Decomposing the Univalence Axiom

Ian Orton[1]
University of Cambridge Dept. Computer Science & Technology, Cambridge, UK
https://orcid.org/0000-0002-9924-0623

Andrew M. Pitts
University of Cambridge Dept. Computer Science & Technology, Cambridge, UK
https://orcid.org/0000-0001-7775-3471

Abstract

This paper investigates Voevodsky's univalence axiom in intensional Martin-Löf type theory. In particular, it looks at how univalence can be derived from simpler axioms. We first present some existing work, collected together from various published and unpublished sources; we then present a new decomposition of the univalence axiom into simpler axioms. We argue that these axioms are easier to verify in certain potential models of univalent type theory, particularly those models based on cubical sets. Finally we show how this decomposition is relevant to an open problem in type theory.

2012 ACM Subject Classification Theory of computation → Type theory

Keywords and phrases dependent type theory, homotopy type theory, univalent type theory, univalence, cubical type theory, cubical sets

Digital Object Identifier 10.4230/LIPIcs.TYPES.2017.7

Related Version `https://arxiv.org/abs/1712.04890`

Supplement Material `https://doi.org/10.17863/CAM.25036` (Agda 2.5.4 source code)

Acknowledgements We would like to thank the anonymous referees for their insightful comments which materially improved the quality of the paper.

1 Introduction

Extensionality is a principle whereby two mathematical objects are deemed to be equal if they have the same observable properties. Often, formal systems for mathematics will include axioms designed to capture this principle. In the context of set theory we have the axiom of extensionality, which tells us that two sets are equal if they contain the same elements. In the context of (univalent) type theory we have Voevodsky's univalence axiom [15, Section 2.10], which tells us, roughly speaking, that two types are equal if they are isomorphic.

In axiomatic set theory the axiom of extensionality is easily formalised as a simple implication $\forall A \forall B((\forall X(X \in A \iff X \in B)) \implies A = B)$. The converse implication follows from the properties of equality, and combining these two implications we deduce that equality of two sets is logically equivalent to them having the same elements. At this point we are done, we now have an extensionality principle for sets and nothing further needs to be assumed.

[1] Supported by a UK EPSRC PhD studentship, funded by grants EP/L504920/1, EP/M506485/1.

The situation is more complicated in a proof relevant setting such as intensional type theory. As with sets we can formalise the statement of interest in the language of type theory as $(A\,B : U) \to A \simeq B \to A = B$, where $A \simeq B$ is the type of *equivalences* between A and B. We then postulate the existence of a term witnessing this statement. As before, the converse implication follows from the properties of the identity type, and hence equality of two types is logically equivalent to them being isomorphic. However, the proof relevant nature of type theory means that what we have described so far will be insufficient. We may want to know how equalities derived using the postulated term compute when passed to the eliminator for identity types. For example, if we convert them back into equivalences do we always get the same equivalence that we started with?

In univalent type theory (UTT), also known as homotopy type theory (HoTT), these problems are resolved by taking a different approach to the statement of the univalence axiom. As mentioned before the converse implication, $(A\,B : U) \to A = B \to A \simeq B$, follows from the properties of the identity type. The approach taken in UTT is to state that for any types A and B this map is itself an equivalence between the types $A = B$ and $A \simeq B$. From this fact we can deduce the existence of a map in the other direction (the original implication of interest), as well as some information about how that map computes.

Merely stating that a certain canonical map is an equivalence is a very concise way to express the univalence axiom. From a mathematical point of view it is appealingly simple and yet powerful. In particular, this statement has the nice property that it is a *mere proposition* [15, Definition 3.3.1] and so there is no ambiguity about the term witnessing the axiom.

However, there are some disadvantages to this way of stating the univalence axiom. For example, verifying the univalence axiom in a model of type theory can be a difficult task. Fully expanded, this seemingly simple statement becomes very large, with many complex subterms. Verifying univalence directly, by computing the interpretation of the statement in the model and explicitly constructing the interpretation of its proof term, may be unfeasible. Instead, one would need to build up several intermediate results about contractibility, equivalences, and possibly new constructions such as *Glueing* [8, Section 6], through a mixture of internal, syntactic and semantic arguments.

The contribution in this paper is a reduction of the usual statement of univalence to a collection of simpler axioms which are more easily verified in certain models of dependent type theory, particularly those based on cubical sets [8, 5, 2, 3, 14, 6]. Importantly, we do not propose these axioms as an alternative statement for the univalence axiom when doing mathematics in univalent type theory. These axioms are designed with the previous goal in mind and are not intended to be mathematically elegant or user-friendly.

In the rest of this paper we begin with some preliminary definitions and notational conventions (Section 2). We then briefly discuss the univalence axiom (Section 3). These sections cover existing work. We then introduce our alternative set of axioms (Section 4), and examine their application to models of type theory (Section 5). Finally, we propose another application of these axioms to an open problem in UTT (Section 6).

Agda formalisation

This work presented in this paper is supported by two separate developments in the Agda proof assistant [1]. The first covers the material in sections 2-4, where Agda is useful for precisely tracking universe levels in many of the theorems. The second covers the material in Section 5, and builds on the development accompanying [13]. In this development we use Agda to verify our constructions in the internal type theory of the cubical sets topos. The

source for both can be found at https://doi.org/10.17863/CAM.25036.

2 Preliminaries

In most of this paper we work in intensional Martin-Löf type theory with dependent sums and products, intentional identity types, and a cumulative hierarchy of universes $U_0 : U_1 : U_2 : \ldots$.

We use the symbol $=$ for the identity type, $\equiv$ for definitional equality and $\triangleq$ when giving definitions. Given $p : x = y$ and $q : y = z$, we write $p \centerdot q : x = z$ for the composition of identity proofs, and $p^{-1} : y = x$ for the inverse proof.

We also assume the principle of function extensionality, which states that two functions $f, g : \prod_{x:A} B(x)$ are equal whenever they are pointwise equal: $f \approx g \triangleq \prod_{x:A} f(x) = g(x)$. That is, that there exists a term:

$$funext_{i,j} : \prod_{A:U_i} \prod_{B:A\to U_j} \prod_{f,g:\Pi_{x:A}B(x)} f \approx g \to f = g$$

for all universe levels i, j.

Note that in the Agda development mentioned previously we do not assume function extensionality in general, but rather we make it an explicit assumption to each theorem. This means that we can see exactly where function extensionality is used, and at which universe levels it needs to hold.

We now recall some standard definitions and results in UTT/HoTT.

▶ **Definition 2.1** (Contractibility). A type A is said to be *contractible* if the type

$$isContr(A) \triangleq \sum_{a_0:A} \prod_{a:A} (a_0 = a)$$

is inhabited. Contractibility expresses the fact that a type has a unique inhabitant.

▶ **Definition 2.2** (Singletons). Given a type A and element $a : A$, we can define

$$sing(a) \triangleq \sum_{x:A} (a = x)$$

to be the type of elements of A which are equal to a. It is easily shown by path induction that the type $sing(a)$ is always contractible.

▶ **Definition 2.3** (Equivalences). An *equivalence* from $A \simeq B$ is a pair (f, e) where $f : A \to B$ and e is a proof that for every $b : B$ the fiber of f at b is contractible. To be precise:

$$A \simeq B \triangleq \sum_{f:A\to B} isEquiv(f)$$

where

$$fib_f(b) \triangleq \sum_{a:A} (f\ a = b) \qquad \text{and} \qquad isEquiv(f) \triangleq \prod_{b:B} isContr(fib_f(b))$$

for $A : U_i$, $B : U_j$ for any i, j.

A simple example of an equivalence is the identity function $id_A : A \to A$ for any type A. To demonstrate that id_A is an equivalence we must show that $\prod_{a:A} isContr(\sum_{x:A}(a = x))$, but this is equivalent to the statement that $sing(a)$ is contractible for all $a : A$.

3 Voevodsky's Univalence Axiom

In this section we introduce Voevodsky's univalence axiom. We then present an existing result which decomposes the univalence axiom into a "naive" form and a computation rule. In Section 4, we will then decompose these two axioms further into five even simpler axioms.

▶ **Definition 3.1** (Coerce and idtoeqv)**.** For all i, and types $A, B : U_i$, there is a canonical map $idtoeqv : (A = B) \to (A \simeq B)$ which is defined by path induction on the proof $A = B$:

$$idtoeqv(\mathtt{refl}) \triangleq id_A$$

where $id_A : A \simeq A$ is the identity map regarded as an equivalence. We can also define a map $\mathtt{coerce} : (A = B) \to A \to B$ either by path induction, or as:

$$\mathtt{coerce}(p, a) \triangleq \mathtt{fst}(idtoeqv(p))(a)$$

where `fst` is the first projection.

▶ **Definition 3.2** (Voevodsky's univalence axiom)**.** The univalence axiom for a universe U_i asserts that for all $A, B : U_i$ the map $idtoeqv : (A = B) \to (A \simeq B)$ is an equivalence.

In light of the following definition we will often refer to the univalence axiom as the *proper* univalence axiom.

▶ **Definition 3.3** (The naive univalence axiom)**.** The naive univalence axiom for a universe U_i gives, for all $A, B : U_i$, a map from equivalences to equalities. In other words, it asserts the existence of an inhabitant of the type:

$$UA_i \triangleq \prod_{A,B:U_i} A \simeq B \to A = B$$

When using a term $ua : UA_i$ we will often omit the first two arguments (A and B). Proofs of naive univalence may also come with an associated computation rule. That is, an inhabitant of the type $UA\beta_i(ua)$, where:

$$UA\beta_i(ua) \triangleq \prod_{A,B:U_i} \prod_{f:A\to B} \prod_{e:isEquiv(f)} \mathtt{coerce}\,(ua(f, e)) = f$$

Next, we give a result which is known in the UTT/HoTT community and has been discussed on the HoTT mailing list. However, the authors are not aware of any existing presentation of a proof in the literature. This result decomposes the proper univalence axiom into the naive version and a computation rule. First we give a lemma which generalises the core construction of this result.

▶ **Lemma 3.4.** *Given $X : U_i$, $Y : X \to X \to U_j$ and a map $f : \prod_{x,x':X} x = x' \to Y(x, x')$ then $f\,x\,x'$ is an equivalence for all $x, x' : X$ iff there exists a map*

$$g : \prod_{x,x':X} Y(x, x') \to x = x'$$

such that for all $x, x' : X$ and $y : Y(x, x')$ we have $f(g(y)) = y$ (we leave the first two arguments to f and g implicit).

Proof. For the backwards direction, assume that we are given g as above. To show that f is an equivalence it suffices to show that f is a bi-invertible map [15, Section 4.3]. To do this we must exhibit both a right and left inverse.

For the left inverse we take $g'(y) \triangleq g(y) \cdot g(f(\mathtt{refl}))^{-1}$. To see that this is indeed a left inverse to f consider an arbitrary $p : x = x'$, we aim to show that $g'(f(p)) = p$. By path induction we may assume that $x \equiv x'$ and $p \equiv \mathtt{refl}$ and therefore we are required to show $g'(f(\mathtt{refl})) = \mathtt{refl}$. However, since $g'(f(\mathtt{refl})) \equiv g(f(\mathtt{refl})) \cdot g(f(\mathtt{refl}))^{-1}$ this goal simplifies to $g(f(\mathtt{refl})) \cdot g(f(\mathtt{refl}))^{-1} = \mathtt{refl}$ which follows immediately from the groupoid laws for identity types.

For the right inverse we take g unchanged and observe that we know $f(g(y)) = y$ for all $y : Y(x, x')$ by assumption. Therefore the map f is an equivalence.

For the forwards direction, given a proof $e : \mathit{isEquiv}(f)$ and $y : Y(x, x')$ we have $\mathtt{fst}(e(y)) : \sum_{p:x=x'} f(p) = y$. We can then define $g(y)$ to be the first component of this and the second component tells us that $f(g(y)) = y$ as required. ◀

▶ **Theorem 3.5.** *Naive univalence, along with a computation rule, is logically equivalent to the proper univalence axiom. That is, there are terms*

$$ua : UA_i, \qquad ua\beta : UA\beta_i(ua)$$

iff for all types $A, B : U_i$, the map $\mathit{idtoeqv} : (A = B) \to (A \simeq B)$ is an equivalence.

Proof. By $ua\beta$ we know that $\mathtt{fst}(\mathit{idtoeqv}(ua(f, e))) = f$ for all $(f, e) : A \simeq B$. Now, since $\mathit{isEquiv}(f)$ is a mere proposition for each f, we can deduce that $\mathit{idtoeqv}(ua(f, e)) = (f, e)$ by [15, Lemma 3.5.1]. Therefore we simply take $X \equiv U_i$, $Y(A, B) \equiv A \simeq B$, $f \equiv \mathit{idtoeqv}$ and $g \equiv ua$ in Lemma 3.4 to deduce the desired result. ◀

4 A new set of axioms

In this section we further decompose the univalence axiom into even simpler axioms. We show that it is equivalent to axioms (1) to (5) given in Table 1. Note that these axioms apply to a specific universe U_i.

	Axiom		Premise(s)		Equality		
(1)	*unit*	:			A	$=$	$\sum_{a:A} 1$
(2)	*flip*	:			$\sum_{a:A} \sum_{b:B} C\ a\ b$	$=$	$\sum_{b:B} \sum_{a:A} C\ a\ b$
(3)	*contract*	:	$\mathit{isContr}\ A$	$\to$	A	$=$	1
(4)	*unitβ*	:			$\mathit{coerce}\ \mathit{unit}\ a$	$=$	$(a, *)$
(5)	*flipβ*	:			$\mathit{coerce}\ \mathit{flip}\ (a, b, c)$	$=$	(b, a, c)

Table 1 ($A, B : U_i$, $C : A \to B \to U_i$, $a : A$, $b : B$ and $c : C\,a\,b$, for some universe U_i)

We begin by decomposing naive univalence, UA_i, into axioms (1)-(3). These axioms also follow from UA_i. Recall that we are taking function extensionality as an ambient assumption.

▶ **Theorem 4.1.** *Axioms (1)-(3) for a universe U_i are together logically equivalent to UA_i.*

Proof. We begin by showing the forwards direction. Assume that we are given axioms (1) to (3). We now aim to define a term $ua : UA_i$. Given arbitrary types $A, B : U_i$ and an

equivalence $(f, e) : A \simeq B$ we define $ua(f, e) : A = B$ as follows:

$$\begin{aligned}
A &= \sum_{a:A} 1 && \text{by (1)} \\
&= \sum_{a:A} \sum_{b:B} f\ a = b && \text{by funext and (3) on } sing(fa) \\
&= \sum_{b:B} \sum_{a:A} f\ a = b && \text{by (2)} \\
&= \sum_{b:B} 1 && \text{by funext and (3) on } fib_f(b) \text{ (contractible by } e) \\
&= B && \text{by (1)}
\end{aligned}$$

where the proof that $A = B$ is given by the concatenation of each step of the above calculation.

The backwards direction follows from the fact that the obvious maps $A \to \sum_{a:A} 1$ and $(\sum_{a:A} \sum_{b:B} C\ a\ b) \to (\sum_{b:B} \sum_{a:A} C\ a\ b)$ are both easily shown to be bi-invertible and hence equivalences, and from the fact that any contractible type is equivalent to 1 [15, Lemma 3.11.3.]. Therefore given $ua : UA_i$ we simply apply it to these equivalences to get the required equalities (1)-(3). ◀

Next, we decompose the computation rule for naive univalence $UA\beta_i$ into axioms (4) and (5). Since $UA\beta_i$ depends on UA_i and axioms (4) and (5) depend on axioms (1) and (2) respectively, we in fact show the logical equivalence between the pair UA_i and $UA\beta_i$, and axioms (1)-(5).

▶ **Lemma 4.2.** *The function* `coerce` *is compositional. That is, given types* $A, B, C : U_i$*, and equalities* $p : A = B$ *and* $q : B = C$ *we have* $\texttt{coerce}(p \cdot q) = \texttt{coerce}(q) \circ \texttt{coerce}(p)$.

Proof. Straightforward by path induction on either of p or q, or on both. ◀

▶ **Theorem 4.3.** *Axioms (1)-(5) for a universe* U_i *are together logically equivalent to* $\sum_{ua:UA_i} UA\beta_i(ua)$.

Proof. For the forwards direction we know from Theorem 4.1 that axioms (1) to (3) allow us to construct a term $ua : UA_i$. If, in addition, we assume axioms (4) and (5) then we can show that for all $(f, e) : A \simeq B$ we have $\texttt{coerce}(ua(f, e)) = f$ as follows.

Since ua was constructed as the concatenation of five equalities then, in light of Lemma 4.2, we have that coercing along $ua(f, e)$ is equal to the result of coercing along each stage of the composite equality $ua(f, e)$. Therefore starting with an arbitrary $a : A$, we can track what happens at each stage of this process like so:

$$a \quad \mapsto \quad (a,\ *) \quad \mapsto \quad (a,\ f\ a,\ \mathit{refl}) \quad \mapsto \quad (f\ a,\ a,\ \mathit{refl}) \quad \mapsto \quad (f\ a,\ *) \quad \mapsto \quad f\ a$$

Therefore we see that for all $a : A$ we have $\texttt{coerce}(ua(f, e))(a) = f(a)$ and hence by function extensionality we have $\texttt{coerce}(ua(f, e)) = f$ as required.

For the reverse direction we assume that we are given $ua : UA_i$ and $ua\beta : UA\beta_i(ua)$. We can now apply Theorem 4.1 to construct terms *unit*, *flip* and *contract* satisfying axioms (1) to (3) from ua.

Since *unit* and *flip* were constructed by applying ua to the obvious equivalences, then by $ua\beta$ we know that applying `coerce` to these equalities will return the equivalences that we started with. From this we can easily construct terms $unit\beta$ and $flip\beta$ satisfying axioms (4) and (5) respectively. ◀

▶ **Corollary 4.4.** *Axioms (1)-(5) for a universe U_i are together logically equivalent to the proper univalence axiom for U_i.*

Proof. By combining Theorems 3.5 and 4.3. ◀

5 Applications in models of type theory

In this section we discuss one reason why the result given in Corollary 4.4 is useful when trying to construct models of univalent type theory. Specifically, we believe that this decomposition is particularly useful for showing that a model of type theory with an interval object (e.g. cubical type theory [8]) supports the univalence axiom. We first explain why we believe this to be the case in general terms, and then give a precise account of what happens in the specific case of the cubical sets model presented in [8]. The arguments given here should translate to many similar models of type theory [5, 2, 3, 14, 6].

Note that we are assuming function extensionality. Every model of univalence must satisfy function extensionality [15, Section 4.9], but it is often much easier to verify function extensionality than the proper univalence axiom in a model of type theory. In particular, function extensionality will hold in any type theory which includes an appropriate interval type, cf. [15, Lemma 6.3.2].

Experience shows that axioms (1), (2), (4) and (5) are simple to verify in many potential models of univalent type theory. To understand why, it is useful to consider the interpretation of $A \simeq B$ in such a model. Propositional equality in the type theory is usually not interpreted as equality in the model's metatheory, but rather as a construction on types e.g. path spaces in models of HoTT. Therefore, writing $[\![X]\!]$ for the interpretation of a type X, an equivalence in the type theory will give rise to morphisms $f : [\![A]\!] \to [\![B]\!]$ and $g : [\![B]\!] \to [\![A]\!]$ which are not exact inverses, but rather are inverses modulo the interpretation of propositional equality, e.g. the existence of paths connecting x and $g(f(x))$, and y and $f(g(y))$ for all $x \in [\![X]\!], y \in [\![Y]\!]$. However, in many models the interpretations of A and $\sum_{a:A} 1$, and of $\sum_{a:A} \sum_{b:B} C\ a\ b$ and $\sum_{b:B} \sum_{a:A} C\ a\ b$ will be isomorphic, i.e. there will be morphisms going back and forth which are inverses up to equality in the model's metatheory. This will be true in any presheaf model of type theory of the kind described in Section 5.1.1, and should be true more generally in any model which validates eta-rules for 1 and Σ.

This means that we can satisfy (1) and (2) by proving that this stronger notion of isomorphism gives rise to a propositional equality between types. Verifying axioms (4) and (5) should then reduce to a fairly straightforward calculation involving two instance of this construction.

This leaves axiom (3), which captures the homotopical condition that every contractible space can be continuously deformed into a point. The hope is that verifying the previous axioms should be fairly straightforward, leaving this as the only non-trivial condition to check.

We now examine what happens in the specific case of cubical sets [8].

5.1 Example: the CCHM model of univalent type theory

In this section we will examine what happens in the case of the Cohen, Coquand, Huber, Mörtberg (CCHM) model of type theory based on cubical sets [8]. To be clear, this model is shown to validate the univalence axiom in the previously cited paper. However, here we give an alternative, hopefully simpler, proof of univalence using the decomposition given in Section 4. We start from the knowledge that cubical sets model a type theory with *Path*

types given by maps out of an interval object $\mathtt{I}$, and where types come equipped with a *composition* operation which is closed under all type formers $(\Sigma, \Pi, Path)$. From this we then show how to validate our axioms, and therefore the proper univalence axiom. This therefore yields an alternative proof of univalence which does not use the glueing construction from [8]. Note however that this construction is still required to show that the universe is fibrant, and hence we cannot completely eliminate the use of glueing from the CCHM model.

For most of this section we will work in the internal language of the cubical sets topos, using a technique developed by the authors in a previous paper [13]. We begin with a brief summary of the cubical sets model and then describe the internal language approach to working with such models. For those unfamiliar with this material we refer the reader to [8] and [13] respectively for further details.

5.1.1 The cubical sets model

Cohen *et al* [8] present a model of type theory using the category $\hat{\mathcal{C}}$ of presheaves on the small category $\mathcal{C}$ whose objects are given by finite sets of symbols, written I, J, K, with $\mathcal{C}(I, J)$ being the set of maps $J \to dm(I)$, where $dm(I)$ is the free De Morgan algebra [4] on the set I.

First we recall the standard way of constructing a presheaf model of type theory [10], note that this is not the final model construction. Take $\hat{\mathcal{C}}$ to be the category of contexts with the types over a context $\Gamma \in \hat{\mathcal{C}}$, written $\mathtt{Ty}(\Gamma)$, given by presheaves on Γ's category of elements. Terms of type $A \in \mathtt{Ty}(\Gamma)$, written $Ter(\Gamma \vdash A)$ are simply global sections of A. Explicitly, this means that a type $A \in \mathtt{Ty}(\Gamma)$ is given by a family of sets $A(I, \rho)$ for every $I \in \mathcal{C}$ and $\rho \in \Gamma(I)$ such that for every $a \in A(I, \rho)$ and $f : J \to I$ we have $A(f)(a) \in A(J, \Gamma(f)(\rho))$ with $A(id_I)(a) = a$ and $A(g \circ f)(a) = A(f)(A(g)(a))$. A term $a \in Ter(\Gamma \vdash A)$ is given by a family $a(I, \rho) \in A(I, \rho)$ for every $I \in \mathcal{C}$ and $\rho \in \Gamma(I)$ such that for all $f : J \to I$ we have $A(f)(a(I, \rho)) = a(J, \Gamma(f)(\rho))$. Following the convention in [8] we will often omit the first argument I and will write functorial actions $\Gamma(f)(\rho)$ simply as ρf.

These constructions all model substitution, context extension, projection, etc, in the correct way, and can be shown to form a category with families (CwF) in the sense of Dybjer [9]. Such a model always supports both dependent sums and products. For example, given types $A \in \mathtt{Ty}(\Gamma)$ and $B \in \mathtt{Ty}(\Gamma.A)$ then the dependent sum $\Sigma AB \in \mathtt{Ty}(\Gamma)$ can be interpreted as

$$\Sigma AB(I, \rho) \triangleq \{(a, b) \mid a \in A(I, \rho),\ b \in B(I, \rho, a)\}$$

As stated in the previous section, any model of this kind will always have $\Sigma A(\Sigma BC)$ being strictly isomorphic to $\Sigma B(\Sigma AC)$, that is, with natural transformations in each direction which are inverses up to equality in the model's metatheory. Furthermore, assuming that the terminal type 1 is interpreted as the terminal presheaf, then the same will be true for the types A and $\Sigma A1$. This is potentially useful when verifying axioms (1), (2), (4) and (5) for the reasons given above.

To get a model of type theory which validates the univalence axiom we restrict our attention to types with an associated *composition structure* [8, Definition 13]. We call such types *fibrant* and write $\mathtt{FTy}(\Gamma)$ for the collection of fibrant types over a context $\Gamma \in \hat{\mathcal{C}}$, and $\mathtt{FTy}_0(\Gamma)$ for the collection of small types. Taking contexts, terms, type formers and substitution as before, we get a new CwF of fibrant types. We delay giving the exact definition of a composition structure until after we introduce the internal type theory of $\hat{\mathcal{C}}$ in the following section.

5.1.2 The internal type theory of $\hat{\mathcal{C}}$

In previous work [13] the authors axiomatised the properties of the cubical sets topos needed to develop a model of univalent type theory. They then showed how many of the constructions used in the model could be replicated using the internal type theory of an elementary topos [12]. Here we will often build on this approach, working mostly in the internal type theory. We now give a brief overview of this approach, and refer the reader to [13] for full details.

We use a concrete syntax inspired by Agda [1]. Dependent function types are written as $(x : A) \to B$ with lambda abstractions written as $\lambda(x : A) \to t$. We use $\{\}$ in place of $()$ to indicate the use of implicit arguments. Dependent product types are written as $(x : A) \times B$ with the pairing operation written as (s, t).

We assume the existence of an interval object $\mathtt{I}$ with endpoints $\mathtt{0}, \mathtt{1} : 1 \to \mathtt{I}$ subject to certain conditions, a class of propositions $\mathtt{Cof} \rightarrowtail \Omega$, closed under $\vee, \wedge$ and $\mathtt{I}$-indexed $\forall$, which we call the *cofibrant* propositions and two internal Russell-style universes $\mathcal{U} : \mathcal{U}_1$, closed under all the type formers of the internal language. Given $\varphi : \Omega$ we write $[\varphi] \triangleq \{\_ : 1 \mid \varphi\}$ for the type whose inhabitation corresponds to the provability of φ, and given a object $\Gamma : \mathcal{U}$ and a cofibrant property $\Phi : \Gamma \to \mathtt{Cof}$ we write $\Gamma | \Phi \triangleq (x : \Gamma) \times [\Phi\, x]$ for the *restriction of* Γ *by* Φ. Given $\varphi : \mathtt{Cof}$, $f : [\varphi] \to A$ and $a : A$ we write $(\varphi, f) \nearrow a$ for $(u : [\varphi]) \to f\, u = a$; thus elements of this type are proofs that the partial element f (with cofibrant domain of definition φ) extends to the totally defined element a. Finally, we write $a \sim a' \triangleq \{p : \mathtt{I} \to A \mid p\, \mathtt{0} = a \wedge p\, \mathtt{1} = a'\}$ for the type of *paths* from $a : A$ to $a' : A$.

As an example of the use of this language we now reproduce the internal definition of a (small) fibration [13, Definition 5.7]:

▶ **Definition 5.1** (CCHM fibrations). A *CCHM fibration* (A, α) over a type $\Gamma : \mathcal{U}$ is a family $A : \Gamma \to \mathcal{U}$ equipped with a fibration structure $\alpha : \mathtt{isFib}\, A$, where $\mathtt{isFib} : \{\Gamma : \mathcal{U}\}(A : \Gamma \to \mathcal{U}) \to \mathcal{U}$ is defined by

$$\mathtt{isFib}\, \{\Gamma\}\, A \triangleq (e : \mathtt{\{0,1\}})(p : \mathtt{I} \to \Gamma) \to \mathtt{Comp}\, e\, (A \circ p)$$

Here $\mathtt{Comp} : (e : \mathtt{\{0,1\}})(A : \mathtt{I} \to \mathcal{U}) \to \mathcal{U}$ is the type of *composition structures* for $\mathtt{I}$-indexed families:

$$\begin{aligned}\mathtt{Comp}\, e\, A \triangleq\ & (\varphi : \mathtt{Cof})(f : [\varphi] \to \Pi_{\mathtt{I}} A) \to \\ & \{a_0 : A\, e \mid (\varphi, f) \mathbin{@} e \nearrow a_0\} \to \{a_1 : A\, \overline{e} \mid (\varphi, f) \mathbin{@} \overline{e} \nearrow a_1\}\end{aligned}$$

where $(\varphi, f) \mathbin{@} e$ is an abbreviation for the term $\lambda u : [\varphi].\, f\, u\, e$ of type $[\varphi] \to A\, e$, and where $\overline{\mathtt{0}} = \mathtt{1}$ and $\overline{\mathtt{1}} = \mathtt{0}$.

We write $\mathtt{Fib}\ \Gamma \triangleq (A : \Gamma \to \mathcal{U}) \times \mathtt{isFib}\ A$ for the type of fibrations over $\Gamma : \mathcal{U}$ and recall that fibrations can be reindexed $(A, \alpha)[\gamma] = (A \circ \gamma, \alpha[\gamma]) : \mathtt{Fib}\ \Delta$ for $\gamma : \Delta \to \Gamma$.

5.1.3 Paths between fibrations

We work in the internal type theory of the cubical sets topos wherever possible, however this approach does have its limitations. In particular this internal approach is unable to describe type theoretic universes [13, Remark 7.5]. Therefore we will not be able to construct elements of the identity type on the universe (the target type of axioms (1)-(3)). Instead, we work with an (externally) equivalent notion of equality between types.

▶ **Definition 5.2** (Path equality between fibrations)**.** Define the type of paths between CCHM fibrations $\_ \sim_{\mathcal{U}} \_ : \{\Gamma : \mathcal{U}\} \to \mathtt{Fib}\,\Gamma \to \mathtt{Fib}\,\Gamma \to \mathcal{U}_1$ by

$$(A,\alpha) \sim_{\mathcal{U}} (B,\beta) \triangleq \{(P,\rho) : \mathtt{Fib}(\Gamma \times \mathtt{I}) \mid (P,\rho)[\langle id, \mathtt{0}\rangle] = (A,\alpha) \wedge (P,\rho)[\langle id, \mathtt{1}\rangle] = (B,\beta)\}$$

To understand why this notion of path is equivalent to the usual notion recall that the universe construction in [8] is given by the usual Hofmann-Streicher universe construction for presheaf categories [11]. This means that there exists a type $\mathcal{U} \in \mathtt{Ty}(\Gamma)$ for all Γ given by $\mathcal{U}(I,\rho) \triangleq \mathtt{FTy}_0(\mathrm{y}I)$ where $\mathrm{y}I$ denotes the Yoneda embedding of I. Every (small) fibrant type $A \in \mathtt{FTy}_0(\Gamma)$ has a code $\ulcorner A \urcorner \in Ter(\Gamma \vdash \mathcal{U})$ and every $a \in Ter(\Gamma \vdash \mathcal{U})$ encodes a type $El\, a \in \mathtt{FTy}_0(\Gamma)$ such that $El\,(\ulcorner A \urcorner) = A$ and $\ulcorner El\, a \urcorner = a$ for all a and A.

Now consider the following: externally, a path $P : A \sim_{\mathcal{U}} B$ corresponds to a fibration $P \in \mathtt{FTy}_0(\Gamma.\mathbb{I})$ such that $P[\langle id, 0\rangle] = A$ and $P[\langle id, 1\rangle] = B$ for some $\Gamma \in \hat{\mathcal{C}}$ and $A, B \in \mathtt{FTy}_0(\Gamma)$. From this data we can construct $p \in Ter(\Gamma \vdash Path\, \mathcal{U}\, \ulcorner A \urcorner\, \ulcorner B \urcorner)$ like so:

$$p(\rho) \triangleq \langle i \rangle\, \ulcorner P \urcorner (\rho s_i, i)$$

for $I \in \mathcal{C}, \rho \in \Gamma(I)$, where $s_i : I, i \to I$ is the obvious inclusion of I in I, i. Note that this does define a path with the correct endpoints since substituting 0 for i we get:

$$(\ulcorner P \urcorner (\rho, 0))f = P(\rho f, 0f) = P(\rho f, 0) = A(\rho f) = (\ulcorner A \urcorner(\rho))f$$

for all $f : J \to I$. The case for $i = 1$ is similar.

Conversely, given $p \in Ter(\Gamma \vdash Path\, \mathcal{U}\, \ulcorner A \urcorner\, \ulcorner B \urcorner)$ we can define $P \in \mathtt{FTy}_0(\Gamma.\mathbb{I})$ with the required properties like so:

$$P(\rho, i) \triangleq (p\ \rho\ i) id_I$$

for $I \in \mathcal{C}, \rho \in \Gamma(I), i \in \mathbb{I}$. Again, note that this has the correct properties, e.g. at 0:

$$P[\langle id, 0\rangle]\ \rho = P(\rho, 0) = (p\ \rho\ 0)\ id_I = (\ulcorner A \urcorner\ \rho) id_I = (El \ulcorner A \urcorner)\rho = A\rho$$

for all $\rho \in \Gamma(I)$. It is easily checked that these two constructions are mutual inverses. Therefore the data described by $\_ \sim_{\mathcal{U}} \_$ corresponds exactly to the data required to describe a path in the universe.

5.1.4 The realignment lemma

Next, we introduce a technical lemma that will be needed in the following sections. For readers familiar with the cubical sets model, it is interesting to note that this is the only place where we use the fact that cofibrant propositions are closed under $\mathtt{I}$-indexed $\forall$ [8, Section 4.1].

▶ **Lemma 5.3** (Realignment lemma)**.** *Given $\Gamma : \mathcal{U}$ and $\Phi : \Gamma \to \mathtt{Cof}$, let $\iota : \Gamma|\Phi \rightarrowtail \Gamma$ be the first projection. For any $A : \Gamma \to \mathcal{U}$, $\beta : \mathtt{isFib}(A \circ \iota)$ and $\alpha : \mathtt{isFib}\, A$, there exists a composition structure $\mathtt{realign}(\Phi, \beta, \alpha) : \mathtt{isFib}\, A$ such that $\beta = \mathtt{realign}(\Phi, \beta, \alpha)[\iota]$.*

Proof. By [13, Theorem 6.13], in which we define $\mathtt{realign}(\Phi, \beta, \alpha)$ by

$$\mathtt{realign}(\Phi, \beta, \alpha)\, e\, p\, \psi\, f\, a \triangleq \alpha\, e\, p\, (\psi \vee (\forall(i : \mathtt{I}).\, \Phi\,(p\, i)))\, (f \cup f')\, a \tag{1}$$

where $f' : [\forall(i : \mathtt{I}).\, \Phi\,(p\, i)] \to \Pi_{\mathtt{I}}(A \circ p)$ is given by $f'\, u \triangleq \mathtt{fill}\ e\, \beta\, (\lambda i \to (p, u\, i))\, \psi\, f\, a$. ◀

In words, given a fibrant type A and an alternative composition structure defined only on some restriction of A, then we can *realign* the original composition structure so that it agrees with the alternative on that restriction.

Note that this construction is stable under reindexing in the following sense: given $\gamma : \Delta \to \Gamma$, $\Phi : \Gamma \to \mathtt{Cof}$, $A : \Gamma \to \mathcal{U}$, $\beta : \mathtt{isFib}(A \circ \iota)$ and $\alpha : \mathtt{isFib}\, A$ then,

$$\begin{aligned}
&\mathtt{realign}(\Phi, \beta, \alpha)[\gamma]\, e\, p\, \psi\, f\, a \\
&\quad = \mathtt{realign}(\Phi, \beta, \alpha)\, e\, (\gamma \circ p)\, \psi\, f\, a \\
&\quad = \alpha\, e\, (\gamma \circ p)\, (\psi \vee (\forall (i : \mathtt{I}).\, \Phi\, ((\gamma \circ p)\, i)))\, (f \cup \mathtt{fill}\ e\, \beta\, (\lambda i \to (\gamma \circ p, u\, i))\, \psi\, f\, a)\, a \\
&\quad = \alpha[\gamma]\, e\, p\, (\psi \vee (\forall (i : \mathtt{I}).\, (\Phi \circ \gamma)(p\, i)))\, (f \cup \mathtt{fill}\ e\, \beta[\langle \gamma, id \rangle]\, (\lambda i \to (p, u\, i))\, \psi\, f\, a)\, a \\
&\quad = \mathtt{realign}(\Phi \circ \gamma, \beta[\langle \gamma, id \rangle], \alpha[\gamma])\, e\, p\, \psi\, f\, a
\end{aligned}$$

Therefore we have

$$\mathtt{realign}(\Phi, \beta, \alpha)[\gamma] = \mathtt{realign}(\Phi \circ \gamma, \beta[\langle \gamma, id \rangle], \alpha[\gamma])$$

5.1.5 Fibrations are closed under isomorphism

▶ **Definition 5.4** (Strict isomorphism)**.** A *strict isomorphism* between two objects $A, B : \mathcal{U}$ is a pair (f, g) where $f : A \to B$ and $g : B \to A$ such that $g \circ f = id$ and $f \circ g = id$. We write $(f, g) : A \cong B$.

This notion lifts to both families and fibrations, and we overload the notation $\_ \cong \_$ like so: when $A, B : \Gamma \to \mathcal{U}$ then we take $A \cong B$ to mean $(x : \Gamma) \to A\, x \cong B\, x$ and when $A, B : \mathtt{Fib}\ \Gamma$ then we take $A \cong B$ to mean $\mathtt{fst}\, A \cong \mathtt{fst}\, B$.

▶ **Lemma 5.5.** *Given a family $A : \Gamma \to \mathcal{U}$ and a fibration $(B, \beta) : \mathtt{Fib}\ \Gamma$, such that $A \cong B$, then we can construct an α such that $(A, \alpha) : \mathtt{Fib}\ \Gamma$.*

Proof. Assume that we are given A and (B, β) as above with an isomorphism $\langle f, g \rangle : A \cong B$. We can then define a composition structure for A as follows:

$$\alpha\ e\ p\ \varphi\ q\ a_0 \triangleq g\ (p\ \overline{e})\ (\beta\ e\ p\ \varphi\ (\lambda u\, i \to f\ (p\, i)\ (q\ u\ i))\ (f\ (p\, e)\ a_0))$$

This construction has the required property that, given $u : [\varphi]$:

$$\begin{aligned}
\alpha\ e\ p\ \varphi\ q\ a_0 &= g\ (p\ \overline{e})\ (\beta\ e\ p\ \varphi\ (\lambda u\, i \to f\ (p\, i)\ (q\ u\ i))\ (f\ (p\, e)\ a_0)) \\
&= g\ (p\ \overline{e})\ (f\ (p\ \overline{e})\ (q\ u\ \overline{e})) \\
&= q\ u\ \overline{e}
\end{aligned}$$

Hence $(A, \alpha) : \mathtt{Fib}\ \Gamma$. ◀

Note that this proof only uses the fact that $g\, x \circ f\, x = id$ and so in fact the lemma holds more generally in the case where $\langle f, g \rangle$ is just a section-retraction pair rather than a full isomorphism. Although here will we only use it in the context of isomorphisms.

5.1.6 Strictification

▶ **Theorem 5.6.** *There exists a term:*

$$\begin{aligned}
\mathtt{strictify} : \{\varphi : \mathtt{Cof}\}(A : [\varphi] \to \mathcal{U})(B : \mathcal{U})(s : (u : [\varphi]) \to (A\, u \cong B)) \to \\
(B' : \mathcal{U}) \times \{s' : B' \cong B \mid \forall (u : [\varphi]).\, A\, u = B' \wedge s\, u = s'\}
\end{aligned}$$

In words, this says that given any object $B : \mathcal{U}$ and any cofibrant partial object $A : [\varphi] \to \mathcal{U}$ such that A is isomorphic to B everywhere it is defined, then one can can construct a new object $B' : \mathcal{U}$ which extends A, is isomorphic to B, and this isomorphism extends the original isomorphism.

Proof. See [13, Theorem 8.4] for a proof that this property holds in the cubical sets model, and more generally in many other models (classically, in all presheaf models). The construction depends on the fact that, externally, cofibrant propositions are pointwise decidable and so we can simply define (B', s') to be either (A, s) or (B, id) by case analysis on φ at each point. ◀

We now lift this strictification property from objects to fibrations.

▶ **Theorem 5.7.** *Given $\Gamma : \mathcal{U}$ and $\Phi : \Gamma \to \mathtt{Cof}$, a partial fibration $A : \mathtt{Fib}(\Gamma|\Phi)$ and a total fibration $B : \mathtt{Fib}\ \Gamma$ with $iso : A \cong B[\iota]$, we can construct a new type and isomorphism:*

$$A' : \mathtt{Fib}(\Gamma) \qquad \textit{and} \qquad iso' \ : \ A' \cong B$$

such that

$$A'[\iota] = A \qquad \textit{and} \qquad iso' \circ \iota = iso$$

where ι is the inclusion $\Gamma|\Phi \rightarrowtail \Gamma$.

Proof. Given $\Gamma : \mathcal{U}$, $\Phi : \Gamma \to \mathtt{Cof}$, $(A, \alpha) : \mathtt{Fib}(\Gamma|\Phi)$ and $(B, \beta) : \mathtt{Fib}\ \Gamma$ with $iso : A \cong B \circ \iota$, we define A', iso' as:

$$A'\, x \triangleq \mathtt{fst}(\mathtt{strictify}(A(x, \_), B\, x, iso(x, \_)))$$

$$iso'\, x \triangleq \mathtt{snd}(\mathtt{strictify}(A(x, \_), B\, x, iso(x, \_)))$$

Now consider the equalities that are required to hold. From the properties of `strictify` we already have that $A' \circ \iota = A$ and $iso' \circ \iota = iso$. Therefore we just need to define a composition structure $\alpha' : \mathtt{isFib}\ A'$ such that $\alpha'[\iota] = \alpha$.

Since $A' \cong B$ and $\beta : \mathtt{isFib}\ B$ we can use Lemma 5.5 to deduce that A' has a composition structure, which we call α'_{pre}. We then define $\alpha' \triangleq \mathtt{realign}(\Phi, \alpha, \alpha'_{pre})$ using Lemma 5.3. ◀

5.1.7 Misaligned paths between fibrations

We now introduce a new relation between fibrations which we call a misaligned path. This is similar to the notion of path between fibrations introduced in Definition 5.2, except that rather than being equal to A and B at the endpoints, the path only need be isomorphic to A and B at the endpoints.

▶ **Definition 5.8** (Misaligned path equality between fibrations)**.** Define the type of misaligned paths between CCHM fibrations $\_ \sim_{\cong} \_ : \{\Gamma : \mathcal{U}\} \to \mathtt{Fib}\,\Gamma \to \mathtt{Fib}\,\Gamma \to \mathcal{U}_1$ by

$$(A, \alpha) \sim_{\cong} (B, \beta) \triangleq ((P, \rho) : \mathtt{Fib}(\Gamma \times \mathtt{I})) \times (A \cong P \circ \langle id, \mathtt{0} \rangle) \times (B \cong P \circ \langle id, \mathtt{1} \rangle)$$

We can show that every misaligned path can be improved to a regular path between fibrations. First, we introduce a new construction on fibrations.

▶ **Definition 5.9.** Given fibrations $A, B : \mathtt{Fib}\ \Gamma$ we define a new fibration

$$A \veebar B \ : \ \mathtt{Fib}((\Gamma \times \mathtt{I})|\Phi) \qquad \text{where} \qquad \Phi(x,i) \triangleq (i = \mathtt{0}) \vee (i = \mathtt{1})$$

given by $(A,\alpha) \veebar (B,\beta) \triangleq (C,\gamma)$ where

$$C : (\Gamma \times \mathtt{I})|\Phi \to \mathcal{U}$$
$$C\,((x,i),u) \triangleq ((\lambda\_ : [i = \mathtt{0}] \to A\,x) \cup (\lambda\_ : [i = \mathtt{1}] \to B\,x))\,u$$

Here C is a sort of disjoint union of the families A and B, observing that $(\Gamma \times \mathtt{I})|\Phi \cong \Gamma + \Gamma$ then we can think of C as essentially being $[A,B] : \Gamma + \Gamma \to \mathcal{U}$.

To see that C is fibrant we observe that the interval $\mathtt{I}$ is internally connected in the sense of $\mathtt{ax_1}$ in [13, Figure 1]. This means that any path $p : \mathtt{I} \to (\Gamma \times \mathtt{I})|\Phi$ must either factor as $p = \langle p', \mathtt{0}, * \rangle$ or as $p = \langle p', \mathtt{1}, * \rangle$. Therefore any composition problem for C must lie either entirely in A, in which case we use α to construct a solution, or entirely in B, in which case we use β. For further detail we refer the reader to [13, Theorem 7.3] where the family C occurs as an intermediate construction.

▶ **Definition 5.10.** Given $\Gamma : \mathcal{U}$, $A, B : \mathtt{Fib}\ \Gamma$ and $D : \mathtt{Fib}(\Gamma \times \mathtt{I})$ with isomorphisms $iso_0 : A \cong D[\langle id, \mathtt{0}\rangle]$ and $iso_1 : B \cong D[\langle id, \mathtt{1}\rangle]$ then define $iso_0 \veebar iso_1 : A \veebar B \cong D[\iota]$ as follows. Given $(x,i,u) : (\Gamma \times \mathtt{I})|\Phi$:

$$(iso_0 \veebar iso_1)\,(x,i,u) : (\mathtt{fst}(A \veebar B))\,(x,i,u) \to (\mathtt{fst}\ D)\,(x,i)$$
$$(iso_0 \veebar iso_1)\,(x,i,u) \triangleq \begin{cases} iso_0\,x & \text{when } u : [i = \mathtt{0}] \\ iso_1\,x & \text{when } u : [i = \mathtt{1}] \end{cases}$$

Observe that for all $A, B : \mathtt{Fib}\ \Gamma$ we have $(A \veebar B)[\langle id, \mathtt{0}, * \rangle] = A$ and $(A \veebar B)[\langle id, \mathtt{1}, * \rangle] = B$, and for all $iso_0 : A \cong D[\langle id, \mathtt{0}\rangle]$ and $iso_1 : B \cong D[\langle id, \mathtt{1}\rangle]$ we have $(iso_0 \veebar iso_1) \circ \langle id, \mathtt{0}, * \rangle = iso_0$ and $(iso_0 \veebar iso_1) \circ \langle id, \mathtt{1}, * \rangle = iso_1$. We now use this construct to show the following result:

▶ **Lemma 5.11.** *There exists a function*

$$\mathtt{improve} : \{\Gamma : \mathcal{U}\}\{A\ B : \mathtt{Fib}\,\Gamma\} \to A \sim_{\cong} B \to A \sim_{\mathcal{U}} B$$

Proof. Take $\Gamma : \mathcal{U}$, $A, B : \mathtt{Fib}\,\Gamma$ and $(P, iso_0, iso_1) : A \sim_{\cong} B$ and observe that $iso_0 \veebar iso_1 : A \veebar B \cong P[\iota]$. Therefore we can use Theorem 5.7 to strictify P in order to get $P' : \mathtt{Fib}(\Gamma \times \mathtt{I})$ such that $P'[\iota] = A \veebar B$, where ι is the restriction $(\Gamma \times \mathtt{I})|\Phi \to \Gamma \times \mathtt{I}$. Now consider reindexing P' along $\langle id, \mathtt{0}\rangle : \Gamma \to \Gamma \times \mathtt{I}$ we get:

$$P'[\langle id, \mathtt{0}\rangle] = P'[\iota \circ \langle id, \mathtt{0}, * \rangle] = P'[\iota][\langle id, \mathtt{0}, * \rangle] = (A \veebar B)[\langle id, \mathtt{0}, * \rangle] = A$$

and similarly $P'[\langle id, \mathtt{1}\rangle] = B$. Therefore we have $P' : A \sim_{\mathcal{U}} B$ as required. ◀

5.1.8 Function extensionality

As discussed previously, function extensionality holds straightforwardly in any type theory which includes an interval object/type with certain computational properties, cf. [15, Lemma 6.3.2]. See [13, Remark 5.16] and [8, Section 3.2] for a proof in the case of cubical type theory.

5.1.9 Axioms (1), (2), (4) and (5)

As discussed previously, we can satisfy axioms (1) and (2) by showing that there is a way to construct paths between strictly isomorphic (fibrant) types $A, B : \mathtt{Fib}\ \Gamma$.

▶ **Theorem 5.12.** *Given fibrations $A, B : \mathtt{Fib}\ \Gamma$ with $iso : A \cong B$ we can construct a path $\mathtt{isopath}(iso) : A \sim_{\mathcal{U}} B$.*

Proof. Given A, B, f, g as above, let $B' \triangleq B[\mathtt{fst}] : \mathtt{Fib}(\Gamma \times \mathtt{I})$ and note that $iso : A \cong B'[\langle id, \mathtt{0}\rangle]$ and $id : B \cong B'[\langle id, \mathtt{1}\rangle]$ where id is the obvious isomorphism $B \cong B$. Therefore we can define

$$\mathtt{isopath}(iso) \triangleq \mathtt{improve}(B[\mathtt{fst}], iso, id) : A \sim_{\mathcal{U}} B$$

as required. Note that, in this case, `improve` will in fact only improve $B[\mathtt{fst}]$ at `0`, since at `1` we improve along the identity, which does nothing. ◀

▶ **Corollary 5.13.** *Axioms (1) and (2) hold in the cubical sets model.*

Proof. The obvious isomorphisms $A \cong A \times 1$ and $\sum_{a:A} \sum_{b:B} C\ a\ b \cong \sum_{b:B} \sum_{a:A} C\ a\ b$ are both clearly strict isomorphisms in the sense of Definition 5.4. Therefore we can construct the required paths $A \sim_{\mathcal{U}} (A \times 1)$ and $(\sum_{a:A} \sum_{b:B} C\ a\ b) \sim_{\mathcal{U}} (\sum_{b:B} \sum_{a:A} C\ a\ b)$. Hence axioms (1) and (2) hold.

Note that in order interpret axioms (1) and (2) using `isopath` we need to know that `isopath` is stable under reindexing (substitution in the type theory). This will be the case because most of the constructions used to define it (strictification, closure under isomorphism, etc) are all performed fiberwise and hence will be stable under reindexing. The only exception is the realignment lemma, which redefines the entire composition structure. However, we previously showed `realign` to be stable under reindexing. Therefore `isopath` will also be stable under reindexing. ◀

We have seen that we can easily satisfy axioms (1) and (2) in the cubical sets model. However, we also need to know what happens when we coerce along these equalities. This can be stated in general for any strictly isomorphic types.

▶ **Theorem 5.14.** *Given fibrations $(A, \alpha), (B, \beta) : \mathtt{Fib}\ \Gamma$ with $\langle f, g\rangle : A \cong B$, coercing along $\mathtt{isopath}(\langle f, g\rangle)$ is (propositionally) equal to applying f.*

Proof. Take $(A, \alpha), (B, \beta), f, g$ as above and let $(P, \rho) = \mathtt{isopath}(\langle f, g\rangle)$. By unfolding the constructions used we can see that ρ was obtained by realigning some ρ_{pre}, which in turn was obtained by transferring $\beta[\mathtt{fst}]$ across the isomorphism:

$$iso'\,(x, i) = \mathtt{snd}(\mathtt{strictify}((A, \beta) \veebar (B, \beta)(x, i, \_), B\,x, (\langle f, g\rangle \veebar id)\,(x, i, \_))) : P\,x \cong B\,x$$

Now consider arbitrary $x : \Gamma, a_0 : A\ x$ and note that

$$iso'\,(x, \mathtt{0}) = (\langle f, g\rangle \veebar id)\,(x, \mathtt{0}) = \langle f, g\rangle\ x = (f\,x, g\,x)$$

and

$$iso'\,(x, \mathtt{1}) = (\langle f, g\rangle \veebar id)\,(x, \mathtt{1}) = (id, id)$$

Now calculate:

$$
\begin{aligned}
&\texttt{coerce}\ \texttt{isopath}(\langle f, g\rangle)\ x\ a_0 \\
&\quad = \rho\ \texttt{0}\ \langle x, id\rangle\ \bot\ \texttt{elim}_{\emptyset}\ a_0 && \text{by unfolding definitions}^2 \\
&\quad = \rho_{pre}\ \texttt{0}\ \langle x, id\rangle\ (\forall i.(i = \texttt{0} \vee i = \texttt{1}))\ q\ a_0 && \text{by Lemma 5.3, for some } q \\
&\quad = \rho_{pre}\ \texttt{0}\ \langle x, id\rangle\ \bot\ \texttt{elim}_{\emptyset}\ a_0 && \text{by definition of } \forall \\
&\quad = \texttt{snd}(iso'(x, \texttt{1}))\ (\beta\ \texttt{0}\ \langle x, id\rangle\ \bot\ \texttt{elim}_{\emptyset}\ (\texttt{fst}(iso'(x, \texttt{0}))\ a_0)) && \text{by Lemma 5.5} \\
&\quad = \beta\ \texttt{0}\ \langle x, id\rangle\ \bot\ \texttt{elim}_{\emptyset}\ (\texttt{fst}(iso'(x, \texttt{0}))\ a_0)) && \text{since } \texttt{snd}(iso'(x, \texttt{1})) = id \\
&\quad = \beta\ \texttt{0}\ \langle x, id\rangle\ \bot\ \texttt{elim}_{\emptyset}\ (f\ x\ a_0) && \text{since } \texttt{fst}(iso'(x, \texttt{0})) = f\ x
\end{aligned}
$$

Since this is merely a trivial/empty composition applied to $f\ x\ a_0$ we can construct a path from $f\ x\ a_0$ to $\texttt{coerce}\ \texttt{isopath}(\langle f, g\rangle)\ x\ a_0$ like so:

$$\texttt{fill}\ \texttt{0}\ \beta\ \langle x, id\rangle\ \bot\ \texttt{elim}_{\emptyset}\ (f\ x\ a_0) : f\ x\ a_0 \sim \texttt{coerce}\ \texttt{isopath}(\langle f, g\rangle)\ x\ a_0$$

Therefore, coercing along $isopath(\langle f, g\rangle)$ is always propositionally equal to applying f. ◀

▶ **Corollary 5.15.** *Axioms (4) and (5) hold in the cubical sets model (for the terms constructed in Corollary 5.13).*

Proof. By Theorem 5.14. ◀

5.1.10 Axiom (3)

In light of the previous section, the only axiom remaining is axiom (3). Our goal here is, given a contractible fibration $A : \texttt{Fib}\ \Gamma$, to define a path $A \sim_{\mathcal{U}} 1$. Note that, for any $\Gamma : \mathcal{U}$, there exists a unique fibration structure $!_1$ such that $(\lambda\_ \to 1, !_1) : \texttt{Fib}(\Gamma)$. Therefore we will ambiguously write $1 : \texttt{Fib}(\Gamma)$ for the pair $(\lambda\_ \to 1, !_1)$.

▶ **Definition 5.16** (The contraction of a family)**.** Given a family $A : \Gamma \to \mathcal{U}$ we define the contraction of A as

$$C_A : \Gamma \times \texttt{I} \to \mathcal{U}$$
$$C_A(x, i) \triangleq [i = \texttt{0}] \to A(x)$$

We now need to show that C_A is fibrant when A is both fibrant and contractible. First, we restate the property of being contractible (Definition 2.1) in the internal type theory.

▶ **Definition 5.17.** A type A is said to be *contractible* if it has a centre of contraction $a_0 : A$ and every element $a : A$ is propositionally equal to a_0, that is, there exists a path $a_0 \sim a$. Therefore a type is contractible if $\texttt{Contr}\ A$ is inhabited, where $\texttt{Contr} : \mathcal{U} \to \mathcal{U}$ is defined by

$$\texttt{Contr}\ A \triangleq (a_0 : A) \times ((a : A) \to a_0 \sim a)$$

We say that a family $A : \Gamma \to \mathcal{U}$ is contractible if each of its fibres is and abusively write

$$\texttt{Contr}\ A \triangleq (x : \Gamma) \to \texttt{Contr}(A\ x)$$

[2] Note that there are different ways to interpret `coerce` in the model. This interpretation is not in general the same as the one obtained by directly interpreting Definition 3.1. However, the two interpretations will always be path equal in the model (the other interpretation will have more trivial/empty compositions), and so the result still holds when using the other interpretation.

Next we recall the notion of an extension structure [13, Definition 6.4].

▶ **Definition 5.18** (Extension structures)**.** The type of extension structures, $\mathtt{Ext} : \mathcal{U} \to \mathcal{U}$, is given by

$$\mathtt{Ext}\, A \triangleq (\varphi : \mathtt{Cof})(f : [\varphi] \to A) \to \{a : A \mid (\varphi, f) \nearrow a\}$$

Having an extension structure for a type $A : \mathcal{U}$ allows us to extend any partial element of A to a total element. As before we say that a family $A : \Gamma \to \mathcal{U}$ has an extension structure if each of its fibres do, and write

$$\mathtt{Ext}\, A \triangleq (x : \Gamma) \to \mathtt{Ext}(A\, x)$$

▶ **Lemma 5.19.** *Any family $A : \Gamma \to \mathcal{U}$ that is both fibrant and contractible is also extendable in the sense of Defintion 5.18.*

Proof. By [13, Lemma 6.6]. ◀

Now we can construct a fibrancy structure for C_A as follows:

▶ **Theorem 5.20.** *If $(A, \alpha) : \mathtt{Fib}\ \Gamma$ is contractible then we can construct a composition structure for C_A.*

Proof. Take $(A, \alpha) : \mathtt{Fib}\ \Gamma$ as above. Since A is both fibrant and contractible we can construct an extension structure $\epsilon : \mathtt{Ext}\, A$. We can then define a composition structure $c_\alpha : \mathtt{isFib}(C_A)$ as follows, given

$$e : \{\mathtt{0},\mathtt{1}\} \qquad p : \mathtt{I} \to \Gamma \times \mathtt{I} \qquad \varphi : \mathtt{Cof} \qquad f : [\varphi] \to \Pi_{\mathtt{I}}(C_A \circ p) \qquad c_0 : C_A\,(p\, e)$$

we must define as term of type $C_A\,(p\,\overline{e})$. Since this type is defined to be $[\mathtt{snd}(p\,\overline{e}) = \mathtt{0}] \to A$, we define the composition like so:

$$(c_\alpha\ e\ p\ \varphi\ f\ c_0)\ u \triangleq \epsilon\ (p\,\overline{e})\ \varphi\ (\lambda v \to f\, v\, \overline{e}\, u)$$

Given $v : [\varphi]$ we have:

$$c_\alpha\ e\ p\ \varphi\ f\ c_0 = \lambda u \to \epsilon\ (p\,\overline{e})\ \varphi\ (\lambda v \to f\, v\, \overline{e}\, u) = \lambda u \to f\, v\, \overline{e}\, u = f\, v\, \overline{e}$$

as required. Therefore we have a defined a valid composition operation for C_A. ◀

▶ **Theorem 5.21.** *There exists a function*

$$\mathtt{contract} : \{\Gamma : \mathcal{U}\}(A : \mathtt{Fib}\ \Gamma) \to \mathtt{Contr}\ A \to A \sim_{\mathcal{U}} 1$$

Proof. Given $\Gamma : \mathcal{U}$, $(A, \alpha) : \mathtt{Fib}\ \Gamma$ and $\epsilon : \mathtt{Contr}\ A$, we obverse that

$$C_A[\langle id, \mathtt{0}\rangle](x) = C_A(x, \mathtt{0}) = [\mathtt{0} = \mathtt{0}] \to A(x) \cong 1 \to A(x) \cong A(x)$$

and

$$C_A[\langle id, \mathtt{1}\rangle](x) = C_A(x, \mathtt{1}) = [\mathtt{1} = \mathtt{0}] \to A(x) \cong \emptyset \to A(x) \cong 1$$

Therefore we have $((C_A, c_\alpha), iso_A, iso_1) : A \sim_{\cong} 1$ where $iso_A : A \cong C_A[\langle id, \mathtt{0}\rangle]$ and $iso_1 : 1 \cong C_A[\langle id, \mathtt{1}\rangle]$ are the obvious isomorphisms indicated above. Hence we can define

$$\mathtt{contract}((A, \alpha), \epsilon) \triangleq \mathtt{improve}((C_A, c_\alpha), iso_A, iso_1) : (A, \alpha) \sim_{\mathcal{U}} 1$$

as required. ◀

▶ **Corollary 5.22.** *Cubical type theory with the cubical sets model supports axiom (3).*

As in Corollary 5.13 we need to check that `contract` is stable under reindexing (substitution). This holds for the same reasons as before, namely that the only non fibrewise construction used in the definition of `contract` is `realign` which we previously showed to be stable under reindexing.

6 An application to an open problem in type theory

In Section 2 we defined *funext* to be the principle which says that two functions $f, g : \prod_{x:A} B(x)$ are equal if they are pointwise equal: $f \approx g \triangleq \prod_{x:A} f(x) = g(x)$. That is, we assumed the existence of a term:

$$\mathit{funext}_{i,j} : \prod_{A:U_i} \prod_{B:A\to U_j} \prod_{f,g:\Pi_{x:A}B(x)} f \approx g \to f = g$$

for all universe levels i, j. This is similar to the statement of naive univalence, UA, from Definition 3.3 and we call this principle naive function extensionality.

As with proper univalence (Definition 3.2), we could have instead stated that the canonical map $\mathit{happly} : (f = g) \to f \approx g$ is an equivalence. In fact, these two formulations turn out to be equivalent.

▶ **Theorem 6.1** (due to Voevodsky)**.** *Naive function extensionality is logically equivalent to the proper function extensionality axiom. That is, the existence of a term:*

$$\mathit{funext}_{i,j} : \prod_{A:U_i} \prod_{B:A\to U_j} \prod_{f,g:\Pi_{x:A}B(x)} f \approx g \to f = g$$

is logically equivalent to the statement that, for all types $A : U_i$, $B : A \to U_j$ and maps $f, g : \prod_{x:A} B(x)$, the map $\mathit{happly} : (f = g) \to (f \approx g)$ is an equivalence.

Proof. For the forwards direction: assuming *funext* as above, it is easy to derive a proof of *weak function extensionality* [15, Definition 4.9.1]. This in turn implies the proper function extensionality axiom by [15, Theorem 4.9.5]. The reverse direction follows trivially. ◀

Compare this result with Theorem 3.5 where we saw that naive univalence with a computation rule is logically equivalent to the proper univalence axiom. In the case of function extensionality we did not need to assume any sort of computation rule about *funext*. Therefore an obvious question is whether this computation rule is in fact necessary in the case of univalence, or whether, as is the case with function extensionality, it is in fact redundant.

▶ **Conjecture 6.2.** *Naive univalence implies the proper univalence axiom. That is, given UA_i, it follows that for all types $A, B : U_i$ the map $\mathit{idtoeqv} : (A = B) \to (A \simeq B)$ is an equivalence.*

To the authors' best knowledge the status of Conjecture 6.2 is currently unknown. It is certainly not inconsistent since there are models where naive univalence fails to hold, such as the *Set*-valued model [10], and models where full univalence holds, such as the cubical sets model [8]. However it is not clear whether Conjecture 6.2 is either a theorem of type theory, cf. the case with function extensionality, or whether there are models which validate UA but which do not validate the proper univalence axiom.

The work presented here may offer an approach to tackling this problem, by reducing it to the following:

▶ **Conjecture 6.3.** *Axioms (1)-(3) imply axioms (4)-(5), for possibly modified unit and flip. That is, if for all $A, B : U_i$, $C : A \to B \to U_i$ we have:*

$$A = \sum_{a:A} 1 \qquad \sum_{a:A}\sum_{b:B} C\ a\ b = \sum_{b:B}\sum_{a:A} C\ a\ b \qquad isContr(A) \to A = 1$$

then there exist terms unit and flip, with types as in Table 1, for which the following equalities hold:

$$\texttt{coerce}\ \mathit{unit}\ a = (a, *) \qquad \texttt{coerce}\ \mathit{flip}\ (a, b, c) = (b, a, c)$$

for all $a : A$, $b : B$ and $c : C\,a\,b$.

▶ **Theorem 6.4.** *In the presence of function extensionality, Conjecture 6.2 and Conjecture 6.3 are logically equivalent.*

Proof. For the forwards direction, assume function extensionality, 6.2 and axioms (1)-(3). By Theorem 4.1 we deduce that naive univalence, UA_i, holds. Therefore by our assumption of 6.2 we deduce the proper univalence axiom for U_i. Hence, by Corollary 4.4, we deduce axioms (1)-(5) (possibly with different proof terms than our existing assumptions of axioms (1)-(3)). Therefore the conclusion of 6.3 holds.

For the reverse direction, assume function extensionality, 6.3 and naive univalence. By Theorem 4.1 we deduce that axioms (1)-(3) hold. Therefore by our assumption of 6.3 we deduce axioms (4)-(5) also hold. Hence, by Corollary 4.4, we deduce the proper univalence axiom. ◀

This result may be useful in tackling the open question of whether Conjecture 6.2 is a theorem of type theory, or whether there are in fact models in which it does not hold. This is because finding models where the conclusions of Conjecture 6.3 do not hold given the assumptions, or showing that no such models exist, seems an easier task. For example, consider the case where the first conclusion fails, that is, where $\texttt{coerce}\ \mathit{unit}\ a \neq (a, *)$ for some $A : U$ and $a : A$. If this is the case then we have $\texttt{fst} \circ (\texttt{coerce}\ \mathit{unit}) : \{A : U\} \to A \to A$ which is not equal to the identity function. We note that the existence of such a term has interesting consequences relating to parametricity and excluded middle [7], and potentially informs our search about the type of models which might invalidate Conjecture 6.2. However, we leave further investigation of this problem to future work.

References

1 Agda Project. URL: `http://wiki.portal.chalmers.se/agda`.

2 C. Angiuli, G. Brunerie, T. Coquand, K.-B. Hou (Favonia), R. Harper, and D. R. Licata. Cartesian cubical type theory (preprint). 2017. URL: `https://github.com/dlicata335/cart-cube/blob/master/cart-cube.pdf`.

3 S. Awodey. A cubical model of homotopy type theory. *arXiv preprint arXiv:1607.06413*, 2016. URL: `https://arxiv.org/abs/1607.06413`.

4 R. Balbes and P. Dwinger. *Distributive Lattices*. University of Missouri Press, 1975.

5 M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets. In *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, volume 26, pages 107–128, 2014.

6 L. Birkedal, A. Bizjak, R. Clouston, H. B. Grathwohl, B. Spitters, and A. Vezzosi. Guarded Cubical Type Theory: Path Equality for Guarded Recursion. In *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:17, 2016.

7 A. B. Booij, M. H. Escardó, P. L. Lumsdaine, and M. Shulman. Parametricity, automorphisms of the universe, and excluded middle. *arXiv preprint arXiv:1701.05617*, 2017. URL: `https://arxiv.org/abs/1701.05617`.

8 C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, volume 69 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:34, 2018.

9 P. Dybjer. Internal type theory. In S. Berardi and M. Coppo, editors, *Types for Proofs and Programs*, volume 1158 of *Lecture Notes in Computer Science*, pages 120–134. Springer Berlin Heidelberg, 1996.

10 M. Hofmann. Syntax and semantics of dependent types. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, Publications of the Newton Institute, pages 79–130. Cambridge University Press, 1997.

11 M. Hofmann and T. Streicher. Lifting Grothendieck universes (unpublished note). 1997. URL: `https://www2.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf`.

12 M. E. Maietti. Modular correspondence between dependent type theories and categories including pretopoi and topoi. *Mathematical Structures in Computer Science*, 15:1089–1149, 2005.

13 I. Orton and A. M. Pitts. Axioms for modelling cubical type theory in a topos. *Logical Methods in Computer Science*, 2018. Special issue for CSL 2016, to appear. URL: `https://arxiv.org/abs/1712.04864`.

14 A. M. Pitts. Nominal presentation of cubical sets models of type theory. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 39. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

15 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations for Mathematics*. Institute for Advanced Study, 2013. URL: `http://homotopytypetheory.org/book`.