

PAPER

On Church’s thesis in cubical assemblies

Andrew W. Swan^{1*}  and Taichi Uemura² 

¹Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA 15213, USA and ²Department of Mathematics, Stockholm University, Stockholm, Sweden

*Corresponding author. Email: wakelin.swan@gmail.com

(Received 24 September 2019; revised 21 January 2022; accepted 11 February 2022; first published online 21 March 2022)

Abstract

We show that Church’s thesis, the axiom stating that all functions on the naturals are computable, does not hold in the cubical assemblies model of cubical type theory. We show that nevertheless Church’s thesis is consistent with univalent type theory by constructing a lex modality in cubical assemblies such that Church’s thesis holds in the corresponding reflective subuniverse.

Keywords: Homotopy type theory; realizability; cubical sets

1. Introduction

One of the main branches of constructive mathematics is that of recursive or ‘Russian’ constructivism, where to justify the existence of mathematical objects, one must show how to compute them. A rather extreme interpretation of this philosophy is the axiom of Church’s thesis, which states that all functions from \mathbb{N} to \mathbb{N} are computable. Despite (or perhaps because of) its highly non-classical nature, it has been well studied by logicians and turns out to be consistent with a wide variety of formal theories for constructive mathematics. This is usually proved using realizability models based on computable functions, starting with Kleene’s model of Heyting arithmetic (Kleene 1945), but with many later variants and generalisations. See, for example, Troelstra and van Dalen (1988, Chapter 4, Section 4) for a standard reference.

When interpreting Church’s thesis in type theory, an additional complication is introduced. Logical statements are usually interpreted in type theory using the propositions-as-types interpretation. Applying this to Church’s thesis would give us the type below.

$$\prod_{f:\mathbb{N}\rightarrow\mathbb{N}} \sum_{e:\mathbb{N}} \prod_{x:\mathbb{N}} \sum_{z:\mathbb{N}} T(e, x, z) \wedge U(z) = f(x)$$

However, it is straightforward to use function extensionality to show that this type is empty.¹ It is therefore impossible in any case to show that the above “untruncated” version of Church’s thesis is consistent with univalence, since univalence implies function extensionality (Univalent Foundations Program 2013, Theorem 4.9.4).

To have any hope of showing Church’s thesis is consistent with univalence, we need a different formulation. We will use the interpretation of logical statements advocated in Univalent Foundations Program (2013, Section 3.7), and commonly used in homotopy type theory and elsewhere. In this approach, one uses the higher inductive type of propositional truncation

at disjunction and existential quantifiers, which ensures that the resulting type is always an hproposition (i.e. that any two of its elements are equal). This yields the following version of Church's thesis, which is the one we will study here.

$$\prod_{f:\mathbb{N}\rightarrow\mathbb{N}} \left\| \sum_{e:\mathbb{N}} \prod_{x:\mathbb{N}} \sum_{z:\mathbb{N}} T(e, x, z) \times U(z) = f(x) \right\|$$

It is well known that Church's thesis holds in the internal logic of Hyland's effective topos (see for instance van Oosten 2008, Section 3.1 for a standard reference). Similar arguments show that in fact it already holds in its simpler subcategory of assemblies, and even in cubical assemblies, when they are viewed as regular locally cartesian closed categories and thereby, following Awodey and Bauer (2004) or Maietti (2005) as models of extensional type theory with propositional truncation. However, the interpretation of cubical type theory in cubical assemblies due to the second author (Uemura 2019) is very different to the interpretation of extensional type theory. We draw attention in particular to the fact that for extensional type theory hpropositions are implemented as maps where in the internal logic each fibre has at most one element.² On the other hand in the interpretation of cubical type theory, each fibre can have multiple elements as long as any two elements are joined by a path, telling us to always treat them as "propositionally equal." For propositional truncation, we do not strictly identify elements by quotienting, but instead add new paths. Our first result is that Church's thesis is in fact false in the interpretation of cubical type theory in cubical assemblies, even though it holds in the internal logic.

To show Church's thesis is consistent with univalence, we will combine cubical assemblies with the work of Rijke, Shulman and Spitters on lex modalities and Σ -closed reflective subuniverses in Rijke et al. (2020). We will construct a lex modality where Church's thesis is forced to hold, and then use properties of cubical assemblies to show that the modality is non-trivial. Our model can also be viewed as a kind of stack model akin to those used by Coquand, Ruch and Sattler for various independence and consistency results, including the independence of countable choice from homotopy type theory (Coquand et al. 2021), although our formulation will be quite different to theirs.

Mike Shulman has pointed out that the overall construction may be seen as analogous to an existing result on exact completions. Menni (2001) showed that under certain conditions, the *ex/reg* completion of a category is equivalent to a category of sheaves in the *ex/lex* completion, over a topology generated by maps that are regular epimorphisms in the original category. The effective topos is the *ex/reg* completion of assemblies, and so a category of sheaves in the *ex/lex* completion of assemblies. Analogously, we think of cubical assemblies as an " ∞ -exact completion" of assemblies. We view the lex modality as analogous to the topology in Menni's construction. Since the lex modality is generated by only some, rather than all, regular epimorphisms we obtain only an approximation to the yet-to-be defined ∞ -effective topos.

As part of our proof, we will give a generalisation of the Coquand–Huber–Mörtberg (Coquand et al. 2018) construction of higher inductive types to arbitrary Orton–Pitts models. This will be used to define the higher inductive types of propositional truncation and nullification, which play an essential role in our proof.

2. Models of Type Theories

In this article, we use models of different kinds of type theories: extensional dependent type theory and intensional dependent type theory (with the univalence axiom). All of them are based on the notion of a category with families (Dybjer 1996).

Definition 1. Let \mathcal{C} be a category. A cwf-structure over \mathcal{C} is a pair (T, E) of presheaves $T : \mathcal{C}^{\text{op}} \rightarrow \mathbf{Set}$ and $E : (\int_{\mathcal{C}} T)^{\text{op}} \rightarrow \mathbf{Set}$ such that, for any object $\Gamma \in \mathcal{C}$ and element $X \in T(\Gamma)$, the presheaf

$$(\mathcal{C}/\Gamma)^{\text{op}} \ni (f : \Delta \rightarrow \Gamma) \mapsto E(\Delta, X \cdot f) \in \mathbf{Set}$$

is representable. The representing object for this presheaf is denoted by $\chi(X) : \{X\} \rightarrow \Gamma$ or $\chi(X) : \Gamma.X \rightarrow \Gamma$. A category with families, *cwf* in short, is a triple $\mathcal{E} = (\mathbb{C}^{\mathcal{E}}, \mathbb{T}^{\mathcal{E}}, \mathbb{E}^{\mathcal{E}})$ such that $\mathbb{C}^{\mathcal{E}}$ is a category with a terminal object and $(\mathbb{T}^{\mathcal{E}}, \mathbb{E}^{\mathcal{E}})$ is a cwf-structure over $\mathbb{C}^{\mathcal{E}}$.

In general, a model \mathcal{E} of a type theory consists of a category with families $(\mathbb{C}^{\mathcal{E}}, \mathbb{T}^{\mathcal{E}}, \mathbb{E}^{\mathcal{E}})$ and algebraic operations on the presheaves $\mathbb{T}^{\mathcal{E}}$ and $\mathbb{E}^{\mathcal{E}}$. An object Γ of $\mathbb{C}^{\mathcal{E}}$ is called a *context*. An element X of $\mathbb{T}^{\mathcal{E}}(\Gamma)$ is called a *type* and written $\Gamma \vdash_{\mathcal{E}} X$. An element a of $\mathbb{E}^{\mathcal{E}}(\Gamma, X)$ is called an *element of type X* and written $\Gamma \vdash_{\mathcal{E}} a : X$. The subscript of $\vdash_{\mathcal{E}}$ is omitted when the model \mathcal{E} is clear from the context. An algebraic operation on those presheaves is expressed by the schema

$$\frac{\Gamma \vdash \mathcal{J}_1 \quad \dots \quad \Gamma \vdash \mathcal{J}_n}{\Gamma \vdash A(\mathcal{J}_1, \dots, \mathcal{J}_n)}$$

where $\Gamma \vdash \mathcal{J}_j$ and $\Gamma \vdash A(\mathcal{J}_1, \dots, \mathcal{J}_n)$ are either of the form $\Gamma.X_1 \dots X_m \vdash Y$ or of the form $\Gamma.X_1 \dots X_m \vdash b : Y$ with $(\Gamma \vdash X_1), \dots, (\Gamma.X_1 \dots X_{m-1} \vdash X_m)$. In this schema, we always assume that the operation $A(\mathcal{J}_1, \dots, \mathcal{J}_n)$ is stable under reindexing: if $f : \Delta \rightarrow \Gamma$ is a morphism in $\mathbb{C}^{\mathcal{E}}$, then we have $A(\mathcal{J}_1, \dots, \mathcal{J}_n) \cdot f = A(\mathcal{J}_1 \cdot f, \dots, \mathcal{J}_n \cdot f)$.

Example 2. Let \mathcal{E} be a cwf. We say \mathcal{E} supports *dependent product types* if it has operations

$$\frac{\Gamma \vdash X \quad \Gamma.X \vdash Y}{\Gamma \vdash \Pi(X, Y)} \quad \frac{\Gamma \vdash X \quad \Gamma.X \vdash Y \quad \Gamma.X \vdash b : Y}{\Gamma \vdash \lambda(X, Y, b) : \Pi(X, Y)}$$

such that the map $\mathbb{E}(\Gamma.X, Y) \ni b \mapsto \lambda(X, Y, b) \in \mathbb{E}(\Gamma, \Pi(X, Y))$ is bijective.

We omit explicit definitions of other well-known type constructors such as dependent sum types, extensional and intensional identity types, inductive types, higher inductive types and universes.

For a model \mathcal{E} of a type theory, we denote by $\llbracket - \rrbracket^{\mathcal{E}}$ the interpretation of the type theory in the model \mathcal{E} .

Definition 3. By a model of univalent type theory, we mean a cwf that supports dependent product types, dependent sum types, intensional identity types, unit type, finite coproducts, natural numbers, propositional truncation and a countable chain

$$\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \dots$$

of univalent universes.

2.1 Internal languages

Formally we will work with models of type theories, but we will construct types and terms of those models in a syntactic way using their internal languages. Let $\mathcal{E} = (\mathbb{C}^{\mathcal{E}}, \mathbb{T}^{\mathcal{E}}, \mathbb{E}^{\mathcal{E}}, \dots)$ be a model of a type theory. For a context $\Gamma \in \mathbb{C}^{\mathcal{E}}$ and a type $\Gamma \vdash X$, we introduce a variable x and write $(\Gamma, x : X)$ for the context $\Gamma.X$. For another type $\Gamma \vdash Y$, the weakening $\Gamma, x : X \vdash Y$ is interpreted as the reindexing $\Gamma.X \vdash Y \cdot \chi(X)$. For an element $\Gamma \vdash a : X$ and a type $\Gamma, x : X \vdash Y(x)$, the substitution $\Gamma \vdash Y(a)$ is interpreted as the reindexing $\Gamma \vdash Y \cdot \bar{a}$, where $\bar{a} : \Gamma \rightarrow \Gamma.X$ is the section of $\Gamma.X \rightarrow \Gamma$ corresponding to the element $\Gamma \vdash a : X$. All type and term constructors of the type theory are soundly interpreted in \mathcal{E} in a natural way. Note that types and terms built in the internal language are stable under reindexing.

2.2 *W*-types with reductions

We will later use *W*-types with reductions to construct higher inductive types. So that we can use them internally in type theory we give below a new, split formulation. This is based on the non-dependent special case of the version in Swan (2018).

Let $\mathcal{E} = (\mathbb{C}^{\mathcal{E}}, \mathbb{T}^{\mathcal{E}}, \mathbb{E}^{\mathcal{E}}, \dots)$ be a model of a type theory with dependent product types, dependent sum types and extensional identity types. A *polynomial with reductions* over a context $\Gamma \in \mathbb{C}^{\mathcal{E}}$ consists of the following data:

- a type $\Gamma \vdash Y$ of *constructors*;
- a type $\Gamma, y : Y \vdash X(y)$ of *arities*;
- a type $\Gamma, y : Y \vdash R(y)$ together with an element $\Gamma, y : Y, r : R(y) \vdash k(y, r) : X(y)$ which we refer to as the *reductions*.

An *algebra* for a polynomial with reductions (Y, X, R, k) over $\Gamma \in \mathbb{C}^{\mathcal{E}}$ is a type $\Gamma \vdash W$ together with an element $\Gamma, y : Y, \alpha : X(y) \rightarrow W \vdash s(y, \alpha) : W$ such that $\Gamma, y : Y, \alpha : X(y) \rightarrow W, r : R(y) \vdash s(y, \alpha) = \alpha(k(y, r))$. Algebras for (Y, X, R, k) form a category in the obvious way and we refer to the initial algebra (if it exists) as the *W-type with reductions* for the polynomial with reductions.

3. Orton–Pitts Construction

Assumption 4. Let \mathcal{E} be a model of dependent type theory that supports dependent product types, dependent sum types, extensional identity types, unit type, finite colimits, natural numbers, propositional truncation and a countable chain of universes. We further assume that every context $\Gamma \in \mathbb{C}^{\mathcal{E}}$ is isomorphic to $1.X$ for some type X over the terminal object 1 . Suppose the following:

- \mathcal{E} has a type $1 \vdash \mathbb{I}$ equipped with two constants 0 and 1 and two binary operators \sqcap and \sqcup ;
- \mathcal{E} has types $1 \vdash \mathbb{F}$ and $\varphi : \mathbb{F} \vdash [\varphi]$ such that $\varphi : \mathbb{F}, x : [\varphi], y : [\varphi] \vdash x = y$. An element of \mathbb{F} is called a cofibrant proposition. We often omit $[-]$ and regard an element $\varphi : \mathbb{F}$ itself as a type;
- \mathbb{I} and \mathbb{F} satisfy ax_1 – ax_9 given by Orton and Pitts (2018);
- \mathbb{F} satisfies propositional extensionality: $\prod_{\varphi, \psi : \mathbb{F}} (\varphi \Leftrightarrow \psi) \Rightarrow (\varphi = \psi)$;
- the exponential functor $(-)^{\mathbb{I}} : \mathbb{C}^{\mathcal{E}} \rightarrow \mathbb{C}^{\mathcal{E}}$ has a right adjoint;
- \mathcal{E} has *W*-types with reductions for those polynomials with reductions where R is of the form $[\varphi]$ for an element $\Gamma, y : Y \vdash \varphi : \mathbb{F}$, and they are stable under reindexing. We say \mathcal{E} has *cofibrant W-types with reductions*.

Note that the axioms in Orton and Pitts (2018) are written in the internal language of an elementary topos, but they are easily translated into dependent type theory with \mathbb{I} and \mathbb{F} as above. We require propositional extensionality which trivially holds when \mathbb{F} is a subobject of the subobject classifier of an elementary topos.

Under these assumptions, following Orton and Pitts, we will build a model $\tilde{\mathcal{E}}$ of univalent type theory as follows:

- the base category $\mathbb{C}^{\tilde{\mathcal{E}}}$ is that of \mathcal{E} ;
- the types $\Gamma \vdash_{\tilde{\mathcal{E}}} X$ are the types $\Gamma \vdash_{\mathcal{E}} X$ equipped with a ‘fibration structure’;
- the elements $\Gamma \vdash_{\tilde{\mathcal{E}}} a : X$ are the elements $\Gamma \vdash_{\mathcal{E}} a : X$ of the underlying type X in \mathcal{E} ;

By the construction given in Orton and Pitts (2018), this model $\tilde{\mathcal{E}}$ supports dependent product types, dependent sum types, identity types, unit type, finite coproducts and natural numbers. For

a countable chain of univalent universes, use the right adjoint to $(-)^{\mathbb{I}}$ as in Licata et al. (2018). It remains to show that \mathcal{E} supports propositional truncation, which will be proved in Section 3.1 using cofibrant W -types with reductions. We call a model of univalent type theory of the form \mathcal{E} an *Orton–Pitts model*.

3.1 Higher inductive types in Orton–Pitts models

We are still working with a model \mathcal{E} of type theory that satisfies Assumption 4. We will show how to construct higher inductive types in \mathcal{E} . Our techniques are fairly general, although we will focus on the higher inductive types (HITs) that we will need for the main theorem. The techniques developed by Coquand, Huber and Mörtberg in Coquand et al. (2018) are already very close to working in arbitrary Orton–Pitts models. The main exception is that the underlying objects for the HITs are given by certain initial algebras, which are constructed directly for cubical sets. This definition does not quite work for cubical assemblies for two reasons. Firstly we are using a different cube category, and secondly we are working internally in assemblies. Rather than proving the same results again for cubical assemblies, we will use a more general approach based on W -types with reductions that covers both cases. The first author already showed in Swan (2018, Section 4) that (non-split) W -types with locally decidable reductions can be constructed in any category of presheaf assemblies and we will see later how to ensure that we get in fact *split* W -types with reductions in presheaf assemblies.

Finally, we will also make some minor adjustments related to the fact that we do not assume the interval object has reversals.

When we construct higher inductive types, we will use formulations based on Path types, following Coquand, Huber and Mörtberg. Technically these formulations can only be stated in cubical type theory, and not in intensional type theory in general. However, it is straightforward to derive versions based on Id types using the equivalence of Path and Id types, which are then valid in \mathcal{E} . We note that although computation rules hold definitionally for both point and path constructors for the Path type versions, after translating to Id types, the definitional equality only holds for point constructors. However, neither definitional equality will be needed for our end result.

We will review a key idea in the techniques developed by Coquand et al. (2018) for constructing higher inductive types. We first recall the Orton–Pitts definition of Kan composition.

Definition 5 (Orton–Pitts). *Suppose we are given a type $\Gamma \vdash_{\mathcal{E}} A$. A Kan composition operator is an operation that takes as input $\varepsilon \in \{0, 1\}$, $\gamma : \mathbb{I} \rightarrow \Gamma$ and $u : \prod_{i:\mathbb{I}} ((i = \varepsilon) \vee \varphi) \rightarrow A(p(i))$ and returns an element $a : A(p(1 - \varepsilon))$ together with a proof that for $p : \varphi$, $a = u(1 - \varepsilon, p)$.*

We now recall the Coquand–Huber–Mörtberg definitions of homogeneous Kan composition and transport. In order to apply their techniques to arbitrary Orton–Pitts models, we will make the following minor adjustment. They work in a setting where the interval type has an involution operation, allowing them to use a ‘one directional’ definition of Kan composition. That is, instead of the definition above, which takes a direction argument $\varepsilon \in \{0, 1\}$, they only consider the simpler definition with ε fixed as 0, where the other direction can be derived using the involution operator, if necessary. In the more general setting of an arbitrary Orton–Pitts model, we need to follow the Orton–Pitts definition of Kan composition, and similarly define homogenous composition and transport so that they take a direction as input.

Definition 6 (Coquand–Huber–Mörtberg). *Suppose we are given a type $\Gamma \vdash_{\mathcal{E}} A$. A homogeneous Kan composition operator is an operation in context Γ that takes as input $\varepsilon \in \{0, 1\}$ and $u : \prod_{i:\mathbb{I}} ((i = \varepsilon) \vee \varphi) \rightarrow A$ and returns $a : A$ such that for $p : \varphi$, $a = u(1 - \varepsilon, p)$.*

Definition 7 (Coquand–Huber–Mörtberg). *Given a type $\Gamma \vdash_{\mathcal{E}} A$, a transport operator is an operation that takes as input $\varepsilon \in \{0, 1\}$, a path $\gamma : \mathbb{I} \rightarrow \Gamma$ such that $\varphi \rightarrow \prod_{i,j:\mathbb{I}} \gamma(i) = \gamma(j)$, and $u_0 : A(\gamma(\varepsilon))$ and returns $u_1 : A(\gamma(1 - \varepsilon))$ such that $\varphi \rightarrow u_0 = u_1$.*

We can now see the key lemma used in the construction of higher inductive types.

Lemma 8 (Coquand–Huber–Mörtberg). *Given both a homogeneous Kan composition operator and a transport operator on a type $\Gamma \vdash_{\mathcal{E}} A$ we can construct a Kan composition operator on $\Gamma \vdash_{\mathcal{E}} A$.*

Proof. This appears as Coquand et al. (2018, Lemma 2.5). We simply observe that given a transport operator for both directions $\varepsilon \in \{0, 1\}$ we can produce also a Kan composition operator for both directions, by applying the Coquand–Huber–Mörtberg argument twice, once for each direction. □

We can also think of types with homogeneous Kan composition as ‘locally fibrant.’ We show below how to use W -types with reductions to construct local fibrant replacements, i.e. how to freely add homogeneous Kan composition to arbitrary types.

Definition 9. *Given a type $\Gamma \vdash_{\mathcal{E}} A$, we define the local fibrant replacement of A , $LFR(A)$ to be the W -type with reductions defined as follows.*

- When $a : A$, we add an element $\text{inc}(a)$ to $LFR(A)$.
- When $\varphi : \mathbb{F}$, $\varepsilon \in \{0, 1\}$ and $u : \sum_{i:\mathbb{I}} ((i = \varepsilon) \vee \varphi) \rightarrow LFR(A)$, we add an element $\text{hcomp}(\varphi, \varepsilon, u)$ to $LFR(A)$.
- If $p : \varphi$ and ε and u are as above then $\text{hcomp}(\varphi, \varepsilon, u)$ reduces to $u(1 - \varepsilon, p)$.

Formally, we define the constructors Y to be the coproduct $A + (\mathbb{F} \times 2)$. We take the arity $X(\text{inc}(a))$ to be the empty type for $a : A$ and $X(\text{inr}(\varphi, \varepsilon))$ to be $\sum_{i:\mathbb{I}} \varphi \vee (i = \varepsilon)$ for $(\varphi, \varepsilon) : \mathbb{F} \times 2$. We take the reductions $R(\text{inc}(a))$ to be \perp for $a : A$ and $R(\text{inr}(\varphi, \varepsilon))$ to be φ together with the map $p : \varphi \rightarrow (1 - \varepsilon, p) : \sum_{i:\mathbb{I}} \varphi \vee (i = \varepsilon)$.

Theorem. The model $\tilde{\mathcal{E}}$ supports suspensions.

Proof. Suppose we are given a type $\Gamma \vdash_{\tilde{\mathcal{E}}} X$. We first construct the naïve suspension, $\text{Susp}_0(X)$ as the pushout below.

$$\begin{array}{ccc}
 X \times 2 & \longrightarrow & 2 \\
 \downarrow & & \downarrow \\
 X \times \mathbb{I} & \longrightarrow & \text{Susp}_0(X)
 \end{array}$$

We next take the local fibrant replacement, to get $LFR(\text{Susp}_0(X))$. This is then an initial $\text{Susp}(X)$ algebra, as defined by Coquand et al. (2018, Section 2.2) and so we can then proceed with the same proof as they do there, observing that their argument produces by symmetry a transport operator in both directions. □

Theorem. The model $\tilde{\mathcal{E}}$ supports propositional truncation.

Proof. Suppose we are given a type $\Gamma \vdash_{\tilde{\mathcal{E}}} A$. We first define the underlying object of $\|A\|$ to be the W -type with reductions defined as follows.

- When $a : A$, we add an element $\text{inc}(a)$ to $\|A\|$.
- When $\varphi : \mathbb{F}$, $\varepsilon \in \{0, 1\}$ and $u : \sum_{i:\mathbb{I}} ((i = \varepsilon) \vee \varphi) \rightarrow \|A\|$, we add an element $\text{hcomp}(\varphi, \varepsilon, u)$ to $\|A\|$.
- If $p : \varphi$ and ε and u are as above, then $\text{hcomp}(\varphi, \varepsilon, u)$ reduces to $u(1 - \varepsilon, p)$.
- If $x, y : \|A\|$ and $i : \mathbb{I}$, then $\|A\|$ contains an element of the form $\text{sq}(x, y, i)$.
- If x, y, i are as above and $i = 0$, then $\text{sq}(x, y, i)$ reduces to x .
- If x, y, i are as above and $i = 1$, then $\text{sq}(x, y, i)$ reduces to y .

Formally, we define this by taking the coproduct of two polynomials with reductions. The first is the one we used before for LFR. The second has constructors $Y := \mathbb{I}$, with the arity defined by $X(i) := 2$, and reductions $R(i) := (i = 0) \vee (i = 1)$ together with the map $p : (i = 0) \vee (i = 1) \vdash k(p) : 2$ defined by $k(p) = 0$ if $p : i = 0$ and $k(p) = 1$ if $p : i = 1$.

The remainder of the proof is the same as the syntactic description of propositional truncation by Coquand et al. (2018, Section 3.3.4). □

We now construct a new higher inductive type, which is a simplified version of the higher inductive type \mathcal{I}_F defined by Rijke et al. (2020, Section 2.2). Given families of types $\Gamma \vdash_{\mathcal{E}} A$ and $\Gamma, a : A \vdash_{\mathcal{E}} B(a)$ we will construct a higher inductive type \mathcal{K}_B^Γ defined as follows.

- When $a : A$ and $f : B(a) \rightarrow \mathcal{K}_B$, we add an element $\text{ext}(a, f)$ to \mathcal{K}_B .
- When $a : A$, $f : B(a) \rightarrow \mathcal{K}_B$ and $b : B(a)$ we add an element $\text{isext}(a, f, b)$ to $\text{Path}(\text{ext}(a, f), f(b))$.

We require that \mathcal{K}_B satisfies the following elimination rule. Suppose we are given a family of types $\Gamma, x : \mathcal{K}_B \vdash_{\mathcal{E}} P(x)$ together with the terms below.

$$\begin{aligned}
 R : \prod_{a:A} \prod_{f:B(a) \rightarrow \mathcal{K}_B} \left(\prod_{b:B(a)} P(f(b)) \right) &\rightarrow P(\text{ext}(a, f)) \\
 S : \prod_{a:A} \prod_{f:B(a) \rightarrow \mathcal{K}_B} \prod_{f':\prod_{b:B(a)} P(f(b))} \prod_{b:B(a)} \prod_{i:\mathbb{I}} &P(\text{isext}(a, f, b)(i))
 \end{aligned}$$

Suppose further that S satisfies the equalities

$$\begin{aligned}
 S(a, f, f', b, 0) &= R(a, f, f') \\
 S(a, f, f', b, 1) &= f'(b)
 \end{aligned}$$

Then we have a choice of term $\Gamma, x : \mathcal{K}_B \vdash s(x) : P(x)$ satisfying the following computation rules for $a : A, f : B(a) \rightarrow \mathcal{K}_B$ and $b : B(a)$.

$$\begin{aligned}
 s(\text{ext}(a, f)) &= R(a, f, s \circ f) \\
 s(\text{isext}(a, f, b)(i)) &= S(a, f, s \circ f, b, i)
 \end{aligned}$$

Moreover the choice of term is strictly preserved by reindexing.

We use the techniques developed by Coquand, Huber and Mörtberg together with W -types with reductions for constructing the actual objects. In order to give \mathcal{K}_B^Γ the structure of a fibration, we need to define a composition operator. We will do this by freely adding an hcomp operator, and then combining it with a transport operator, which we will explicitly define.

Definition 10. Let $\Gamma \vdash_{\mathcal{E}} X$ be a type. We define the naïve cone, $\text{Cone}(X)$ to be the following pushout³.

$$\begin{array}{ccc} X & \longrightarrow & 1 \\ \downarrow (1_X, \delta_0) & \lrcorner & \downarrow \text{inl} \\ X \times \mathbb{I} & \xrightarrow{\text{inr}} & \text{Cone}(X) \end{array}$$

We can now define \mathcal{K}_B^Γ to be the following W -type with reductions.

- When $a : A, c : \text{Cone}(B(a))$ and $f : B(a) \rightarrow \mathcal{K}_B$, we add an element $\text{pastecone}(a, c, f)$ to \mathcal{K}_B .
- If a, c, f are as above and c is of the form $\text{inr}(b, 1)$ for $b : B(a)$, then $\text{pastecone}(a, c, f)$ reduces to $f(b)$.
- When $\varphi : \mathbb{F}, \varepsilon : \{0, 1\}$ and $u : \sum_{i:\mathbb{I}} ((i = 0) \vee \varphi) \rightarrow \mathcal{K}_B$, we add an element $\text{hcomp}(\varphi, \varepsilon, u)$ to \mathcal{K}_B .
- If $p : \varphi$, and ε and u are as above then $\text{hcomp}(\varphi, \varepsilon, u)$ reduces to $u(1 - \varepsilon, p)$.

To check that this really is a W -type with reductions, we need to define the polynomial with reductions. We take it to be the coproduct of the following two polynomials with reductions.

We define the first component of the coproduct as follows. We take the constructors Y to be $\sum_{a:A} \text{Cone}(B(a))$ and the arities $X(a, c)$ to be $B(a)$. We take the reductions $R(a, \text{inl}(\ast))$ to be \perp and $R(a, \text{inr}(b, i))$ to be $(i = 1)$ together with the map $R(a, \text{inr}(b, i)) \vdash b : B(a)$. Note that $R : (\sum_{a:A} \text{Cone}(B(a))) \rightarrow \mathbb{F}$ is well-defined because we have $(0 = 1) = \perp$ by propositional extensionality.

The second component in the coproduct is the polynomial with reductions that we used for local fibrant replacement.

Lemma 11. We construct a transport operator for \mathcal{K}_B^Γ .

Proof. We give a proof for the direction $\varepsilon = 0$. The construction for the other direction is exactly the same by symmetry.

Suppose we are given $\varphi : \mathbb{F}$ and a path γ in Γ which is constant on φ . We need to define a transport operator, which is a map $t : \mathcal{K}_{B(\gamma(0))} \rightarrow \mathcal{K}_{B(\gamma(1))}$ such that t is the identity when φ is true. Formally this map can be defined by giving an appropriate algebra structure on $\mathcal{K}_{B(\gamma(1))}$ and then using the initiality of $\mathcal{K}_{B(\gamma(0))}$. However, for clarity we will present the proof as an argument by higher recursion on the definition of $\mathcal{K}_{B(\gamma(0))}$.

We need to show how to define $t(\text{pastecone}(a, c, f))$ and $t(\text{hcomp}(\psi, \varepsilon, u))$, and then check that the definition respects the reduction equations. For the latter, we define the transport operator so that it preserves the hcomp structure, which determines it uniquely, following Coquand et al. (2018). For the former, we recall that $\text{Cone}(B(a))$ was defined as a pushout, and so we can split into a further two cases. Either c is of the form $\text{inl}(\ast)$, or it is of the form $\text{inr}(b, i)$ where $b : B(a)$ and $i : \mathbb{I}$. Now in addition to the reduction equation, we have to also satisfy $t(\text{inl}(\ast)) = t(\text{inr}(b, 0))$ in order to eliminate out of the pushout.

Write t_A for the transport $A(\gamma(0)) \rightarrow A(\gamma(1))$ and t_B for the transport $\prod_{a:A(\Gamma(0))} B(a) \rightarrow B(t_A(a))$ ensuring that $t_A(a) = a$ and $t_B(b) = b$ when $\varphi = \top$, for all $a : A(\gamma(0))$ and $b : B(a)$. Write t_B^{-1} for the homotopy inverse $\prod_{a:A(\Gamma(0))} B(t_A(a)) \rightarrow B(a)$, again ensuring that $t_B^{-1}(b) = b$ when $\varphi = \top$. Since we are only guaranteed the existence of a homotopy inverse, not a strict inverse, we do not necessarily have $t_B^{-1} \circ t_B = 1_{B(a)}$. We can however construct paths $p : \prod_{a:A(\Gamma(0))} \prod_{b:B(a)} \mathbb{I} \rightarrow B(a)$ satisfying for all $a : A(\Gamma(0))$ and $b : B(a)$ that $p(a, b, 0) = t_B^{-1}(t_B(b))$ and $p(a, b, 1) = b$. Furthermore, we may assume that for any a, b and i , if $\varphi = \top$ then $p(a, b, i) = b$.

We define $t(\text{pastecone}(a, \text{inl}(\ast), f))$ to be of the form $\text{pastecone}(t_A(a), \text{inl}(\ast), f')$, where we still need to define a function $f' : B(t_A(a)) \rightarrow \mathcal{K}_{B(\gamma(1))}$. Note that we may assume by recursion that

for each $b : B(a)$, $t(f(b))$ has already been defined and belongs to $\mathcal{K}_{B(\gamma(1))}$. Hence, we can simply define f' to be $t \circ f \circ t_B^{-1}$.

The obvious first attempt at defining $t(\text{pastecone}(a, \text{inr}(b, i), f))$ would be $\text{pastecone}(t_A(a), \text{inr}(t_B(b), i), t \circ f \circ t_B^{-1})$. Note however that this does not satisfy the reduction equations. This is because when $i = 1$, $\text{pastecone}(a, \text{inr}(b, i), f)$ reduces to $f(b)$ and $\text{pastecone}(t_A(a), \text{inr}(t_B(b), i), t \circ f \circ t_B^{-1})$ reduces to $t(f(t_B^{-1}(t_B(b))))$ which is not necessarily strictly equal to $t(f(b))$. We fix this using the `hcomp` constructor, following the construction of homotopy pushouts in Coquand et al. (2018, Section 2.3). We define $\psi : \mathbb{F}$ to be $\varphi \vee (i = 0) \vee (i = 1)$. We then define $u : \sum_{j:\mathbb{I}} (\psi \vee (j = 0)) \rightarrow \mathcal{K}_{B(\gamma(a))}$ as follows.

$$u(j, *) \equiv \begin{cases} \text{pastecone}(t_A(a), \text{inr}(t_B(b), i), t \circ f \circ t_B^{-1}) & j = 0 \\ \text{pastecone}(a, \text{inr}(b, i), t \circ f) & \varphi = \top \\ \text{pastecone}(t_A(a), \text{inl}(*), t \circ f \circ t_B^{-1}) & i = 0 \\ t(f(p(a, b, j))) & i = 1 \end{cases}$$

We then define $t(\text{pastecone}(a, \text{inr}(b, i), f))$ to be `hcomp`($\psi, 0, u$). The reduction equation for `hcomp` then ensures that we do satisfy the reduction equation for `pastecone` and also retain the necessary equations for the pushout and furthermore ensures that the resulting map $t : \mathcal{K}_{B(\gamma(0))} \rightarrow \mathcal{K}_{B(\gamma(1))}$ is (one direction of) a transport operator. □

Theorem. We construct a fibration structure for each \mathcal{K}_B , which is strictly preserved by reindexing.

Proof. By Lemmas 11 and 8. □

Lemma 12. We construct terms `ext` and `isext` for \mathcal{K}_B that satisfy the appropriate equations.

Proof.

$$\begin{aligned} \text{ext}(a, f) &:\equiv \text{pastecone}(a, \text{inl}(*), f) \\ \text{isext}(a, f, b)(i) &:\equiv \text{pastecone}(a, \text{inr}(b, i), f) \end{aligned}$$

□

Lemma 13. \mathcal{K}_B satisfies the necessary induction principle.

Proof. Suppose we are given a family of types $\Gamma, x : \mathcal{K}_B \vdash_{\mathcal{E}} P(x)$ together with the terms below.

$$\begin{aligned} R : \prod_{a:A} \prod_{f:B(a) \rightarrow \mathcal{K}_B} \left(\prod_{b:B(a)} P(f(b)) \right) &\rightarrow P(\text{ext}(a, f)) \\ S : \prod_{a:A} \prod_{f':\prod_{b:B(a)} P(f(b))} \prod_{b:B(a)} \prod_{i:\mathbb{I}} &P(\text{isext}(a, f, b)(i)) \end{aligned}$$

We need to define a term $\Gamma, x : \mathcal{K}_B \vdash s(x) : P(x)$ satisfying the appropriate equalities. We define s by higher recursion on the construction of \mathcal{K}_B . We first deal with the case $s(\text{pastecone}(a, c, f))$. Recalling that $\text{Cone}(B(a))$ is defined as a pushout, we can split into the two cases $c = \text{inl}(*)$ and $c = \text{inr}(b, i)$ for some $b : B(a)$ and $i : \mathbb{I}$.

We define

$$\begin{aligned} s(\text{pastecone}(a, \text{inl}(*), f)) &:\equiv R(a, f, s \circ f) \\ s(\text{pastecone}(a, \text{inr}(b, i), f)) &:\equiv S(a, f, s \circ f, b, i) \end{aligned}$$

It is straightforward to check that this does preserve the reduction and pushout equations and so does give a well defined map. One can show it is a section again by higher recursion and the computation rules are satisfied by definition.

Finally, to define $s(\text{hcomp}(\varphi, \varepsilon, u))$, we use the fibration structure on $\Gamma, x : \mathcal{K}_B \vdash_{\mathcal{E}} P(x)$. □

3.2 Internal cubical models

Let \mathcal{S} be a model of dependent type theory with dependent product types, dependent sum types, extensional identity types, unit type, finite colimits, W -types and a countable chain of universes. We also assume that every context of \mathcal{S} is isomorphic to $1.X$ for some type $1 \vdash_{\mathcal{S}} X$. In particular, the category $\mathbb{C}^{\mathcal{S}}$ is finitely complete so that internal categories in $\mathbb{C}^{\mathcal{S}}$ make sense. Let \square denote the internal category in $\mathbb{C}^{\mathcal{S}}$ in which the objects are the natural numbers and the morphisms from n to m are the order-preserving functions $2^n \rightarrow 2^m$. Note that \mathcal{S} has a natural number object since it has W -types. Spitters (2016) has observed that \square is equivalent to the opposite of the category of free finitely generated distributive lattices. We will refer to internal presheaves over \square as *internal cubical objects*.

Theorem. Under those assumptions, the category of internal cubical objects in \mathcal{S} is part of a model of type theory that satisfies Assumption 4.

Example 14. Let \mathcal{A} be a partial combinatory algebra. It is well-known that the category $\mathbf{Asm}(\mathcal{A})$ of assemblies on \mathcal{A} is part of a model of type theory with dependent product types, dependent sum types, extensional identity types, unit type and finite colimits. It is also known that $\mathbf{Asm}(\mathcal{A})$ has W -types (an explicit construction is found in van den Berg 2006, Section 2.2). Assuming a countable chain of Grothendieck universes in the set theory, $\mathbf{Asm}(\mathcal{A})$ has a countable chain of universes, since for each Grothendieck universe one may implement a universe in assemblies as the uniform assembly on the set of small assemblies. Thus, the category $\mathbf{CAsm}(\mathcal{A})$ of internal cubical objects in $\mathbf{Asm}(\mathcal{A})$ is part of a model of type theory that satisfies Assumption 4.

It is shown in Orton and Pitts (2018) that, when $\mathcal{S} = \mathbf{Set}$, the category of presheaves over \square satisfies all the axioms of Orton and Pitts if we take \mathbb{F} to be the presheaf of locally decidable propositions. The proof works for an arbitrary \mathcal{S} and one can show that the category of internal cubical objects in \mathcal{S} is part of a model of type theory satisfying Assumption 4 except the existence of cofibrant W -types with reductions (see also Uemura 2019). To construct cofibrant W -types with reductions, we recall the following from Swan (2018).

Theorem. Let \mathcal{E} be a locally cartesian closed category with finite colimits and disjoint coproducts and W -types, and let \mathbf{C} be an internal category in \mathcal{E} . Then the category $\mathcal{P}(\mathbf{C})$ of internal presheaves over \mathbf{C} has all locally decidable W -types with reductions.

We furthermore observe that one can show that this construction is stable under pullback up to isomorphism using a technique similar to the one used by Gambino and Hyland for ordinary W -types (Gambino and Hyland 2004). The reason is that pointed polynomial endofunctors are stable under pullback because they are constructed from Σ types, Π types and pushouts, all of which are preserved by pullback, and in locally cartesian closed categories, the initial algebras of such pointed endofunctors are also stable under pullback. However, to ensure that the construction is strictly preserved requires a little more work.

We show how to use the non-split version above to construct split W -types with reductions. The essential idea is to carry out the construction given above “pointwise,” expanding out the method suggested by Coquand et al. (2018, Section 2.2). Since we define cubical sets here as a category of presheaves in the usual, contravariant sense, we work with contravariant presheaves

here, although the original proof in Swan (2018) is phrased in terms of covariant presheaves. We also make minor adjustments to fit with the ‘split’ version appearing in Section 2.2.

Suppose that we are given a context $\Gamma \in \mathcal{P}(\mathbf{C})$ together with a type $Y \in \mathcal{P}(\int_{\mathbf{C}} \Gamma)$, a type $X \in \mathcal{P}(\int_{\mathbf{C}} \{Y\})$, a locally decidable monomorphism $R \rightarrow Y$ and a map $k : \prod_{y:R} X(y)$ over $\int_{\mathbf{C}} \Gamma$.

We need to show how to define a strict version of the W -type with reductions $W(Y, X, R)$. We will refer to the new strict version as $W'(Y, X, R)$. This should be an element of $\mathcal{P}(\int_{\mathbf{C}} \Gamma)$, so in particular we need to define a family of types $W'(Y, X, R)(c, \gamma)$ indexed by objects c of \mathbf{C} and elements $\gamma : \Gamma(c)$.

We fix such a c and γ . We first note that we have a locally decidable polynomial with reductions $Y_\gamma, X_\gamma, R_\gamma$ in the internal presheaf category $\mathcal{P}(\int_{\mathbf{C}} \mathbf{C}(-, c))$ given by reindexing along the map $\mathbf{C}(-, c) \rightarrow \Gamma$ given by Yoneda. We then carry out the ‘non-strict’ construction to get a presheaf $W(Y_\gamma, X_\gamma, R_\gamma)$ on $\int_{\mathbf{C}} \mathbf{C}(-, c)$ and finally we define $W'(Y, X, R)(c, \gamma)$ to be $W(Y_\gamma, X_\gamma, R_\gamma)(c, 1_c)$.

For completeness, we unfold the definitions to obtain the following explicit description of $W'(Y, X, R)(c, \gamma)$. We first define the dependent W -type N_0 of *normal forms* indexed by the objects (d, f) of $\int_{\mathbf{C}} \mathbf{C}(-, c)$.

If (d, f) is an object of $\int_{\mathbf{C}} \mathbf{C}(-, c)$, we add an element to $N_0(d, f)$ of the form $\text{sup}(y, \alpha)$ whenever y is an element of $Y(d, \Gamma(f)(\gamma))$ that does not belong to the subobject $R(d, \Gamma(f)(\gamma))$ and α is an element of the following type.

$$\prod_{g:e \rightarrow d} N_0(e, f \circ g)^{X(f \circ g, \Gamma(f \circ g)(\gamma), Y(g)(y))}$$

The next step is to define maps $N_0(d, f) \rightarrow N_0(e, f \circ g)$ whenever $g : e \rightarrow d$ and $f : d \rightarrow c$ in \mathbf{C} . Say that we are given an element of $N_0(d, f)$ of the form $\text{sup}(y, \alpha)$. We recall that $N_0(g)(\text{sup}(y, \alpha))$ is defined by splitting into cases depending on whether or not y belongs to the subobject $R(d, \Gamma(f)(\gamma))$. If it does, we define $N_0(g)(\text{sup}(y, \alpha))$ to be $\alpha(g, k(y))$. Otherwise, we define $N_0(g)(\text{sup}(y, \alpha))$ to be $\text{sup}(Y(g)(y), \alpha')$ where $\alpha'(h, x)$ is defined to be $\alpha(g \circ h, x)$.

We then define $N(d, f)$ for each $f : d \rightarrow c$ to be the subobject of $N_0(d, f)$ consisting of hereditarily natural elements and verify that this does indeed define a presheaf on $\int_{\mathbf{C}} \mathbf{C}(-, c)$. But this is identical to Swan (2018, Section 4) so we omit the details.

If we then define $W'(Y, X, R)(c, \gamma)$ to be $N(c, 1_c)$, then this is strictly stable under reindexing by definition.

One can construct by recursion an isomorphism between $N(d, f)$ and $W(Y, X, R)(d, \Gamma(f)(\gamma))$ for each $f : d \rightarrow c$. In particular, this gives us an isomorphism between $N(c, 1_c)$ and $W(Y, X, R)(c, \gamma)$, and so we have a canonical isomorphism between $W'(Y, X, R)(c, \gamma)$ and $W(Y, X, R)(c, \gamma)$. It follows that we can assign an initial algebra structure to $W'(Y, X, R)(c, \gamma)$ by transferring the algebra structure on $W(Y, X, R)(c, \gamma)$ via the isomorphism.

3.3 Discrete types

We introduce a class of types in an Orton–Pitts model for future use. Let \mathcal{E} be a model of type theory satisfying Assumption 4.

Definition 15. A type $1 \vdash X$ is said to be discrete if the map $\lambda x. \lambda i. x : X \rightarrow X^{\mathbb{I}}$ is an isomorphism.

The proofs of the following propositions are found in Uemura (2019).

Proposition 16. Every discrete type $1 \vdash X$ carries a fibration structure.

Proposition 17. If a type $1 \vdash X$ has decidable equality, then it is discrete.

Corollary 18. The natural number object in \mathcal{E} is discrete.

4. Church’s Thesis

We consider a dependent type theory with dependent product types, dependent sum types, identity types, unit type, disjoint finite coproducts, propositional truncation and natural numbers. In such a dependent type theory, one can define Kleene’s computation predicate $T(e, x, z)$ and result extraction function $U(z)$ as primitive recursive functions $T : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbf{2}$ and $U : \mathbb{N} \rightarrow \mathbb{N}$. The statement $T(e, x, z)$ means that z codes a computation on Turing machine e with input x and $U(z)$ is the output of the computation. *Church’s Thesis* is the following axiom.

$$\forall f:\mathbb{N} \rightarrow \mathbb{N} \exists e:\mathbb{N} \forall x:\mathbb{N} \exists z:\mathbb{N} T(e, x, z) \wedge U(z) = f(x)$$

Since the type $\sum_{z:\mathbb{N}} T(e, x, z) \times U(z) = f(x)$ is a proposition, Church’s Thesis is equivalent to the type

$$\prod_{f:\mathbb{N} \rightarrow \mathbb{N}} \left\| \sum_{e:\mathbb{N}} \prod_{x:\mathbb{N}} \sum_{z:\mathbb{N}} T(e, x, z) \times U(z) = f(x) \right\|.$$

4.1 Failure of Church’s thesis in internal cubical models

Let \mathcal{S} be a model of type theory as in Section 3.2. We have seen that the category $\mathcal{P}(\square)$ of internal cubical objects in \mathcal{S} is part of a model of type theory satisfying Assumption 4. In this section, we show the following theorem.

Theorem. The negation of Church’s Thesis holds in the model of univalent type theory $\widetilde{\mathcal{P}(\square)}$.

To prove Theorem 4.1, we recall from Uemura (2019) the notion of a codiscrete presheaf and some useful propositions. The constant presheaf functor $\Delta : \mathcal{S} \rightarrow \mathcal{P}(\square)$ extends to a morphism of cwf’s and preserves (at least up to isomorphism) several type constructors. Here, we only need the following.

Proposition 19. *The morphism $\Delta : \mathcal{S} \rightarrow \mathcal{P}(\square)$ of cwf’s preserves dependent product types, dependent sum types, extensional identity types and natural number objects.*

A constant presheaf ΔX is regarded as a type in $\widetilde{\mathcal{P}(\square)}$ by the following proposition and Proposition 16.

Proposition 20. *Constant presheaves are discrete.*

For types $1 \vdash_{\mathcal{S}} X$ and $x : X \vdash_{\mathcal{S}} Y(x)$, one can define a type $x : \Delta X \vdash_{\mathcal{P}(\square)} \nabla_X Y(x)$ called the codiscrete presheaf which has the following properties.

Proposition 21. *∇_X is the right adjoint to the evaluation functor $(-)_0$ at $0 \in \square$: for any type $x : \Delta X \vdash_{\mathcal{P}(\square)} Z(x)$, we have a natural bijection between the set of elements $x : \Delta X, z : Z(x) \vdash_{\mathcal{P}(\square)} b : \nabla_X Y(x)$ and the set of elements $x : X, z : Z_0(x) \vdash_{\mathcal{S}} b : Y(x)$. Note that $(\Delta X)_0 = X$ and thus Z_0 is a type in \mathcal{S} over X .*

Proposition 22. *For a type $x : X \vdash_{\mathcal{S}} Y(x)$, the type $x : \Delta X \vdash_{\mathcal{P}(\square)} \nabla_X Y(x)$ has a composition structure and is a proposition in $\widetilde{\mathcal{P}(\square)}$.*

Proof of Theorem 4.1. We define types $f : \mathbb{N} \rightarrow \mathbb{N} \vdash C(f) \equiv \sum_{e:\mathbb{N}} \prod_{x:\mathbb{N}} \sum_{z:\mathbb{N}} T(e, x, z) \times U(x) = f(x)$ and $f : \mathbb{N} \rightarrow \mathbb{N} \vdash C(f) \equiv \|C(f)\|$. Let N denote the natural number object in \mathcal{S} . Then

Church's Thesis is interpreted in $\widetilde{\mathcal{P}(\square)}$ as $\prod_{f:\Delta(N \rightarrow N)} \llbracket C \rrbracket^{\widetilde{\mathcal{P}(\square)}}(f)$ by Proposition 19. We will construct two functions in $\widetilde{\mathcal{P}(\square)}$:

- $\prod_{f:\Delta(N \rightarrow N)} \llbracket C \rrbracket^{\widetilde{\mathcal{P}(\square)}}(f) \rightarrow \nabla_{N \rightarrow N} \llbracket C \rrbracket^{\mathcal{S}}(f)$;
- $\left(\prod_{f:\Delta(N \rightarrow N)} \nabla_{N \rightarrow N} \llbracket C \rrbracket^{\mathcal{S}}(f) \right) \rightarrow \mathbf{0}$.

Then we readily get a function $\left(\prod_{f:\Delta(N \rightarrow N)} \llbracket C \rrbracket^{\widetilde{\mathcal{P}(\square)}}(f) \right) \rightarrow \mathbf{0}$.

For the former one, it suffices to give a function $\llbracket C \rrbracket^{\widetilde{\mathcal{P}(\square)}}(f) \rightarrow \nabla_{N \rightarrow N} \llbracket C \rrbracket^{\mathcal{S}}(f)$ for all $f : \Delta(N \rightarrow N)$ by the recursion principle of the propositional truncation because the codomain is a proposition by Proposition 22. By the adjunction $(-)_0 \dashv \nabla_{N \rightarrow N}$, it suffices to give a function $\llbracket C \rrbracket_0^{\widetilde{\mathcal{P}(\square)}} \rightarrow \llbracket C \rrbracket^{\mathcal{S}}$ but we have an isomorphism $\llbracket C \rrbracket^{\widetilde{\mathcal{P}(\square)}} \cong (\Delta \llbracket C \rrbracket^{\mathcal{S}})_0 = \llbracket C \rrbracket^{\mathcal{S}}$ by Proposition 19.

For the latter function, observe that $\prod_{f:\Delta(N \rightarrow N)} \nabla_{N \rightarrow N} \llbracket C \rrbracket^{\mathcal{S}}(f) \cong \nabla_1 \left(\prod_{f:N \rightarrow N} \llbracket C \rrbracket^{\mathcal{S}}(f) \right)$ and that $\nabla_1 \mathbf{0} \cong \mathbf{0}$. Then we apply ∇_1 to the function $\left(\prod_{f:N \rightarrow N} \llbracket C \rrbracket^{\mathcal{S}}(f) \right) \rightarrow \mathbf{0}$ in \mathcal{S} obtained from the inconsistency of Church's Thesis with the axiom of choice and function extensionality. \square

5. Null Types

Let \mathcal{E} be a model of univalent type theory. Based on Rijke, Shulman and Spitters' null types (Rijke et al. 2020), we define a notion of null structure as follows.

Let $a : A \vdash B(a)$ be a proposition in \mathcal{E} . For a type $\Gamma \vdash X$ in \mathcal{E} , we define a proposition $\Gamma \vdash \text{isNull}_B(X)$ as

$$\Gamma \vdash \prod_{a:A} \text{isEquiv}(\lambda(x : X). \lambda(b : B(a)). x)$$

and call a term of $\text{isNull}_B(X)$ a *B-null structure on X*. A *B-null type* is a type $\Gamma \vdash X$ equipped with a *B-null structure* n on X . That is, a *B-null type* has a witness that the canonical map $X \rightarrow X^{B(a)}$ is an equivalence for each a .

Definition 23. We define a *cfw* \mathcal{E}_B as follows:

- the contexts are those of \mathcal{E} ;
- the types are the *B-null types* in \mathcal{E} ;
- the elements of $\Gamma \vdash_{\mathcal{E}_B} X$ are those of the underlying type X in \mathcal{E} .

We have the obvious forgetful morphism $\mathcal{E}_B \rightarrow \mathcal{E}$ of *cfw*'s.

For a proposition $a : A \vdash B(a)$, a *nullification operator* assigns

- each type $\Gamma \vdash X$ a *B-null type* $\Gamma \vdash \mathcal{L}_B X$ and an element $\Gamma \vdash \eta_X : X \rightarrow \mathcal{L}_B X$; and
- each pair of type $\Gamma \vdash X$ and *B-null type* $\Gamma \vdash Y$ an element $\Gamma \vdash e : \text{isEquiv}(\lambda(f : \mathcal{L}_B X \rightarrow Y). f \circ \eta_X)$.

We also require that a nullification operator is preserved by reindexing.

We review some properties of null types. See Rijke et al. (2020) for further details.

Proposition 24. *Let $\Gamma \vdash X$ and $\Gamma, x : X \vdash Y(x)$ be types in \mathcal{E} .*

- *There exists a term of type $\Gamma \vdash (\prod_{x:X} \text{isNull}_B(Y(x))) \rightarrow \text{isNull}_B(\prod_{x:X} Y(x))$.*
- *There exists a term of type $\Gamma \vdash \text{isNull}_B(X) \rightarrow (\prod_{x:X} \text{isNull}_B(Y(x))) \rightarrow \text{isNull}_B(\sum_{x:X} Y(x))$.*
- *There exists a term of type $\Gamma \vdash \text{isNull}_B(X) \rightarrow \prod_{x_0, x_1 : X} \text{isNull}_B(\text{Id}_X(x_0, x_1))$.*

Consequently, \mathcal{E}_B supports dependent product, dependent sum and intensional identity types preserved by the morphism $\mathcal{E}_B \rightarrow \mathcal{E}$.

For a universe U , we define a subuniverse U_B of U as

$$U_B \equiv \{X : U \mid \text{isNull}_B(X)\}.$$

Proposition 25. *The universe U_B has a B -null structure.*

Proof. Our condition that each $B(a)$ is a proposition corresponds to Rijke, Shulman and Spitters’ notion of *topological modality*. They prove in Rijke et al. (2020, Corollary 3.11 and Theorem 3.12) that any such modality is lex and hence that the universe of modal types is itself modal. \square

Proposition 26. *If a nullification operator \mathcal{L}_B exists, then it preserves propositions.*

Proof. This is true for any modality by Rijke et al. (2020, Lemma 1.28). \square

Corollary 27. *Suppose that \mathcal{E} has a nullification operator \mathcal{L}_B . Then $a : A \vdash \mathcal{L}_B B(a)$ is contractible.*

Proof. Since $\mathcal{L}_B B(a)$ is a proposition, it suffices to find an element of $\prod_{a:A} \mathcal{L}_B B(a)$. Assume that $a : A$ is given. Since $\mathcal{L}_B B(a)$ is B -null, it is enough to give a function $B(a) \rightarrow \mathcal{L}_B B(a)$, so take the constructor $\eta_{B(a)} : B(a) \rightarrow \mathcal{L}_B B(a)$. \square

Corollary 28. *Suppose that \mathcal{E} has a nullification operator \mathcal{L}_B . Then $X \mapsto \mathcal{L}_B \|X\|$ gives propositional truncation in the model \mathcal{E}_B .*

Proof. By Proposition 26, $\mathcal{L}_B \|X\|$ is a proposition. For any B -null proposition Z , we have equivalences

$$\begin{aligned} (\mathcal{L}_B \|X\| \rightarrow Z) &\simeq (\|X\| \rightarrow Z) \\ &\simeq (X \rightarrow Z). \end{aligned}$$

\square

5.1 Null types in Orton–Pitts models

Let $\tilde{\mathcal{E}}$ be an Orton–Pitts model.

Definition 29. *A type $a : A \vdash B(a)$ in \mathcal{E} or $\tilde{\mathcal{E}}$ is said to be well-supported if the propositional truncation $a : A \vdash_{\mathcal{E}} \|B(a)\|$ taken in the model \mathcal{E} of extensional dependent type theory is inhabited.*

Proposition 30. *Let $1 \vdash_{\mathcal{E}} X$ be a type and $a : A \vdash_{\tilde{\mathcal{E}}} B(a)$ a proposition. If X is discrete and B is well-supported, then X has a B -null structure.*

Proof. We show that, for any $a : A$, the function $k_a := \lambda x. \lambda b. x : X \rightarrow (B(a) \rightarrow X)$ is an isomorphism in the internal language of \mathcal{E} . Since B is well-supported, k_a is injective. To prove surjectivity, we assume that $f : B(a) \rightarrow X$ is given. By the well-supportedness of B , there exists

some element $b : B(a)$. We show that $f = k_a(f(b))$. Assume $b' : B(a)$ is given. Since B is a proposition in $\tilde{\mathcal{E}}$, we have a path $p : \mathbb{I} \rightarrow B(a)$ such that $p0 = b'$ and $p1 = b$. By the discreteness of X the path $f \circ p : \mathbb{I} \rightarrow X$ is constant, which implies that $f(b') = f(b)$. Hence, we have $f = k_a(f(b))$ by function extensionality. \square

We easily deduce the following corollaries.

Corollary 31. *If B is well-supported, then $\mathbf{0}$ has a B -null structure.*

Corollary 32. *If B is well-supported, then \mathbb{N} has a B -null structure.*

Proposition 33. *Let $a : A \vdash_{\tilde{\mathcal{E}}} B(a)$ be a proposition and $\Gamma \vdash_{\tilde{\mathcal{E}}} X$ and $\Gamma \vdash_{\tilde{\mathcal{E}}} Y$ types. Then there exists a term of type*

$$\Gamma \vdash \text{isNull}_B(X) \rightarrow \text{isNull}_B(Y) \rightarrow \text{isNull}_B(X + Y).$$

Proof. We proceed in the internal language of \mathcal{E} . Suppose that X and Y has a B -null structure. Assume that $a : A$ is given. Since the function $(X + Y) \rightarrow (B(a) \rightarrow (X + Y))$ factors as

$$\begin{array}{ccc} X + Y & \xrightarrow{\quad\quad\quad} & B(a) \rightarrow (X + Y) \\ & \searrow & \nearrow \Phi \\ & (B(a) \rightarrow X) + (B(a) \rightarrow Y), & \end{array}$$

it suffices to show that the function $\Phi : ((B(a) \rightarrow X) + (B(a) \rightarrow Y)) \rightarrow (B(a) \rightarrow (X + Y))$ is an isomorphism. The injectivity of Φ follows from the well-supportedness of B . To prove the surjectivity, we assume that $f : B(a) \rightarrow (X + Y)$ is given. We show that $(\forall_{b:B(a)} fb \in X) \vee (\forall_{b:B(a)} fb \in Y)$. Since B is well-supported, there exists some element $b_0 : B(a)$. We know that $fb_0 \in X \vee fb_0 \in Y$. Suppose that $fb_0 \in X$. Assume $b : B(a)$ is given. Since B is a proposition in \mathcal{E} , we have a path $p : \mathbb{I} \rightarrow B(a)$ such that $p0 = b_0$ and $p1 = b$. Since the exponential functor $(-)^{\mathbb{I}}$ preserves colimits because it has a right adjoint, we have $(\forall_{i:\mathbb{I}} f(pi) \in X) \vee (\forall_{i:\mathbb{I}} f(pi) \in Y)$. Now $f(p0) \in X$ and thus we have $\forall_{i:\mathbb{I}} f(pi) \in X$. In particular, $fb \in X$. In a similar manner, we have $\forall_{b:B(a)} fb \in Y$ assuming $fb_0 \in Y$. Hence, we get $(\forall_{b:B(a)} fb \in X) \vee (\forall_{b:B(a)} fb \in Y)$. \square

By Corollary 18, and Propositions 24, 30 and 33, for any type X defined in dependent type theory only using dependent product types, dependent sum types, identity types, unit type, disjoint finite coproducts and natural numbers, the underlying type of $\llbracket X \rrbracket^{\tilde{\mathcal{E}}_B}$ is equal to $\llbracket X \rrbracket^{\tilde{\mathcal{E}}}$ for any well-supported proposition B in $\tilde{\mathcal{E}}$. In particular, in this case if $\llbracket X \rrbracket^{\tilde{\mathcal{E}}}$ is inhabited, then so is $\llbracket X \rrbracket^{\tilde{\mathcal{E}}_B}$.

Example 34. *Markov's Principle* is the following axiom.

$$\forall \alpha : \mathbb{N} \rightarrow \mathbf{2} \neg \neg (\exists n : \mathbb{N} \alpha(n)) \rightarrow \exists n : \mathbb{N} \alpha(n)$$

It is equivalent to the type

$$\prod_{\alpha : \mathbb{N} \rightarrow \mathbf{2}} \prod_{p : (\prod_{n:\mathbb{N}} \alpha(n) \rightarrow \mathbf{0}) \rightarrow \mathbf{0}} \left\| \sum_{n:\mathbb{N}} \alpha(n) \right\|.$$

For a decidable predicate $\alpha : \mathbb{N} \rightarrow \mathbf{2}$, the proposition $\left\| \sum_{n:\mathbb{N}} \alpha(n) \right\|$ is equivalent to the type $\sum_{n:\mathbb{N}} \alpha(n) \times \prod_{k:\mathbb{N}} \alpha(k) \rightarrow n \leq k$ which is defined without propositional truncation. Hence, if the model \mathcal{E} of extensional dependent type theory satisfies Markov's Principle, then so does the model $\tilde{\mathcal{E}}_B$ of univalent type theory for any well-supported proposition B in $\tilde{\mathcal{E}}$.

We now show how to define nullification operators in Orton–Pitts models. Following Rijke et al. (2020, Section 2.2), we will first define an operator \mathcal{J}_B , although we will only consider the case of nullification, since that is all we need here.

Lemma 35. *For types $a : A \vdash_{\tilde{\mathcal{E}}} B(a)$ and $\Gamma \vdash_{\tilde{\mathcal{E}}} X$, we have the higher inductive $\mathcal{J}_B(X)$ defined as follows.*

- When $x : X$, then $\mathcal{J}_B(X)$ contains an element $\alpha_X^B(x)$.
- When $a : A$ and $f : B(a) \rightarrow \mathcal{K}_B$, then $\mathcal{J}_B(X)$ contains an element $\text{ext}(a, f)$.
- When $a : A$, $f : B(a) \rightarrow \mathcal{K}_B$ and $b : B(a)$ then $\text{Id}(\text{ext}(a, f), f(b))$ contains an element $\text{isext}(a, f, b)$.

Proof. $\mathcal{J}_B(X)$ differs from \mathcal{K}_B by having an extra point constructor $\alpha_X^B : X \rightarrow \mathcal{J}_B(X)$. We define A' to be the type $A + X$ and define the family of types $a : A' \vdash B'(a)$ as follows.

$$\begin{aligned} B'(\text{inl}(a)) &::= B(a) \\ B'(\text{inr}(x)) &::= 0 \end{aligned}$$

We can then take $\mathcal{J}_B(X)$ to be $\mathcal{K}_{B'}$, as defined in Section 3.1. We take $\alpha_X^B(x)$ to be $\text{ext}(\text{inr}(x), \perp_{\mathcal{K}_B})$ where $\perp_{\mathcal{K}_B}$ is the unique map from 0 to \mathcal{K}_B . □

Theorem. $\tilde{\mathcal{E}}$ has a nullification operator \mathcal{L}_B for every type $a : A \vdash_{\tilde{\mathcal{E}}} B(a)$.

Proof. This follows from Rijke et al. (2020, Theorem 2.18), observing that for the case of nullification the pushout appearing there is just a suspension, which we have already shown how to implement in Theorem 3.1, and we showed in Lemma 35 how to implement their \mathcal{J} operator. □

6. Church’s Thesis in Null Types

Consider a dependent type theory with dependent product types, dependent sum types, identity types, unit type, disjoint finite coproducts, propositional truncation and natural numbers. Let $a : A \vdash B(a)$ be a type in this type theory where A and B are definable only using dependent product types, dependent sum types, identity type of $\mathbf{2}$, unit type, finite coproducts and natural numbers. We define $a : A \vdash C(a) := \|B(a)\|$. For an Orton–Pitts model $\tilde{\mathcal{E}}$, the underlying types of the interpretations $\llbracket A \rrbracket^{\tilde{\mathcal{E}}}$ and $\llbracket B \rrbracket^{\tilde{\mathcal{E}}}$ are $\llbracket A \rrbracket^{\mathcal{E}}$ and $\llbracket B \rrbracket^{\mathcal{E}}$, respectively.

Theorem. Let $\tilde{\mathcal{E}}$ be an Orton–Pitts model. Suppose that $\llbracket B \rrbracket^{\tilde{\mathcal{E}}}$ is well-supported. Then the proposition C holds in the model of univalent type theory $\tilde{\mathcal{E}}_{\llbracket C \rrbracket^{\tilde{\mathcal{E}}}}$.

Proof. Let $D = \llbracket C \rrbracket^{\tilde{\mathcal{E}}}$. By assumption $\llbracket B \rrbracket^{\tilde{\mathcal{E}}}$ is well-supported and so is its truncation D . By Corollary 18 and Propositions 24, 30 and 33, the underlying types of $\llbracket A \rrbracket^{\tilde{\mathcal{E}}^D}$ and $\llbracket B \rrbracket^{\tilde{\mathcal{E}}^D}$ are equal to $\llbracket A \rrbracket^{\tilde{\mathcal{E}}}$ and $\llbracket B \rrbracket^{\tilde{\mathcal{E}}}$, respectively. Hence, we have $\llbracket C \rrbracket^{\tilde{\mathcal{E}}^D} = \mathcal{L}_D \llbracket C \rrbracket^{\tilde{\mathcal{E}}}$ by Corollary 28, and this type is inhabited by Corollary 27. □

Corollary 36. *Let \mathcal{S} be a model of type theory as in Section 3.2. If the proposition C holds in \mathcal{S} , then C also holds in the model of univalent type theory $\tilde{\mathcal{E}}_{\llbracket C \rrbracket^{\tilde{\mathcal{E}}}}$ where $\mathcal{E} = \mathcal{P}(\square)$.*

Proof. Since $C = \|B\|$ holds in \mathcal{S} , the type $\llbracket B \rrbracket^{\mathcal{S}}$ is well-supported. Then $\llbracket B \rrbracket^{\tilde{\mathcal{E}}} = \llbracket B \rrbracket^{\mathcal{E}}$ is also well-supported because the constant presheaf functor $\mathcal{S} \rightarrow \mathcal{E}$ preserves all structures of the type theory. Then use Theorem 6. \square

Example 37. Recall that Church’s Thesis is equivalent to the type

$$\prod_{f:\mathbb{N} \rightarrow \mathbb{N}} \left\| \sum_{e:\mathbb{N}} \prod_{x:\mathbb{N}} \sum_{z:\mathbb{N}} T(e, x, z) \times U(z) = f(x) \right\|.$$

Also note that the equality of natural numbers is decidable, and thus there exists a function $=_{\mathbb{N}}: \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbf{2}$ such that the type $U(z) = f(x)$ is equivalent to $(U(z) =_{\mathbb{N}} f(x)) = 1$. Therefore, Church’s Thesis is equivalent to a type of the form

$$\prod_{a:A} \|B(a)\|$$

with a type $a : A \vdash B(a)$ definable only using dependent product types, dependent sum types, identity of $\mathbf{2}$, unit type, finite coproducts and natural numbers. Since Church’s Thesis holds in the category $\mathbf{Asm}(\mathcal{K}_1)$ of assemblies on Kleene’s first model \mathcal{K}_1 , by Corollary 36 the model of univalent type theory $\tilde{\mathcal{E}}_{\llbracket C \rrbracket^{\tilde{\mathcal{E}}}}$ satisfies Church’s Thesis where $\mathcal{E} = \mathbf{CAsm}(\mathcal{K}_1)$.

We can now prove our second main result, which informally says that univalent type theory is consistent with the main principles of Recursive Constructive Mathematics.

Theorem. Martin–Löf type theory remains consistent when all of the following extra structure and axioms are added.

- (1) Propositional truncation.
- (2) The axiom of univalence.
- (3) Church’s Thesis.
- (4) Markov’s Principle.

Proof. We prove consistency by constructing a model where all of the above holds and where there is no element of type \perp . Consider the Orton–Pitts model \mathcal{E} with $\mathcal{E} = \mathbf{CAsm}(\mathcal{K}_1)$. We have seen that \mathcal{E} satisfies Church’s Thesis in Example 37. It remains to show that \mathcal{E} satisfies Markov’s Principle and \perp is empty in this model.

Using well supportness again, and Example 34 we see that to show Markov’s principle holds, it suffices to show it holds in cubical assemblies (as a model of extensional type theory). Again, we observe that the type corresponding to Markov’s principle is preserved by the constant presheaves functor, and so it suffices to show that Markov’s principle holds in assemblies, which is again a standard argument.

Using well supportness once more, and Corollary 31, we see that \perp is the same in null types as in cubical assemblies. It follows that it has no global sections, i.e. there is no element of type \perp in the model. \square

We can use Theorem 6 for other principles.

Example 38. Brouwer’s Continuity Principle is the following axiom.

$$\forall_{F:(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}} \forall_{\alpha:\mathbb{N} \rightarrow \mathbb{N}} \exists_{n:\mathbb{N}} \forall_{\beta:\mathbb{N} \rightarrow \mathbb{N}} (\forall_{m:\mathbb{N}} m < n \rightarrow \alpha(m) = \beta(m)) \rightarrow F(\alpha) = F(\beta)$$

The standard ordering $<$ on \mathbb{N} is decidable, and thus Brouwer's Continuity Principle is an instance of Theorem 6.

We obtain a new proof of the following result due to Coquand et al. (2021).

Theorem. Martin–Löf type theory remains consistent when all of the following extra structure and axioms are added.

- (1) Propositional truncation.
- (2) The axiom of univalence.
- (3) Brouwer's Continuity Principle.

Proof. This is the same as for Theorem 6. See e.g. van Oosten (2008, Proposition 3.1.6) for a proof that Brouwer's principle holds in the effective topos (the same proof applies for assemblies). \square

The original proof by Coquand, Ruch and Sattler used cubical sheaves. Mike Shulman has pointed out that it can also be proved using the main result of Shulman (2019): there are well-known examples of Grothendieck toposes where Brouwer's continuity principle holds, and using a Grothendieck $(\infty, 1)$ -topos whose corresponding 1-topos of hsets is such a topos, one obtains a model of univalent type theory where Brouwer's continuity principle holds. Both the Coquand–Ruch–Sattler proof and Shulman's proof can also be said to use modalities, although in a very different way to our proof.

7. Conclusion and Further Work

We have constructed a model of type theory that satisfies the main axiom of homotopy type theory (univalence) and the main axioms of recursive constructive mathematics (Church's thesis and Markov's principle). However, in both fields, there are additional axioms that are natural to consider, but which we have left for future work.

With regards to homotopy type theory, we expect that the remaining higher inductive types appearing in Univalent Foundations Program (2013) can be implemented following the technique suggested in Rijke et al. (2020, Remark 3.24) together with the technique of Coquand et al. (2018) for constructing the necessary higher inductive types in cubical assemblies.

The situation with the remaining axioms of recursive constructive mathematics is more difficult. The axiom of countable choice is often included, but it is unclear whether countable choice holds in our model, or how to adjust the model to ensure countable choice does hold. The other main axiom of recursive constructive mathematics is extended Church's thesis, which states that certain partial functions from \mathbb{N} to \mathbb{N} are computable. The main issue here is that it is unclear what is the most natural way to formulate partial functions in homotopy type theory. Much progress on this has been made by Escardó and Knapp (2017). However, as they show, a weak form of countable choice is needed for their definition to work as expected. We expect that for any reasonable formulation of extended Church's thesis Theorem 6 can be used to construct a model where it holds.

Another open problem is to find a good definition of $(\infty, 1)$ -effective topos, which should be to the effective topos what $(\infty, 1)$ -toposes are to Grothendieck toposes. In particular, the effective topos should be recovered as the localisation of the hsets in the $(\infty, 1)$ -effective topos, and commonly seen theorems and definitions in the effective topos should be special cases of corresponding higher versions. One possible definition is cubical assemblies. We can now see another possibility in the form of reflective subuniverses of cubical assemblies. However, our definition is dependent on a particular choice of axioms that satisfy the necessary conditions to apply

Theorem 6, so we leave open the problem of finding a ‘natural’ definition that satisfies axioms such as Church’s thesis without needing to ensure they hold in the definition. One possible approach is to try to find a version of our construction that applies to all well supported maps at once. Another possibility is to further develop an alternative, more direct approach by Van den Berg (2020).

Acknowledgements. The first author is grateful for some helpful discussions on higher inductive types with Simon Huber, Anders Mörtberg and Christian Sattler at the Hausdorff Research Institute for Mathematics during the trimester program Types, Sets and Constructions. We are grateful to Benno van den Berg, Thierry Coquand and Mike Shulman for helpful comments and corrections.

Financial Support. The second author is supported by the research programme ‘The Computational Content of Homotopy Type Theory’ with project number 613.001.602, which is financed by the Netherlands Organisation for Scientific Research (NWO).

Notes

- 1 Surprisingly, however, this version is consistent with intensional type theory as long as one drops the ξ rule, which was proved by Ishihara, Maietti, Maschio and Streicher in Ishihara et al. (2018). They leave the case where one has the ξ rule but not function extensionality open and to the authors’ knowledge it remains an open problem.
- 2 Equivalently, the map is a monomorphism, and for this reason hpropositions are sometimes referred to as *mono types*.
- 3 Pushout in the usual categorical sense, not the homotopy pushout.

References

- Awodey, S. and Bauer, A. (2004). Propositions as [types]. *Journal of Logic and Computation* **14** (4) 447–471.
- Coquand, T., Huber, S. and Mörtberg, A. (2018). On higher inductive types in cubical type theory. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS’18*, New York, NY, USA, ACM, 255–264.
- Coquand, T., Ruch, F. and Sattler, C. (2021). Constructive sheaf models of type theory. *Mathematical Structures in Computer Science*, 1–24.
- Dybjer, P. (1996). Internal type theory. In: Berardi, S. and Coppo, M. (eds.) *Types for Proofs and Programs: International Workshop, TYPES’95 Torino, Italy, June 5–8, 1995 Selected Papers*, Berlin, Heidelberg, Springer, 120–134.
- Escardó, M. H. and Knapp, C. M. (2017). Partial elements and recursion via dominances in univalent type theory. In: Goranko, V. and Dam, M. (eds.) *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 82, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 21:1–21:16.
- Gambino, N. and Hyland, M. (2004). Wellfounded trees and dependent polynomial functors. In: Berardi, S., Coppo, M. and Damiani, F. (eds.) *Types for Proofs and Programs: International Workshop, TYPES 2003, Torino, Italy, April 30–May 4, 2003, Revised Selected Papers*, Berlin, Heidelberg, Springer, 210–225.
- Ishihara, H., Maietti, M. E., Maschio, S. and Streicher, T. (2018). Consistency of the intensional level of the minimalist foundation with Church’s thesis and axiom of choice. *Archive for Mathematical Logic* **57** (7) 873–888.
- Kleene, S. C. (1945). On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic* **10** (4) 109–124.
- Licata, D. R., Orton, I., Pitts, A. M. and Spitters, B. (2018). Internal universes in models of homotopy type theory. In: Kirchner, H. (ed.) *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 108, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 22:1–22:17.
- Maietti, M. E. (2005). Modular correspondence between dependent type theories and categories including pretopoi and topoi. *Mathematical Structures in Computer Science* **15** (06) 1089–1149.
- Menni, M. (2001). Closure operators in exact completions. *Theory and Applications of Categories* **8** (21) 522–540.
- Orton, I. and Pitts, A. M. (2018). Axioms for modelling cubical type theory in a topos. *Logical Methods in Computer Science* **14** (4) 1–33.
- Rijke, E., Shulman, M. and Spitters, B. (2020). Modalities in homotopy type theory. *Logical Methods in Computer Science* **16** (1) 2:1–2:79.
- Shulman, M. (2019). All $(\infty, 1)$ -toposes have strict univalent universes. arXiv preprint, [arXiv:1904.07004](https://arxiv.org/abs/1904.07004).
- Spitters, B. (2016). Cubical sets and the topological topos. arXiv preprint, [arXiv:1610.05270](https://arxiv.org/abs/1610.05270).
- Swan, A. W. (2018). W types with reductions and the small object argument. arXiv preprint, [arXiv:1802.07588](https://arxiv.org/abs/1802.07588).
- Troelstra, A. and van Dalen, D. (1988). *Constructivism in Mathematics, Volume I*, Studies in Logic and the Foundations of Mathematics, vol. 121, Elsevier.

- Uemura, T. (2019). Cubical assemblies, a univalent and impredicative universe and a failure of propositional resizing. In: Dybjer, P., Santo, J. E. and Pinto, L. (eds.) *24th International Conference on Types for Proofs and Programs (TYPES 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 130, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 7:1–7:20.
- Univalent Foundations Program (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. <http://homotopytypetheory.org/book>, Institute for Advanced Study.
- van den Berg, B. (2006). *Predicative Topos Theory and Models for Constructive Set Theory*. Phd thesis, University of Utrecht.
- van den Berg, B. (2020). Univalent polymorphism. *Annals of Pure and Applied Logic* **171** (6) 102793, 29.
- van Oosten, J. (2008). *Realizability: An Introduction to its Categorical Side*, Studies in Logic and the Foundations of Mathematics, vol. 152, Elsevier.