference implementation of the connections inside of Cartesian cubical type theory is available in the standard library of redtt [1, `prelude.connection`]. Having defined these, we build a homotopy with the same boundaries as $v^k_{m,n}$ by induction on m and n, iteratively using these two constructions in place of the connections.

To construct $v^k_{m,n}$ where the length of $\vec{i}$ is m and that of $\vec{j}$ is n, we first do induction over an auxiliary number $z \le m$ to define a term w of type $\mathbb{I}^{z+m+n+1} \to \mathsf{E}^{m+n+2} A$. when $z = 0$, we simply define the term to be $w_0 := \lambda \vec{x} : \mathbb{I}^{m+n+1}.u_{m+n+1}(\vec{x})$. Now, assume we have defined $w_z : \mathbb{I}^{z+m+n+1} \to \mathsf{E}^{m+n+2} A$. To define the term $w_{z+1} : \mathbb{I}^{z+1+m+n+1} \to \mathsf{E}^{m+n+2} A$, we isolate and curry the variable x_{z+1} from $\vec{x}$ (where x_{z+1} will ultimately correspond to the $(z+1)$-th element in the m-length list $\vec{i}$), and then use the $\wedge$-connection construction on $\lambda.x_{z+1}.\lambda \vec{x} \setminus x_{z+1}.w_z(\vec{x})$, adding a new dimension variable for the connection at position x_{z+1} and resulting in w_{z+1} of the desired type. Given we can do this construction for any $z \le m$, we in particular can do it for m itself, resulting in a term $w : \mathbb{I}^{2m+n+1} \to E^{m+n+2} A$. Lastly, we repeat the above construction starting with w instead of u, using n instead of m, using the $\vee$-connection instead of $\wedge$, and working from the variables indexed at the end of $\vec{x}$ instead of the beginning to define a term $w' : \mathbb{I}^{2m+2n+1} \to E^{m+n+2} A$.

To finish, we abstract w' over a fresh dimension variable k and substitute back in $\vec{i}$, $\vec{j}$ and k in the correct positions of its arguments corresponding to the definition of $v^k_{m,n}$ that uses connections, resulting in a cube that has identical boundaries. Given this construction has the identical boundary as that built using connections, the family defined using this also satisfies the constraints required to be an element of $\mathrm{D}^2 A$, and is the path we set out to build. □

$$
\begin{array}{rcl}
\mathrm{Ddua} & : & \forall\{A\ B : \mathsf{U}_{\mathrm{Kan}}\}\{f : [\![A]\!]_{\mathrm{Kan}} \to [\![B]\!]_{\mathrm{Kan}}\}\{i : \mathbb{2}\} \\
 & & \to \mathrm{D}(\mathrm{duahom}'\ A\ B\ f\ i) \\
 & & \to \mathrm{duahom}'\ (\mathrm{D}A)\ (\mathrm{D}B)\ (\mathrm{Dfun}\ f)\ i \\
\mathrm{Ddua}\ g & := & \mathrm{glue}[i = \mathbb{0}_\mathbb{2} \mapsto g, i = \mathbb{1}_\mathbb{2} \mapsto \mathrm{Dfun\ unglue}\ g] \\
 & & (\mathrm{Dfun\ unglue}\ g) \\
 & & \\
\mathrm{Ddua\text{-}iso} & : & \mathrm{isIso}\ \_\ \_\ \mathrm{Ddua} \\
\mathrm{DPath\text{-}iso} & : & \{A : \mathsf{U}_{\mathrm{Kan}}\} \to (a_0\ a_1 : [\![A]\!]_{\mathrm{Kan}}) \\
 & & \to \mathrm{Iso}\ (\mathrm{DPath}_{\mathrm{D}A}\ \eta a_0\ \eta a_1)\ (\mathrm{D}(\mathrm{DPath}_A\ a_0\ a_1)) \\
\mathrm{DHom\text{-}iso} & : & \{A : \mathsf{U}_{\mathrm{Kan}}\} \to (a_0\ a_1 : [\![A]\!]_{\mathrm{Kan}}) \\
 & & \to \mathrm{Iso}\ (\mathrm{DHom}_{\mathrm{D}A}\ \eta a_0\ \eta a_1)\ (\mathrm{D}(\mathrm{DHom}_A\ a_0\ a_1)) \\
\mathrm{Path\text{-}}\eta\mathrm{\text{-}eq} & : & \forall\{A : \mathsf{U}_{\mathrm{Kan}}\}\{a_0\ a_1 : [\![A]\!]_{\mathrm{Kan}}\}(p : \mathrm{DPath}_A\ a_0\ a_1) \\
 & & \to (\mathrm{DPath\text{-}iso}\ a_0\ a_1) \circ \eta_A \circ p = \eta_{\mathrm{DPath}_A\ a_0\ a_1\ p} \\
\mathrm{Hom\text{-}}\eta\mathrm{\text{-}eq} & : & \forall\{A : \mathsf{U}_{\mathrm{Kan}}\}\{a_0\ a_1 : [\![A]\!]_{\mathrm{Kan}}\}(p : \mathrm{DHom}_A\ a_0\ a_1) \\
 & & \to (\mathrm{DHom\text{-}iso}\ a_0\ a_1) \circ \eta_A \circ p = \eta_{\mathrm{DHom}_A\ a_0\ a_1\ p}
\end{array}
$$

Figure 3. Additional Axioms and Definitions for Cobar Operator

Theorem 5.7 (Corollary 4.1 in [13])**.** *The cobar operator satisfies the axioms in Figure 2.*

Proof. The proof that cobar operator is a lex operator (and thus satisfies the axioms) can be found in [13]. The only proof that depends on connections is that corresponding to Proposition 5.6, which we have proven above. □

To complete the proofs that stacks are closed under path and morphism types, we add a few isomorphisms and equations for η to our model, shown in Figure 3.

Lemma 5.8. *For any representable* $\Psi = \mathbb{I}^n \times 2^m$ *and Kan fibration* $A : \Psi \to \mathsf{U}_{\mathsf{Kan}}$ *in context* Γ, $\mathsf{D}(\Pi x : \Psi.A\ x)$ *is isomorphic to* $\Pi x : \Psi.\mathsf{D}(A\ x)$.

Proof. As our representables are all give by products of $\mathbb{I}$ and 2, it is sufficient to just show it holds for each interval. First, consider $\mathsf{E}(\Pi i : \mathbb{I}.A\ i)$. For each $X \in \widehat{\square}_{\mathrm{Ded}}$, $\rho \in \Gamma(X)$, $u \in \mathsf{E}(\Pi i : \mathbb{I}.A\ i)\rho$ and $f \in \widehat{\square}_{\mathrm{Ded}}(Y, X)$, $u(f)$ is a term in $\Pi i : \mathbb{I}.(A\ i)\rho f$. We can reorder the arguments to get an element of $\Pi i : \mathbb{I}.\mathsf{E}(A\ i)$ by $(i, X, f) \mapsto u(X, f, i)$, which is clearly an isomorphism. Similarly, for $\mathsf{E}(\Pi i : 2.A\ i)$, for each $X \in \widehat{\square}_{\mathrm{Ded}}$, $u \in \mathsf{E}(\Pi i : 2.A\ i)\rho$ and $f \in \widehat{\square}_{\mathrm{Ded}}(Y, X)$, $u(f)$ is a term in $\Pi i : 2.(A\ i)\rho f$. This is equivalent to considering $u(f \times \mathrm{id}_2)$ a term in $A(Y \times 2)$, and thus we can also pull out the 2 to the outside with Yoneda and rearrange the arguments to get a term in $\Pi i : 2.\mathsf{E}(A\ i)$. Given how $\mathsf{D}A$ is constructed using $\mathsf{E}A$, we can iterate this isomorphism and commute arguments with the indexing lists of interval variables to get the isomorphisms between $\mathsf{D}(\Pi i : \mathbb{I}.A\ i)$ and $\Pi i : \mathbb{I}.\mathsf{D}(A\ i)$, along with between $\mathsf{D}(\Pi i : 2.A\ i)$ and $\Pi i : 2.\mathsf{D}(A\ i)$. As this is simply reordering the arguments to the families, the equations required by D still hold. □

Lemma 5.9. *For any representable* $\Psi = \mathbb{I}^n \times 2^m$, *Kan fibration* $A : \Psi \to \mathsf{U}_{\mathsf{Kan}}$ *in context* Γ, *cofibration* β *only depending on* Ψ, *and partial term* $a : \Pi x : \Psi.\beta \to A\ x$, *the type* $\mathsf{D}(\Pi x : \Psi.(A\ x)[\beta \mapsto a\ x])$ *is isomorphic to* $\Pi x : \Psi.(\mathsf{D}(A\ x))[\beta \mapsto \eta_{A\ x}\ a\ x]$.

Proof. Given how $\mathsf{D}(A\ x)$ is defined using E, we can unfold the definition and use the above isomorphism to see that $\mathsf{D}(\Pi x : \Psi.(A\ x)[\beta \mapsto a\ x])$ is a restriction of the type $\Pi x : \Psi.\mathsf{D}(A\ x)$ given by $\Pi x : \Psi.(\mathsf{D}(A\ x))[\beta \mapsto \eta_{A\ x}\ a\ x]$. □

Corollary 5.10. *The axioms DPath-iso, DHom-iso, Path-η-eq and Hom-η-eq from Figure 3 hold.*

Lemma 5.11 (`Descent.Path.Path0-isStack`)**.** *If the fibers of a Kan fibration* $A : \mathbb{I} \to \mathsf{U}_{\mathsf{Kan}}$ *are stacks, then for any terms* a_0 *in* $A\ \mathbb{0}_{\mathbb{I}}$ *and* a_1 *in* $A\ \mathbb{1}_{\mathbb{I}}$ *the type* $\mathsf{DPath}_A\ a_0\ a_1$ *is a stack.*

Proof. To construct the inverse to η, we use the isomorphism from Corollary 5.10 and construct an inverse to the function $\eta_A \circ - : \mathsf{DPath}_A\ a_0\ a_1 \to \mathsf{DPath}_{\mathsf{D}\circ A}\ (\eta_{A\ \mathbb{0}_{\mathbb{I}}}\ a_0)\ (\eta_{A\ \mathbb{1}_{\mathbb{I}}}\ a_1)$. Given a path p in $\mathsf{DPath}_{\mathsf{D}\circ A}\ (\eta_{A\ \mathbb{0}_{\mathbb{I}}}\ a_0)\ (\eta_{A\ \mathbb{1}_{\mathbb{I}}}\ a_1)$ and i in 2, the inverse just applies $\eta^{-1}_{A\ i}$ to $p\ i$, using the fact that $A\ i$ is a stack to compose with the path in $\mathsf{Path}_{A\ i}\ (\eta^{-1}_{A\ i}(\eta\ (p\ i)))\ (p\ i)$ when i equals $\mathbb{0}_{\mathbb{I}}$ or $\mathbb{1}_{\mathbb{I}}$. As $(\eta_{A\ i}, \eta^{-1}_{A\ i})$ form an adjoint equivalence for every i, it follows that this is also an adjoint equivalence and thus an equivalence. □

Analogously, we have

Lemma 5.12 (`Descent.Hom.Hom0-isStack`)**.** *If the fibers of a Kan fibration* $A : 2 \to \mathsf{U}_{\mathsf{Kan}}$ *are stacks, then for any terms* a_0 *in* $A\ \mathbb{0}_2$ *and* a_1 *in* $A\ \mathbb{1}_2$ *the type* $\mathsf{DHom}_A\ a_0\ a_1$ *is a stack.*

As with path and hom types, we add an isomorphism as an axiom in our model to facilitate the proof that stacks are closed under duahom′.

Lemma 5.13. *Consider Kan types* A *and* B, *function* $f : [\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}$ *and* $i : 2$. *The function* Ddua *defined in Figure 3 is an isomorphism.*

Proof. Observe that, for every object $\Psi = \mathbb{I}^m \times 2^n$, the type $\Psi \to \mathsf{duahom}'\ A\ B\ f\ i$ is isomorphic to $\mathsf{duahom}'\ (\Psi \to A)\ (\Psi \to B)\ (f \circ -)\ i$. Unfolding the definition of D, we see the action of Ddua is an application of this isomorphism, and thus is an isomorphism itself. □

Lemma 5.14 (`Descent.FunGlue.Glue-isStack`)**.** *Given Kan types* A *and* B, *a function* $f : [\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}$ *and a term* $x : 2$, *if* A *and* B *are stacks, then so is the type* $\mathsf{duahom}'\ A\ B\ f\ i$.

We use the following property below:

Theorem 5.15 (Theorem 5.1 in [13])**.** *Given stacks* A *and* B *in* $\mathsf{U}_{\mathsf{Kan}}$ *and a function* $f : [\![A]\!]_{\mathsf{Kan}} \to [\![B]\!]_{\mathsf{Kan}}$, *if for every* $X \in \widehat{\square}_{\mathrm{Ded}}$ $f(-, X)$ *is an equivalence of cubical sets, then* f *is an equivalence of bicubical sets.*

5.3 Completing Directed Univalence

Next, we show that $\mathsf{U}_{\mathsf{covstack}}$ satisfies directed univalence.

First, as $\mathsf{U}_{\mathsf{covstack}}$ is the restriction of $\mathsf{U}_{\mathsf{cov}}$ by a homotopy proposition isStack, the proof of the retraction in $\mathsf{U}_{\mathsf{cov}}$ lifts into $\mathsf{U}_{\mathsf{covstack}}$ (see `DirUnivalenceReflectionStack.agda`). To complete the equivalence, it suffices to show that duaηfun is an equivalence; then, by function extensionality and path-univalence for $\mathsf{U}_{\mathsf{covstack}}$, we get a path in $2 \to \mathsf{U}_{\mathsf{covstack}}$. To construct this, we will externally justify the following axiom:

Definition 5.16 (Covariant Equivalence Axiom)**.**

$$\begin{aligned}\mathsf{covEquivAx} : \ & \Pi p\ q{:}2 \to \mathsf{U}_{\mathsf{covstack}}.\\ & \Pi f{:}(\Pi i : 2.[\![p\ i]\!]_{\mathsf{covstack}} \to [\![q\ i]\!]_{\mathsf{covstack}}).\\ & \Pi e_0{:}\mathsf{isEquiv}\ (f\ \mathbb{0}_2).\Pi e_1{:}\mathsf{isEquiv}\ (f\ \mathbb{1}_2).\\ & \Pi i{:}2.\mathsf{isEquiv}\ (f\ i)\end{aligned}$$

which says that any map between covariant fibrations that is an equivalence at $\mathbb{0}_2$ and at $\mathbb{1}_2$ is an equivalence. Before justifying the axiom, we use it to complete the internal directed univalence equivalence.

Lemma 5.17 (`DirUnivalence.dua`η)**.** *For every pair of types* $A\ B : \mathsf{U}_{\mathsf{covstack}}$ *and every morphism* $p : \mathsf{Hom}_{\mathsf{U}_{\mathsf{covstack}}}\ (A, B)$, *the map* dua$\eta$fun *defined in Lemma 4.4 is an equivalence, so by path-univalence for* $\mathsf{U}_{\mathsf{covstack}}$ *we obtain*

$$\mathsf{dua}\eta : \mathsf{Path}_{\mathsf{Hom}_{\mathsf{U}_{\mathsf{covstack}}}\ (A,B)}\ (p, \mathsf{duahom}\ A\ B\ (\mathsf{dcoe}\ A\ B\ p))$$

Proof. To build this path, we use covEquivAx with duaηfun, as on both $\mathbb{0}_2$ and $\mathbb{1}_2$ duaηfun is strictly equal to the identity function and thus is an equivalence. We will denote the proofs that it is an equivalence on the endpoints by e_0 and e_1 respectively. As the paths at the endpoints are induced by the identity functions, we can also construct paths between the the paths induced by duaηfun at $\mathbb{0}_2$ and $\mathbb{1}_2$ and the identity

paths at A and B, which we will denote by p_0 and p_1. We now construct the path duaη as follows:

$$\begin{aligned}&\lambda i\ j.(\mathsf{com}^{\mathbb{0}_\mathbb{I} \mapsto \mathbb{1}_\mathbb{I}} (\lambda\_.\mathsf{Path}_{\mathsf{U}_{\mathsf{covstack}}}\ (p\ j, \mathsf{duahom}\ A\ B\ (\mathsf{dcoe}\ A\ B\ p)\ j))\\ &\qquad [j = \mathbb{0}_2 \mapsto \lambda i.p_0\ i,\\ &\qquad\ j = \mathbb{1}_2 \mapsto \lambda i.p_1\ i]\\ &\qquad \mathsf{ua}\ (\mathsf{covEquivAx}\ \_\ \_\ \mathsf{dua}\eta\mathsf{fun}\ e_0\ e_1\ j))\ i\end{aligned}$$

□

Theorem 5.18 (`DirUnivalence.dua`). *For all $A\ B : \mathsf{U}_{\mathsf{covstack}}$, we have an equivalence*

$$\mathsf{dua} : \mathsf{Equiv}\ (\llbracket A \rrbracket_{\mathsf{covstack}} \to \llbracket B \rrbracket_{\mathsf{covstack}})\ (\mathsf{Hom}_{\mathsf{U}_{\mathsf{covstack}}}\ (A, B))$$

Proof. As we have functions between $\llbracket A \rrbracket_{\mathsf{covstack}} \to \llbracket B \rrbracket_{\mathsf{covstack}}$ and $\mathsf{Hom}_{\mathsf{U}_{\mathsf{covstack}}}\ (A, B)$ that are inverses up to homotopy, we have a quasi-equivalence and thus an equivalence. □

Putting all of these results together, we conclude:

Theorem 5.19 (Main Theorem). *There exists a constructive model of a type theory in bicubical sets with a universe of fibrant types ($\mathsf{U}_{\mathsf{Kan}}$) and a universe of covariant fibrations ($\mathsf{U}_{\mathsf{covstack}}$) such that:*

- $\mathsf{U}_{\mathsf{covstack}}$ *has a decode function into* $\mathsf{U}_{\mathsf{Kan}}$*;*
- $\mathsf{U}_{\mathsf{Kan}}$ *is closed under* Π, Σ, DPath, DHom *and contains codes for (smaller)* $\mathsf{U}_{\mathsf{Kan}}$ *and* $\mathsf{U}_{\mathsf{covstack}}$*;*
- $\mathsf{U}_{\mathsf{covstack}}$ *is closed under* Π *(with a fixed closed domain),* Σ, DPath *and* DHom*;*
- $\mathsf{U}_{\mathsf{Kan}}$ *and* $\mathsf{U}_{\mathsf{covstack}}$ *are both path univalent;*
- $\mathsf{U}_{\mathsf{covstack}}$ *is morphism (directed path) univalent.*

The model described by Theorem 5.19 could be presented in such a way that the user only sees fibrant (Kan) types, with a homotopy type theory living in $\mathsf{U}_{\mathsf{Kan}}$, and a fixed collection of covariant type constructors living in $\mathsf{U}_{\mathsf{covstack}}$. That is, the type constructors that land in $\mathsf{U}_{\mathsf{covstack}}$ are special in two ways relative to the general universe $\mathsf{U}_{\mathsf{Kan}}$ that they are drawn from—they are covariant families, and they are families of stacks. The reason we do not restrict attention to a universe $\mathsf{U}_{\mathsf{KanStack}}$ of Kan stacks, is that at present we only know that $\mathsf{U}_{\mathsf{covstack}}$ itself is in $\mathsf{U}_{\mathsf{Kan}}$. In our current approach, to make $\mathsf{U}_{\mathsf{covstack}}$ internally extensible (by proving an internal covariance predicate as in Definitions 3.2/3.3), we would also need to expose the isStack predicate and its definition or closure conditions, so that the user can also prove that a type family is a family of stacks. If we could show that $\mathsf{U}_{\mathsf{covstack}}$ is itself a stack, then we could hide all of the stacks from the user by interpreting the homotopy type theory in $\mathsf{U}_{\mathsf{KanStack}}$.

5.4 The Equivalence Axiom in Bicubical Sets

Having used the equivalence axiom to complete our main theorem, we now provide an external constructive proof justifying its inclusion in the model. While we do not define a model structure structure on bicubical sets, we do use some aspects of the type-theoretic model structure on cubical sets (see [5, 14]) in the following definitions and proofs.

Definition 5.20. A morphism $f \in \mathbf{Set}^{\square^{\mathrm{op}}_{\mathrm{path}} \times \square^{\mathrm{op}}_{\mathrm{Ded}}}$ is a *left map* if, for every $n \in \mathbb{N}$, the commuting square of cubical sets shown below is a homotopy pullback.

$$\begin{array}{ccc} A(\_, n) & \xrightarrow{\mathbb{0}_2} & A(\_, 0) \\ {\scriptstyle f(\_,n)}\downarrow & & \downarrow{\scriptstyle f(\_,0)} \\ \Gamma(\_, n) & \xrightarrow[\mathbb{0}_2]{} & \Gamma(\_, 0) \end{array}$$

The map denoted by $\mathbb{0}_2$ is that induced by the substitution sending all directed interval variables to $\mathbb{0}_2$. A morphism is a *left fibration* if it is both a left map and a Kan fibration.

This definition unfolds to the equivalent statement that, for any directed n-cube in Γ and any point in A over its zero vertex, one can construct a unique (up to homotopy) directed n-cube in A over that in Γ.

The internal definition of relCov for a type family $A : \Gamma \to$ Type picks out the same morphisms of bicubical sets.

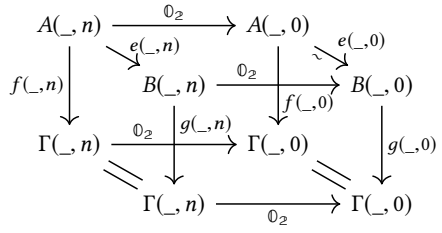
Lemma 5.21. *Given some internal type family $B : A \to \mathsf{U}_{\mathsf{cov}}$ in an open context Γ (i.e. $\Gamma \vdash B : A \to \mathsf{U}_{\mathsf{cov}}$), the corresponding external morphism $\Sigma x : \Gamma.\Sigma y : A\ x.B\ x\ y \to \Sigma x : \Gamma.A\ x$ is a left map.*

Proof. As described in Definition 3.3, the definition of relCov picks out type families for which, given any morphism p in Γ and term in A over p at $\mathbb{0}_2$, there is a unique morphism in A over p; however, as any instance of relCov can be used in an open context and the category $\square_{\mathrm{Ded}}$ is the free Cartesian category generated from $\mathbb{2}$, the instance can be used n many times in a context with (at least) n many directed interval variables to construct the solution for the directed n-cube. Thus, the external morphism is a left map. □

The covariant equivalence axiom is a direct corollary of a general lemma from [35]. As with the definition of left map, the lemma originally was stated for the Reedy model structure on bisimplicial sets. The proof itself only uses the definition of left map as a level wise homotopy pullback over the zero cells (Lemma 5.21), and the fact that weak equivalences are level wise weak equivalences (Theorem 5.15); thus, it also holds in our setting. The lemma is as follows.

Lemma 5.22. *Given bicubical sets Γ, A and B, two left fibrations $f : A \to \Gamma$ and $g : B \to \Gamma$, and a morphism $e : A \to B$ such that $f = g \circ e$, the morphism e is a weak equivalence of bicubical sets when $e(\_, 0) : A(\_, 0) \to B(\_, 0)$ is a weak equivalence of cubical sets.*

Proof. For each $n \in \mathbb{N}$ consider the following diagram.



As both the front and back squares are homotopy pullback diagrams and $e(\_, 0)$ is a weak equivalence of cubical sets, it follows that $e(\_, n)$ is also a weak equivalence of cubical sets. As weak equivalences are level wise weak equivalences of cubical sets, e is a weak equivalence of bicubical sets. □

Corollary 5.23. *Consider bicubical sets Γ, A and B, two left fibrations $f : A \to \Gamma \times \mathbb{2}$ and $g : B \to \Gamma \times \mathbb{2}$, and a morphism $e : A \to B$ such that $(f, p) = (g, q) \circ e$. Let $A_{\mathbb{2}(0)}$ denote the homotopy pullback of $f(\_, 0)$ and the $\mathbb{0}_{\mathbb{2}}$-projection $\Gamma \times \mathbb{2}(\_, 0) \hookrightarrow \Gamma(\_, 0) \times \mathbb{2}(\_, 0)$ (and define $B_{\mathbb{2}(0)}$ analogously) as shown in the following diagram.*

$$\begin{array}{ccc} A_{\mathbb{2}(0)} & \xrightarrow{\mathbb{0}_{\mathbb{2}}} & A(\_, 0) \\ {\scriptstyle f_{\mathbb{2}(0)}}\downarrow & & \downarrow{\scriptstyle f(\_,0)} \\ \Gamma \times \mathbb{2}(\_, 0) & \xrightarrow[\mathbb{0}_{\mathbb{2}}]{} & \Gamma(\_, 0) \times \mathbb{2}(\_, 0) \end{array}$$

Let $e_{\mathbb{2}(0)} : A_{\mathbb{2}(0)} \to B_{\mathbb{2}(0)}$ be the morphism induced by e. If $e_{\mathbb{2}(0)}$ is a weak equivalence of bicubical sets, then e is also a weak equivalence of bicubical sets.

Proof. As $e_{\mathbb{2}(0)}$ is a weak equivalence of bicubical sets, it is a level wise weak equivalence of cubical sets. In particular, $e_{\mathbb{2}(0)}(\_, 0) : A_{\mathbb{2}(0)}(\_, 0) \to B_{\mathbb{2}(0)}(\_, 0)$ is a weak equivalence. Observe that $A_{\mathbb{2}(0)}(\_, 0) = A(\_, 0)$ (and similarly for B), and $e_{\mathbb{2}(0)}(\_, 0) = e(\_, 0)$. As $e(\_, 0)$ is a weak equivalence of cubical sets, Lemma 5.22 allows us to conclude that e is a weak equivalence of bicubical sets. □

Theorem 5.24. covEquivAx *holds in our model.*

Proof. In order to see Corollary 5.23 as a justification of the covariant equivalence axiom (Definition 5.16), we must make a couple observations. First, from Lemma 5.21, we know that the internal term $\Gamma \vdash p : \mathbb{2} \to \mathsf{U}_{\mathsf{cov}}$ corresponds to the external left fibration $\Sigma x : \Gamma.\Sigma i : \mathbb{2}.p\ x\ i \to \Gamma \times \mathbb{2}$ (and analogously for q). Thus, in the external corollary, we can instantiate A and B with the external closed bicubical sets $\Sigma x : \Gamma.\Sigma i : \mathbb{2}.p\ x\ i$ and $\Sigma x : \Gamma.\Sigma i : \mathbb{2}.q\ x\ i$. Let A and B denote these bicubical sets in the following proof. The second item of note is that the pullback defining $A_{\mathbb{2}(0)}$ in the statement of the corollary corresponds to the restriction of $A \to \Gamma \times \mathbb{2}$ to the components over the two endpoints $\mathbb{0}_{\mathbb{2}}$ and $\mathbb{1}_{\mathbb{2}}$ of $\mathbb{2}$. Thus, the internal proofs e_0 and e_1 in the open context Γ demonstrating f is an equivalence over the endpoints of the directed interval matches up with the equivalence needed in the statement of Corollary 5.23.

We also need to confirm that the proof of covEquivAx is stable under substitution, and thus can be an internal axiom. Given covEquivAx uses Theorem 5.15 to construct the equivalence from the input data, we first show that for any A, B and $f : A \to B$ where f is a point-wise equivalence, the term of isEquiv f built using Theorem 5.15 is stable under (strict) pullbacks. As the construction in Theorem 5.15 described in Theorem 5.1 of [13] is computed point-wise with respect to $\square_{\mathrm{Ded}}$ and pullbacks of bicubical sets are also point-wise pullbacks of cubical sets, the pullback commutes with the construction and thus Theorem 5.15 commutes with substitution. The second construction that must be stable under pullbacks is the level-wise weak equivalence defined in Lemma 5.22. In particular, given cubical sets Γ, A and B, left fibrations $f : A \to \Gamma$ and $g : B \to \Gamma$, a morphism $e : A \to B$ such that $f = g \circ e$, and a proof v_0 : isEquiv $e(\_, 0)$, we need to know that every v_n : isEquiv $e(\_, n)$ created by the proof is stable under pullback. Again, as each v_n is computed separately for each object in $\square_{\mathrm{Ded}}$, the pullback commutes with the point-wise argument, and as every v_n is constructed as the solution of a covariant filling problem (which in particular commutes with pullbacks), the construction in Lemma 5.22 is also stable under substitution. □

6 Future Work

We expect that the internal language axioms in Section 2 can be given an operational semantics following [3, 21]: the new axioms $\mathbb{1}$to$\mathbb{2}$triv and $\mathbb{2}$mono are used only to refute certain possibilities in showing that the definitions of Lemmas 4.1 and 4.2 type check, and the cobar construction used to complete directed univalence is entirely constructive. We also are interested in developing the constructive theory of inductive and higher inductive types in a directed setting, opening up the potential for directed type theory to be a useful setting for the formal verification of computational structures.

For the use of our constructive model for formalization of higher category theory, we conjecture that our type theory on bicubical sets can support a model of $(\infty, 1)$-categories in the same way that the Segal/Rezk types [32] in bisimplicial sets do. First, we have observed that the bicubical sets satisfying the Segal or Rezk condition also weakly satisfy a sheaf condition on the directed cubes that makes them equivalent to simplicial-cubical sets. Thus, if we use the equivariant Cartesian cubical sets model proposed by Awodey, Cavallo, Coquand, Riehl and Sattler as the base cubical type theory—which is Quillen equivalent to the simplicial model [30]—it seems likely that in the resulting model Segal/Rezk bicubical sets would be equivalent to Segal/Rezk bisimplicial sets.

Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research under Grant No. FA9550-17-1-0363, No. FA9550-15-1-0053, and No. FA9550-16-1-0292. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Air Force Office of Scientific Research. We thank Evan Cavallo, Thierry Coquand, Emily Riehl, Mitchell Riley, Christian Sattler, Mike Shulman, Jonathan Weinberger and the anonymous reviewers for their helpful insights and feedback.

References

[1] [n.d.]. redtt. https://github.com/RedPRL/redtt.

[2] Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Kuen-Bang Hou (Favonia), Robert Harper, and Daniel R. Licata. 2019. Syntax and Models of Cartesian Cubical Type Theory. (2019). Available from https://github.com/dlicata335/cart-cube/blob/master/cart-cube.pdf.

[3] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. 2017. Computational Higher Type Theory III: Univalent Universes and Exact Equality. (2017). arXiv:1712.01800.

[4] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. 2018. Cartesian Cubical Computational Type Theory: Constructive Reasoning with Paths and Equalities.. In *Computer Science Logic*.

[5] Steve Awodey. 2019. A Quillen model structure on the category of cartesian cubical sets. (2019). Talk at *Homotopy Type Theory 2019*, available from https://hott.github.io/HoTT-2019.

[6] S. Awodey and M. Warren. 2009. Homotopy theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society* (2009).

[7] Marc Bezem, Thierry Coquand, and Simon Huber. 2013. A model of type theory in cubical sets. (September 2013). Preprint.

[8] Lars Birkedal, Ales Bizjak, Ranald Clouston, Hans Bugge Grathwohl, Bas Spitters, and Andrea Vezzosi. 2016. Guarded Cubical Type Theory: Path Equality for Guarded Recursion. (2016). arXiv:1606.05223.

[9] Ulrik Buchholtz and Jonathan Weinberger. 2019. Type-theoretic modalities for synthetic $(\infty, 1)$-categories. (2019). Talk at Homotopy Type Theory 2019.

[10] Evan Cavallo and Robert Harper. 2019. Parametric Cubical Type Theory. (2019). arXiv:1901.00489.

[11] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. 2018. Cubical Type Theory: a constructive interpretation of the univalence axiom. In *Post-proceedings of the 21st International Conference on Types for Proofs and Programs (TYPES 2015)*, T. Uustalu (Ed.). 5:1–5:34. doi: 10.4230/LIPIcs.TYPES.2015.5.

[12] Thierry Coquand. 2019. Constructive Presheaf Models of Univalence. (2019). Talk at Homotopy Type Theory 2019.

[13] Thierry Coquand and Fabian Ruch. 2019. Constructive sheaf models of type theory. (2019). arXiv:1912.10407.

[14] Thierry Coquand and Christian Sattler. 2016. A model structure on some presheaf categories. (2016). Available from cse.chalmers.se/~coquand/mod2.pdf.

[15] Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, Samuel Mimram, and Martin Raussen. 2016. *Directed algebraic topology and concurrency*. Springer.

[16] Marcelo Fiore, Gordon Plotkin, and Daniele Turi. 1999. Abstract Syntax and Variable Binding. In *IEEE Symposium on Logic in Computer Science*.

[17] Nicola Gambino and Simon Henry. 2019. Towards a constructive simplicial model of Univalent Foundations. (2019). https://arxiv.org/abs/1905.06281.

[18] M. Grandis. 2009. *Directed Algebraic Topology: Models of non-reversible worlds*. Cambridge University Press.

[19] Martin Hofmann. 1995. *Extensional Concepts in Intensional Type Theory*. Ph.D. Dissertation. University of Edinburgh.

[20] Martin Hofmann. 1999. Semantical analysis of higher-order abstract syntax. In *IEEE Symposium on Logic in Computer Science*.

[21] Simon Huber. 2018. Canonicity for Cubical Type Theory. *Journal of Automated Reasoning* (2018). https://doi.org/10.1007/s10817-018-9469-1

[22] Daniel R. Licata and Robert Harper. 2011. 2-Dimensional Directed Type Theory. In *Mathematical Foundations of Programming Semantics (MFPS)*.

[23] Daniel R. Licata, Ian Orton, Andrew M. Pitts, and Bas Spitters. 2018. Internal Universes in Models of Homotopy Type Theory. In *International Conference on Formal Structures for Computation and Deduction*.

[24] Paige Randall North. 2018. Towards a directed homotopy type theory. (2018). arXiv:1807.10566.

[25] Andreas Nuyts. 2015. *Towards a directed HoTT based on 4 kinds of variance*. Master's thesis. KU Leuven.

[26] Andreas Nuyts. 2017. A Model of Parametric Dependent Type Theory in Bridge/Path Cubical Sets. (2017). arXiv:1706.04383.

[27] Ian Orton and Andrew M. Pitts. 2016. Axioms for Modelling Cubical Type Theory in a Topos. In *Computer Science Logic*.

[28] I. Orton and A. M. Pitts. 2018. Axioms for Modelling Cubical Type Theory in a Topos. *Logical Methods in Computer Science* 14, 4:23 (Dec. 2018), 1–33. Special issue for CSL 2016.

[29] Emily Riehl. 2014. *Categorical Homotopy Theory*. Cambridge University Press.

[30] Emily Riehl. 2019. The equivariant uniform kan fibration model of cubical homotopy type theory. (2019). Talk at Homotopy Type Theory 2019.

[31] Emily Riehl, Evan Cavallo, and Christian Sattler. 2018. On the directed univalence axiom. Talk at the AMS Special Session on Homotopy Type Theory, Joint Mathematics Meetings.

[32] Emily Riehl and Michael Shulman. 2018. A type theory for synthetic ∞-categories. *Higher Structures* 1, 1 (2018).

[33] Emily Riehl and Dominic Verity. 2014. The theory and practice of Reedy categories. *Theory and Applications of Categories* 29, 9 (2014).

[34] Christian Sattler. 2018. Do cubical models of type theory also model homotopy types? (2018). Talk at Workshop on Types, Homotopy Type theory, and Verification, Hausdorff Institute for Mathematics.

[35] Christian Sattler and Emily Riehl. 2018. Directed univalence for the left fibration classifier. (2018). Private correspondence.

[36] Michael Shulman. 2009. Homotopy limits and colimits and enriched homotopy theory. (2009). https://arxiv.org/abs/math/0610194.

[37] Michael Shulman. 2019. All $(\infty, 1)$-toposes have strict univalent universes. (2019). arXiv:1904.07004.

[38] The Univalent Foundations Program, Institute for Advanced Study. 2013. *Homotopy Type Theory: Univalent Foundations Of Mathematics*. Available from homotopytypetheory.org/book.

[39] Vladimir Voevodsky. 2006. A very short note on homotopy λ-calculus. *Unpublished* (September 2006), 1–7. http://www.math.ias.edu/vladimir/files/2006_09_Hlambda.pdf

[40] Matthew Z. Weaver and Daniel R. Licata. 2020. A Constructive Model of Directed Univalence in Bicubical Sets (extended version). (2020). Available from https://dlicata.wescreates.wesleyan.edu/.