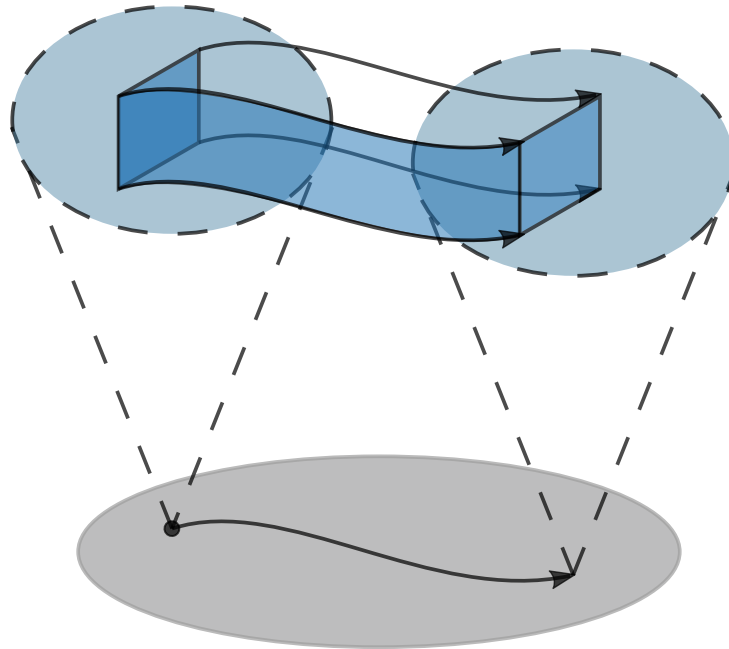


Cubical models of homotopy type theory — *an internal approach*



Richard Ian Orton

November 2018

This dissertation is submitted for the degree of Doctor of Philosophy

CHURCHILL
COLLEGE 

 UNIVERSITY OF
CAMBRIDGE

Abstract

This thesis presents an account of the cubical sets model of homotopy type theory using an internal type theory for elementary topoi.

Homotopy type theory is a variant of Martin-Löf type theory where we think of types as spaces, with terms as points in the space and elements of the identity type as paths. We actualise this intuition by extending type theory with Voevodsky’s univalence axiom which identifies equalities between types with homotopy equivalences between spaces.

Voevodsky showed the univalence axiom to be consistent by giving a model of homotopy type theory in the category of Kan simplicial sets in a paper with Kapulkin and Lumsdaine. However, this construction makes fundamental use of classical logic in order to show certain results. Therefore this model cannot be used to explain the computational content of the univalence axiom, such as how to compute terms involving univalence.

This problem was resolved by Cohen, Coquand, Huber and Mörtberg, who presented a new model of type theory in Kan cubical sets which validated the univalence axiom using a constructive metatheory. This meant that the model provided an understanding of the computational content of univalence. In fact, the authors present a new type theory, cubical type theory, where univalence is provable using a new “glueing” type former. This type former comes with appropriate definitional equalities which explain how the univalence axiom should compute. In particular, Huber proved that any term of natural number type constructed in this new type theory must reduce to a numeral.

This thesis explores models of type theory based on the cubical sets model of Cohen et al. It gives an account of this model using the internal language of toposes, where we present a series of axioms which are sufficient to construct a model of cubical type theory, and hence a model of homotopy type theory. This approach therefore generalises the original model and gives a new and useful method for analysing models of type theory.

We also discuss an alternative derivation of the univalence axiom and show how this leads to a potentially simpler proof of univalence in any model satisfying the axioms mentioned above, such as cubical sets.

Finally, we discuss some shortcomings of the internal language approach with respect to constructing univalent universes. We overcome these difficulties by extending the internal language with an appropriate modality in order to manipulate global elements of an object.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.



Richard Ian Orton

January 23, 2019

Publications and collaborations: The work presented in this thesis is largely based on the publications listed below and is the result of work done in collaboration with my coauthors.

- Ian Orton, Andrew Pitts, *Axioms for modelling cubical type theory in a topos*, Logical Methods in Computer Science (LMCS), 2018. Special issue for CSL 2016, to appear; arXiv:1712.04864.
- Ian Orton, Andrew Pitts, *Decomposing the univalence axiom*, Leibniz International Proceedings in Informatics (LIPIcs), 2018. Post-proceedings for TYPES 2017, to appear; arXiv:1712.04890.
- Dan Licata, Ian Orton, Andrew Pitts, Bas Spitters, *Internal universes in models of homotopy type theory*, Leibniz International Proceedings in Informatics (LIPIcs), 2018. Proceedings for FSCD 2018. arXiv:1801.07664.

Acknowledgements

First and foremost I would like to thank my supervisor Andrew Pitts for his guidance and collaboration over the past three years. Our many meetings and discussions were invaluable and I have learned an incredible amount from him over the course of my PhD. Andy was always available to offer advice and guidance while leaving me free to tackle interesting problems as I saw fit. He has been both a fantastic supervisor and coauthor.

I would also like to thank my other coauthors, Dan Licata and Bas Spitters, for the chance to collaborate, as well as for their insights and discussions, on our paper and many other things. I also owe thanks to countless other members of the HoTT/UF community for detailed discussions, both over email and over coffee at various workshops and conferences, including (but not limited to): Thorsten Altenkirch, Carlo Angiuli, Steve Awodey, Guillaume Brunerie, Paolo Capriotti, Thierry Coquand, Floris van Doorn, Martín Escardó, Kuen-Bang Hou (Favonia), Jonas Frey, Simon Huber, Nicolai Kraus, Peter LeFanu Lumsdaine, Anders Mörtberg, Egbert Rijke, Christian Sattler and Jon Sterling.

In addition, I would like to thank my colleagues at the Computer Lab. My cohort of PhD students: Matthew Daggitt, Dylan McDermott, Hugo Paquet and Philip Saville, for many hours of discussion, many lunches at the Cavendish and generally for being excellent friends over the past three years. I would also like to thank the various other CL members, ex-members and visitors for their insights and friendship: Mohammad Abdulaziz, Aurore Alcolei, Simon Castellan, Pierre Clairambault, Marcelo Fiore, Tamara von Glehn, Tim Griffin, Ohad Kammar, Tamas Kispeter, Guillaume Munch-Maccagnoni and Jean Pichon-Pharabod. I would also like to thank my friends in the computer architecture group, Sam Ainsworth and Colin Rothwell, for their friendship over the past seven years and for, in Colin's words, their "entertainingly singular take on life in the Computer Lab."

Over the course of my PhD I have had fantastic opportunities to travel and collaborate with researchers from across the world. In particular I would like to thank Steve Awodey and Carnegie Mellon University for hosting me in Pittsburgh, and Bas Spitters and Aarhus University for hosting me in Aarhus. I would also like to thank the EPSRC for their generous funding and Lise Gough for dealing with my many travel requests.

Finally I would like to thank my family and friends for their support, and particularly my wonderful partner Zoe Pillidge for her companionship over the past several years.

Contents

1	Introduction	13
1.1	Layperson’s introduction	13
1.2	Expert’s introduction	16
1.3	Structure of this thesis	18
2	Type theory: an introduction	19
2.1	Univalent type theory	19
2.1.1	Martin-Löf type theory	20
2.1.2	Identity types and UIP	21
2.1.3	Voevodsky’s univalence axiom	22
2.1.4	Function extensionality	24
2.1.5	An alternative formulation of univalence	25
2.1.6	The failure of UIP	26
2.2	Cubical type theory	27
2.2.1	The formal interval \mathbb{I}	27
2.2.2	Path types	28
2.2.3	The face lattice \mathbb{F}	28
2.2.4	Kan composition	30
2.2.5	Glueing	30
2.2.6	Univalence	31
3	Models of Martin-Löf type theory	33
3.1	Categories with Families (CwFs)	33
3.1.1	Additional structure on a CwF	34
3.2	Presheaf models of type theory	36
3.3	A model in cubical sets	38
3.3.1	The interval and face lattice	39
3.3.2	Fibrant types	40
3.3.3	Path types	41
3.3.4	Glueing	42

3.3.5	A univalent universe	42
4	The internal type theory of a topos	45
4.1	Elementary topoi	45
4.1.1	Examples	46
4.2	The internal type theory	46
4.3	The proof assistant Agda	51
4.4	A remark on impredicativity	52
5	Axioms for modelling cubical type theory in a topos	55
5.1	The axioms	56
5.1.1	The interval I	56
5.1.2	Cofibrant propositions	57
5.1.3	The strictness axiom	62
5.2	Path types	63
5.3	Cohen-Coquand-Huber-Mörtberg (CCHM) fibrations	65
5.3.1	Composition and filling structures	65
5.3.2	Properties of fibrations	69
5.3.3	Type formers and simple datatypes	71
5.3.4	Path and identity types	75
5.4	Glueing	78
5.5	Univalence	84
5.6	Satisfying the axioms	89
5.6.1	The interval object ($\mathbf{ax}_1\text{--}\mathbf{ax}_4$)	89
5.6.2	Cofibrant propositions ($\mathbf{ax}_5\text{--}\mathbf{ax}_8$) and the strictness axiom (\mathbf{ax}_9)	90
5.7	Related work	94
6	Decomposing the univalence axiom	97
6.1	The decomposition	97
6.2	Applications in models of type theory	100
6.2.1	Strictification	101
6.2.2	Misaligned paths between fibrations	102
6.2.3	Function extensionality	104
6.2.4	Axioms (1), (2), (4) and (5)	104
6.2.5	Axiom (3)	106
6.3	Conclusion	108
7	Internal universes	111
7.1	The “no-go” theorem for internal universes	111
7.2	Crisp type theory	113

7.3	Universes from tiny intervals	116
7.4	Applications	123
7.5	Conclusion	125
8	Conclusion	127
8.1	Future work (higher inductive types)	128
	Bibliography	133

Chapter 1

Introduction

This introduction is divided into two sections. The first offers an introduction for a layperson. This is intended for friends, family, or anyone else with a school-level background in mathematics who happens to stumble across this thesis. My hope is that these few pages will offer at least some insight into what I've been doing for the past three years.

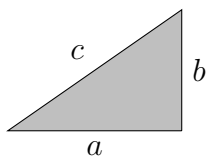
The second introduction is aimed at experts, who may be more familiar with some of the technical background involved: type theory, category theory, etc. It is intended to be accessible to anyone with a background in mathematics or theoretical computer science.

Neither introduction contains significant technical detail. This is covered in chapters 2, 3 and 4, which provide more detailed background on existing work.

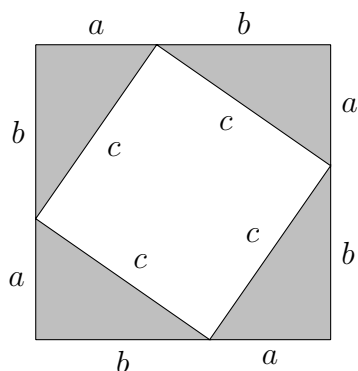
1.1 Layperson's introduction

What does it mean to prove a theorem in mathematics? Take, for example, Pythagoras' theorem: that for any right-angled triangle whose sides have lengths a , b and c , with c being the longest side, then it is always the case that $a^2 + b^2 = c^2$. Most people will have learned this fact in school, but how do we know that it is true? How can we *prove* this theorem?

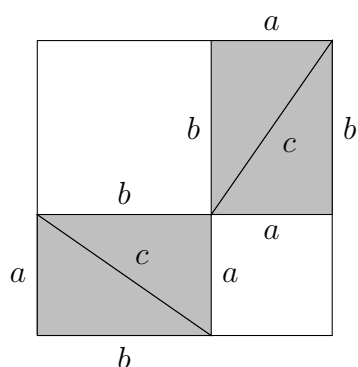
In this case we could start by considering any right-angled triangle and labelling its sides a , b and c :



We can then take four copies of this triangle and arrange them together to form a square, like so:



Note that the white, unfilled region in the centre is a square with sides of length c . Therefore the area of this unfilled space is c^2 . Next, we rearrange the triangles inside the same outer square like so:



Now the amount of unfilled space is given by the sum of the areas of the two white squares, specifically $a^2 + b^2$. But we have not changed the amount of filled/unfilled space. Therefore the unfilled area in this diagram, $a^2 + b^2$, is the same as the unfilled area in the previous diagram, c^2 , and hence $a^2 + b^2 = c^2$.

Is this a proof of Pythagoras' theorem? Can you be sure that this will work for any values a , b and c , and not just the ones that happen to have been used in the diagram? How do you know that this is not a trick or an optical illusion, and perhaps the triangles in the second diagram are different from those in the first? Does the argument about the diagrams having the same amount of "unfilled space" really make sense?

These questions all stem from a more fundamental question: what exactly is a proof? Can we come up with a precise definition of what qualifies as a proof, and then use that to decide whether or not the argument above is a valid proof?

One way to approach this question is to think of a proof as a sequence of logical steps, where each step has to be correct according to some predefined set of rules. First, we fix a set of very basic facts as a starting point, for example that $1 + 1 = 2$. We take these facts to be true without any proof and call such things *axioms*. Next, we fix a set of rules which tell us how we can prove new facts from old facts. For example, a rule might say that if you prove that $x = y$ and you prove that x has some property, e.g. that x is an even

number, then you get a proof that y has that property too, e.g. that y is also even. Then, to prove a theorem we start with the axioms and apply the rules in order to prove new facts. The aim is to repeat this process, gradually proving more and more complicated things, until eventually we prove the theorem itself.

A specific instance of such axioms and rules is referred to as a *foundational system* for mathematics. Once the axioms and rules are fixed then we can ask questions such as: can we prove Pythagoras' theorem using these axioms and rules? It could be the case that we are not able to prove the theorem because our axioms are missing some important fact, or perhaps we are missing a rule that we need for a crucial step of the proof. In this case we say that the theorem is *independent*¹ of the foundational system. On the other hand, it could be that we have accidentally included an axiom which is not actually true, or a rule which doesn't work in all cases. In this case it could be that, not only can we prove Pythagoras' theorem, but we can also prove things which are clearly false, for example that $0 = 1$ or that 7 is an even number. We call such a system *inconsistent*.

One example of a foundational system is homotopy type theory (HoTT). HoTT is an extension of an existing foundational system known as Martin-Löf type theory, or simply "type theory", where we add one new axiom: Voevodsky's univalence axiom. Intuitively, this axiom states that it doesn't matter how you encode data in your proofs. For example, you can encode numbers in decimal form: 0, 1, 2, 3, 4, etc, or in binary form: 0, 1, 10, 11, 100, etc. If you prove a theorem about numbers using one encoding, then it should apply just as well to the other encoding. Traditionally there is no way to prove this in general using the rules of type theory. Instead, you would need to repeat the proof using the other encoding, making adjustments where necessary. In HoTT, the univalence axiom tells you that once you prove that the two encodings really are equivalent then you can treat them as being the same. Therefore the proof that you had using one encoding will automatically apply to the other encoding.

We want to use HoTT as a basis for doing mathematics. However, before doing so it is crucial to have a good understanding of the answers to questions such as: what can we and can't we prove using HoTT, which theorems are independent, is HoTT consistent, etc. In order to investigate these questions we study what are known as *models* of HoTT. Intuitively, a model is a sort of translation from the rules and axioms of HoTT into a more traditional form of mathematics (e.g. category theory) where it becomes easier to study properties of the rules and axioms, and therefore easier to answer the sort of questions discussed above.

One such model involves translating from HoTT into a class of mathematical structures known as *cubical sets*. Accordingly, this is known as the cubical sets model of HoTT. In fact there are several related models, all of which work in similar ways, and collectively we

¹Technically, independence means that we can't prove *or disprove* the theorem.

refer to these as “cubical models of homotopy type theory”, a.k.a the first half of the title of this thesis.

So far, everything described in this introduction covers existing work. The contribution of this thesis is captured by the second half of its title: “an internal approach”. This refers to the use of a technique known as using the *internal language* in order to simplify some aspects of the existing work. At a high level, this thesis takes some existing work (the cubical sets model of type theory) and applies a different technique (internal languages) to the problem. Doing so offers several advantages such as: simplifying the existing work, making it easier to understand certain aspects, generalising the existing work so that it now applies in other situations, and providing a better understanding of why the existing approach works.

1.2 Expert’s 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 axiomatic set theory we have the axiom of extensionality, which tells us that two sets are equal if they contain the same elements. This can be stated formally as:

$$\forall A. \forall B. ((\forall x. (x \in A \iff x \in B)) \implies A = B)$$

In the context of (univalent/homotopy) type theory we have Voevodsky’s univalence axiom [62, Section 2.10], which tells us, roughly speaking, that two types are equal if they are isomorphic. This can be stated in the language of type theory as:

$$\prod_{A, B : U} A \simeq B \rightarrow A = B$$

where $A \simeq B$ is the type of *equivalences* between A and B . So, for example, we might have a type of decimal natural numbers, \mathbb{N}_{dec} , with terms $0, 1, 2, 3, \dots : \mathbb{N}_{dec}$ and a type of binary natural numbers, \mathbb{N}_{bin} , with terms $0, 1, 10, 11, \dots : \mathbb{N}_{bin}$. It is easy to show how to swap back and forth between the two encodings, that is, that these two types are equivalent. Therefore by the univalence axiom we can deduce that $\mathbb{N}_{dec} = \mathbb{N}_{bin}$.

It might seem strange at first for these types to be equal. For example, 7 is an element of \mathbb{N}_{dec} , but not an element of \mathbb{N}_{bin} , so how can the two types be equal? The reason why this axiom is consistent is because the rules of type theory prevent you from forming statements such as “7 is an element of \mathbb{N}_{bin} ”. The rules ensure that all statements are phrased in a suitably generic way so that they can only make use of the externally observable properties

of a type, and not the specific encoding used.

Let's consider an example of a valid statement in type theory: the property of being a commutative monoid. We say that a type X forms a commutative monoid iff there exists an operation $_ + _ : X \rightarrow X \rightarrow X$ which is associative, commutative and comes with a unit $e : X$. This definition is well-formed in type theory because it is applicable to any arbitrary type; it does not make any assumptions about what the elements of X might look like.

It is possible to prove, in type theory, that \mathbb{N}_{dec} forms a commutative monoid, where $+$ is addition and e is 0. Since \mathbb{N}_{dec} and \mathbb{N}_{bin} are just different encodings of the same object (the natural numbers) then it should also be the case that \mathbb{N}_{bin} also forms a commutative monoid. However, without the univalence axiom, proving this would require manually reworking the proof for \mathbb{N}_{dec} by converting back and forth between the two encodings. The power of the univalence axiom is that it allows us to simply transport the monoid structure from \mathbb{N}_{dec} to \mathbb{N}_{bin} by virtue of the fact that $\mathbb{N}_{dec} \simeq \mathbb{N}_{bin}$ and hence $\mathbb{N}_{dec} = \mathbb{N}_{bin}$.

The simplest way to extend type theory with this new principle of univalence is simply to add it as an axiom. That is, to postulate the existence of a term which witnesses the principle of univalence. However, this approach presents a problem. Type theory is a constructive system whose primitives have *computational content*. This means that not only do we know how to construct terms, we also know how to compute with these terms. An example of such behaviour is that if we have a proof of a statement of the form “there exists a natural number n such that ...” then we should be able to actually compute n from the proof. However, by simply stating univalence as an axiom we do not know, a priori, how expressions involving univalence should compute.

In order to understand how such expressions should compute we study models of univalent type theory. Once we have a model of UTT then we can potentially compute the value of an expression simply by looking at its interpretation in the model. Or, more likely, we could use the model to justify new computation rules which we could add to the type theory in order to recover good computational properties.

The first model of univalent type theory, the simplicial sets model [39], makes inherent use of classical logic to decide certain undecidable propositions. Therefore, while it tells us that the univalence axiom is consistent, it does not in fact provide a good basis for restoring the computational content of type theory, since evaluating an expression in the model is not computable.

To resolve these issues Cohen et al. introduced the cubical sets model of type theory [18]. This new model is presented using a constructive metatheory, in which evaluating an expression in the model is computable, thereby providing a way to recover the computational content of type theory. This model provides the basis for most of the work presented in this thesis.

In general terms, this thesis analyses the cubical sets model using the internal language

of the cubical sets topos. There are several advantages to this approach. In particular, this approach: makes certain aspects of the cubical sets model clearer and easier to understand; allows us to generalise the existing model by axiomatising the properties required to construct a model of univalent type theory; and makes it clear which of these properties are required to model which aspects of the type theory.

The process of axiomatising the model is not always straightforward. For example, we will see in Chapter 7 that the problem of constructing univalent universes presents a particular challenge for this approach. However, we believe that doing so is a worthwhile endeavour. This approach provides a solid foundation for many applications. For example, in Chapter 6 we use this approach to give an alternative presentation of the univalence axiom. Other authors have also applied this work to other applications such as proving independence results [61].

1.3 Structure of this thesis

This thesis begins with some background information:

- In Chapter 2 we explain univalent/homotopy type theory in more detail and discuss an extension of it known as cubical type theory.
- In Chapter 3 we define what exactly a model of type theory is, and we briefly explain the cubical sets model of Cohen et al. [18].
- In Chapter 4 we explain exactly what an internal language is, specifically in the case of a topos.

We then cover the new results which form the main contribution of this thesis:

- In Chapter 5 we show how to use the internal language to axiomatise the cubical sets model of type theory in any elementary topos with appropriate structure. However, we are not able to define a univalent universe, although we do prove a universe-agnostic form of univalence.
- In Chapter 6 we give a decomposition of the univalence axiom into some alternative axioms and show that these are satisfied in any topos with the extra structure specified in Chapter 5. This is essentially a simplification of the universe-agnostic proof of univalence mentioned previously.
- In Chapter 7 we explain how to extend the internal language with a modality so as to be able to axiomatise certain global properties of the topos. Using this modality we show how to construct a univalent universe, thus resolving the shortcomings of the preceding chapters.

Chapter 2

Type theory: an introduction

In this chapter we present some background material necessary for reading the rest of this thesis. In particular, we introduce univalent type theory, also known as homotopy type theory, and explain Voevodsky’s univalence axiom. We then introduce the system of cubical type theory which can be seen as an extension, or refinement, of standard univalent type theory. The key feature of cubical type theory is that it provides a computational interpretation for the univalence axiom.

2.1 Univalent type theory

Univalent type theory (UTT), also known as homotopy type theory (HoTT), is a variant of intensional Martin-Löf type theory where we formalise the common mathematical practice of identifying isomorphic structures by introducing a new axiom known as Voevodsky’s *univalence axiom*. In doing so we are forced to rethink many of our intuitions about the nature of equality in intensional type theory.

In this section we will review the basic aspects of Martin-Löf type theory, including the notion of intensional identity types and will discuss a new intuition for thinking about elements of these identity types as paths in some topological space. We will see that this intuition leads us to question the principle that any two proofs of equality are themselves equal, often referred to as the *uniqueness of identity proofs* (UIP). We will then discuss Voevodsky’s univalence axiom, which captures the notion that two spaces are equivalent precisely when one can be continuously deformed into the other and vice-versa. Finally, we will see that the univalence axiom is in fact inconsistent with the principle of UIP.

The information presented here is intended as a summary. For the avoidance of doubt, the type theory that we describe here is exactly the one defined in Appendix A of the HoTT book [62].

2.1.1 Martin-Löf type theory

Martin-Löf type theory (MLTT) is a foundational system for mathematics based on types rather than the more usual notion of sets. It allows propositions to be encoded as types, whose terms are then proofs of the proposition in question. This interpretation is often referred to as the Curry-Howard correspondence, or sometimes as propositions-as-types. MLTT includes many components which we can use to build and manipulate types and terms. These include:

Dependent pair types - Given a type A and a family of types $B(x)$ ranging over $x : A$ we can form the dependent pair type, $\sum_{x:A} B(x)$, of pairs (a, b) where $a : A$ and $b : B(a)$. These come equipped with first and second projections `fst`, `snd` with the usual computation rules. In the case where B does not depend on $x : A$ we simply write $A \times B$ for the type $\sum_{x:A} B$, recovering the usual simple product type.

Dependent product types - Given a type A and a family of types $B(x)$ ranging over $x : A$ we can form the dependent product type, $\prod_{x:A} B(x)$, of functions $\lambda x. b(x)$ which map terms $a : A$ to terms $b(a) : B(a)$. In the case where B does not depend on $x : A$ we write $A \rightarrow B$ for the type $\prod_{x:A} B$, recovering the usual function type.

Identity types - Given a type A and two terms $x, y : A$ we can form the type $\text{Id}_A(x, y)$ of proofs that x is equal to y . We know that everything must be equal to itself, by reflexivity, and so for any A and $x : A$ we have a proof `reflx : IdA(x, x)` which asserts that x is equal to itself. We will also write $x =_A y$, or simply $x = y$, for type $\text{Id}_A(x, y)$. When we have a proof $p : \text{Id}_A(x, y)$ we say that x and y are *propositionally* equal, as opposed to being mutually convertible via a sequence of $\beta\eta$ -reductions, in which case we say that they are *definitionally* equal and write $x \equiv y$.

Base types: $0, 1, \mathbb{B}, \mathbb{N}, \dots$ - We have the initial (or empty) type 0 which has no inhabitants and from which there always exists a unique map $0_A : 0 \rightarrow A$ for any type A . We also have the terminal (or unit) type 1 which has a single inhabitant, $*$: 1 , and for which there always exists a unique map $1_A : A \rightarrow 1$ for any type A . Finally we have common datatypes such as the type of booleans, \mathbb{B} , with terms `true` : \mathbb{B} and `false` : \mathbb{B} , and the type of natural numbers \mathbb{N} which has a terms $0 : \mathbb{N}$ and `succ`(n) : \mathbb{N} for any $n : \mathbb{N}$. These datatypes have the usual induction and computation rules.

Universes - We assume a countable hierarchy of universes $U_0 : U_1 : U_2 : \dots$ which we take to be à la Russell. That is, we identify types A with elements of the universe $A : U$. We sometimes write U to ambiguously refer to any of the U_i where the index can be inferred from the context, or where we mean to work polymorphically in universe levels.

We encode mathematical statements in the usual way, interpreting universal quantification with Π and existential quantification with Σ . So, for example, the statement “there exist infinitely many prime numbers” may be encoded as:

$$\prod_{n:\mathbb{N}} \sum_{p:\mathbb{N}} \left((n < p) \times \prod_{k:\mathbb{N}} ((1 < k) \times (k < p)) \rightarrow \neg(\text{rem}(p, k) = 0) \right)$$

Note this encoding yields a constructive notion of proof. A term of the above type (i.e. a proof of the proposition) is a function which given any $n : \mathbb{N}$ returns a pair (p, u) where $p : \mathbb{N}$ is a prime number which is greater than n (with both facts witnessed by u). This means that such a proof not only tells us of the existence of such a p , but actually computes its value.

2.1.2 Identity types and UIP

As mentioned above, for any type A and terms $x, y : A$ we can form the identity type $\text{Id}_A(x, y)$. We make use of identity proofs using the *J-eliminator*. This eliminator states that given a type $C(x, y, p)$ depending on terms $x, y : A$ and $p : x = y$ then we can inhabit $C(x, y, p)$ simply by giving a term of type $C(x, x, \text{refl}_x)$. Alternatively, in order to prove a property, C , about some y which is propositionally equal to x (as witnessed by p), then it suffices to assume that $y \equiv x$ and $p \equiv \text{refl}_x$.

A standard intuition for thinking about type theory is that types represent sets and their terms represent the elements of that set. Under this interpretation we might expect the identity type to correspond to the empty set if x and y are distinct, and the singleton set if x and y in fact represent the same element. This gives us a simple interpretation of the `refl` constructor as the single element of the singleton set. It is also easy to see how to interpret the J-eliminator since a proof $p : \text{Id}_A(x, y)$ tells us that x and y represent the same element and `reflx` and p both denote the unique element of the singleton set, therefore $C(x, y, p)$ must in fact denote the same set as $C(x, x, \text{refl}_x)$.

In fact, under this interpretation it should be the case that, since the identity type represents a set containing at most one element, then any $p, q : \text{Id}_A(x, y)$ must in fact denote the same element. So we would expect that the nested identity type, $\text{Id}_{\text{Id}_A(x, y)}(p, q)$, should always be inhabited. This principle is often referred to as *uniqueness of identity proofs* (UIP). However, Hofmann and Streicher [29, 32] showed that this principle does not follow from the usual rules of type theory by exhibiting a model in the category of groupoids (categories where every morphism is invertible). In this model: types are interpreted as groupoids, terms $x, y : A$ as elements of the groupoid $x, y \in \text{obj } A$ and the identity type $\text{Id}_A(x, y)$ as the discrete groupoid on $\text{hom}_A(x, y)$. The principle of UIP fails to hold since a groupoid can have multiple distinct morphisms between two objects.

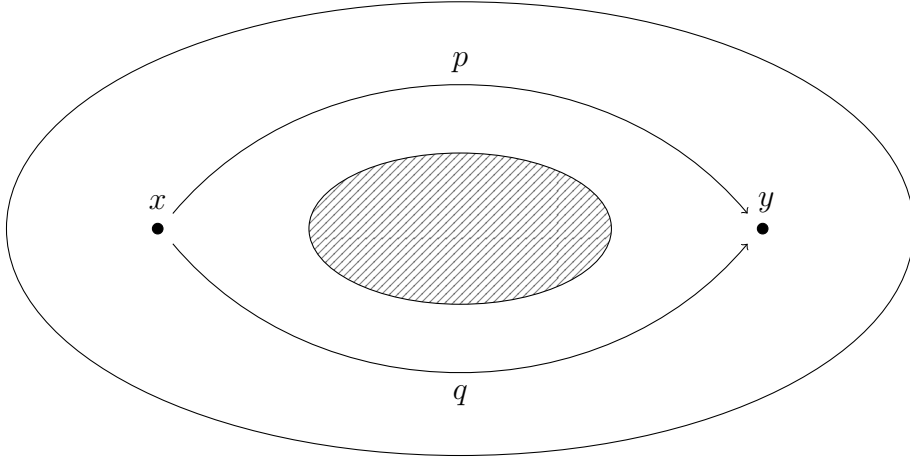


Figure 2.1: Two distinct paths between points x and y

Much of the work on UTT/HoTT is inspired by the intuition that, rather than thinking of types as sets with their terms as elements of the set, we should instead think of types as behaving like topological spaces with their terms as points in the space. We then view elements of the identity type, $p : \mathbf{Id}_A(x, y)$, as representing paths from the point x to the point y . So we think of p as behaving like a continuous function $p : \mathbb{I} \rightarrow A$, where \mathbb{I} is the unit interval $[0, 1]$, with $p(0) = x$ and $p(1) = y$. We then view elements of nested identity types, $\alpha : \mathbf{Id}_{\mathbf{Id}_A(x, y)}(p, q)$ as homotopies $\alpha : \mathbb{I} \times \mathbb{I} \rightarrow A$ such that $\alpha(0, i) = p(i)$ and $\alpha(1, i) = q(i)$ with $\alpha(i, 0) = x$ and $\alpha(i, 1) = y$ for all $i \in \mathbb{I}$.

Under this interpretation of type theory the principle of UIP no longer seems obvious. Indeed, by considering the example shown in Figure 2.1 we see why it is natural to reject this principle entirely. The two points x and y represent two elements of the type A . The two paths shown in the figure are both from x to y and so under this interpretation would correspond to elements of the identity type $p, q : \mathbf{Id}_A(x, y)$. However, because of the hole in the centre of the space, there is clearly no way to continuously deform one into the other without moving either endpoint. Therefore there can be no homotopy from one to the other, and hence, there can be no inhabitant of the nested identity type $\mathbf{Id}_{\mathbf{Id}_A(x, y)}(p, q)$.

2.1.3 Voevodsky’s univalence axiom

In the previous section we discussed the new intuition behind UTT/HoTT, and why this might lead us to question the principle of UIP. However, we did not actually extend type theory in any way so as to formalise this intuition. This is achieved via Voevodsky’s univalence axiom [62, Section 2.10], which identifies the type of equalities between types $\mathbf{Id}_U(A, B)$ (also written $A = B$) with the type of equivalences $A \simeq B$.

The univalence axiom captures the homotopical intuition that two spaces with the same homotopy group (i.e. two spaces with an equivalence between them) are connected by a path in some classifying space. That is, we view U as the classifying space for (small)

homotopy types.

From a logical perspective we can think of the univalence axiom as formalising the common mathematical practice of identifying isomorphic objects. For example, a mathematician might prove that a certain mathematical object, e.g. a group A , is isomorphic to another, B , where we already know that B possesses some property. The mathematician will then informally conclude that A therefore also possesses this property, often without checking that the property in question is preserved under isomorphism. The univalence axiom formalises this form of reasoning by essentially stating that every construction in type theory is preserved under equivalence.

The exact statement of univalence involves some subtleties, particularly regarding the definition of an equivalence between types. We now give the exact definition, starting with the definition of an equivalence, and the auxiliary notion of a contractible type.

Definition 2.1.1 (Contractibility). A type A is said to be *contractible* if the type

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

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

Definition 2.1.2 (Equivalences). An *equivalence* $A \simeq B$ is a pair (f, e) where $f : A \rightarrow B$ and e is a proof that for every $b : B$ the fibre of f at b is contractible. To be precise:

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

where

$$\mathbf{fib}_f(b) \triangleq \sum_{a:A} (f\ a = b) \quad \text{and} \quad \mathbf{isEquiv}(f) \triangleq \prod_{b:B} \mathbf{isContr}(\mathbf{fib}_f(b))$$

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

A simple example of an equivalence is the identity function $id_A : A \rightarrow A$ for any type A . To demonstrate that id_A is an equivalence we must show that $\prod_{a:A} \mathbf{isContr}(\sum_{x:A} (x = a))$, a fact which is easily shown using the J-eliminator.

Note that there are many competing notions of equivalence [62, Chapter 4]. We have selected this one because it simplifies some of the proofs in Chapter 6. It is also worth noting that the traditional definition of isomorphism is not a well-behaved notion of equivalence in UTT. Specifically, defining $\mathbf{isEquiv}(f)$ as the existence of a map $g : B \rightarrow A$ such that $\prod_{a:A} g(f\ a) = a$ and $\prod_{b:B} f(g\ b) = b$ is not well-behaved. This is because this notion, referred to as a quasi-inverse, is not a mere proposition [62, Theorem 4.1.3] and hence there might be multiple distinct proofs that f has a quasi-inverse. This has several

undesirable consequences. For example, the type of quasi-inverse equivalences is not a subtype of the type $A \rightarrow B$, and in fact the univalence axiom stated using quasi-inverse equivalences is inconsistent [62, Exercise 4.6].

Next, observe that there always exists a canonical map from the type of equalities between two types to the type of equivalences between them.

Definition 2.1.3 (idtoeqv). For all i , and types $A, B : U_i$, there is a canonical map $\text{idtoeqv} : (A = B) \rightarrow (A \simeq B)$ which is defined by path induction¹ on the proof $A = B$:

$$\text{idtoeqv}(\text{refl}_A) \triangleq \text{id}_A$$

where $\text{id}_A : A \simeq A$ is the identity map regarded as an equivalence.

Finally, we formalise the univalence axiom as the statement that this map is itself an equivalence.

Definition 2.1.4 (Voevodsky’s univalence axiom). The univalence axiom for a universe U_i asserts that for all $A, B : U_i$ the map $\text{idtoeqv} : (A = B) \rightarrow (A \simeq B)$ is an equivalence.

Note that the univalence axiom is a statement about a particular universe U_i . In general, when working in UTT, we assume that all universes are univalent unless explicitly stated otherwise. This assumption is what separates univalent/homotopy type theory from vanilla Martin-Löf type theory.

2.1.4 Function extensionality

We now recall the principle of function extensionality in type theory:

Definition 2.1.5 (Function extensionality). The principle of function extensionality states that two functions $f, g : \prod_{x:A} B(x)$ are equal whenever they are pointwise equal. That is, we have a map:

$$f \approx g \rightarrow f = g \quad \text{where} \quad f \approx g \triangleq \prod_{x:A} f(x) = g(x)$$

We say that we “assume function extensionality” when we are assuming the existence of a term:

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

for all universe levels i, j .

¹This is simply the J-eliminator described in Section 2.1.2. We use this name in reference to the intuition of equality proofs as paths.

Note that function extensionality is not derivable from the rules of standard Martin-Löf type theory. However, it is derivable from the univalence axiom [62, Section 4.9].

2.1.5 An alternative formulation of univalence

While Definition 2.1.4 is very concise it is sometimes easier to work with a slightly different, but equivalent, statement of the univalence axiom which we present here. In light of the following definition we will sometimes refer to the univalence axiom as the *proper* univalence axiom.

Definition 2.1.6 (Coerce). For all i , and types $A, B : U_i$, we can define a map `coerce` : $(A = B) \rightarrow A \rightarrow B$ either by path induction, or as:

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

where `fst` is the first projection.

Definition 2.1.7 (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:

$$\mathbf{UA}_i \triangleq \prod_{A, B : U_i} A \simeq B \rightarrow A = B$$

When using a term $ua : \mathbf{UA}_i$ 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 $\mathbf{UA}\beta_i(ua)$, where:

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

Next, we give a result which is well-known in the UTT/HoTT community. This result decomposes the proper univalence axiom into the naive version and a computation rule. Note that this result requires the principle of function extensionality to hold. First we give a lemma which generalises the core construction of this result.

Lemma 2.1.8. *Given $X : U_i$, $Y : X \rightarrow X \rightarrow U_j$ and a map $f : \prod_{x, x' : X} x = x' \rightarrow Y(x, x')$ then $f x x'$ is an equivalence for all $x, x' : X$ iff there exists a map*

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

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 [62, Section 4.3]. To do this we must exhibit both a right and left inverse.

For the left inverse we take $g'(y) \triangleq g(y) \cdot g(f(\mathbf{refl}))^{-1}$. 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 $p \equiv \mathbf{refl}$ and therefore we are required to show $g'(f(\mathbf{refl})) = \mathbf{refl}$. However, since $g'(f(\mathbf{refl})) \equiv g(f(\mathbf{refl})) \cdot g(f(\mathbf{refl}))^{-1}$ this goal simplifies to $g(f(\mathbf{refl})) \cdot g(f(\mathbf{refl}))^{-1} = \mathbf{refl}$ 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 $e : \mathbf{isEquiv}(f)$ and $y : Y(x, x')$ we have $\mathbf{fst}(e(y)) : \sum_{p:x=x'} f(p) = y$. 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. \square

Theorem 2.1.9. *Assuming function extensionality, naive univalence, along with a computation rule, is logically equivalent to the proper univalence axiom. That is, there are terms*

$$ua : \mathbf{UA}_i, \quad ua\beta : \mathbf{UA}\beta_i(ua)$$

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

Proof. By $ua\beta$ we know that $\mathbf{fst}(\mathbf{idtoeqv}(ua(f, e))) = f$ for all $(f, e) : A \simeq B$. Now, since $\mathbf{isEquiv}(f)$ is a mere proposition for each f , we can deduce that $\mathbf{idtoeqv}(ua(f, e)) = (f, e)$ by [62, Lemma 3.5.1]. Therefore we simply take $X \equiv U_i$, $Y(A, B) \equiv A \simeq B$, $f \equiv \mathbf{idtoeqv}$ and $g \equiv ua$ in Lemma 2.1.8 to deduce the desired result. \square

2.1.6 The failure of UIP

We previously discussed how the new intuition behind UTT/HoTT led us to reject the principle of UIP. We then formalised this intuition by way of Voevodsky's univalence axiom. We now briefly observe that univalence is sufficient to render UIP inconsistent.

Consider the type of equivalences on the booleans, $\mathbb{B} \simeq \mathbb{B}$. This type certainly contains at least two distinct elements: the identity $id_{\mathbb{B}} : \mathbb{B} \simeq \mathbb{B}$ and the negation map $neg : \mathbb{B} \simeq \mathbb{B}$ (leaving the proof terms witnessing the equivalences implicit). By univalence we know that the type of equalities $\mathbb{B} =_U \mathbb{B}$ is equivalent to the type $\mathbb{B} \simeq \mathbb{B}$, therefore we deduce the existence of two distinct proofs $\mathbb{B} =_U \mathbb{B}$, but this is clearly incompatible with UIP.

2.2 Cubical type theory

This section introduces the notion of cubical type theory, which is type theory extended with an *interval object* and certain *composition* operations. As with “type theory” the phrase “cubical type theory” can refer to any of several related systems, all with slightly different notions of interval and composition. In this thesis we will take the system given by Cohen et al. [18] as the canonical example of a cubical type theory. This section provides an overview of the work presented in [18].

The motivation for studying cubical type theory is to find a system in which the univalence axiom can be given a constructive interpretation. This means that proofs involving univalence should always compute in a suitable sense. For example, we would expect that any closed term $M : \mathbb{N}$ should reduce to a numeral, that is, something of the form $\text{succ}^n 0$ for some (external) natural number n . This property is often called canonicity, and was proved for cubical type theory by Huber [33].

2.2.1 The formal interval \mathbb{I}

Cubical type theory introduces the notion of an interval object, \mathbb{I} , which represents the unit interval $[0, 1]$. The terms of \mathbb{I} are elements of the free De Morgan algebra [9] on a countable set of variables (written i, j, k, \dots). That is, they are the terms built from the following grammar:

$$r, s ::= 0 \mid 1 \mid i \mid 1 - r \mid r \wedge s \mid r \vee s$$

where i is a variable, quotiented by the equations making $r \wedge s$ and $r \vee s$ into the meet and join with 0 and 1 as the least and greatest elements respectively, and making $1 - r$ into an involution satisfying:

$$1 - 0 = 1 \quad 1 - 1 = 0$$

$$1 - (r \vee s) = (1 - r) \wedge (1 - s) \quad 1 - (r \wedge s) = (1 - r) \vee (1 - s)$$

Following the intuition that \mathbb{I} represents the unit interval, we think of $r \wedge s$ and $r \vee s$ as representing the operations *min* and *max* respectively.

We can extend contexts with variables ranging over \mathbb{I} , so if Γ is a valid context then so is $\Gamma, i : \mathbb{I}$. We say that r is well-formed in a context Γ , written $\Gamma \vdash r : \mathbb{I}$, iff r only refers to variables declared in Γ .

One thing to note is that, while the interval behaves much like a type in many respects, it is a distinct sort of object and cannot be used as if it were a type, e.g. we cannot form types such as $\mathbb{N} \rightarrow \mathbb{I}$, and in particular \mathbb{I} will not have a *composition operation* which we

introduce in Section 2.2.4.

2.2.2 Path types

Following the intuition that equalities should be interpreted as paths, cubical type theory uses the interval to define the notion of the path type, $\mathbf{Path}_A x y$, which intuitively consists of functions $p : \mathbb{I} \rightarrow A$ such that $p0 \equiv x$ and $p1 \equiv y$. Since the interval is not a type then we cannot directly define such functions and so the syntax is extended with separate abstraction and application rules for paths. Path abstraction is written as $\langle i \rangle t$ and application is written as $t r$ where r is built from the grammar given in the previous section. We have the expected computation rule $(\langle i \rangle t) r \equiv t[r/i]$.

A nice feature of these path types, and a good example of using the syntax described above, is that one can derive the principle of function extensionality for paths. That is, given a proof $p : \prod_{x:A} \mathbf{Path}_B (f x) (g x)$ then we can define:

$$\langle i \rangle \lambda x. p x i : \mathbf{Path}_{A \rightarrow B} f g$$

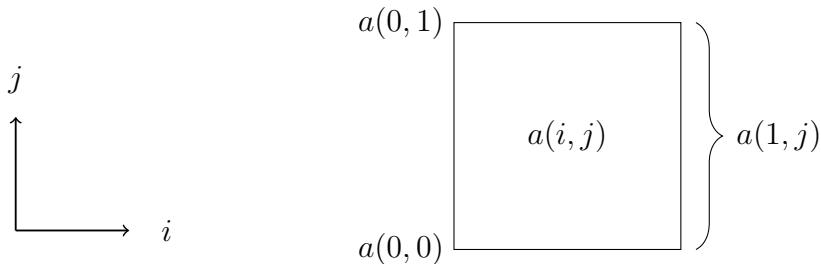
This is well-typed because if we substitute 0 for i in the path abstraction then we have:

$$\begin{aligned} \lambda x. p x 0 &\equiv \lambda x. f x && \text{since } p x : \mathbf{Path}_B (f x) (g x) \text{ and hence } p x 0 \equiv f x \\ &\equiv f && \text{by } \eta\text{-conversion} \end{aligned}$$

and similarly when substituting 1 for i we get $\lambda x. p x 1 \equiv g$. Therefore $\langle i \rangle \lambda x. p x i$ has type $\mathbf{Path}_{A \rightarrow B} f g$.

2.2.3 The face lattice \mathbb{F}

Recall the intuition behind univalent type theory, where we think of the terms of a type as corresponding to points in some topological space. Now that we have extended type theory with an interval object we can begin to talk, not only about points, but also about lines, squares, cubes, and so on. For example, consider a term $i, j : \mathbb{I} \vdash a(i, j) : A$ in a context with two interval variables, i and j . We can think of a as defining a square, parameterised by the dimensions i and j , in the space corresponding to A , like so:



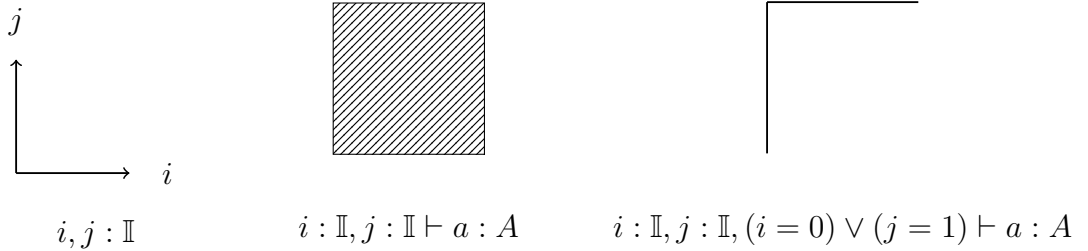
By substituting 0 or 1 for the variables i and j we can access the corners and sides of the square as shown in the diagram above.

Cubical type theory introduces the notion of the face lattice, \mathbb{F} , which extends our ability to talk about points, lines, squares, etc, to allows us to talk about certain shapes built up as the union of faces, edges and corners of (n -dimensional) cubes, known as “sub-polyhedra” of a cube. The terms of the face lattice are the elements of the free distributive lattice on symbols ($i = 0$) and ($i = 1$) for every variable i , quotiented by the relation $(i = 0) \wedge (i = 1) = 0_{\mathbb{F}}$. These elements can be described by the following grammar:

$$\varphi, \psi ::= 0_{\mathbb{F}} \mid 1_{\mathbb{F}} \mid i = 0 \mid i = 1 \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

where i is any variable. We say that a formula φ is well-formed in a context Γ , written $\Gamma \vdash \varphi : \mathbb{F}$, iff φ only refers to variables declared in Γ .

Whenever we have a well-formed formula $\Gamma \vdash \varphi : \mathbb{F}$ then we can form the restricted context $\Gamma, \varphi \vdash$ which allows us to talk about sub-polyhedra of a type. For example, we can think of a term $i : \mathbb{I}, j : \mathbb{I} \vdash a : A$ as a square indexed by the dimensions i and j . If we wish to restrict our attention to just two sides of this square then we can consider a in the restricted context $i : \mathbb{I}, j : \mathbb{I}, (i = 0) \vee (j = 1) \vdash a : A$.



Importantly this context restriction defines a congruence on terms. So we may have two terms $\Gamma \vdash a : A$ and $\Gamma \vdash a' : A$ which are not judgementally equal, in that $\Gamma \vdash a \not\equiv a' : A$, but which become equal in the restricted context, i.e. $\Gamma, \varphi \vdash a \equiv a' : A$. An example would be the terms $\Gamma \vdash i \vee j : \mathbb{I}$ and $\Gamma \vdash i \wedge (1 - j) : \mathbb{I}$ which are not equal according to the equations of the free De Morgan algebra. However, if we restrict the context with the formula $j = 0$ then we have

$$\Gamma, (j = 0) \vdash i \vee j \equiv i \vee 0 \equiv i \equiv i \wedge 1 \equiv i \wedge (1 - j) \quad : \quad \mathbb{I}$$

and hence the two terms become equal under the restriction $j = 0$. In this case we say that they *agree* on the formula $j = 0$.

We can also use context restriction to talk about partial elements of a type. Given a type $\Gamma \vdash A$ and a term $\Gamma, \varphi \vdash u : A$ we say that u is a partial element of A whose extent of definition (or simply, “extent”) is φ . We also say that a term $\Gamma \vdash a : A$ (which we sometimes refer to as a total element of A) extends u iff $\Gamma, \varphi \vdash u \equiv a : A$. Finally, we

introduce a new judgement $\Gamma \vdash a : A[\varphi \mapsto u]$ to mean that a is a total element which extends u , or explicitly: $\Gamma \vdash a : A$ and $\Gamma, \varphi \vdash u \equiv a : A$.

2.2.4 Kan composition

In homotopy theory, topological spaces are often represented as certain presheaves, such as simplicial or cubical sets. In particular attention is often restricted to those presheaves satisfying some extra condition, sometimes referred to as the Kan filling condition, which ensures that the presheaf represents some “well-behaved” topological space. Following the intuition that types in UTT/HoTT should represent well-behaved topological spaces, cubical type theory also introduces such a Kan-like condition for types. In fact this condition is a generalisation of the one given for cubical sets by Kan himself [38].

The condition states that any partial line in a type which is extensible at one end is also extensible at the other. What this means is perhaps made clearer by the exact definition of the *composition operation*. A composition is well formed according to the following judgement:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \quad \Gamma, i : \mathbb{I} \vdash A \quad \Gamma, \varphi, i : \mathbb{I} \vdash u : A \quad \Gamma \vdash a_0 : A(i0)[\varphi \mapsto u(i0)]}{\Gamma \vdash \text{comp}^i A [\varphi \mapsto u] a_0 : A(i1)[\varphi \mapsto u(i1)]}$$

What this says is that for any type $\Gamma, i : \mathbb{I} \vdash A$ which may depend on the interval \mathbb{I} , and face formula $\varphi : \mathbb{F}$ then given a partial line in A , i.e. a term $\Gamma, \varphi, i : \mathbb{I} \vdash u : A$, and a total element which extends u at 0, i.e. $\Gamma \vdash a_0 : A(i0)[\varphi \mapsto u(i0)]$, then we get a total element which extends u at 1, that is a term of type $A(i1)[\varphi \mapsto u(i1)]$. Following Cohen et al [18], we use the notation $(i0)$ for the substitution $[0/i]$.

Using this composition operation, along with the structure of the interval, it is possible to interpret the J-eliminator for path types², meaning that the path type does indeed model all the rules that we expect for an equality type.

2.2.5 Glueing

The final ingredient of cubical type theory is the glueing construction. This is similar to the composition operation but instead of composing terms along (partial) lines it involves composing types along (partial) equivalences. The glueing construction is then used to define two things. Firstly, it is used to define a composition operation for the universe, and secondly, it is used to interpret the univalence axiom in cubical type theory.

Here we simply present an overview of the Glueing construction. For further details see either the original paper [18] or Section 5.4 of this thesis. Firstly, the formation rule

²With a propositional computation rule. See Definition 3.1.3 for further details on what this means.

for the **Glue** type is as follows:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \quad \Gamma \vdash B \quad \Gamma, \varphi \vdash A \quad \Gamma, \varphi \vdash f : \mathbf{Equiv} \ A \ B}{\Gamma \vdash \mathbf{Glue} \ [\varphi \mapsto (A, f)] \ B}$$

This rule states that given any face formula $\Gamma \vdash \varphi : \mathbb{F}$, any total type $\Gamma \vdash B$ and any partial type $\Gamma, \varphi \vdash A$ which is equivalent to B everywhere that it is defined, as witnessed by $\Gamma, \varphi \vdash f : \mathbf{Equiv} \ A \ B$, then we get a new type $\Gamma \vdash \mathbf{Glue} \ [\varphi \mapsto (A, f)] \ B$. We should think of $\mathbf{Glue} \ [\varphi \mapsto (A, f)] \ B$ as an extension of the partial type A . In particular we have that this type is equal to A whenever it is defined, that is: $\mathbf{Glue} \ [1_{\mathbb{F}} \mapsto (A, f)] \ B \equiv A$. Here we use the notation $\mathbf{Equiv} \ A \ B$ from [18], but this is exactly the type $A \simeq B$ defined in Section 2.1.3.

2.2.6 Univalence

Using the features described in the previous sections it is possible to prove the univalence axiom inside of cubical type theory. This means that the axiom now reduces to more primitive concepts in the type theory which provide much better computational behaviour – the intended aim of cubical type theory. There are many different ways to approach the proof of univalence, three different approaches are given in the original paper [18] (including the appendix) and two are given in this thesis: one using glueing (Section 5.5) and one which avoids the glueing construction (Chapter 6).

Here we just give a flavour of how glueing can be used to prove the univalence axiom. A necessary step in establishing the univalence axiom is to define, either explicitly or implicitly, an inverse to the map $\mathbf{pathToEquiv} : \mathbf{Path}_U \ A \ B \rightarrow \mathbf{Equiv} \ A \ B$ which is defined exactly as in Definition 2.1.3 but using \mathbf{Path} -types in place of \mathbf{Id} -types. Using glueing, we can define such a map, $\mathbf{equivToPath} : \mathbf{Equiv} \ A \ B \rightarrow \mathbf{Path}_U \ A \ B$, like so:

$$\mathbf{equivToPath} \ f \triangleq \langle i \rangle \mathbf{Glue} \ [(i = 0) \mapsto (A, f), (i = 1) \mapsto (B, id_B)] \ B$$

where $id_B : \mathbf{Equiv} \ B \ B$ is the identity function regarded as an equivalence. Note that when $i = 0$ we have

$$\mathbf{Glue} \ [(0 = 0) \mapsto (A, f), (0 = 1) \mapsto (B, id_B)] \ B \equiv \mathbf{Glue} \ [1_{\mathbb{F}} \mapsto (A, f)] \ B \equiv A$$

and when $i = 1$ we have

$$\mathbf{Glue} \ [(1 = 0) \mapsto (A, f), (1 = 1) \mapsto (B, id_B)] \ B \equiv \mathbf{Glue} \ [1_{\mathbb{F}} \mapsto (B, id_B)] \ B \equiv B$$

and therefore $\mathbf{equivToPath} \ f$ does indeed define a path from A to B . This establishes the first part of the naive univalence axiom (Definition 2.1.7). Next, we would need to

show the computation rule: that coercing along `equivToPath` f is `Path`-equal to applying the function f . Then, using Theorem 2.1.9, we would have a proof of the full univalence axiom. Again, we will revisit this proof in more detail in the main body of the thesis.

Chapter 3

Models of Martin-Löf type theory

In the previous chapter we discussed Martin-Löf type theory and various extensions to it such as the univalence axiom, and cubical features such as an interval and composition operations. In this chapter we describe how to present categorical models of Martin-Löf type theory and discuss a particular model in the category of cubical sets which will model all of the extensions described in Section 2.2, and hence provides a model of the univalence axiom.

3.1 Categories with Families (CwFs)

There are many competing notions for how to present a model of dependent type theory. In this thesis we will use Dybjer's notion of a Category with Families (CwF) [23]. We use slightly different notation from Dybjer and we separate his functor into the category of families into two different functors, but otherwise the definition is unchanged. We first recall the definition of the category of elements of a presheaf $P : \mathbf{C}^{op} \rightarrow \mathbf{Set}$.

Definition 3.1.1 (The category of elements). Given any presheaf $P : \mathbf{C}^{op} \rightarrow \mathbf{Set}$, the category of elements of P , written $\int P$, has as objects pairs (X, p) where $X \in \mathbf{C}$ and $p \in P(X)$. A morphism $f : (X, p) \rightarrow (X', p')$ is simply a morphism $X \rightarrow X'$ in \mathbf{C} such that $P(f)(p') = p$.

Definition 3.1.2 (Category with Families). A *category with families* consists of the following data:

- A category of contexts \mathbf{C} , whose objects represents the contexts of type theory and whose morphisms represent substitutions between contexts. This category is required to have a terminal object, 1 , which represents the empty context.
- A functor $\mathbf{Ty} : \mathbf{C}^{op} \rightarrow \mathbf{Set}$ which, to every context Γ , assigns a set $\mathbf{Ty}(\Gamma)$ representing the types in context Γ . Given any morphism $\gamma : \Delta \rightarrow \Gamma$ and type $A \in \mathbf{Ty}(\Gamma)$,

the action of $\mathbf{T}\mathbf{y}$ on morphisms gives us the interpretation of substitution and we suggestively write $A[\gamma]$ for $\mathbf{T}\mathbf{y}(\gamma)(A) \in \mathbf{T}\mathbf{y}(\Delta)$. The functorial nature of $\mathbf{T}\mathbf{y}$ ensures that substitution is strictly associative.

- A functor $\mathbf{Ter} : (\int \mathbf{T}\mathbf{y})^{op} \rightarrow \mathbf{Set}$ which to every context Γ and type $A \in \mathbf{T}\mathbf{y}(\Gamma)$ assigns a set $\mathbf{Ter}(\Gamma, A)$ representing the terms of type A in context Γ . We will write $\mathbf{Ter}(\Gamma \vdash A)$ for the set $\mathbf{Ter}(\Gamma, A)$. Once again, the action of \mathbf{Ter} on morphisms models substitution in a strictly associative way, and given $\gamma : \Delta \rightarrow \Gamma$, $A \in \mathbf{T}\mathbf{y}(\Gamma)$ and $a \in \mathbf{Ter}(\Gamma \vdash A)$ we write $a[\gamma]$ for $\mathbf{Ter}(\gamma)(a) \in \mathbf{Ter}(\Delta \vdash A[\gamma])$.
- A context extension operation \dots which takes a context $\Gamma \in \mathbf{C}$ and a type $A \in \mathbf{T}\mathbf{y}(\Gamma)$ and returns an object $\Gamma.A \in \mathbf{C}$ with a morphism $\mathfrak{p} : \Gamma.A \rightarrow \Gamma$ representing context weakening, and a term $\mathfrak{q} \in \mathbf{Ter}(\Gamma.A \vdash A[\mathfrak{p}])$ representing the newly added variable of type A . This operation must satisfy the following universal property: for any substitution $\gamma : \Delta \rightarrow \Gamma$ and term $a \in \mathbf{Ter}(\Delta \vdash A[\gamma])$ there exists a unique morphism $\langle \gamma, a \rangle : \Delta \rightarrow \Gamma.A$ such that $\mathfrak{p} \circ \langle \gamma, a \rangle = \gamma$ and $\mathfrak{q}[\langle \gamma, a \rangle] = a$.

For notational convenience we will sometimes express properties of a CwF in the form of rules where we will write the more familiar $\Gamma \vdash A$ for $A \in \mathbf{T}\mathbf{y}(\Gamma)$ and $\Gamma \vdash a : A$ for $a \in \mathbf{Ter}(\Gamma \vdash A)$. For example, we could have stated the existence of the morphism $\langle \gamma, a \rangle : \Delta \rightarrow \Gamma.A$ like so:

$$\frac{\gamma : \Delta \rightarrow \Gamma \quad \Gamma \vdash A \quad \Delta \vdash a : A[\gamma]}{\langle \gamma, a \rangle : \Delta \rightarrow \Gamma.A}$$

We will also write $[a_1, a_2, \dots, a_n] : \Gamma \rightarrow \Gamma.A_1.A_2 \dots A_n$ for the substitution inductively defined as:

$$\begin{aligned} [] &\triangleq id \\ [a_1, \dots, a_n, a_{n+1}] &\triangleq \langle [a_1, \dots, a_n], a_{n+1} \rangle \end{aligned}$$

where $\Gamma.A_1.A_2 \dots A_n \vdash A_{n+1}$ and $\Gamma \vdash a_{n+1} : A_{n+1}[a_1, \dots, a_n]$.

3.1.1 Additional structure on a CwF

A CwF provides a basic framework for interpreting the types and terms of dependent type theory. However, by default a CwF will not support the interpretation of type formers such as dependent pairs, dependent functions, or intensional identity types. Such type formers are specified as additional structure on the CwF.

As an example we give the extra structure required for a CwF to support intensional identity types. In doing so we will draw attention to a subtlety in the definition of identity types which will be important later in this thesis, specifically the difference between

identity types with a *definitional*, or *judgemental*, computational rule vs. those with a *propositional* computation rule. We will refer to these as strict and weak intensional identity types respectively. For the definition of other extra structure on a CwF, such as dependent pairs, dependent functions and type-theoretic universe, we refer the reader to an article by Hofmann [30].

Definition 3.1.3 (Intensional identity types on a CwF). We say that a CwF supports *intensional identity types* iff, for every type $\Gamma \vdash A$ and term $\Gamma \vdash a : A$, the CwF is closed under the following structure:

- A type former for the identity type:

$$\frac{\Gamma \vdash b : A}{\Gamma \vdash \text{Id}_A(a, b)}$$

- A term witnessing reflexivity:

$$\overline{\Gamma \vdash \text{refl}_A(a) : \text{Id}_A(a, a)}$$

- A term witnessing the induction principle:

$$\frac{\Gamma.A. \text{Id}_{A[p]}(a[p], q) \vdash C \quad \Gamma \vdash c_0 : C[a, \text{refl}_A(a)] \quad \Gamma \vdash b : A \quad \Gamma \vdash u : \text{Id}_A(a, b)}{\Gamma \vdash J_A(a, C, c_0, b, u) : C[b, u]}$$

such that all of these constructions commute with substitution:

$$\begin{aligned} \text{Id}_A(a, b)[\gamma] &= \text{Id}_{A[\gamma]}(a[\gamma], b[\gamma]) \\ \text{refl}_A(a)[\gamma] &= \text{refl}_{A[\gamma]}(a[\gamma]) \\ J_A(a, C, c_0, b, u)[\gamma] &= J_{A[\gamma]}(a[\gamma], C[\gamma], c_0[\gamma], b[\gamma], u[\gamma]) \end{aligned}$$

for all $\gamma : \Delta \rightarrow \Gamma$, where $\gamma' : \Delta.A[\gamma]. \text{Id}_{A[\gamma \circ p]}(a[\gamma \circ p], q) \rightarrow \Gamma.A. \text{Id}_{A[p]}(a[p], q)$ is given by $\gamma' \triangleq \langle \langle \gamma \circ p, q \rangle \circ p, q \rangle$. Identity types must then satisfy one of the following conditions:

- We say that identity types are *strict* if they satisfy:

$$J_A(a, C, c_0, a, \text{refl}_A(a)) = c_0$$

- We say that identity types are *weak* if they are closed under the following rule:

$$\frac{\Gamma.A. \text{Id}_{A[p]}(a[p], q) \vdash C \quad \Gamma \vdash c_0 : C[a, \text{refl}_A(a)]}{\Gamma \vdash \text{JEq}_A(a, C, c_0) : \text{Id}_{C[a, \text{refl}_A(a)]}(J_A(a, C, c_0, a, \text{refl}_A(a)), c_0)}$$

satisfying the obvious stability condition:

$$\mathbf{JEq}_A(a, C, c_0)[\gamma] = \mathbf{JEq}_{A[\gamma]}(a[\gamma], C[\gamma], c_0[\gamma])$$

for all $\gamma : \Delta \rightarrow \Gamma$, where γ' is as above.

Note that strong identity types always satisfy the condition for weak identity types since if $J_A(a, C, c_0, a, \mathbf{refl}_A(a)) = c_0$ then we can simply define $\mathbf{JEq}_A(a, C, c_0) \triangleq \mathbf{refl}_{C[a, \mathbf{refl}_A(a)]}(c_0)$.

3.2 Presheaf models of type theory

In this section we show how any presheaf category, $\widehat{\mathbf{C}} = \mathbf{Set}^{\mathbf{C}^{op}}$, admits the structure of a category with families. Following the notation used in [18], we write I, J, K and f, g, h for the objects and morphisms respectively of the category \mathbf{C} .

Definition 3.2.1 (The CwF associated with a presheaf category $\widehat{\mathbf{C}}$). Given any small category \mathbf{C} we define a category with families as follows:

- The category of contexts is given by the functor category $\widehat{\mathbf{C}} = \mathbf{Set}^{\mathbf{C}^{op}}$. Every such presheaf category has a terminal object given by the presheaf which is constantly a one element set.
- The functor $\mathbf{Ty} : \widehat{\mathbf{C}}^{op} \rightarrow \mathbf{Set}$ is given by presheaves on the category of elements of Γ , that is:

$$\mathbf{Ty}(\Gamma) \triangleq \mathbf{obj}(\mathbf{Set}^{(\int \Gamma)^{op}})$$

Explicitly, this means that a type $A \in \mathbf{Ty}(\Gamma)$ is given by a family of sets $A(I, \rho)$ for every $I \in \mathbf{C}, \rho \in \Gamma(I)$ such that for every $f : J \rightarrow I$ we have,

$$A(f) : A(I, \rho) \rightarrow A(J, \Gamma(f)(\rho))$$

such that $A(id) = id$ and $A(g \circ f) = A(f) \circ A(g)$.

Note that a natural transformation (substitution) between two presheaves (contexts), $\gamma : \Delta \rightarrow \Gamma$, yields a functor between the categories of elements, $\int \gamma : \int \Delta \rightarrow \int \Gamma$. Therefore the action on morphisms, $\mathbf{Ty}(\gamma) : \mathbf{Ty}(\Gamma) \rightarrow \mathbf{Ty}(\Delta)$, is simply given by precomposition with $(\int \gamma)^{op} : (\int \Delta)^{op} \rightarrow (\int \Gamma)^{op}$. That is, given $\delta \in \Delta(I)$, define:

$$A[\gamma](I, \delta) \triangleq A(I, \gamma_I(\delta))$$

This gives a functor $\mathbf{Ty} : \widehat{\mathbf{C}}^{op} \rightarrow \mathbf{Set}$ as required.

- The functor $\mathbf{Ter} : (\int \mathbf{Ty})^{op} \rightarrow \mathbf{Set}$ is given by global sections of A in the presheaf category $\mathbf{Set}^{(\int \Gamma)^{op}}$, that is:

$$\mathbf{Ter}(\Gamma \vdash A) \triangleq \mathbf{hom}(1, A)$$

Explicitly, writing $a(I, \rho)$ for $a_{(I, \rho)}(*)$, this means that a term $a \in \mathbf{Ter}(\Gamma \vdash A)$ is given by a family,

$$a(I, \rho) \in A(I, \rho)$$

for every $I \in \mathbf{C}$ and $\rho \in \Gamma(I)$, such that for all $f : J \rightarrow I$ we have:

$$A(f)(a(I, \rho)) = a(J, \Gamma(f)(\rho))$$

Here and elsewhere we omit the application of $* \in 1(I)$ in uses of a .

As before, the action of \mathbf{Ter} on a morphism $\gamma : (\Delta, A') \rightarrow (\Gamma, A)$ is given by a form of precomposition. That is, given $\delta \in \Delta(I)$, define:

$$a[\gamma](I, \delta) \triangleq a(I, \gamma_I(\delta)) \in A(I, \gamma_I(\delta)) = A[\gamma](I, \delta) = A'(I, \delta)$$

This gives a functor $\mathbf{Ter} : (\int \mathbf{Ty})^{op} \rightarrow \mathbf{Set}$ as required.

- Given $\Gamma \in \widehat{\mathbf{C}}$ and $A \in \mathbf{Ty}(\Gamma)$, the extended context $\Gamma.A \in \widehat{\mathbf{C}}$ is given by,

$$(\Gamma.A)(I) \triangleq \{(\rho, a) \mid \rho \in \Gamma(I), a \in A(I, \rho)\}$$

with the action on morphisms $f : J \rightarrow I$ given by:

$$(\Gamma.A)(f)(\rho, a) \triangleq (\Gamma(f)(\rho), A(f)(a))$$

The map $\mathbf{p} : \Gamma.A \rightarrow \Gamma$ and the term $\mathbf{q} \in \mathbf{Ter}(\Gamma.A \vdash A[\mathbf{p}])$ are simply given by the (pointwise) first and second projections:

$$\mathbf{p}_I(\rho, a) \triangleq \rho \qquad \mathbf{q}(I, (\rho, a)) \triangleq a$$

These definitions satisfy the required universal property, with the unique morphism $\langle \gamma, a \rangle : \Delta \rightarrow \Gamma.A$ given by,

$$\langle \gamma, a \rangle_I(\delta) \triangleq (\gamma_I(\delta), a(I, \delta))$$

for $I \in \mathbf{C}$ and $\delta \in \Delta(I)$. This is well-defined since if $a \in \mathbf{Ter}(\Delta \vdash A[\gamma])$ then $a(I, \delta) \in A[\gamma](I, \delta) = A(I, \gamma_I(\delta))$ and hence $(\gamma_I(\delta), a(I, \delta)) \in (\Gamma.A)(I)$.

Following the notation used in [18] we often omit the first argument to types and terms, e.g. given $A \in \mathbf{Ty}(\Gamma)$ and $\rho \in \Gamma(I)$ we write $A(\rho)$ for $A(I, \rho)$. We will also write the action of functors on morphisms $\Gamma(f)(\rho)$ simply as ρf .

It is easy to show that any CwF of this form admits dependent sums, dependent products, strict intensional identity types and universes. We omit the details of these constructions but refer the reader to Huber’s thesis for further detail [34, Section 1.2]. In the following section we explain how the cubical sets model of type theory [18] is constructed. In doing so we will make use of these of these dependent sums and products. However, the standard interpretation of identity types will validate UIP, and hence will not validate the univalence axiom. Therefore we will use an alternative interpretation for the identity types, which we will call *path types*. We will also modify the standard universe construction in order to construct a univalent universe.

3.3 A model in cubical sets

In this section we outline the presheaf model of type theory given by Cohen et al. [18] which models all of the features of cubical type theory, as discussed in Section 2.2.

Recall that a *De Morgan algebra* is a distributive lattice equipped with a function $d \mapsto 1 - d$ which is involutive $1 - (1 - d) = d$ and satisfies De Morgan’s Law $1 - (d_1 \vee d_2) = (1 - d_1) \wedge (1 - d_2)$; see [9, Chapter XI]. A homomorphism of De Morgan algebras is a function preserving finite meets and joins and the involution function. Let \mathbf{DM} denote the category of De Morgan algebras and homomorphisms and $\mathbf{dM}(I)$ the free De Morgan algebra on a set I .

Definition 3.3.1 (The category of cubes, \square). Fix a countably infinite set \mathbb{D} whose elements we call *names* and write as i, j, k, \dots . The objects of \square are the finite subsets of \mathbb{D} , which we write as I, J, K, \dots . The morphisms $\square(I, J)$ are all functions $J \rightarrow \mathbf{dM}(I)$. Such functions are in bijection with the De Morgan algebra homomorphisms $\mathbf{dM}(J) \rightarrow \mathbf{dM}(I)$ and the composition in \square of $f \in \square(I, J)$ with $g \in \square(J, K)$ is the composition in \mathbf{Set} of $g : K \rightarrow \mathbf{dM}(J)$ with the homomorphism $\mathbf{dM}(J) \rightarrow \mathbf{dM}(I)$ corresponding to f .

Thus \square is equivalent to the opposite of the full subcategory of \mathbf{DM} whose objects are the free, finitely generated De Morgan algebras. Hence it is the algebraic theory of De Morgan algebra as a Lawvere theory [41, 2]: it is a category with finite products equipped with an internal De Morgan algebra (whose underlying object is $\{i\}$ for some chosen $i \in \mathbb{D}$) and is universal among such categories.

Definition 3.3.2. A *cubical set* is a presheaf on the category of cubes.

The category of cubical sets, $\mathbf{Set}^{\square^{op}}$, forms the category of contexts of a CwF which will allow us to interpret all of the features of cubical type theory. In subsequent chapters

we will axiomatise certain properties of this category and prove the previous statement in an axiomatic way. In the rest of this section we provide a very brief outline of the original approach taken by Cohen et al. [18].

We first introduce some notation that we will use in this section. For any set of symbols $I \in \square$ and symbol $i \notin I$ we write I, i as an abbreviation for $I \cup \{i\}$. We also write $(i0)$ for the map $I \rightarrow I, i$ which sends i to 0 and is the identity everywhere else. Similarly we write $(i1) : I \rightarrow I, i$ for the map which sends i to 1. Here 0 and 1 are the least and greatest elements of the free De Morgan algebra $\mathbf{dM}(I)$. Finally, we write $s_i : I, i \rightarrow I$ for the inclusion $I \subseteq \mathbf{dM}(I, i)$.

In the following sections we write \mathbf{Ty} , \mathbf{Ter} and \dots to refer to the standard CwF on the category of cubical sets, as described in Definition 3.2.1.

3.3.1 The interval and face lattice

In order to model the additional features of cubical type theory we need to describe how to interpret the interval, \mathbb{I} , and the face lattice, \mathbb{F} , in the CwF associated to the category of cubical sets. We will define both as cubical sets and then observe that any cubical set X can be viewed as a type in any context Γ by taking $X' \in \mathbf{Ty}(\Gamma)$ to be $X'(I, \rho) \triangleq X(I)$. We then interpret terms $\Gamma \vdash i : \mathbb{I}$ and $\Gamma \vdash \varphi : \mathbb{F}$ as elements of $\mathbf{Ter}(\Gamma \vdash \mathbb{I})$ and $\mathbf{Ter}(\Gamma \vdash \mathbb{F})$. Context extensions of the form $\Gamma.\mathbb{I}$ are simply interpreted as normal context extensions.

We define \mathbb{I} to be the cubical set defined by $\mathbb{I}(I) \triangleq \mathbf{dM}(I)$ which sends morphisms to their associated De Morgan algebra homomorphisms. An alternative characterisation is that \mathbb{I} is the Yoneda embedding of a single element set, $\mathbf{y}\{i\} = \square(-, \{i\})$. All of the required structure on the interval $0, 1, - \vee -, - \wedge -, \text{etc}$, extends to \mathbb{I} . For example, we get a map $\vee : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$ like so:

$$\vee_I(r, s) \triangleq r \vee s$$

which is natural since, for any $f \in \square(J, I) = I \rightarrow \mathbf{dM}(J)$, we have $\overline{f}(r \vee s) = \overline{f}(r) \vee \overline{f}(s)$ where $\overline{f} : \mathbf{dM}(I) \rightarrow \mathbf{dM}(J)$ is the De Morgan homomorphism corresponding to f .

We define \mathbb{F} to be the cubical set which sends a set of symbols I to the set of face formulae, as defined in Section 2.2.3, which only mention variables in I . The action of \mathbb{F} on a morphism $f \in \square(J, I) = I \rightarrow \mathbf{dM}(J)$ is simply the act of substituting variables $i \in I$ for $f(i) \in \mathbf{dM}(J)$ and then performing the following rewrites,

$$\begin{array}{llll} (r \vee s) = 1 & \mapsto & (r = 1) \vee (s = 1) & (r \vee s) = 0 & \mapsto & (r = 0) \wedge (s = 0) \\ (r \wedge s) = 1 & \mapsto & (r = 1) \wedge (s = 1) & (r \wedge s) = 0 & \mapsto & (r = 0) \vee (s = 0) \\ (1 - r) = 1 & \mapsto & r = 0 & (1 - r) = 0 & \mapsto & r = 1 \end{array}$$

in order to get a well-formed face formula.

Given $\varphi \in \text{Ter}(\Gamma \vdash \mathbb{F})$ we define the restricted context Γ, φ to be the cubical set:

$$(\Gamma, \varphi)(I) \triangleq \{\rho \in \Gamma(I) \mid \varphi(I, \rho) = 1_{\mathbb{F}}\}$$

We now have an interpretation for the interval \mathbb{I} from Section 2.2.1 and the face lattice \mathbb{F} from Section 2.2.3. However, with what we have described so far we are not currently able to interpret either the univalence axiom or the Kan composition operation from Section 2.2.4. To do this we introduce the notion of a fibrant type.

3.3.2 Fibrant types

Since every presheaf category forms a CwF as described in Definition 3.2.1 we see that the category of cubical sets, $\widehat{\square}$, forms a model of intensional type theory with all the usual type formers, including intentional identity types. However, the standard interpretation of identity types in this CwF will not coincide with the desired interpretation as maps out of the interval \mathbb{I} . In particular, as we remarked at the end of Section 3.2, these identity types will validate the principle of UIP and hence will not allow us to interpret the univalence axiom.

Another problem with the standard CwF construction is that we will not be able to interpret the composition operation described in Section 2.2.4. To resolve this issue we construct a new CwF out of the canonical one, with the same contexts and terms but only those types which have sufficient structure to interpret the composition operation. We call such types “fibrant” and write $\text{FTy}(\Gamma)$ for the set of types over a context Γ in this new CwF. A fibrant type will be a non-fibrant type with a composition structure, defined below. First we introduce the notion of a partial element.

Definition 3.3.3. Given $I \in \square$, $i \notin I$, $\rho \in \Gamma(I, i)$, $\varphi \in \mathbb{F}(I)$ a *partial element* of $A(\rho)$ is a family $u_f \in A(\rho f)$ for every $f : J \rightarrow I, i$ such that $\mathbb{F}(s_i \circ f)(\varphi) = 1_{\mathbb{F}}$, with the property that for any $g : K \rightarrow J$ we have $u_{f \circ g} = A(g)(u_f)$. We call φ the *extent* of the partial element.

Note that the term *partial element* is usually defined more generally than in the above definition. Indeed we will make extensive use of partial elements in the rest of this thesis. However, for the purposes of this chapter we restricted to a simplified form. See [18] for a more general account in this context, and later chapters of this thesis for an account using the internal language of topoi.

Definition 3.3.4 (Composition structure [18, Definition 13]). A *composition structure* for $A \in \text{Ty}(\Gamma)$ is an operation, comp , as follows. Given $I \in \square$, $i \notin I$, $\rho \in \Gamma(I, i)$, $\varphi \in \mathbb{F}(I)$, u a partial element of $A(\rho)$ of extent φ and $a_0 \in A(\rho(i0))$ such that $A(f)(a_0) = u_{(i0) \circ f}$ for

all $f : J \rightarrow I$ such that $\mathbb{F}(f)(\varphi) = 1_{\mathbb{F}}$, we have

$$\mathbf{comp}(I, i, \rho, \varphi, u, a_0) \in A(\rho(i1))$$

which is *uniform* in the sense that for any $f : J \rightarrow I$ and $j \notin J$ we have

$$\mathbf{comp}(I, i, \rho, \varphi, u, a_0)f = \mathbf{comp}(J, j, \rho(f, i = j), \varphi f, u(f, i = j), a_0 f)$$

and

$$\mathbf{comp}(I, i, \rho, 1_{\mathbb{F}}, u, a_0) = u_{(i1)}$$

where $(f, i = j) : J, j \rightarrow I, i$ is the morphism that maps i to j and otherwise acts like f , and $u(f, i = j)$ is the partial element given by $u(f, i = j)_g \triangleq u_{(f \circ g, i = j)}$.

Definition 3.3.5 (Fibrant types). A *fibrant type* consists of a pair $(A, \mathbf{comp}_A) \in \mathbf{FTy}(\Gamma)$ where $A \in \mathbf{Ty}(\Gamma)$ and \mathbf{comp}_A is a composition structure for A .

Theorem 3.3.6. *Replacing \mathbf{Ty} with \mathbf{FTy} in the standard \mathbf{CwF} on the category of cubical sets yields a new \mathbf{CwF} with dependent pairs, dependent functions and natural numbers. Further, this \mathbf{CwF} supports an interpretation of the composition operation described in Section 2.2.4.*

Proof. The first part is [18, Theorem 14]. The second is shown in [18, Section 8.3]. Note that this depends on the fact that composition structures can be reindexed, in the sense that, given $(A, \mathbf{comp}_A) \in \mathbf{FTy}(\Gamma)$ and $\gamma : \Delta \rightarrow \Gamma$, we get a composition structure $\mathbf{comp}_A[\gamma]$ for the type $A[\gamma]$ and hence $(A[\gamma], \mathbf{comp}_A[\gamma]) \in \mathbf{FTy}(\Delta)$. \square

3.3.3 Path types

We now define path types as described in Section 2.2.2. This definition is equivalent to the one given in the section entitled ‘‘Semantic Path Types’’ in [18].

Definition 3.3.7. Given a type $A \in \mathbf{Ty}(\Gamma)$ and terms $a, b \in \mathbf{Ter}(\Gamma \vdash A)$ define the path type $\mathbf{Path}_A(a, b) \in \mathbf{Ty}(\Gamma)$ as follows. Given $\rho \in \Gamma(I)$, let $\mathbf{Path}_A(a, b)(\rho)$ be the set of equivalences classes generated by pairs (i, w) with $i \notin I$ and $w \in A(\rho s_i)$ such that $w(i0) = a(\rho)$ and $w(i1) = b(\rho)$, where we identify (i, w) with (i', w') iff $w' = w(i = j)$. The action of $\mathbf{Path}_A(a, b)$ on a morphism $f : (J, \rho f) \rightarrow (I, \rho)$ is given by $(i, w) f \triangleq (j, w(f, i = j))$ for $j \notin J$.

While the above definition applies to any (not necessarily fibrant) type, we require that the type A is fibrant in order to conclude that \mathbf{Path}_A is fibrant. When we restrict to the \mathbf{CwF} of fibrant types then these path types correctly model weak intensional identity

types (i.e. with just a propositional computation rule). For example, given $A \in \mathbf{Ty}(\Gamma)$ and $a \in \mathbf{Ter}(\Gamma \vdash A)$, the $\mathbf{refl}_A(a)$ constructor can be defined as,

$$\mathbf{refl}_A(a)(I, \rho) \triangleq (i, a(\rho s_i))$$

where $i \notin I$. These results are summarised below.

Theorem 3.3.8. *The CwF of fibrant types supports weak intensional identity types given by \mathbf{Path}_A . More specifically, given $(A, \mathbf{comp}_A) \in \mathbf{FTy}(\Gamma)$ and $a, b \in \mathbf{Ter}(\Gamma \vdash A)$, we can define a new composition structure $\mathbf{comp}_{\mathbf{Path}_A(a,b)}$ such that $(\mathbf{Path}_A(a, b), \mathbf{comp}_{\mathbf{Path}_A(a,b)}) \in \mathbf{FTy}(\Gamma)$. Moreover this type supports all the required structure of weak intensional identity types for a CwF as described in Definition 3.1.3.*

Proof. By [18], where the J-eliminator is defined using the composition structures attached to types in the CwF of fibrant types. □

3.3.4 Glueing

The next feature of cubical type theory is the glueing construction described in Section 2.2.5. The definition of the glueing construction, and the proofs that it preserves fibrancy and has the correct properties are all quite involved. Indeed, one of the aims of this thesis is to analyse these proofs and hopefully to give a simpler presentation. As such we do not reproduce the original versions here. Instead we simply state the following theorem.

Theorem 3.3.9. *The CwF of fibrant types supports the interpretation of a glueing operation as defined in Section 2.2.5.*

Proof. By [18]. □

3.3.5 A univalent universe

The final aspect of cubical type theory which must be modelled is the existence of a univalent universe. As discussed in Section 2.2.6, we can prove that the universe is univalent from the other features of cubical type theory, particularly the glueing construction. However, this proof requires the existence of a universe which is closed under all the type formers, including glueing. Therefore, given the previous results, to model a univalent universe it suffices to model a universe which is closed under all the type formers.

In fact, Cohen et al. construct a universe containing all small fibrant types (for some notion of smallness in the metatheory). Since all of the type formers preserve smallness this universe will therefore be closed under the necessary type formers and hence will be univalent by the argument given in [18, Section 7].

The construction is a slightly modified version of the Hofmann-Streicher universe construction [31]. This definition relies on the existence of a notion of smallness in the ambient metatheory, e.g. a Grothendieck universe [14]. Given this, we say that a type $A \in \mathbf{Ty}(\Gamma)$ is small iff its fibres $A(\rho)$ are small for all $\rho \in \Gamma(I)$. We write $\mathbf{Ty}_0(\Gamma)$ and $\mathbf{FTy}_0(\Gamma)$ for the collections of small types and small fibrant types respectively. We can then define a universe of small types like so:

Definition 3.3.10. The universe in the cubical sets model is defined as follows. Define a cubical set $\mathcal{V} : \square^{\text{op}} \rightarrow \mathbf{Set}$ as the composition $\mathcal{V} \triangleq \mathbf{FTy}_0 \circ \mathbf{y}^{\text{op}}$. This means that, on objects, the functor gives the collection of fibrant types over the Yoneda embedding of an object:

$$\mathcal{V}(I) \triangleq \mathbf{FTy}_0(\mathbf{y}I)$$

As mentioned above, any cubical set can be regarded as a type in the empty (and therefore any other) context and so we have $\mathcal{V} \in \mathbf{Ty}(1)$. We then define a decoding function $El : \mathbf{Ter}(\Gamma \vdash \mathcal{V}) \rightarrow \mathbf{FTy}_0(\Gamma)$ by taking $El(a) \triangleq (El_a, \mathbf{comp}_a)$ where $El_a \in \mathbf{Ty}_0(\Gamma)$ is defined as

$$El_a(I, \rho) \triangleq \mathbf{fst}(a(\rho))(I, id_I)$$

and \mathbf{comp}_a is a composition structure for El_a defined as

$$\mathbf{comp}_a(I, i, \rho, \varphi, u, a_0) \triangleq \mathbf{snd}(a(\rho))(I, i, id, \varphi, u, a_0)$$

Finally we define a coding function $\ulcorner _ \urcorner : \mathbf{FTy}_0(\Gamma) \rightarrow \mathbf{Ter}(\Gamma \vdash \mathcal{V})$ like so:

$$\ulcorner (A, \mathbf{comp}_A) \urcorner(\rho) \triangleq (A, \mathbf{comp}_A)[\bar{\rho}]$$

where $\bar{\rho} : \mathbf{y}I \rightarrow \Gamma$ is $\rho \in \Gamma(I)$ transported across the Yoneda lemma. We can then easily check that these coding and decoding functions are inverses. Given $a : \mathbf{Ter}(\Gamma \vdash \mathcal{V})$ and $\rho \in \Gamma(I)$ then for any $f : J \rightarrow I$ we have

$$\begin{aligned} El_a[\bar{\rho}](f) &= El_a(\rho f) \\ &= \mathbf{fst}(a(\rho f))(id_J) \\ &= \mathbf{fst}(a(\rho))(f) \end{aligned}$$

and so $El_a[\bar{\rho}] = \mathbf{fst}(a(\rho))$. Similarly for the second component we have

$$\begin{aligned} \mathbf{comp}_a[\bar{\rho}](J, j, f, \varphi, u, a_0) &= \mathbf{comp}_a(J, j, \rho f, \varphi, u, a_0) \\ &= \mathbf{snd}(a(\rho f))(I, i, id, \varphi, u, a_0) \\ &= \mathbf{snd}(a(\rho))(I, i, f, \varphi, u, a_0) \end{aligned}$$

and so $\text{comp}_a[\bar{\rho}] = \text{snd}(a(\rho))$. Therefore

$$\begin{aligned} \ulcorner \text{El}(a) \urcorner(\rho) &= (\text{El}_a, \text{comp}_a)[\bar{\rho}] \\ &= (\text{fst}(a(\rho)), \text{snd}(a(\rho))) \\ &= a(\rho) \end{aligned}$$

and hence we have $\ulcorner \text{El}(a) \urcorner = a$. In the other direction, given $(A, \text{comp}_A) \in \text{FTy}_0(\Gamma)$, then for any $\rho \in \Gamma(I)$ we have

$$\begin{aligned} \text{fst}(\text{El}(\ulcorner(A, \text{comp}_A) \urcorner))(\rho) &= \text{fst}(\ulcorner(A, \text{comp}_A) \urcorner(\rho))(id_I) \\ &= \text{fst}((A, \text{comp}_A)[\bar{\rho}])(id_I) \\ &= A[\bar{\rho}](id_I) \\ &= A(\rho) \end{aligned}$$

and so $\text{fst}(\text{El}(\ulcorner(A, \text{comp}_A) \urcorner)) = A$. Then for the other projection we have

$$\begin{aligned} \text{snd}(\text{El}(\ulcorner(A, \text{comp}_A) \urcorner))(I, i, \rho, \varphi, u, a_0) &= \text{snd}(\ulcorner(A, \text{comp}_A) \urcorner(\rho))(I, i, id, \varphi, u, a_0) \\ &= \text{snd}((A, \text{comp}_A)[\bar{\rho}])(I, i, id, \varphi, u, a_0) \\ &= \text{comp}_A[\bar{\rho}](I, i, id, \varphi, u, a_0) \\ &= \text{comp}_A(I, i, \rho, \varphi, u, a_0) \end{aligned}$$

and so $\text{snd}(\text{El}(\ulcorner(A, \text{comp}_A) \urcorner)) = \text{comp}_A$. Therefore $\text{El}(\ulcorner(A, \text{comp}_A) \urcorner) = (A, \text{comp}_A)$ and hence the coding and decoding functions are mutual inverses. Therefore $(\text{FTy}_0, \mathcal{V}, \text{El}, \ulcorner _ \urcorner)$ define a universe in the sense of Huber [34, Definition 1.1.4].

Note that the above proves that $(\text{FTy}_0, \mathcal{V}, \text{El}, \ulcorner _ \urcorner)$ is a universe in the original CwF of (not necessarily fibrant) types. In order to conclude that it is also a universe in the CwF of fibrant types we need to show that the type \mathcal{V} is fibrant. This follows from the fact that the universe contains all small fibrant types and hence is closed under glueing. We can then use the glueing operation to define a composition structure on the universe as in [18, Section 7.1].

Theorem 3.3.11. *The CwF of fibrant types supports the interpretation of a universe of small fibrant types. Moreover, this universe is univalent.*

Proof. The first part is shown in the previous definition. The fact that the universe is univalent follows from the fact that it is closed under glueing. This means that we can apply the argument sketched in Section 2.2.6. See [18, Section 7] for full details or the main body of this thesis for an account of this fact using the internal language of topoi. \square

Chapter 4

The internal type theory of a topos

The previous chapters described the formal system of cubical type theory and its model in the topos of cubical sets. In the subsequent chapters of this thesis I present an account of this model working in the internal language of topoi. In order to do so I first explain what exactly this language is and how it can be used to work with models of cubical type theory.

4.1 Elementary topoi

We begin by introducing the notion of an (elementary) topos. A topos is a category which behaves like the category of sets. Indeed, the category of sets is the canonical example of a topos. Every topos comes equipped with an object known as the *subobject classifier*:

Definition 4.1.1. In any category \mathcal{E} with all finite limits, a subobject classifier is an object $\Omega \in \mathcal{E}$ equipped with a monomorphism $\top : 1 \rightarrow \Omega$, such that for every monomorphism $m : X \rightarrow Y$ there exists a unique morphism $\chi_X : Y \rightarrow \Omega$ making the following diagram into a pullback square:

$$\begin{array}{ccc} X & \xrightarrow{m} & Y \\ \downarrow & \lrcorner & \downarrow \chi_X \\ 1 & \xrightarrow{\top} & \Omega \end{array}$$

We say that the map χ_X *classifies* the monomorphism $m : X \rightarrow Y$, or simply that χ_X classifies the subobject X . Every subobject of Y is classified uniquely (up-to isomorphism) by a map $Y \rightarrow \Omega$.

Definition 4.1.2. A topos is a cartesian closed category with all finite limits and a subobject classifier.

Note that while the definition is quite simple and concise it in fact implies many other properties. For example, every topos is locally cartesian closed (every slice category is cartesian closed), has all finite colimits, and admits the interpretation of extensional type theory as an internal language. It is this final property of topoi that we will make extensive use of in the rest of this thesis.

4.1.1 Examples

We now briefly describe two examples of topoi. Firstly, the canonical example of a topos: the category of sets. Secondly, we explain why any presheaf category forms a topos.

Example 4.1.3 (The topos \mathbf{Set}). *\mathbf{Set} is well known to be cartesian closed and have all finite limits, therefore we simply outline definition of the subobject classifier. In \mathbf{Set} the subobject classifier Ω is given by the two element set $\{0, 1\}$. The classifying map $\chi_X : Y \rightarrow \{0, 1\}$ for a subobject $m : X \rightarrowtail Y$ is essentially just the characteristic map for X regarded as a subset of Y , explicitly:*

$$\chi_X(y) \triangleq \begin{cases} 1 & \text{if } \exists x \in X. m(x) = y \\ 0 & \text{otherwise} \end{cases}$$

Example 4.1.4 (Presheaf topoi). *Given any small category, \mathbf{C} , the presheaf category, $\widehat{\mathbf{C}}$ forms a topos. Again, we only describe the construction of the subobject classifier Ω . In the case of presheaves, $\Omega(I)$ is defined to be the set of sieves on I . That is, sets of morphisms into I which are closed under precomposition with arbitrary morphisms. The classifying map $\chi_X : Y \rightarrow \Omega$ for a subobject $m : X \rightarrowtail Y$ is given by:*

$$(\chi_X)_I(y) \triangleq \{f : J \rightarrow I \mid \exists x \in X(J). m_J(x) = Y(f)(y)\}$$

This set is closed under precomposition because, given $f : J \rightarrow I$ with $x \in X(J)$ such that $m_J(x) = Y(f)(y)$ and any $g : K \rightarrow J$, then taking $x' = X(g)(x)$ we have $m_K(x') = m_K(X(g)(x)) = Y(g)(m_J(x)) = Y(g)(Y(f)(y)) = Y(f \circ g)(y)$ and hence $f \circ g$ is also in the sieve.

4.2 The internal type theory

As mentioned above, every topos possesses a powerful internal language which may be presented in several ways. The most common presentation is in terms of a many-sorted higher-order logic, often called the Mitchell-Bénabou language [40]. However, in this thesis we use a presentation in terms of extensional Martin-Löf type theory along the lines of those discussed in [48].

This language includes a type for every object of the topos \mathcal{E} , and all the usual type formers of MLTT: dependent pairs, dependent products and identity types. Note that these identity types are extensional, which means not only that UIP always holds, but in fact propositional equality coincides with definitional equality and so, for example, if we know that $b : B a$ and that $a = a'$ then we can conclude $b : B a'$ with no explicit coercion. Extensional identity types also satisfy function extensionality.

In order to interpret this language in the topos we rely on the categorical semantics of dependent type theory in terms of categories with families, as described in Section 3.1. For each topos \mathcal{E} one can find a CwF with the same objects, such that the category of families at each object X is equivalent to the slice category \mathcal{E}/X . This can be done in a number of different ways; for example [53, Example 6.14], or the more recent references [39, Section 1.3], [46] and [7], which cater for categories more general than a topos (and for contextual/comprehension categories rather than CwFs in the first two cases). Here we sketch the construction given in [53, Example 6.14]:

Definition 4.2.1 (The CwF associated with any topos). Given a topos \mathcal{E} with subobject classifier $\top : 1 \rightarrow \Omega$, we construct a CwF whose category of contexts is simply given by \mathcal{E} . For each object $\Gamma \in \mathcal{E}$ define $\mathbf{Ty}(\Gamma)$ as the collection of pairs (A, a) where $A \in \mathcal{E}$ and $a : \Gamma \times A \rightarrow \Omega$. The set of terms $\mathbf{Ter}(\Gamma \vdash (A, a))$ then consists of morphisms $t : \Gamma \rightarrow A$ making the following diagram commute¹:

$$\begin{array}{ccc} \Gamma & \xrightarrow{(id, t)} & \Gamma \times A \\ \downarrow & & \downarrow a \\ 1 & \xrightarrow{\top} & \Omega \end{array}$$

Given any $\gamma : \Delta \rightarrow \Gamma$ we can reindex types $(A, a) \in \mathbf{Ty}(\Gamma)$ and terms $t \in \mathbf{Ter}(\Gamma \vdash A)$ like so:

$$(A, a)[\gamma] = (A, a \circ (\gamma, id)) \in \mathbf{Ty}(\Delta) \quad t[\gamma] = t \circ \gamma \in \mathbf{Ter}(\Delta \vdash (A, a)[\gamma])$$

The reindexed term, $t[\gamma]$, satisfies the condition required to be an element of the set $\mathbf{Ter}(\Delta \vdash (A, a)[\gamma])$ since:

$$a \circ (\gamma, id) \circ (id, t \circ \gamma) = a \circ (\gamma, t \circ \gamma) = a \circ (id, t) \circ \gamma = \top \circ 1_\Gamma \circ \gamma = \top \circ 1_\Delta$$

It is easy to see that this definition of reindexing is functorial in a strict sense, without any need for a coherent choice of pullbacks in \mathcal{E} , since reindexing of both types and terms is defined using composition. The extended context $\Gamma.(A, a)$ is simply given by the subobject

¹We use the notation $(-, -)$ for the pairing map, rather than $\langle -, - \rangle$, to avoid confusion with the unique morphism that exists as part of the CwF structure.

classified by a , ie. the domain of $\bar{a} : \Gamma.(A, a) \rightarrow \Gamma \times A$. The projection $\mathbf{p} : \Gamma.(A, a) \rightarrow \Gamma$ and generic element $\mathbf{q} \in \mathbf{Ter}(\Gamma.(A, a) \vdash A[\mathbf{p}])$ are then given by:

$$\mathbf{p} \triangleq \mathbf{fst} \circ \bar{a} \qquad \mathbf{q} \triangleq \mathbf{snd} \circ \bar{a}$$

Finally, we show how to construct the unique map into an extended context. Given $\Delta, \Gamma \in \mathcal{E}$, $(A, a) \in \mathbf{Ty}(\Gamma)$, $\gamma : \Delta \rightarrow \Gamma$, then $t \in \mathbf{Ter}(\Delta \vdash (A, a)[\gamma])$ is a morphism $t : \Delta \rightarrow A$ such that $a \circ (\gamma, id) \circ (id, t) = \top \circ 1_\Delta$. Therefore, the outer square in the following diagram commutes,

$$\begin{array}{ccc}
 \Delta & \xrightarrow{(\gamma, t)} & \Gamma \times A \\
 \downarrow \langle \gamma, t \rangle & \searrow & \downarrow a \\
 \Gamma.(A, a) & \xrightarrow{\bar{a}} & \Gamma \times A \\
 \downarrow \lrcorner & & \downarrow a \\
 1 & \xrightarrow{\top} & \Omega
 \end{array}$$

and hence we get a unique map $\langle \gamma, t \rangle : \Delta \rightarrow \Gamma.(A, a)$ satisfying $\mathbf{p} \circ \langle \gamma, t \rangle = \gamma$ and $\mathbf{q}[\langle \gamma, t \rangle] = t$, as required.

Using the objects, families and elements of this CwF as a signature, we get an internal type theory along the lines of those discussed in [48], canonically interpreted in the above CwF in the standard fashion [30]. This CwF supports dependent pairs, dependent functions and strict extensional identity types. In addition, the universal property of the subobject classifier gives rise to *comprehension subtypes*.

Definition 4.2.2 (Comprehension subtypes). Given $\Gamma \vdash A$ and $\Gamma, x : A \vdash \varphi(x) : \Omega$, then we can form the comprehension subtype $\Gamma \vdash \{x : A \mid \varphi(x)\}$ with the following introduction and elimination rules:

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash \varphi(t)}{\Gamma \vdash t : \{x : A \mid \varphi(x)\}} \qquad \frac{\Gamma \vdash t : \{x : A \mid \varphi(x)\}}{\Gamma \vdash t : A}$$

Note that there are no explicit coercions between the original type and the comprehension subtype.

Comprehension subtypes can be interpreted in the CwF described in Definition 4.2.1 like so: given $\Gamma \in \mathcal{E}$, $(A, a) \in \mathbf{Ty}(\Gamma)$ and $\varphi \in \mathbf{Ter}(\Gamma.(A, a) \vdash \Omega)$ we interpret the comprehension subtype as the pair $(A, a') \in \mathbf{Ty}(\Gamma)$ where a' is the dependent conjunction of a and φ . More precisely, we have that $\varphi : \Gamma.(A, a) \rightarrow \Omega$ classifies a monomorphism $\bar{\varphi} : \cdot \rightarrow \Gamma.(A, a)$. Therefore we have $\bar{a} \circ \bar{\varphi} : \cdot \rightarrow \Gamma \times A$ where $\bar{a} : \Gamma.(A, a) \rightarrow \Gamma \times A$ is the monomorphism classified by a . We then take $a' = \chi_{\bar{a} \circ \bar{\varphi}} : \Gamma \times A \rightarrow \Omega$.

We now show the soundness of the introduction and elimination rules. For the introduction rule, first observe that $a' \circ \bar{a} = \varphi$ because, by the definition of a' , the left diagram is a pullback square and therefore, since \bar{a} is mono, so is the right diagram.

$$\begin{array}{ccc}
\Gamma.(A, a') \xrightarrow{\bar{\varphi}} \Gamma.(A, a) \xrightarrow{\bar{a}} \Gamma \times A & & \Gamma.(A, a') \xrightarrow{\bar{\varphi}} \Gamma.(A, a) \\
\downarrow \lrcorner & & \downarrow \lrcorner \\
1 \xrightarrow{\top} \Omega & \xrightarrow{\bar{a} \text{ mono}} & 1 \xrightarrow{\top} \Omega \\
& & \downarrow a' \quad \downarrow a' \circ \bar{a}
\end{array}$$

This means that $a' \circ \bar{a}$ classifies the monomorphism $\bar{\varphi}$. However, since classifying morphisms are unique we have $a' \circ \bar{a} = \varphi$. Given the hypotheses of the introduction rule: $t \in \mathbf{Ter}(\Gamma \vdash (A, a))$ such that $\varphi \circ \langle id, t \rangle = \top \circ 1_\Gamma$, we have:

$$a' \circ \langle id, t \rangle = a' \circ \bar{a} \circ \langle id, t \rangle = \varphi \circ \langle id, t \rangle = \top \circ 1_\Gamma$$

Therefore we deduce $t \in \mathbf{Ter}(\Gamma \vdash (A, a'))$.

Next, we consider the elimination rule. Given any term $t \in \mathbf{Ter}(\Gamma \vdash (A, a'))$ we have $\langle id, t \rangle = \bar{a}' \circ \langle id, t \rangle = \bar{a} \circ \bar{\varphi} \circ \langle id, t \rangle$ and so,

$$a \circ \langle id, t \rangle = a \circ \bar{a} \circ \bar{\varphi} \circ \langle id, t \rangle = \top \circ 1_{\Gamma.(A, a)} \circ \bar{\varphi} \circ \langle id, t \rangle = \top \circ 1_\Gamma$$

hence $t \in \mathbf{Ter}(\Gamma \vdash (A, a))$. Therefore $\mathbf{Ter}(\Gamma \vdash (A, a')) \subseteq \mathbf{Ter}(\Gamma \vdash (A, a))$, validating the elimination rule.

Definition 4.2.3 (Subsingletons). Taking $A = 1$ in the previous definition we have, for each $\varphi : \Omega$, a type whose inhabitation corresponds to provability of φ :

$$[\varphi] \triangleq \{- : 1 \mid \varphi\}$$

We will use this type extensively in the rest of this thesis.

In following chapters we will show how to model one type theory (cubical type theory) using a different type theory (the internal language of a topos). We use different syntax in order to help differentiate between the two. When working in the object language (cubical type theory) we use the notation introduced in Chapter 2 (Σ , Π , etc). When we make definitions and postulates in this internal language for \mathcal{E} we instead use a concrete syntax inspired by Agda [4]. Dependent function types are written as $(x : A) \rightarrow B$; their canonical terms are function abstractions, written as $\lambda(x : A) \rightarrow t$. Dependent product types are written as $(x : A) \times B$; their canonical terms are pairs, written as (s, t) . In the text we use this language informally, similar to the way that Homotopy Type Theory is presented in [62]. For example, the typing contexts of the judgements in the formal version, such

as $[x_0 : A_0, x_1 : A_1(x_0), x_2 : A_2(x_0, x_1)]$, become part of the running text in phrases like “given $x_0 : A_0, x_1 : A_1(x_0)$ and $x_2 : A_2(x_0, x_1)$, then...”

In the internal type theory the subobject classifier Ω of the topos becomes an impredicative universe of propositions, with logical connectives ($\top, \perp, \neg, \wedge, \vee, \Rightarrow$), quantifiers ($\forall(x : A), \exists(x : A)$) and extensional equality ($=$) satisfying UIP, as well as function and proposition extensionality properties. These propositions will not (in general) satisfy the law of excluded middle or the axiom of choice. However, they always satisfy the *axiom of unique choice* for any type A . This can be expressed in the internal type theory as the existence of a term:

$$\text{uc} : (\varphi : A \rightarrow \Omega) \rightarrow [\exists!(a : A). \varphi a] \rightarrow \{a : A \mid \varphi a\} \quad (4.1)$$

where $\exists!(a : A). \varphi a \triangleq \exists(a : A).(\varphi a \wedge \forall(a' : A). \varphi a' \Rightarrow a = a')$.

Finally, we will assume \mathcal{E} comes with an internal full subtopos \mathcal{U} . In the internal language we use \mathcal{U} as a Russell-style universe (that is, if $A : \mathcal{U}$, then A itself denotes a type) containing Ω and closed under forming products, exponentials and comprehension subtypes. Note that not every topos will contain such an internal full subtopos. However, this assumption is mostly just a convenience to avoid having to quantify externally over the objects, families and elements of the CwF associated with \mathcal{E} . Most of the results in this thesis should apply in topoi without an internal universe, the exception being the work on modelling type theoretic universes described in Chapter 7 which does require an internal universe from which we construct the new universe of fibrant types.

We will also adopt a couple of useful notational conventions from Agda. Firstly, function arguments that are written with infix notation are indicated by the placeholder notation “_”; for example $_ \sqcap _ : \mathbb{I} \rightarrow \mathbb{I} \rightarrow \mathbb{I}$ applied to $i, j : \mathbb{I}$ is written $i \sqcap j$. Secondly, we use the convention that braces $\{\}$ indicate implicit arguments; for example, the application of ax_9 in Figure 5.4 to $\varphi : \text{Cof}, A : [\varphi] \rightarrow \mathcal{U}, B : \mathcal{U}$ and $s : (u : [\varphi]) \rightarrow (A u \cong B)$ is written $\text{ax}_9 A B s$, or $\text{ax}_9 \{\varphi\} A B s$ if φ cannot be deduced from the context.

Remark 4.2.4. *Note that the standard interpretation of the syntax of type theory in an arbitrary CwF [30] assumes that the syntax is annotated with additional typing information in certain places. For example, if $t : (x : A) \rightarrow B$ and $u : A$ then the application, $t u$, should in fact be annotated with A and B , e.g. $t u$ should be written as $\text{app}_{x:A.B}(t, u)$. When using the internal language in this thesis we will omit these annotations in most places and conjecture that any missing annotations can be inferred from the context. We believe that this conjecture is supported by the existence of the Agda development described in the following section where such annotations are either inferred or given explicitly in the code.*

4.3 The proof assistant Agda

Many of the definitions, lemmas and theorems presented in this thesis have been formalised in the proof assistant Agda [4]. This Agda development can be found at <https://doi.org/10.17863/CAM.35681>.

One advantage to the approach of working in the internal type theory of a topos, rather than working externally with the objects and morphisms of the topos, is that this approach better lends itself to machine-assisted formalisation. To understand why, consider what formalising a model constructed externally would entail. Firstly, we would need to define all the mathematical structures (categories, functors, etc) and formalise all their relevant properties used in the model construction. Next, we would have to formalise the actual model itself, using the structures defined in the previous step. By working in the internal language we can essentially remove the entire first step. This is because, rather than formalising the notion of a topos and its internal language in a proof assistant, we simply use the proof assistant as if it were the internal language of a topos.

Agda is a proof assistant based on an intensional form of MLTT. While the language provided by Agda is not quite the same as the internal type theory of a topos we found ways to adapt it so as to make it sufficiently close. Agda does not require explicit type annotations of the kind described in Remark 4.2.4, but instead infers these types as part of its type-checking algorithm. Given these adaptations, we were able to formalise all of our arguments fairly straightforwardly.

The first difference between the two languages is that Agda's identity types are intensional whereas the internal type theory's are extensional. This means that we are required to do more work in the formalisation than in the proofs presented in this thesis. For example, we might have to explicitly coerce between propositionally equal types. Something which is implicit in the internal language. However, this difference did not cause any real issues when formalising the work presented here. We also added some additional extensionality principles to Agda using postulates, specifically function and propositional extensionality.

We also made modest use of the facility for user-defined rewriting in recent versions of Agda [17], in order to make certain postulated equalities definitional, rather than just propositional, thereby eliminating a few proofs in favour of computation.

The second difference is that Agda does not, by default, include an impredicative universe of propositions corresponding to the subobject classifier Ω . However, we persuaded it to provide such a universe of propositions using a method due to Escardo [24]. This method works by using Agda's type-in-type pragma to define Ω as the type of all (small) mere propositions [62, Section 3.3]. That is, a proposition in Ω is given by a pair (prf, equ) where $prf : Set$ is the type of proofs for the proposition and $equ : (u v : prf) \rightarrow u = v$ witnesses the fact that any two proofs are equal. The exact definition can be seen in Figure

```

{-# OPTIONS --type-in-type #-}
record  $\Omega$  : Set where
  constructor prop
  field
    prf : Set
    equ : (u v : prf)  $\rightarrow$  u  $\equiv$  v

```

Figure 4.1: The definition of Ω in Agda

4.1. This construction gives us an intensional, proof-relevant version of the subobject classifier in the sense that two proofs $u, v : P$ for $P : \Omega$ are not definitionally equal but only propositionally equal, and are explicitly passed around in proofs.

Usually a definition like this would mean that Ω would be an element of Set_1 (i.e. in the next universe), however, the use of the type-in-type option means that Agda accepts the definition of Ω as an element of Set . In general, the use of type-in-type is unsound since it allows us to encode a type of all types and then replay Russel’s paradox. However, in our development we only use type-in-type for the definition of Ω , which is separated into a separate module. The type-in-type option is then not enabled outside of this module, and we believe this restricted use to be consistent.

4.4 A remark on impredicativity

While having a universe of propositions satisfying propositional extensionality is an essential part of the constructions presented in this thesis, we believe that the impredicative nature of Ω is in fact unnecessary. Instead, we could simply work in a predicative style using types in \mathcal{U} which are mere propositions in the sense of the HoTT book [62, Section 3.3] and postulate the necessary uses of propositional extensionality.

For example, later we will introduce a certain class of propositions known as *cofibrant* propositions. This class of propositions is specified by a map $\text{cof} : \Omega \rightarrow \Omega$. However, they could instead have been specified by a map $\text{cof} : \mathcal{U} \rightarrow \mathcal{U}$ with axioms stating that: $\text{cof } \varphi$ is always a mere proposition, cofibrant types are always mere propositions, and cofibrant types always satisfy propositional extensionality. Formally, we could postulate:

$$\begin{aligned}
 \text{isPropcof} &: (\varphi : \mathcal{U})(u\ v : \text{cof } \varphi) \rightarrow u \equiv v \\
 \text{cofisProp} &: (\varphi : \mathcal{U})(_ : \text{cof } \varphi)(x\ y : \varphi) \rightarrow x \equiv y \\
 \text{cofExt} &: (\varphi\ \psi : \mathcal{U})(_ : \text{cof } \varphi)(_ : \text{cof } \psi) \rightarrow (\varphi \rightarrow \psi) \rightarrow (\psi \rightarrow \varphi) \rightarrow \varphi \equiv \psi
 \end{aligned}$$

where $_ \equiv _$ is the internal (strict) identity type.

We believe that this approach would be sufficient to carry out the constructions presented in this thesis. In particular, the material presented in Chapter 7, which is presented in the impredicative style, is adapted from a paper [44] where the same material

is presented in the predicative style.

The advantage of using an impredicative style is that it simplifies certain aspects of the presentation. For example, in the impredicative presentation we will define the collection of cofibrant propositions as

$$\mathbf{Cof} \triangleq \{\varphi : \Omega \mid \mathbf{cof} \varphi\}$$

and we have $\mathbf{Cof} : \mathcal{U}$. In the impredicative presentation we would instead define this collection as

$$\mathbf{Cof} \triangleq (A : \mathcal{U}) \times \mathbf{cof} A$$

and then we would have $\mathbf{Cof} : \mathcal{U}_1$ where \mathcal{U}_1 is some larger internal universe with $\mathcal{U} : \mathcal{U}_1$. This happens because this type is a dependent product over the original universe \mathcal{U} and hence must be in a larger universe. This means that accounting for universe levels in various definitions becomes more complicated. The impredicative presentation also more closely matches the true internal language of a topos.

Conversely, the advantage of the predicative style is twofold. Firstly, some mathematicians reject impredicativity on philosophical grounds, either outright or because defining the subobject classifier in the category of sets requires classical logic in the metatheory (the definition of χ_X in Example 4.1.3 requires deciding the statement $\exists x \in X. m(x) = y$). The other, more practical, advantage is that there are potentially more models of the predicative axioms than of the impredicative ones. For example, the predicative axioms could potentially be satisfied by a category which does not form a topos, but merely something weaker such as a locally cartesian closed category or a stratified pseudotopos [50] (or any other notion of “predicative topos”). Certainly, any impredicative model will also be a predicative model since, given $\mathbf{cof} : \Omega \rightarrow \Omega$, we can define $\mathbf{cof}' : \mathcal{U} \rightarrow \mathcal{U}$ as

$$\mathbf{cof}' A \triangleq \{\varphi : \Omega \mid \mathbf{cof} \varphi \wedge [\varphi] = A\}$$

and show that \mathbf{cof}' satisfies the three properties mentioned above.

Overall, we opted for the impredicative presentation in this thesis but we included this section to highlight the fact that our approach does not depend on this impredicativity and that we are confident that the results here should generalise to any predicative setting.

For the Agda development, instead of using the type-in-type option, we could simply have defined Ω to be in the next universe up, and then worked with a larger type of propositions. However, as mentioned above, the approach that we have taken more closely matches the true internal language of a topos and also avoids additional boilerplate code relating to universe levels.

Chapter 5

Axioms for modelling cubical type theory in a topos

In Chapter 4 we saw how every topos admits a rich internal language based on extensional Martin-Löf type theory. In the rest of this thesis we show how the model of cubical type theory described in Section 3.3 can be presented using this internal language. In doing so we implicitly generalise the model given by Cohen et al. [18] to any topos with the appropriate structure. In order to specify this additional structure we take an axiomatic approach, giving a set of axioms which we require to hold in the internal language. Our presentation of the model then makes use of these axioms, rather than assuming that we are always working in the topos described in Section 3.3.

This approach has several advantages. Firstly, it allows for a better understanding of exactly which properties of the topos of cubical sets make it suitable for modelling cubical type theory; which properties are essential, which are superfluous. Secondly, it allows us to see exactly how these properties are used and for which constructions. For example, we will see that a closure property of a certain class of propositions is only required to model the definitional computation rule for the J-eliminator, and can be dispensed with if one does not demand this property of the type theory. Thirdly, this approach makes it easier to find new models of cubical type theory. This is because the axioms clarify the properties that a topos must possess in order to model cubical type theory. Indeed, there is a direct and fairly obvious model in the Lawvere theory defined by a subset of the axioms, although this is in fact very similar to the model given by Cohen et al. [18]. Alternatively, rather than looking for new topoi which model cubical type theory we may want to check if existing topoi of interest, such as the topos of simplicial sets [39], are also potential models. Using the work presented here this task is simplified to checking a few simple properties of the topos.

5.1 The axioms

In this section we present the axioms that we require to hold in the internal type theory of a topos \mathcal{E} . We provide an overview of each axiom, giving some intuition as to its purpose and we explain where it is used in the construction of a model of cubical type theory. This allows us to see that certain axioms are only required for modelling specific parts of cubical type theory, for example definitional identity types (Section 5.3.4). These axioms can therefore be dropped when, for example, looking for models of cubical type theory with only propositional identity types (Section 5.2). For ease of reference the axioms are collected together at the end of this section in Figure 5.4, written in the language described in Chapter 4.

5.1.1 The interval \mathbb{I}

We begin by axiomatising the structure needed to model type theory with an interval object \mathbb{I} , as described in Section 2.2.1. Firstly, we assume the existence of an object $\mathbb{I} : \mathcal{U}$ which will be the interpretation of \mathbb{I} in the model¹. We assume that \mathbb{I} has sufficient structure to model the required operations on the interval. Specifically, we assume that \mathbb{I} comes equipped with morphisms $0, 1 : 1 \rightarrow \mathbb{I}$ and $-\sqcap-, -\sqcup- : \mathbb{I} \rightarrow \mathbb{I} \rightarrow \mathbb{I}$ satisfying axioms \mathbf{ax}_1 – \mathbf{ax}_4 in Figure 5.1.

Axiom \mathbf{ax}_1 expresses that the interval \mathbb{I} is internally connected, in the sense that any decidable subset of its elements is either empty or the whole of \mathbb{I} . This implies that if a path in an inductive datatype has a certain constructor form at one point of the path, it has the same form at any other point. This is used at the end of Section 5.3.3 to show that the natural number object in the topos is fibrant (that is, denotes a type) and that fibrations are closed under binary coproducts. It also gets used in proving properties of the glueing construct in Section 5.4. Together with axiom \mathbf{ax}_2 , connectedness of \mathbb{I} implies that there is no path from $\mathbf{inl}*$ to $\mathbf{inr}*$ in $1 + 1$ and hence that the path-based model of Martin-Löf type theory determined by the axioms is logically non-degenerate.

Axioms \mathbf{ax}_3 and \mathbf{ax}_4 endow \mathbb{I} with a form of *connection algebra* structure [15]. They capture some very simple properties of the minimum and maximum operations on the unit interval $[0, 1]$ of real numbers that suffice to ensure contractibility of singleton types (Section 5.2) and, in combination with subsequent axioms, to define path lifting from composition for fibrations (see Section 5.3.1). In the model of [18] the connection algebra structure is given by the lattice structure of the interval, taking $-\sqcap-$ to be binary meet, $-\sqcup-$ to be binary join and using the fact that 0 and 1 are respectively least and greatest elements.

¹To be precise, since objects of the topos in fact model contexts in the type theory, \mathbb{I} will actually be the interpretation of the singleton context $i : \mathbb{I} \vdash$

The interval $\mathbb{I} : \mathcal{U}$ is connected

$$\mathbf{ax}_1 : [\forall(\varphi : \mathbb{I} \rightarrow \Omega). (\forall(i : \mathbb{I}). \varphi i \vee \neg\varphi i) \Rightarrow (\forall(i : \mathbb{I}). \varphi i) \vee (\forall(i : \mathbb{I}). \neg\varphi i)]$$

has distinct end-points $0, 1 : \mathbb{I}$

$$\mathbf{ax}_2 : [\neg(0 = 1)]$$

and a connection algebra structure $-\sqcap-, -\sqcup- : \mathbb{I} \rightarrow \mathbb{I} \rightarrow \mathbb{I}$

$$\mathbf{ax}_3 : [\forall(i : \mathbb{I}). 0 \sqcap x = 0 = x \sqcap 0 \wedge 1 \sqcap x = x = x \sqcap 1]$$

$$\mathbf{ax}_4 : [\forall(i : \mathbb{I}). 0 \sqcup x = x = x \sqcup 0 \wedge 1 \sqcup x = 1 = x \sqcup 1].$$

Figure 5.1: The axioms for the interval

Remark 5.1.1 (De Morgan involution). *Note that in Section 2.2.1 and in the model of [18] \mathbb{I} is not just a lattice, but also has an involution operation $1 - (-) : \mathbb{I} \rightarrow \mathbb{I}$ (so that $(1 - (1 - i)) = i$) making \sqcup the De Morgan dual of \sqcap , in the sense that $i \sqcup j = 1 - ((1 - i) \sqcap (1 - j))$. Although this involution structure is convenient, it is not really necessary for the constructions that follow. Instead we just give a 0-version and a 1-version of certain concepts; for example, “composing from 1 to 0” as well as “composing from 0 to 1” in Section 5.3.1.*

Axioms \mathbf{ax}_2 – \mathbf{ax}_4 , along with two more subsequent axioms, will allow us to show that fibrations provide a model of Π - and Σ -types; and furthermore to show that the path types determined by the interval object \mathbb{I} (Section 5.2) satisfy the rules for identity types propositionally [20, 63].

5.1.2 Cofibrant propositions

Next, we need to axiomatise the properties of the face lattice \mathbb{F} from cubical type theory. One approach to doing this would be to assume existence of a object modelling \mathbb{F} with certain algebraic properties, as we did for \mathbb{I} in the previous section. However, instead we take a different approach based on the original intuition behind the face lattice as specifying the input to a Kan-like filling problem, as described in see Section 2.2.4.

Kan filling and other cofibrancy conditions on collections of subspaces have to do with extending maps defined on a subspace to maps defined on the whole space. Here we take “subspaces of spaces” to mean subobjects of objects in toposes. Since subobjects are classified by morphisms to Ω , it is possible to consider collections of subobjects that are specified generically by certain propositions. More specifically, given a property of propositions, $\mathbf{cof} : \Omega \rightarrow \Omega$, we get a corresponding collection of propositions

$$\mathbf{Cof} \triangleq \{\varphi : \Omega \mid \mathbf{cof} \varphi\} \tag{5.1}$$

Using this intuition we internalise the face lattice \mathbb{F} as such a collection of propositions. We can then axiomatise the properties that we require of this collection in order to construct a model of cubical type theory. These properties are shown in Figure 5.2. Axioms \mathbf{ax}_5 , \mathbf{ax}_6 and \mathbf{ax}_7 closely mirror to the last four cases in the grammar, given in Section 2.2.3, which generates elements of the face lattice. In this presentation \mathbf{ax}_7 makes explicit that the conjunction may be dependent, something which is not clear in the syntactic presentation. Axiom \mathbf{ax}_8 states that cofibrant propositions are closed under I-indexed quantification. As with the dependency in \mathbf{ax}_7 , this is something which is not explicitly added to the face lattice in the syntactic presentation, but rather is something which happens to be true for the freely generated set of terms which define \mathbb{F} . Axiom \mathbf{ax}_8 is used to prove the realignment lemma (Lemma 5.3.10) which is used in the definition of the weak form of glueing to ensure that the induced fibration structure extends the fibration structure on the family that we are “glueing”.

Consider the class of monomorphisms $m : A \multimap B$ whose classifying morphism

$$\lambda(y : B) \rightarrow \exists(x : A). m x = y : B \rightarrow \Omega$$

factors through $\mathbf{Cof} \multimap \Omega$. We call such monomorphisms *cofibrations*. Kan-like filling properties have to do with when a morphism $A \rightarrow X$ can be extended along a cofibration $m : A \multimap B$. Instead, working in the internal language of \mathcal{E} , we will consider when partial elements whose domains of definition are in \mathbf{Cof} can be extended to totally defined elements. Recall that in intuitionistic logic, partial elements of a type A are often represented by sub-singletons, that is, by functions $s : A \rightarrow \Omega$ satisfying

$$\forall(x x' : A). s x \wedge s x' \Rightarrow x = x'$$

However, it will be more convenient to work with an extensionally equivalent representation as dependent pairs $\varphi : \Omega$ and $f : [\varphi] \rightarrow A$, as in the next definition. The proposition φ is the extent of the partial element; in terms of sub-singletons it is equal to $\exists(x : A). s x$.

Definition 5.1.2 (Cofibrant partial elements, $\square A$). We call elements of type \mathbf{Cof} *cofibrant propositions*. Given a type $A : \mathcal{U}$, we define the type of *cofibrant partial elements* of A to be

$$\square A \triangleq (\varphi : \mathbf{Cof}) \times ([\varphi] \rightarrow A) \tag{5.2}$$

An *extension* of such a partial element $(\varphi, f) : \square A$ is an element $a : A$ together with a proof of the following relation:

$$(\varphi, f) \nearrow a \triangleq \forall(u : [\varphi]). f u = a \tag{5.3}$$

Note that by taking $i = 0$ in axiom \mathbf{ax}_5 we have $\mathbf{cof}(0 = 0)$ (that is, $\mathbf{cof} \top$) and

Cofibrant propositions $\mathbf{Cof} = \{\varphi : \Omega \mid \mathbf{cof} \varphi\}$ (where $\mathbf{cof} : \Omega \rightarrow \Omega$)
include end-point-equality

$$\mathbf{ax}_5 : [\forall(i : \mathbf{I}). \mathbf{cof}(i = 0) \wedge \mathbf{cof}(i = 1)]$$

and are closed under binary disjunction

$$\mathbf{ax}_6 : [\forall(\varphi \psi : \Omega). \mathbf{cof} \varphi \Rightarrow \mathbf{cof} \psi \Rightarrow \mathbf{cof}(\varphi \vee \psi)]$$

dependent conjunction

$$\mathbf{ax}_7 : [\forall(\varphi \psi : \Omega). \mathbf{cof} \varphi \Rightarrow (\varphi \Rightarrow \mathbf{cof} \psi) \Rightarrow \mathbf{cof}(\varphi \wedge \psi)]$$

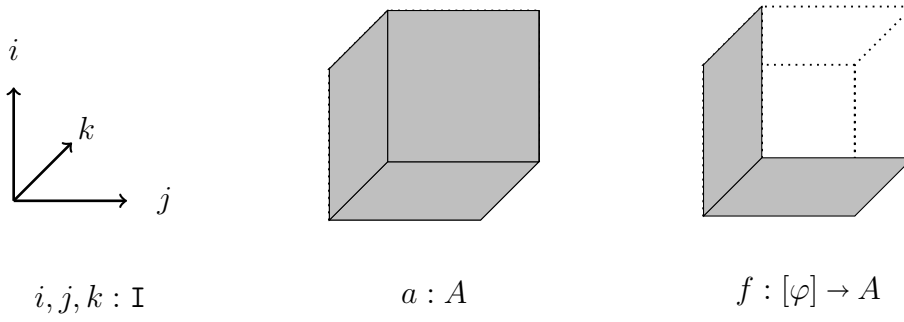
and universal quantification over \mathbf{I}

$$\mathbf{ax}_8 : [\forall(\varphi : \mathbf{I} \rightarrow \Omega). (\forall(i : \mathbf{I}). \mathbf{cof}(\varphi i)) \Rightarrow \mathbf{cof}(\forall(i : \mathbf{I}). \varphi i)].$$

Figure 5.2: Axioms for cofibrant propositions

$\mathbf{cof}(0 = 1)$; and combining the latter with axiom \mathbf{ax}_2 we deduce also that $\mathbf{cof} \perp$ holds. Therefore \top and \perp are both cofibrant propositions, corresponding to the first two cases in the grammar which generates elements of the face lattice (Section 2.2.3). So $A \multimap A$ and $\emptyset \multimap A$ are always cofibrations, where \emptyset is the initial object. Since $\mathbf{cof} \top$ holds, for every $a : A$ there is a *total* cofibrant partial element $(\top, \lambda_- \rightarrow a) : \square A$ with a the unique element that extends $(\top, \lambda_- \rightarrow a)$. Since $\mathbf{cof} \perp$ holds, every object A has an empty cofibrant partial element given by $(\perp, \mathbf{elim}_\emptyset) : \square A$ such that every $a : A$ is an extension of $(\perp, \mathbf{elim}_\emptyset)$. (For any $B : \mathcal{U}$, $\mathbf{elim}_\emptyset : [\perp] \rightarrow B$ denotes the unique function given by initiality of $[\perp]$.)

Example 5.1.3. *It is helpful to think of variables of type \mathbf{I} as names of dimensions in space, so that working in a context $i_1, \dots, i_n : \mathbf{I}$ corresponds to working in n dimensions. Assume that we are working in a context with $i, j, k : \mathbf{I}$; this therefore corresponds to working in three dimensions. We think of an element $i, j, k : \mathbf{I} \vdash a : A$ as a cube in the space A , as shown below. Let $\varphi \triangleq (i = 0) \vee (j = 0) \vee (j = 1 \wedge k = 1)$. From \mathbf{ax}_5 – \mathbf{ax}_7 we have $i, j, k : \mathbf{I} \vdash \varphi : \mathbf{Cof}$. We think of φ as specifying certain faces and edges of a cube, in this case the bottom face ($i = 0$), the left face ($j = 0$) and the front-right edge ($j = 1 \wedge k = 1$), as in the right-hand picture below. Then a cofibrant partial element $f : [\varphi] \rightarrow A$ can be thought of as a partial cube, only defined on the region specified by φ .*



Definition 5.1.4 (Join of compatible partial elements). Say that two partial elements $f : [\varphi] \rightarrow A$ and $g : [\psi] \rightarrow A$ are *compatible* if they agree wherever they are both defined:

$$(\varphi, f) \sim (\psi, g) \triangleq \forall (u : [\varphi])(v : [\psi]). f u = g v \quad (5.4)$$

In that case we can form their join $f \cup g : [\varphi \vee \psi] \rightarrow A$, such that

$$\forall (u : [\varphi]). (f \cup g) u = f u \quad \forall (v : [\psi]). (f \cup g) v = g v$$

To see why, consider the following pushout square in the topos:

$$\begin{array}{ccc}
 [\varphi \wedge \psi] & \xrightarrow{\quad} & [\psi] \\
 \downarrow & & \downarrow \\
 [\varphi] & \xrightarrow{\quad} & [\varphi \vee \psi] \\
 & \searrow f & \downarrow g \\
 & & A
 \end{array}$$

(A dashed arrow labeled $f \cup g$ points from $[\varphi \vee \psi]$ to A)

The outer square commutes because $(\varphi, f) \sim (\psi, g)$ holds and then $f \cup g$ is the unique induced morphism out of the pushout. Note that axiom \mathbf{ax}_6 in Figure 5.2 implies that the collection of cofibrant partial elements is closed under taking binary joins of compatible partial elements.

The following lemma gives an alternative characterization of axioms \mathbf{ax}_7 and \mathbf{ax}_8 . Since we noted above that $\mathbf{cof} \top$ holds, part (i) of the lemma tells us that cofibrations form a *dominance* in the sense of synthetic domain theory [56]; we only use this property of \mathbf{Cof} in order to construct definitional identity types from propositional identity types (see Section 5.3.4).

Lemma 5.1.5. (i) *Axiom \mathbf{ax}_7 is equivalent to requiring the class of cofibrations to be closed under composition.*

(ii) *Axiom \mathbf{ax}_8 is equivalent to requiring the class of cofibrations to be closed under exponentiation by \mathbf{I} .*

Proof. For part (i), first suppose that \mathbf{ax}_7 holds and that $f : A \rightarrow B$ and $g : B \rightarrow C$ are cofibrations. So both $\forall(b : B). \mathbf{cof}(\exists(a : A). f a = b)$ and $\forall(c : C). \mathbf{cof}(\exists(b : B). g b = c)$ hold and we wish to prove $\forall(c : C). \mathbf{cof}(\exists(a : A). g(f a) = c)$. Note that for $b : B$ and $c : C$

$$\begin{aligned} g b = c &\Rightarrow (\exists(a : A). g(f a) = c) = (\exists(a : A). f a = b) \quad \text{since } g \text{ is a monomorphism} \\ &\Rightarrow \mathbf{cof}(\exists(a : A). g(f a) = c) = \mathbf{cof}(\exists(a : A). f a = b) = \top \end{aligned}$$

So for $\varphi \triangleq \exists(b : B). g b = c$ and $\psi \triangleq \exists(a : A). g(f a) = c$, we have $\mathbf{cof} \varphi$ and $\varphi \Rightarrow \mathbf{cof} \psi$. Therefore by \mathbf{ax}_7 we get $\mathbf{cof}(\varphi \wedge \psi)$, which is equal to $\mathbf{cof}(\psi)$ since $\psi \Rightarrow \varphi$. So we do indeed have $\forall(c : C). \mathbf{cof}(\exists(a : A). g(f a) = c)$.

Conversely, suppose cofibrations are closed under composition and that $\varphi, \psi : \Omega$ satisfy $\mathbf{cof} \varphi$ and $\varphi \Rightarrow \mathbf{cof} \psi$. That $\mathbf{cof} \varphi$ holds is equivalent to the monomorphism $[\varphi] \rightarrow 1$ being a cofibration; and since

$$\varphi \Rightarrow (\psi = \varphi \wedge \psi) \Rightarrow (\mathbf{cof} \psi = \mathbf{cof}(\varphi \wedge \psi))$$

from $\varphi \Rightarrow \mathbf{cof} \psi$ we get $\varphi \Rightarrow \mathbf{cof}(\varphi \wedge \psi)$ and hence the monomorphism $[\varphi \wedge \psi] \rightarrow [\varphi]$ is a cofibration. Composing these monomorphisms, we have that $[\varphi \wedge \psi] \rightarrow 1$ is a cofibration, that is, $\mathbf{cof}(\varphi \wedge \psi)$ holds.

For part (ii), first suppose that \mathbf{ax}_8 holds and that $f : A \rightarrow B$ is a cofibration. We have to show that $\mathbf{I} \rightarrow f : (\mathbf{I} \rightarrow A) \rightarrow (\mathbf{I} \rightarrow B)$ is also a cofibration. Given $\beta : \mathbf{I} \rightarrow B$ we have

$$\begin{aligned} &(\forall i : \mathbf{I})(\exists a : A). f a = \beta i \\ &\Rightarrow (\forall i : \mathbf{I})(\exists! a : A). f a = \beta i \quad \text{(since } f \text{ is a monomorphism)} \\ &\Rightarrow (\exists \alpha : \mathbf{I} \rightarrow A)(\forall i : \mathbf{I}). f(\alpha i) = \beta i \quad \text{(by unique choice in the topos)} \\ &\Rightarrow (\forall i : \mathbf{I})(\exists a : A). f a = \beta i \end{aligned}$$

so that $(\forall i : \mathbf{I})(\exists a : A). f a = \beta i$ is equal to $(\exists \alpha : \mathbf{I} \rightarrow A)(\forall i : \mathbf{I}). f(\alpha i) = \beta i$; and the latter is equal to $(\exists \alpha : \mathbf{I} \rightarrow A). (\mathbf{I} \rightarrow f) \alpha = \beta$ by function extensionality in the topos. Since f is a cofibration, for each $i : \mathbf{I}$ we have $\mathbf{cof}(\exists(a : A). f a = \beta i)$. Hence by axiom \mathbf{ax}_8 we also have $\mathbf{cof}((\forall i : \mathbf{I})(\exists a : A). f a = \beta i)$, that is, $\mathbf{cof}(\exists \alpha : \mathbf{I} \rightarrow A). (\mathbf{I} \rightarrow f) \alpha = \beta$, as required for $\mathbf{I} \rightarrow f$ to be a cofibration.

Conversely, suppose cofibrations are closed under $\mathbf{I} \rightarrow (-)$ and that $\varphi : \mathbf{I} \rightarrow \Omega$ satisfies $(\forall i : \mathbf{I}). \mathbf{cof}(\varphi i)$. The latter implies that $\{i : \mathbf{I} \mid \varphi i\} \rightarrow \mathbf{I}$ is a cofibration. Hence so is the monomorphism $(\mathbf{I} \rightarrow \{i : \mathbf{I} \mid \varphi i\}) \rightarrow (\mathbf{I} \rightarrow \mathbf{I})$. Since $id : \mathbf{I} \rightarrow \mathbf{I}$ is in the image of this monomorphism iff $(\forall i : \mathbf{I}). \varphi i$ holds, we have $\mathbf{cof}((\forall i : \mathbf{I}). \varphi i)$, as required for axiom \mathbf{ax}_8 . \square

Strictness axiom: any cofibrant-partial type A that is isomorphic to a total type B everywhere that A is defined, can be extended to a total type B' that is isomorphic to B :

$$\mathbf{ax}_9 : \{\varphi : \mathbf{Cof}\}(A : [\varphi] \rightarrow \mathcal{U})(B : \mathcal{U})(s : (u : [\varphi]) \rightarrow (A u \cong B)) \rightarrow (B' : \mathcal{U}) \times \{s' : B' \cong B \mid \forall (u : [\varphi]). A u = B' \wedge s u = s'\}$$

Figure 5.3: The strictness axiom

5.1.3 The strictness axiom

Our final axiom is distinct from the others in that it is less about axiomatising some essential aspect of cubical type theory and more about addressing a weakness of the internal language approach. This final axiom, \mathbf{ax}_9 , states that any cofibrant-partial type A that is isomorphic to a total type B everywhere that A is defined, can be extended to a total type B' that is isomorphic to B . The exact definition of the axiom is given in Figure 5.3, and the definition of isomorphism is as follows:

Definition 5.1.6 (Isomorphisms). Given objects $A, B : \mathcal{U}$, an isomorphism between A and B is a function $f : A \rightarrow B$ that has a two-sided inverse. Let $A \cong B$ be the type of isomorphisms between A and B , defined by

$$A \cong B \triangleq \{f : A \rightarrow B \mid (\exists g : B \rightarrow A) (g \circ f = id) \wedge (f \circ g = id)\}$$

We say that A and B are isomorphic if there exists an isomorphism $f : A \cong B$. We say that two families $A, B : \Gamma \rightarrow \mathcal{U}$ are isomorphic if each of their fibres are, and we abusively write

$$A \cong B \triangleq (x : \Gamma) \rightarrow A x \cong B x$$

Isomorphisms have inverses up to the extensional equality of the internal type theory, in contrast to the notion of equivalences which we introduce later, which will only have inverses up to path equality. In addition these inverses are unique, and hence, using unique choice, we can always construct the inverse to an isomorphism. Therefore, given $f : A \cong B$, we will write $f^{-1} : B \cong A$ for this inverse of f .

To gain some intuition for what this axiom is doing consider the following example. We have a family of types over the interval $B : \mathbf{I} \rightarrow \mathcal{U}$ and a single type $A : \mathcal{U}$ such that $A \cong B 0$. We would like to be able to, in a sense, “overwrite” the value of B at 0 , replacing it with A . This should then give us a family $B' : \mathbf{I} \rightarrow \mathcal{U}$ which is isomorphic to the original family B , but such that $A = B' 0$. Given axiom \mathbf{ax}_9 then we can define $B' i : \mathcal{U}$ to be the first projection of $\mathbf{ax}_9 \{i = 0\} (\lambda_-. \rightarrow A) (B i) (\lambda_-. \rightarrow s)$ where $s : A \cong B 0$ is the isomorphism witnessing that A is isomorphic to B at 0 . The second projection gives

us the new isomorphism witnessing that B' is isomorphic to B , and in particular this new isomorphism is simply the old one when $i = 0$.

A priori, without axiom \mathbf{ax}_9 , there is no way to perform such a construction in the internal type theory. Indeed, such a construction seems quite strange from a categorical point of view, where we usually only work “up to isomorphism”. However, this axiom will be required later in order to validate certain definitional equalities which must hold in the type theory. In that sense the need for axiom \mathbf{ax}_9 is reminiscent of the issues that arise when interpreting substitution in type theory as pullback in a categorical model.

Axiom \mathbf{ax}_9 will be used to regain the strict form of glueing used by Cohen et al. [18]. Its validity in presheaf models depends on a construction in the external metatheory that cannot be replicated internally; see Theorem 5.6.3 for details.

5.2 Path types

Given $A : \mathcal{U}$, we call elements of type $\mathbb{I} \rightarrow A$ *paths* in A . The *path type* associated with A is $_ \sim _ : A \rightarrow A \rightarrow \mathcal{U}$ where

$$a_0 \sim a_1 \triangleq \{p : \mathbb{I} \rightarrow A \mid p\ 0 = a_0 \wedge p\ 1 = a_1\} \quad (5.5)$$

Can these types be used to model the rules for Martin-Löf identity types? We can certainly interpret the identity introduction rule (reflexivity), since degenerate paths given by constant functions

$$\mathbf{k}\ a\ i \triangleq a \quad (5.6)$$

satisfy $\mathbf{k} : \{A : \mathcal{U}\}(a : A) \rightarrow a \sim a$. However, we need further assumptions to interpret the identity elimination rule, otherwise known as path induction, described in Section 2.1.2. Coquand has given an alternative (propositionally equivalent) formulation of identity elimination in terms of substitution functions $a_0 \sim a_1 \rightarrow P\ a_0 \rightarrow P\ a_1$ and contractibility of singleton types $(a_1 : A) \times (a_0 \sim a_1)$; see [11, Figure 2]. The connection algebra structure gives the latter, since using \mathbf{ax}_3 and \mathbf{ax}_4 we have

$$\begin{aligned} \mathbf{ctr} : \{A : \mathcal{U}\}\{a_0\ a_1 : A\}(p : a_0 \sim a_1) &\rightarrow (a_0, \mathbf{k}\ a_0) \sim (a_1, p) \\ \mathbf{ctr}\ p\ i &\triangleq (p\ i, \lambda j \rightarrow p(i \sqcap j)) \end{aligned} \quad (5.7)$$

However, to get suitably behaved substitution functions we have to consider families of types endowed with some extra structure; and that structure has to lift through the type-forming operations (products, functions, identity types, etc). This is what the definitions in the next section achieve.

The interval $\mathbf{I} : \mathcal{U}$ is connected

$$\mathbf{ax}_1 : [\forall(\varphi : \mathbf{I} \rightarrow \Omega). (\forall(i : \mathbf{I}). \varphi i \vee \neg\varphi i) \Rightarrow (\forall(i : \mathbf{I}). \varphi i) \vee (\forall(i : \mathbf{I}). \neg\varphi i)]$$

has distinct end-points $0, 1 : \mathbf{I}$

$$\mathbf{ax}_2 : [\neg(0 = 1)]$$

and a connection algebra structure $-\sqcap-, -\sqcup- : \mathbf{I} \rightarrow \mathbf{I} \rightarrow \mathbf{I}$

$$\mathbf{ax}_3 : [\forall(i : \mathbf{I}). 0 \sqcap x = 0 = x \sqcap 0 \wedge 1 \sqcap x = x = x \sqcap 1]$$

$$\mathbf{ax}_4 : [\forall(i : \mathbf{I}). 0 \sqcup x = x = x \sqcup 0 \wedge 1 \sqcup x = 1 = x \sqcup 1].$$

Cofibrant propositions $\mathbf{Cof} = \{\varphi : \Omega \mid \mathbf{cof} \varphi\}$ (where $\mathbf{cof} : \Omega \rightarrow \Omega$)
include end-point-equality

$$\mathbf{ax}_5 : [\forall(i : \mathbf{I}). \mathbf{cof}(i = 0) \wedge \mathbf{cof}(i = 1)]$$

and are closed under binary disjunction

$$\mathbf{ax}_6 : [\forall(\varphi \psi : \Omega). \mathbf{cof} \varphi \Rightarrow \mathbf{cof} \psi \Rightarrow \mathbf{cof}(\varphi \vee \psi)]$$

dependent conjunction

$$\mathbf{ax}_7 : [\forall(\varphi \psi : \Omega). \mathbf{cof} \varphi \Rightarrow (\varphi \Rightarrow \mathbf{cof} \psi) \Rightarrow \mathbf{cof}(\varphi \wedge \psi)]$$

and universal quantification over \mathbf{I}

$$\mathbf{ax}_8 : [\forall(\varphi : \mathbf{I} \rightarrow \Omega). (\forall(i : \mathbf{I}). \mathbf{cof}(\varphi i)) \Rightarrow \mathbf{cof}(\forall(i : \mathbf{I}). \varphi i)].$$

Strictness axiom: any cofibrant-partial type A
that is isomorphic to a total type B everywhere that A is defined,
can be extended to a total type B' that is isomorphic to B :

$$\mathbf{ax}_9 : \{\varphi : \mathbf{Cof}\}(A : [\varphi] \rightarrow \mathcal{U})(B : \mathcal{U})(s : (u : [\varphi]) \rightarrow (A u \cong B)) \rightarrow (B' : \mathcal{U}) \times \{s' : B' \cong B \mid \forall(u : [\varphi]). A u = B' \wedge s u = s'\}$$

Figure 5.4: All the axioms

5.3 Cohen-Coquand-Huber-Mörtberg (CCHM) fibrations

In this section we show how to generalise the notion of fibration introduced in Definition 3.3.4 from the particular presheaf model considered there to any topos with an interval object as in the previous sections. To do so we use the notion of *cofibrant proposition* from Figure 5.4 to internalise the composition and filling operations described in [18]. We show that this notion is closed under forming Σ -, Π - and **Path**-types, as well as basic datatypes; finally we show how to define identity types with a definitional computation rule, **Id**-types, and we show that fibrations are also closed under forming **Id**-types.

5.3.1 Composition and filling structures

Given an interval-indexed family of types $A : \mathbb{I} \rightarrow \mathcal{U}$, we think of elements of the dependent function type $\Pi_{\mathbb{I}} A \triangleq (i : \mathbb{I}) \rightarrow A i$ as dependently typed paths. We call elements of type $\square(\Pi_{\mathbb{I}} A)$ *cofibrant-partial paths*. Given $(\varphi, f) : \square(\Pi_{\mathbb{I}} A)$, we can evaluate it at a point $i : \mathbb{I}$ of the interval to get a cofibrant partial element $(\varphi, f) @ i : \square(A i)$:

$$(\varphi, f) @ i \triangleq (\varphi, \lambda(u : [\varphi]) \rightarrow f u i) \quad (5.8)$$

An operation for *filling from 0* in $A : \mathbb{I} \rightarrow \mathcal{U}$ takes any $(\varphi, f) : \square(\Pi_{\mathbb{I}} A)$ and any $a_0 : A 0$ with $(\varphi, f) @ 0 \nearrow a_0$ and extends (φ, f) to a dependently typed path $g : \Pi_{\mathbb{I}} A$ with $g 0 = a_0$. This is a form of uniform *Homotopy Extension and Lifting Property* (HELP) [49, Chapter 10, Section 3] stated internally in terms of cofibrant propositions rather than externally in terms of cofibrations. A feature of our internal approach compared with Cohen et al. is that their *uniformity* condition on composition/filling operations (Definition 3.3.4), which allows one to avoid the non-constructive aspects of the classical notion of Kan filling [10], becomes automatic when the operations are formulated in terms of the internal collection **Cof** of cofibrant propositions.

Since we are not assuming any structure on the interval for reversing paths (see Remark 5.1.1), we also need to consider the symmetric notion of filling from 1. Let

$$\{0, 1\} \triangleq \{i : \mathbb{I} \mid i = 0 \vee i = 1\} \quad (5.9)$$

Note that because of axiom **ax**₂, this is isomorphic to the object of Booleans, $1 + 1$ and hence there is a function

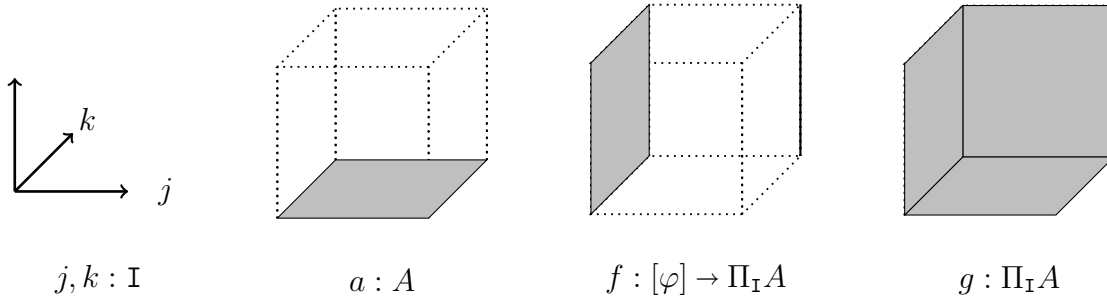
$$\bar{} : \{0, 1\} \rightarrow \{0, 1\} \quad (5.10)$$

satisfying $\bar{0} = 1$ and $\bar{1} = 0$. In what follows, instead of using path reversal we parameterise definitions with $e : \{0, 1\}$ and use (5.10) to interchange 0 and 1.

Definition 5.3.1 (Filling structures). Given $e : \{0, 1\}$, the type $\text{Fill } e A : \mathcal{U}$ of *filling structures* for an \mathbb{I} -indexed families of types $A : \mathbb{I} \rightarrow \mathcal{U}$, is defined by:

$$\begin{aligned} \text{Fill } e A \triangleq & (\varphi : \text{Cof})(f : [\varphi] \rightarrow \Pi_{\mathbb{I}} A)(a : \{a' : A e \mid (\varphi, f) @ e \nearrow a'\}) \rightarrow \\ & \{g : \Pi_{\mathbb{I}} A \mid (\varphi, f) \nearrow g \wedge g e = a\} \end{aligned} \quad (5.11)$$

Example 5.3.2. For some intuition as to why such an operation is referred to as *filling*, consider the following example. For simplicity, assume that $A : \mathbb{I} \rightarrow \mathcal{U}$ is a constant family $A \triangleq \lambda(- : \mathbb{I}) \rightarrow A'$. Recall that we think of variables of type \mathbb{I} as dimensions in space; so that, given an element $a : A$ in an ambient context $j, k : \mathbb{I}$, we think of a as a square in the space A . We are interested in extending this two dimensional square to a three dimensional cube as indicated below.



However, let us imagine that we already know how to extend a on certain faces and edges of the cube, for example, on the faces/edges specified by $\varphi \triangleq (j = 0) \vee (j = 1 \wedge k = 1)$. This means that we have a cofibrant partial path $f : [\varphi] \rightarrow \Pi_{\mathbb{I}} A$ which agrees with a where they are both defined, that is $(\varphi, f) @ 0 \nearrow a$. Note that f is a partial path rather than partial element because, on the faces/edges where it is defined, it must be defined at all points along the new dimension by which we are extending a , i.e. φ cannot depend on this new dimension. A filling for this data is a cube $g : \Pi_{\mathbb{I}} A$ which agrees with the faces/edges that we started with. That is, it extends f and agrees with a at the base of the cube: $(\varphi, f) \nearrow g$ and $g 0 = a$.

A notable feature of [18] compared with preceding work [11] is that such filling structure can be constructed from a simpler *composition* structure that just produces an extension at one end of a cofibrant-partial path from an extension at the other end. We will deduce this using axioms ax_3 – ax_6 from the following, which is the main notion of this chapter.

Definition 5.3.3 (CCHM fibrations). A *CCHM fibration* (A, α) over a type $\Gamma : \mathcal{U}$ is a family $A : \Gamma \rightarrow \mathcal{U}$ equipped with a fibration structure $\alpha : \text{isFib } A$, where $\text{isFib} : \{\Gamma : \mathcal{U}\}(A : \Gamma \rightarrow \mathcal{U}) \rightarrow \mathcal{U}$ is defined by

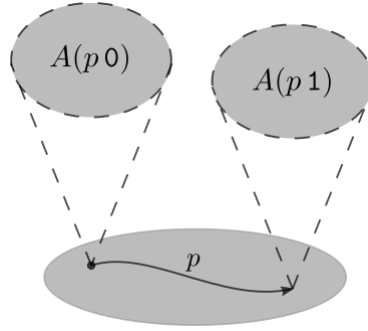
$$\text{isFib } \{\Gamma\} A \triangleq (e : \{0, 1\})(p : \mathbb{I} \rightarrow \Gamma) \rightarrow \text{Comp } e (A \circ p) \quad (5.12)$$

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

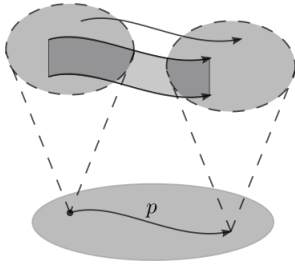
families:

$$\begin{aligned} \text{Comp } e A \triangleq (\varphi : \text{Cof})(f : [\varphi] \rightarrow \Pi_{\mathbf{I}} A) \rightarrow \\ \{a_0 : A e \mid (\varphi, f) @ e \nearrow a_0\} \rightarrow \{a_1 : A \bar{e} \mid (\varphi, f) @ \bar{e} \nearrow a_1\} \end{aligned} \quad (5.13)$$

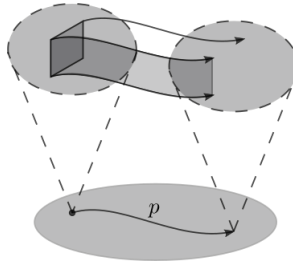
Example 5.3.4. In Example 5.3.2 we considered the case where the family A was constant. We now consider the non-constant case, providing some intuition for the role of Γ and $p : \mathbf{I} \rightarrow \Gamma$ in the definition of CCHM fibrations (5.12). Following the intuition that fibrations should correspond to topological spaces parameterised by another topological space, we think of Γ as a space, called the base of the fibration, and $p : \mathbf{I} \rightarrow \Gamma$ as a path in this space. We have a space, Ax , lying above x for every $x : \Gamma$, called the fibre at x . In particular we have spaces $A(p0)$ and $A(p1)$ lying over the start and end of the path p as shown below:



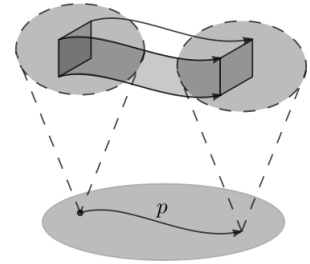
As in Example 5.3.2, consider working in a context with two dimensions $j, k : \mathbf{I}$, where $\varphi \triangleq (j = 0) \vee (j = 1 \wedge k = 1)$. Further, take $e = 0$. This means that $f : [\varphi] \rightarrow \Pi_{\mathbf{I}}(A \circ p)$ defines one face and one edge of a cube as in Example 5.3.2, except that these no longer exist inside a single fibre, but rather they exist in the total space of A , lying over the path p , as shown in the leftmost diagram below. Then a_0 defines a square in the space $A(p0)$, as shown in the central diagram, where the condition $(\varphi, f) @ e \nearrow a_0$ ensures that f properly aligns with the relevant edge and corner of a_0 . The result of the composition operation, $\alpha 0 p \varphi f a_0 : A(p1)$, therefore defines a square in the fibre $A(p1)$, where the condition $(\varphi, f) @ \bar{e} \nearrow a_1$ ensures that this square is properly aligned with f , as shown in the rightmost diagram below.



$$f : [\varphi] \rightarrow \Pi_{\mathbf{I}}(A \circ p)$$



$$a_0 : A(p0)$$



$$\alpha 0 p \varphi f a_0 : A(p1)$$

Definition 5.3.5 (The CwF of CCHM fibrations). Let $\mathbf{Fib} \Gamma$ be the type of CCHM fibrations over an object Γ , defined by

$$\mathbf{Fib} \Gamma \triangleq (A : \Gamma \rightarrow \mathcal{U}) \times \mathbf{isFib} A \quad (5.14)$$

CCHM fibrations are closed under re-indexing: given $\gamma : \Delta \rightarrow \Gamma$ and $A : \Gamma \rightarrow \mathcal{U}$, we get a function $_{-}[\gamma] : \mathbf{isFib} A \rightarrow \mathbf{isFib}(A \circ \gamma)$ defined by $\alpha[\gamma] e p \triangleq \alpha e (\gamma \circ p)$. Therefore we get a function $_{-}[_] : (\Delta \rightarrow \Gamma) \rightarrow \mathbf{Fib} \Gamma \rightarrow \mathbf{Fib} \Delta$ given by

$$(A, \alpha)[\gamma] \triangleq (A \circ \gamma, \alpha[\gamma]) \quad (5.15)$$

which is functorial: $(A, \alpha)[id] = (A, \alpha)$ and $(A, \alpha)[g \circ f] = (A, \alpha)[g][f]$. It follows that \mathbf{Fib} has the structure of a Category with Families by taking families to be CCHM fibrations $(A, \alpha) : \mathbf{Fib} \Gamma$ over each $\Gamma : \mathcal{U}$ and elements of such a family to be dependent functions in $(x : \Gamma) \rightarrow Ax$.

Remark 5.3.6 (Fibrant objects). We say $A : \mathcal{U}$ is a fibrant object if we have a fibration structure for the constant family $\lambda(- : 1) \rightarrow A$ over the terminal object 1 . Note that if $(A, \alpha) : \mathbf{Fib} \Gamma$ is a fibration, then for each $x : \Gamma$ the type $Ax : \mathcal{U}$ is fibrant, with the fibration structure given by reindexing α by the map $\lambda(- : 1) \rightarrow x : 1 \rightarrow \Gamma$. However the converse is not true: having a family of fibration structures, that is, an element of $(x : \Gamma) \rightarrow \mathbf{isFib}(\lambda(- : 1) \rightarrow Ax)$, is weaker than having a fibration structure for $A : \Gamma \rightarrow \mathcal{U}$. To see why, consider the family, $P : \mathbf{I} \rightarrow \mathcal{U}$ defined by

$$P i \triangleq [0 = i] \quad (5.16)$$

For each $i : \mathbf{I}$ the fibre $P i : \mathcal{U}$ is a fibrant object, with a fibration structure, $\rho_i : \mathbf{isFib}(\lambda(- : 1) \rightarrow P i)$, given by $\rho_i e p \phi f x \triangleq x$. However, it is not possible to construct a $\rho : \mathbf{isFib} P$. For if it were, then we could define $\rho 0 id \perp \mathbf{elim}_0 * : [0 = 1]$; combined with \mathbf{ax}_2 , this would lead to contradiction. This example is explored in more detail later, in Section 7.1.

If $\alpha : \mathbf{Fill} e A$, then $\lambda \varphi f a \rightarrow \alpha \varphi f a \bar{e} : \mathbf{Comp} e A$ and so every filling structure gives rise to a composition structure. Conversely, the composition structure of a CCHM fibration gives rise to filling structure:

Lemma 5.3.7 (Filling structure from composition structure). Given $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$, $e : \{0, 1\}$, $\alpha : \mathbf{isFib} A$ and $p : \mathbf{I} \rightarrow \Gamma$, there is a filling structure $\mathbf{fill} e \alpha p : \mathbf{Fill} e (A \circ p)$ that agrees with α at \bar{e} , that is:

$$\begin{aligned} \forall (\varphi : \mathbf{Cof})(f : [\varphi] \rightarrow \Pi_{\mathbf{I}} A)(a : A(p e)). \\ (\varphi, f) @ e \nearrow a \Rightarrow \mathbf{fill} e \alpha p \varphi f a \bar{e} = \alpha e p \varphi f a \quad (5.17) \end{aligned}$$

Furthermore, \mathbf{fill} is stable under re-indexing in the sense that for all $\gamma : \Delta \rightarrow \Gamma$ and $p : \mathbb{I} \rightarrow \Delta$

$$\mathbf{fill} e \alpha (\gamma \circ p) = \mathbf{fill} e (\alpha[\gamma]) p \quad (5.18)$$

Proof. The construction of filling from composition follows [18, Section 4.4], but just using the connection algebra structure on \mathbb{I} (axioms \mathbf{ax}_3 and \mathbf{ax}_4), rather than a De Morgan algebra structure. Suppose $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$, $e : \{0, 1\}$, $\alpha : \mathbf{isFib} A$, $p : \mathbb{I} \rightarrow \Gamma$, $\varphi : \mathbf{Cof}$, $f : [\varphi] \rightarrow \Pi_{\mathbb{I}}(A \circ p)$, $a : A(pe)$ with $(\varphi, f) @ e \nearrow a$, and $i : \mathbb{I}$. Then using Definition 5.1.4 we can define

$$\mathbf{fill} e \alpha p \varphi f a i \triangleq \alpha e (p' i) (\varphi \vee i = e) (f' i \cup g i) a \quad (5.19)$$

where

$$\begin{aligned} p' &: \mathbb{I} \rightarrow \mathbb{I} \rightarrow \Gamma \\ p' i j &\triangleq p(i \sqcap_e j) \\ f' &: (i : \mathbb{I}) \rightarrow [\varphi] \rightarrow \Pi_{\mathbb{I}}(A \circ (p' i)) \\ f' i u j &\triangleq f u (i \sqcap_e j) \\ g &: (i : \mathbb{I}) \rightarrow \{g' : [i = e] \rightarrow \Pi_{\mathbb{I}}(A \circ (p' i)) \mid (\varphi, f' i) \smile (i = e, g')\} \\ g i v j &\triangleq a \end{aligned}$$

and where \sqcap_e is given by $\sqcap_0 \triangleq \sqcap$ and $\sqcap_1 \triangleq \sqcup$. Finally, property (5.18) is immediate from definitions (5.15) and (5.19). \square

Compared with [11], the fact that filling can be defined from composition considerably simplifies the process of lifting fibration structure through the usual type-forming constructs, as the following two theorems demonstrate. Their proofs are internalisations of those in [18, Section 4.5], except that we avoid the use Cohen et al. make of De Morgan involution.

5.3.2 Properties of fibrations

We now prove some properties of fibrations that will be useful later. First, we show that the class of fibrations is closed under isomorphism. We will say that two fibrations (A, α) and $(B, \beta) : \mathbf{Fib} \Gamma$ are isomorphic if the underlying families A and B are, and we abusively write

$$(A, \alpha) \cong (B, \beta) \triangleq A \cong B$$

where $A \cong B$ is as in Definition 5.1.6.

Lemma 5.3.8. *Given a family $A : \Gamma \rightarrow \mathcal{U}$ and a fibration $(B, \beta) : \mathbf{Fib} \Gamma$, such that $A \cong B$, then we can construct α such that $(A, \alpha) : \mathbf{Fib} \Gamma$.*

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

$$\alpha e p \varphi q a_0 \triangleq f^{-1} (p \bar{e}) (\beta e p \varphi (\lambda u i \rightarrow f (p i) (q u i)) (f (p e) a_0))$$

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

$$\begin{aligned} \alpha e p \varphi q a_0 &= f^{-1} (p \bar{e}) (\beta e p \varphi (\lambda u i \rightarrow f (p i) (q u i)) (f (p e) a_0)) \\ &= f^{-1} (p \bar{e}) (f (p \bar{e}) (q u \bar{e})) \\ &= q u \bar{e} \end{aligned}$$

Therefore $(\varphi, q) @ \bar{e} \nearrow \alpha e p \varphi q a_0$, and hence $(A, \alpha) : \mathbf{Fib} \Gamma$. □

Note that this proof only uses the fact that $f^{-1} x \circ f x = id$ and so in fact the lemma holds more generally in the case where $\langle f, f^{-1} \rangle$ is just a section-retraction pair rather than a full isomorphism. Although in this thesis we will only use it in the context of isomorphisms.

Next, we show that it is possible to adapt or *realign* composition structures on a family. First, we introduce the notion of *cofibrant-partial type-families*:

Definition 5.3.9 (Cofibrant-partial families). Given an object $\Gamma : \mathcal{U}$ and a cofibrant property $\Phi : \Gamma \rightarrow \mathbf{Cof}$ define the *restriction of Γ by Φ* to be

$$\Gamma|\Phi \triangleq (x : \Gamma) \times [\Phi x] \tag{5.20}$$

Thus $\Gamma|\Phi : \mathcal{U}$ and there is a monomorphism $\iota : \Gamma|\Phi \rightarrow \Gamma$ given by first projection. (Note that $\Gamma|\Phi$ is isomorphic to the comprehension subtype $\{x : \Gamma \mid \Phi x\}$, but we use the above representation to make proofs of Φx more explicit in various constructions.) Then given an object $\Gamma : \mathcal{U}$ and a cofibrant property $\Phi : \Gamma \rightarrow \mathbf{Cof}$, a *cofibrant-partial type-family* over Γ is a family A of types over the restriction $\Gamma|\Phi$, that is $A : (\Gamma|\Phi) \rightarrow \mathcal{U}$.

Lemma 5.3.10 (Realignment lemma). *Given $\Gamma : \mathcal{U}$ and $\Phi : \Gamma \rightarrow \mathbf{Cof}$, let $\iota : \Gamma|\Phi \rightarrow \Gamma$ be the first projection. For any $A : \Gamma \rightarrow \mathcal{U}$, $\beta : \mathbf{isFib}(A \circ \iota)$ and $\alpha : \mathbf{isFib} A$, there exists a composition structure $\mathbf{realign}(\Phi, \beta, \alpha) : \mathbf{isFib} A$ such that $\beta = \mathbf{realign}(\Phi, \beta, \alpha)[\iota]$.*

Proof. Given $\Gamma, \Phi, A, \beta, \alpha$ as above, using axiom \mathbf{ax}_8 (and \mathbf{ax}_6) we can define $\mathbf{realign}(\Phi, \beta, \alpha)$ by

$$\mathbf{realign}(\Phi, \beta, \alpha) e p \psi f g \triangleq \alpha e p (\psi \vee (\forall (i : \mathbf{I}). \Phi (p i))) (f \cup f') g \tag{5.21}$$

where $f' : [\forall (i : \mathbf{I}). \Phi (p i)] \rightarrow \Pi_{\mathbf{I}}(A \circ p)$ is given by $f' u \triangleq \mathbf{fill} e \beta (\lambda i \rightarrow (p, u i)) \psi f g$. The fact that f and f' are compatible follows from the fact that we use ψ and f in the

definition of f' and so by the properties of filling f' must agree with f wherever they are mutually defined. \square

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 sense that, given the following data:

$$\gamma : \Delta \rightarrow \Gamma, \quad \Phi : \Gamma \rightarrow \mathbf{Cof}, \quad A : \Gamma \rightarrow \mathcal{U}, \quad \beta : \mathbf{isFib}(A \circ \iota), \quad \alpha : \mathbf{isFib} A$$

then,

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

Therefore we have

$$\mathbf{realign}(\Phi, \beta, \alpha)[\gamma] = \mathbf{realign}(\Phi \circ \gamma, \beta[\gamma \times id], \alpha[\gamma])$$

5.3.3 Type formers and simple datatypes

Theorem 5.3.11 (Fibrant Σ -types). *There is a function*

$$\begin{aligned} \mathbf{isFib}_\Sigma : \{ \Gamma : \mathcal{U} \} \{ A_1 : \Gamma \rightarrow \mathcal{U} \} \{ A_2 : (x : \Gamma) \times A_1 x \rightarrow \mathcal{U} \} \rightarrow & \quad (5.22) \\ \mathbf{isFib} A_1 \rightarrow \mathbf{isFib} A_2 \rightarrow \mathbf{isFib}(\Sigma A_1 A_2) \end{aligned}$$

where $\Sigma A_1 A_2 x \triangleq (a_1 : A_1 x) \times A_2(x, a_1)$. The function is stable under re-indexing, in the sense that for all $\gamma : \Delta \rightarrow \Gamma$

$$(\mathbf{isFib}_\Sigma \alpha_1 \alpha_2)[\gamma] = \mathbf{isFib}_\Sigma(\alpha_1[\gamma])(\alpha_2[\gamma \times id]) \quad (5.23)$$

Hence the category with families given by CCHM fibrations supports the interpretation of Σ -types [30, Definition 3.18].

Proof. The construction of \mathbf{isFib}_Σ makes use of the filling operation from Lemma 5.3.7. Given $\Gamma : \mathcal{U}$, $A_1 : \Gamma \rightarrow \mathcal{U}$, $A_2 : (x : \Gamma) \times A_1 x \rightarrow \mathcal{U}$, $\alpha_1 : \mathbf{isFib} A_1$, $\alpha_2 : \mathbf{isFib} A_2$, $e : \{0, 1\}$, $p : \mathbf{I} \rightarrow \Gamma$, $\varphi : \mathbf{Cof}$, $f : [\varphi] \rightarrow \Pi_{\mathbf{I}}((\Sigma A_1 A_2) \circ p)$ and $(a_1, a_2) : (\Sigma A_1 A_2)(p e)$ with

$(\varphi, f) @ e \nearrow (a_1, a_2)$, define

$$\mathbf{isFib}_\Sigma \alpha_1 \alpha_2 e p \varphi f (a_1, a_2) \triangleq (\alpha_1 e p \varphi f_1 a_1, \alpha_2 e q \varphi f_2 a_2) \quad (5.24)$$

where

$$\begin{aligned} f_1 &: [\varphi] \rightarrow \Pi_I(A_1 \circ p) \\ f_1 u i &\triangleq \mathbf{fst}(f u i) \\ q &: I \rightarrow (x : \Gamma) \times A_1 x \\ q &\triangleq \langle p, \mathbf{fill} e \alpha_1 p \varphi f_1 a_1 \rangle \\ f_2 &: [\varphi] \rightarrow \Pi_I(A_2 \circ q) \\ f_2 u i &\triangleq \mathbf{snd}(f u i) \end{aligned}$$

Thus $\mathbf{isFib}_\Sigma \alpha_1 \alpha_2 e p \varphi f (a_1, a_2) : (\Sigma A_1 A_2)(p \bar{e})$; and since

$$\begin{aligned} \forall(u : [\varphi]). f_1 u \bar{e} &= \alpha_1 e p \varphi f_1 a_1 = \mathbf{fill} e \alpha_1 p \varphi f_1 a_1 \bar{e} \\ \forall(u : [\varphi]). f_2 u \bar{e} &= \alpha_2 e q \varphi f_2 a_2 \end{aligned}$$

hold, it follows that

$$(\varphi, f) @ \bar{e} \nearrow \mathbf{isFib}_\Sigma \alpha_1 \alpha_2 e p \varphi f (a_1, a_2).$$

Hence $\mathbf{isFib}_\Sigma \alpha_1 \alpha_2 : \mathbf{isFib}(\Sigma A_1 A_2)$. Finally, property (5.23) follows from (5.18) and (5.24). \square

Theorem 5.3.12 (Fibrant Π -types). *There is a function*

$$\begin{aligned} \mathbf{isFib}_\Pi : \{\Gamma : \mathcal{U}\} \{A_1 : \Gamma \rightarrow \mathcal{U}\} \{A_2 : (x : \Gamma) \times A_1 x \rightarrow \mathcal{U}\} \rightarrow \\ \mathbf{isFib} A_1 \rightarrow \mathbf{isFib} A_2 \rightarrow \mathbf{isFib}(\Pi A_1 A_2) \end{aligned} \quad (5.25)$$

where $\Pi A_1 A_2 x \triangleq (a_1 : A_1 x) \rightarrow A_2(x, a_1)$. This function is stable under re-indexing (cf. 5.23) and hence the category with families given by CCHM fibrations supports the interpretation of Π -types [30, Definition 3.15].

Proof. Given $\Gamma : \mathcal{U}$, $A_1 : \Gamma \rightarrow \mathcal{U}$, $A_2 : (x : \Gamma) \times A_1 x \rightarrow \mathcal{U}$, $\alpha_1 : \mathbf{isFib} A_1$, $\alpha_2 : \mathbf{isFib} A_2$, $e : \{0, 1\}$, $p : I \rightarrow \Gamma$, $\varphi : \mathbf{Cof}$, $f : [\varphi] \rightarrow \Pi_I((\Pi A_1 A_2) \circ p)$, $g : (\Pi A_1 A_2)(p e)$ with $(\varphi, f) @ e \nearrow g$ and $a_1 : A_1(p \bar{e})$, using Lemma 5.3.7 we define

$$\mathbf{isFib}_\Pi \alpha_1 \alpha_2 e p \varphi f g a_1 \triangleq \alpha_2 e q \varphi f_2 a_2 \quad (5.26)$$

where

$$\begin{aligned}
f_1 &: \Pi_{\mathbf{I}}(A_1 \circ p) \\
f_1 &\triangleq \mathbf{fill} \bar{e} \alpha_1 p \perp \mathbf{elim}_\emptyset a_1 \\
q &: \mathbf{I} \rightarrow (x : \Gamma) \times A_1 x \\
q &\triangleq \langle p, f_1 \rangle \\
f_2 &: [\varphi] \rightarrow \Pi_{\mathbf{I}}(A_2 \circ q) \\
f_2 u i &\triangleq f u i (f_1 i) \\
a_2 &: \{a'_2 : A_2(q e) \mid (\varphi, f_2) @ e \nearrow a'_2\} \\
a_2 &\triangleq g(f_1 e)
\end{aligned}$$

Since we know that $f_1 \bar{e} = \mathbf{fill} \bar{e} \alpha_1 p \perp \mathbf{elim}_\emptyset a_1 \bar{e} = a_1$, therefore we have

$$\mathbf{isFib}_\Pi \alpha_1 \alpha_2 e p \varphi f g a_1 : A_2(q \bar{e}) = A_2(p \bar{e}, f_1 \bar{e}) = A_2(p \bar{e}, a_1) \quad (5.27)$$

Furthermore, since $(\varphi, f_2) @ \bar{e} \nearrow \alpha_2 e q \varphi f_2 a_2$, for any $u : [\varphi]$ we have

$$f u \bar{e} a_1 = f u \bar{e} (f_1 \bar{e}) = f_2 u \bar{e} = \alpha_2 e q \varphi f_2 a_2 = \mathbf{isFib}_\Pi \alpha_1 \alpha_2 e p \varphi f g a_1 \quad (5.28)$$

Since (5.27) and (5.28) hold for all $a_1 : A_1(p \bar{e})$, from the first it follows that

$$\mathbf{isFib}_\Pi \alpha_1 \alpha_2 e p \varphi f g : (\Pi A_1 A_2)(p \bar{e})$$

and from the second that $(\varphi, f) @ \bar{e} \nearrow \mathbf{isFib}_\Pi \alpha_1 \alpha_2 e p \varphi f g$. Therefore we have that (5.26) does give an element of $\mathbf{isFib}(\Pi A_1 A_2)$. Finally, stability of $\mathbf{isFib}_\Pi \alpha_1 \alpha_2$ under re-indexing follows from (5.18). \square

These theorems allow us to construct fibration structures for Σ - and Π -types, given fibration structures for their constituent types. But are there any fibration structures to begin with? We answer this question by showing that the natural number object \mathbf{N} in the topos is always fibrant. This is proved for the topos of cubical sets $\widehat{\square}$ in [11, Section 4.5] by defining a composition structure by primitive recursion. We give a more elementary proof using the fact that the interval object in $\widehat{\square}$ satisfies axiom \mathbf{ax}_1 (see Theorem 5.6.1).

Theorem 5.3.13 (N is fibrant). *If \mathbf{N} is an object with decidable equality, then there is a function $\mathbf{isFib}_\mathbf{N} : \{\Gamma : \mathcal{U}\} \rightarrow \mathbf{isFib}(\lambda(- : \Gamma) \rightarrow \mathbf{N})$. In particular, if the topos \mathcal{E} has a natural number object $1 \xrightarrow{\mathbf{Z}} \mathbf{N} \xrightarrow{\mathbf{S}} \mathbf{N}$, then the category with families given by CCHM fibrations has a natural number object.*

Proof. Suppose $\Gamma : \mathcal{U}$, $e : \{0, 1\}$, $p : \mathbf{I} \rightarrow \Gamma$, $\varphi : \mathbf{Cof}$, $f : [\varphi] \rightarrow \Pi_{\mathbf{I}}(\lambda_- \rightarrow \mathbf{N})$ and $n : \mathbf{N}$ with $(\varphi, f) @ e \nearrow n$. By assumption on \mathbf{N} , for each $u : [\varphi]$ the property $\lambda(i : \mathbf{I}) \rightarrow (f u i = n) : \mathbf{I} \rightarrow \Omega$

is decidable; hence by axiom \mathbf{ax}_1 and the fact that $fue = n$, we also have $fue = n$. Therefore we can get $\mathbf{isFib}_{\mathbb{N}}ep\varphi fn : \{n' : \mathbb{N} \mid (\varphi, f) @ \bar{e} \nearrow n'\}$ just by defining: $\mathbf{isFib}_{\mathbb{N}}ep\varphi fn \triangleq n$. For the last part of the theorem we use the fact that in a topos with natural number object, equality of numbers is decidable. \square

A similar use of axiom \mathbf{ax}_1 suffices to prove:

Theorem 5.3.14 (Fibrant coproducts). *Writing $A_1 \xrightarrow{\mathbf{inl}} A_1 + A_2 \xleftarrow{\mathbf{inr}} A_2$ for the coproduct of A_1 and A_2 in \mathcal{E} , we lift this to families of types, $-\uplus - : \{\Gamma : \mathcal{U}\}(A_1 A_2 : \Gamma \rightarrow \mathcal{U}) \rightarrow \Gamma \rightarrow \mathcal{U}$, by defining $(A_1 \uplus A_2)x \triangleq A_1x + A_2x$. Then there is a function*

$$\mathbf{isFib}_{\uplus} : \{\Gamma : \mathcal{U}\}\{A_1 A_2 : \Gamma \rightarrow \mathcal{U}\} \rightarrow \mathbf{isFib} A_1 \rightarrow \mathbf{isFib} A_2 \rightarrow \mathbf{isFib}(A_1 \uplus A_2) \quad (5.29)$$

and this fibration structure on coproducts is stable under re-indexing. Hence the category with families given by CCHM fibrations has binary coproducts.

Proof. The proof makes use of the principle of unique choice, which holds in the internal type theory of a topos:

$$\mathbf{uc} : (A : \mathcal{U})(\varphi : A \rightarrow \Omega) \rightarrow [\exists!(a : A). \varphi a] \rightarrow \{a : A \mid \varphi a\} \quad (5.30)$$

where $\exists!(a : A). \varphi a \triangleq \exists(a : A). \varphi a \wedge \forall(a' : A). \varphi a' \Rightarrow a = a'$.

Suppose we have $\Gamma : \mathcal{U}$, $A_1 A_2 : \Gamma \rightarrow \mathcal{U}$, $\alpha_1 : \mathbf{isFib} A_1$, $\alpha_2 : \mathbf{isFib} A_2$, $e : \{0, 1\}$, $p : \mathbf{I} \rightarrow \Gamma$, $\varphi : \mathbf{Cof}$, $g : [\varphi] \rightarrow \Pi_{\mathbf{I}}((A_1 \uplus A_2) \circ p)$ and $c : A_1(pe) + A_2(pe)$ with $(\varphi, g) @ e \nearrow c$. Note that for all $u : [\varphi]$ and $i : \mathbf{I}$

$$\begin{aligned} P_1, P_2 &: [\varphi] \rightarrow \mathbf{I} \rightarrow \Omega \\ P_1 u i &\triangleq \exists!(a_1 : A_1(pi)). g u i = \mathbf{inl} a_1 \\ P_2 u i &\triangleq \exists!(a_2 : A_2(pi)). g u i = \mathbf{inr} a_2 \end{aligned}$$

are complementary propositions ($P_1 u i \wedge P_2 u i = \perp$ and $P_1 u i \vee P_2 u i = \top$); and hence by \mathbf{ax}_1 we have that $(\forall(i : \mathbf{I}). P_1 u i) \vee (\forall(i : \mathbf{I}). P_2 u i)$. Either $c = \mathbf{inl} a_1$ for some $a_1 : A_1(pe)$, or $c = \mathbf{inr} a_2$ for some $a_2 : A_2(pe)$. In the first case, since $\forall(u : [\varphi]). g u e = c$, it follows that $\forall(u : [\varphi])(i : \mathbf{I}). P_1 u i$; then using \mathbf{uc} we get some $f_1 : [\varphi] \rightarrow \Pi_{\mathbf{I}}(A_1 \circ p)$ with $\forall(u : [\varphi])(i : \mathbf{I}). g u i = \mathbf{inl}(f_1 u i)$ and we can define

$$\mathbf{isFib}_{\uplus} \alpha_1 \alpha_2 e p \varphi g (\mathbf{inl} a_1) \triangleq \mathbf{inl}(\alpha_1 e p \varphi f_1 a_1)$$

Similarly if $c = \mathbf{inr} a_2$, then there is some $f_2 : [\varphi] \rightarrow \Pi_{\mathbf{I}}(A_2 \circ p)$ with $\forall(u : [\varphi])(i : \mathbf{I}). g u i = \mathbf{inr}(f_2 u i)$ and we can define

$$\mathbf{isFib}_{\uplus} \alpha_1 \alpha_2 e p \varphi g (\mathbf{inr} a_2) \triangleq \mathbf{inr}(\alpha_2 e p \varphi f_2 a_2).$$

□

5.3.4 Path and identity types

Theorem 5.3.15 (Fibrant path types). *There is a function*

$$\mathbf{isFib}_{\text{Path}} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \mathbf{isFib} A \rightarrow \mathbf{isFib}(\text{Path } A) \quad (5.31)$$

where $\text{Path } A : (x : \Gamma) \times (Ax \times Ax) \rightarrow \mathcal{U}$ is given by

$$\text{Path } A(x, (a_0, a_1)) \triangleq a_0 \sim a_1 \quad (5.32)$$

and where \sim is as in (5.5). This fibration structure on path types is stable under re-indexing, in the sense that for all $\gamma : \Delta \rightarrow \Gamma$

$$(\mathbf{isFib}_{\text{Path}} \alpha)[\gamma \times (id \times id)] = \mathbf{isFib}_{\text{Path}}(\alpha[\gamma]) \quad (5.33)$$

Proof. Given $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$, $\alpha : \mathbf{isFib} A$, $e : \{0, 1\}$, $p : \mathbf{I} \rightarrow (x : \Gamma) \times (Ax \times Ax)$, $\varphi : \text{Cof}$, $f : [\varphi] \rightarrow \Pi_{\mathbf{I}}((\text{Path } A) \circ p)$, $q : \text{Path } A(p e)$ with $(\varphi, f) @ e \nearrow q$ and $i : \mathbf{I}$, suppose $p = \langle p', \langle q_0, q_1 \rangle \rangle$ where $p' : \mathbf{I} \rightarrow \Gamma$ and $q_0, q_1 : \Pi_{\mathbf{I}}(A \circ p)$, and define

$$\mathbf{isFib}_{\text{Path}} \alpha e p \varphi f q i \triangleq \alpha e p' (\varphi \vee i = 0 \vee i = 1) (f' \cup f_0 \cup f_1) (q i) \quad (5.34)$$

where

$$\begin{aligned} f' &: [\varphi] \rightarrow \Pi_{\mathbf{I}}(A \circ p') \\ f' u j &\triangleq f u j i \\ f_0 &: \{g : [i = 0] \rightarrow \Pi_{\mathbf{I}}(A \circ p') \mid (\varphi, f') \smile (i = 0, g)\} \\ f_0 - &\triangleq q_0 \\ f_1 &: \{g : [i = 1] \rightarrow \Pi_{\mathbf{I}}(A \circ p') \mid (\varphi \vee i = 0, f' \cup f_0) \smile (i = 1, g)\} \\ f_1 - &\triangleq q_1 \end{aligned}$$

Thus for each $i : \mathbf{I}$ we have $\mathbf{isFib}_{\text{Path}} \alpha e p \varphi f q i : A(p' \bar{e})$, so that $\mathbf{isFib}_{\text{Path}} \alpha e p \varphi f q : \mathbf{I} \rightarrow A(p' \bar{e})$. Since $\alpha e p' : \text{Compe}(A \circ p')$, we have

$$\begin{aligned} \forall (u : [\varphi \vee i = 0 \vee i = 1]). (f' \cup f_0 \cup f_1) u \bar{e} = \\ \alpha e p' (\varphi \vee i = 0 \vee i = 1) (f' \cup f_0 \cup f_1) (q i) = \mathbf{isFib}_{\text{Path}} \alpha e p \varphi f q i \end{aligned}$$

Hence $\mathbf{isFib}_{\text{Path}} \alpha e p \varphi f q 0 = q_0 \bar{e}$ and $\mathbf{isFib}_{\text{Path}} \alpha e p \varphi f q 1 = q_1 \bar{e}$, giving a path from

$q_0 \bar{e}$ to $q_1 \bar{e}$ and so

$$\mathbf{isFib}_{\mathbf{Path}} \alpha e p \varphi f q : \mathbf{Path} A (p' \bar{e}, (q_0 \bar{e}, q_1 \bar{e})) = \mathbf{Path} A (p \bar{e})$$

and furthermore, $(\varphi, f) @ \bar{e} \nearrow \mathbf{isFib}_{\mathbf{Path}} \alpha e p \varphi f q$. Thus $\mathbf{isFib}_{\mathbf{Path}} \alpha : \mathbf{isFib}(\mathbf{Path} A)$. Finally, property (5.33) is immediate from (5.34) and the definition of $[-]$ (5.15). \square

These path types in the CwF of CCHM fibrations (Definition 5.3.5) satisfy the Coquand formulation of identity types with propositional computation properties [11, Figure 2]. Thus in addition to the contractibility of singleton types (5.7), we get *substitution functions* for transporting elements of a fibration along a path

$$\begin{aligned} \mathbf{subst} &: \{\Gamma : \mathcal{U}\} \{A : \Gamma \rightarrow \mathcal{U}\} \{\alpha : \mathbf{isFib} A\} \{x_0 x_1 : \Gamma\} \rightarrow (x_0 \sim x_1) \rightarrow A x_0 \rightarrow A x_1 \quad (5.35) \\ \mathbf{subst} p a &\triangleq \alpha 0 p \perp \mathbf{elim}_\emptyset a \end{aligned}$$

using the cofibrant partial elements $(\perp, \mathbf{elim}_\emptyset)$ mentioned after Definition 5.1.2. By Lemma 5.3.7 we have that these substitution functions satisfy a propositional computation rule for constant paths (5.6):

$$\begin{aligned} \mathbf{H} &: \{\Gamma : \mathcal{U}\} \{A : \Gamma \rightarrow \mathcal{U}\} \{\alpha : \mathbf{isFib} A\} \{x : \Gamma\} (a : A x) \rightarrow (a \sim \mathbf{subst}(\mathbf{k} x) a) \quad (5.36) \\ \mathbf{H} a &\triangleq \mathbf{fill} 0 \alpha (\mathbf{k} x) \perp \mathbf{elim}_\emptyset a \end{aligned}$$

Remark 5.3.16 (Function extensionality). *As one might expect from [62, Lemma 6.3.2], the path types of Theorem 5.3.15 satisfy function extensionality. Given $A : \mathcal{U}$, $B : A \rightarrow \mathcal{U}$, $f, g : (x : A) \rightarrow B x$ and $p : (x : A) \rightarrow (f x \sim g x)$, we get a path $\mathbf{funext} p : f \sim g$ in $(x : A) \rightarrow B x$ given by*

$$\mathbf{funext} p i \triangleq \lambda(x : A) \rightarrow p x i$$

for all $i : \mathbf{I}$.

To get Martin-Löf identity types with standard definitional, rather than propositional computation properties from these path types, we use a version of Swan's construction [60] like the one in Section 9.1 of [18], but only using the connection algebra structure on \mathbf{I} , rather than a De Morgan algebra structure. This is the only place that axiom \mathbf{ax}_7 is used; we need the fact that the universe given by \mathbf{Cof} and $[-] : \mathbf{Cof} \rightarrow \mathcal{U}$ is closed under dependent products:

Lemma 5.3.17. *The following element of type Ω is provable: $\forall(\varphi : \Omega)(f : [\varphi] \rightarrow \Omega). \mathbf{cof} \varphi \Rightarrow (\forall(u : [\varphi]). \mathbf{cof}(f u)) \Rightarrow \mathbf{cof}(\exists(u : [\varphi]). f u)$.*

Proof. Note that if $u : [\varphi]$ then $(\exists(v : [\varphi]). f v) = f u$ and hence $\mathbf{cof}(\exists(v : [\varphi]). f v) = \mathbf{cof}(f u)$. So $\forall(u : [\varphi]). \mathbf{cof}(f u)$ equals $\varphi \Rightarrow \mathbf{cof}(\exists(v : [\varphi]). f v)$. Therefore from

$\text{cof } \varphi$ and $\forall(u : [\varphi]). \text{cof}(f u)$ by axiom ax_7 we get $\text{cof}(\varphi \wedge \exists(v : [\varphi]). f v)$ and hence $\text{cof}(\exists(v : [\varphi]). f v)$, since $(\exists(v : [\varphi]). f v) \Rightarrow \varphi$. \square

Theorem 5.3.18 (Fibrant identity types). *Define identity types by:*

$$\begin{aligned} \text{Id} &: \{\Gamma : \mathcal{U}\}(A : \Gamma \rightarrow \mathcal{U}) \rightarrow (x : \Gamma) \times (A x \times A x) \rightarrow \mathcal{U} & (5.37) \\ \text{Id } A(x, (a_0, a_1)) &\triangleq (p : \text{Path } A(x, (a_0, a_1))) \times \{\varphi : \text{Cof} \mid \varphi \Rightarrow \forall(i : \mathbf{I}). p i = a_0\} \end{aligned}$$

Then there is a function $\text{isFib}_{\text{Id}} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \text{isFib } A \rightarrow \text{isFib}(\text{Id } A)$ and the fibration $(\text{Id } A, \text{isFib}_{\text{Id}} A)$ can be given the structure of Martin-Löf identity types in the CwF of $CCHM$ fibrations [30, Definition 3.19].

Proof. Given $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$ and $\alpha : \text{isFib } A$, using Theorems 5.3.11 and 5.3.15 we define $\text{isFib}_{\text{Id}} \alpha \triangleq \text{isFib}_{\Sigma}(\text{isFib}_{\text{Path}} \alpha) \beta$, where $\beta : \text{isFib } \Phi$ with

$$\begin{aligned} \Phi &: (y : (x : \Gamma) \times (A x \times A x)) \times \text{Path } A y \rightarrow \mathcal{U} \\ \Phi((x, (a_0, a_1)), p) &\triangleq \{\varphi : \text{Cof} \mid \varphi \Rightarrow \forall(i : \mathbf{I}). p i = a_0\} \end{aligned}$$

and the fibration structure β mapping $e : \{0, 1\}$, $p : \mathbf{I} \rightarrow (y : (x : \Gamma) \times (A x \times A x)) \times \text{Path } A y$, $\varphi : \text{Cof}$, $f : [\varphi] \rightarrow \Pi_{\mathbf{I}}(\Phi \circ p)$ and $\varphi' : \Phi(p e)$ with $(\varphi, f) @ e \nearrow \varphi'$ to the element

$$\beta e p \varphi f \varphi' \triangleq \exists(u : [\varphi]). f u \bar{e}$$

(using Lemma 5.3.17 to see that this is well defined). We get the usual introduction, elimination and computation rules for these identity types as follows. Since $\top : \text{Cof}$ holds by axiom ax_5 , identity introduction

$$\text{refl} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\}\{x : \Gamma\}(a : A x) \rightarrow \text{Id } A(x, (a, a)) \quad (5.38)$$

can be defined by $\text{refl } a \triangleq (\lambda a i \rightarrow a, \top)$. Identity elimination

$$\begin{aligned} \text{J} &: \{\Gamma : \mathcal{U}\}(A : \Gamma \rightarrow \mathcal{U})(x : \Gamma)(a_0 : A x)(B : (a : A x) \times \text{Id } A(x, (a_0, a)) \rightarrow \mathcal{U}) & (5.39) \\ &(\beta : \text{isFib } B)(a_1 : A x)(e : \text{Id } A(x, (a_0, a_1))) \rightarrow B(a_0, \text{refl } a_0) \rightarrow B(a_1, e) \end{aligned}$$

is given by

$$\text{J } A x a_0 B \beta a_1(p, \varphi) b \triangleq \beta 0 \langle p, q \rangle \varphi f b$$

where $q : (i : \mathbf{I}) \rightarrow \text{Id } A x(a_0, p i)$ is $q i j \triangleq (p(i \sqcap j), \varphi \vee i = 0)$ and $f : [\varphi] \rightarrow \Pi_{\mathbf{I}}(B \circ \langle p, q \rangle)$ is $f u i \triangleq b$. (In the above element, since $(p, \varphi) : \text{Id } A(x, (a_0, a_1))$ we have $p 0 = a_0$, $p 1 = a_1$ and $\varphi \Rightarrow \forall(i : \mathbf{I}). p i = a_0$; hence $\varphi \Rightarrow \forall(i : \mathbf{I}). q i = \text{refl } a_0$, so that f is well-defined.) Note that by axioms ax_3 and ax_4 we have $q 0 = \text{refl } a_0$ and $q 1 = (p, \varphi)$, so that $\text{J } A x a_0 B \beta a_1(p, \varphi) b = \beta 0 \langle p, q \rangle \varphi f b : B(p 1, q 1) = B(a_1, (p, \varphi))$, as required.

Furthermore, since $(\varphi, f) @ 1 \nearrow \beta 0 \langle p, q \rangle \varphi f b$, we have

$$\forall (u : [\varphi]). b = f u 1 = J A x a_0 B \beta a_1(p, \varphi) b$$

So when $(p, \varphi) = \mathbf{refl} a_0 = (\mathbf{k} a_0, \top)$ and hence $a_1 = a_0$, we have

$$J A x a_0 B \beta a_0(\mathbf{refl} a_0) b = b \tag{5.40}$$

In other words the computation property for identity elimination holds as a judgemental equality and not just a propositional one. Finally, to correctly support the interpretation of intensional identity types, one needs stability of $(\mathbf{Id} A, \mathbf{isFib}_{\mathbf{Id}} A)$, \mathbf{refl} and $J A$ under re-indexing; but this follows from the stability properties of \mathbf{isFib}_{Σ} and $\mathbf{isFib}_{\text{Path}}$. \square

5.4 Glueing

In this section we give an internal presentation of the *glueing* construction given by Cohen et al. [18]. Glueing is similar to a composition structure (Definition 5.3.3) for type-families, except that instead of partial paths of types it involves partial equivalences between types. Glueing is crucial for the constructions relating to univalence [62] given in Section 5.5.

We begin by defining the glueing construction for cofibrant-partial types, that is, for functions $A : [\varphi] \rightarrow \mathcal{U}$ where $\varphi : \mathbf{Cof}$:

Definition 5.4.1 (Glueing). Given $\varphi : \mathbf{Cof}$, $A : [\varphi] \rightarrow \mathcal{U}$, $B : \mathcal{U}$ and $f : (u : [\varphi]) \rightarrow A u \rightarrow B$, the type $\mathbf{Glue} \varphi A B f : \mathcal{U}$ is defined to be

$$\mathbf{Glue} \varphi A B f \triangleq (a : (u : [\varphi]) \rightarrow A u) \times \{b : B \mid \forall (u : [\varphi]). f u (a u) = b\} \tag{5.41}$$

Elements of this type consist of pairs (a, b) where a is a partial element of the partial type A and b is an element of type B , such that f applied to a gives a partial element of B that extends to b . When $\varphi = \top$ then A and f are both total and so $\mathbf{Glue} \varphi A B f$ essentially consists of pairs $(a, f a)$ for every $a : A$ and hence is clearly isomorphic to A . When $\varphi = \perp$ then A and f are both uniquely determined and $\mathbf{Glue} \varphi A B f$ will consist of pairs $(\mathbf{elim}_{\emptyset}, b)$ for every $b : B$ and hence is clearly isomorphic to B .

We now extend this glueing operation from cofibrant-partial types to cofibrant-partial type-families:

Definition 5.4.2 (Glueing for families). We lift the glueing operation from types to type-families as follows. Given $\Gamma : \mathcal{U}$, $\Phi : \Gamma \rightarrow \mathbf{Cof}$, $A : \Gamma | \Phi \rightarrow \mathcal{U}$, $B : \Gamma \rightarrow \mathcal{U}$ and

$f : (x : \Gamma)(v : [\Phi x]) \rightarrow A(x, v) \rightarrow Bx$, define the family $\mathbf{Glue} \Phi A B f : \Gamma \rightarrow \mathcal{U}$ by

$$\mathbf{Glue} \Phi A B f x \triangleq \mathbf{Glue} (\Phi x) (A(x, -)) (Bx) (f(x, -)) \quad (5.42)$$

The glueing construction works for any map $f : (x : \Gamma)(v : [\Phi x]) \rightarrow A(x, v) \rightarrow Bx$. However, we want to see that this construction lifts to the CwF of CCHM fibrations. This means that $\mathbf{Glue} \Phi A B f$ should have a fibration structure whenever A and B do and this puts some requirements on f . We begin by introducing the notion of an *extension structure*:

Definition 5.4.3 (Extension structures). The type of extension structures, $\mathbf{Ext} : \mathcal{U} \rightarrow \mathcal{U}$, is given by

$$\mathbf{Ext} A \triangleq (\tilde{a} : \square A) \rightarrow \{a : A \mid \tilde{a} \nearrow a\}$$

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

$$\mathbf{Ext} A \triangleq (x : \Gamma) \rightarrow \mathbf{Ext}(Ax)$$

An extension structure for $A : \Gamma \rightarrow \mathcal{U}$ is similar to having a composition structure for A in the sense that both allow us to extend partial elements; and in fact every family with an extension structure is a fibration. However, an extension structure does not require a total element from which we extend/compose and so is in fact a stronger notion than a composition structure. First note that having an extension structure for $A : \mathcal{U}$ implies that A is inhabited, because we can always extend the empty partial element. Further, given any element $a : A$ we can use the extension structure to show that it is path equal to the extension of the empty partial element. Together these facts tell us that A is *contractible*:

Definition 5.4.4 (Contractibility, cf. Definition 2.1.1). 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 $\mathbf{Contr} A$ is inhabited, where $\mathbf{Contr} : \mathcal{U} \rightarrow \mathcal{U}$ is defined by

$$\mathbf{Contr} A \triangleq (a_0 : A) \times ((a : A) \rightarrow a_0 \sim a)$$

As with extension structures, we say that a family $A : \Gamma \rightarrow \mathcal{U}$ is contractible if each of its fibres is and write

$$\mathbf{Contr} A \triangleq (x : \Gamma) \rightarrow \mathbf{Contr}(Ax)$$

As mentioned above, having an extension structure for a family $A : \Gamma \rightarrow \mathcal{U}$ implies that A is both fibrant and contractible. In fact the converse is true as well (cf. [18, Lemma 5]):

Lemma 5.4.5. *There are functions*

$$\text{fromExt} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \text{Ext } A \rightarrow \text{isFib } A \times \text{Contr } A \quad (5.43)$$

$$\text{toExt} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \text{isFib } A \rightarrow \text{Contr } A \rightarrow \text{Ext } A \quad (5.44)$$

Proof. Given $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$ and $\varepsilon : \text{Ext } A$ we define $\alpha : \text{isFib } A$

$$\alpha \ e \ p \ \varphi \ f \ a_0 \triangleq \varepsilon \ (p \ \bar{e}) \ ((\varphi, f) \ @ \ \bar{e})$$

For every $x : \Gamma$ we use the totally undefined cofibrant-partial element $(\perp, \text{elim}_\emptyset) : \square A$ to define $a_0 : A$

$$a_0 \triangleq \varepsilon \ x \ (\perp, \text{elim}_\emptyset)$$

For each $a : Ax$ and $i : \mathbb{I}$, we have $\lambda_- \rightarrow a : [i = 1] \rightarrow Ax$; so we get a path $p_a : \mathbb{I} \rightarrow Ax$

$$p_a \ i \triangleq \varepsilon \ x \ (i = 1, \lambda_- \rightarrow a)$$

By the definition of $\text{Ext } A$ we have $p_a \ 1 = \varepsilon \ x \ (\top, \lambda_- \rightarrow a) = a$, and by ax_2 we have $p_a \ 0 = \varepsilon \ x \ (\perp, \text{elim}_\emptyset) = a_0$. Therefore $p_a : a_0 \sim a$. Hence Ax is contractible. Together this shows that, given $\varepsilon : \text{Ext } A$, we can define elements of type $\text{isFib } A$ and $\text{Contr } A$. Therefore there is a function $\text{fromExt} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \text{Ext } A \rightarrow \text{isFib } A \times \text{Contr } A$.

Conversely, given $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$, $\alpha : \text{isFib } A$, $\langle a_0, p \rangle : \text{Contr } A$, note that for any $x : \Gamma$, $\varphi : \text{Cof}$ and $f : [\varphi] \rightarrow Ax$ we have $(px) \circ f : [\varphi] \rightarrow (\mathbb{I} \rightarrow Ax)$ such that $\forall (u : [\varphi]). ((px) \circ f) \ u : a_0 \ x \sim f \ u$; therefore $(\varphi, (px) \circ f) \ @ \ 0 \nearrow (a_0 \ x)$ and so defining

$$\varepsilon \ x \ (\varphi, f) \triangleq \alpha \ 0 \ (\lambda_- \rightarrow x) \ \varphi \ ((px) \circ f) \ (a_0 \ x)$$

we get $\varepsilon \ x \ (\varphi, f) : Ax$. Furthermore, since $(\varphi, (px) \circ f) \ @ \ 1 = (\varphi, f)$ by the type of α we get $(\varphi, f) \nearrow \varepsilon \ x \ (\varphi, f)$. Thus $\varepsilon \ x \ (\varphi, f) : \text{Ext}(Ax)$, as required. \square

We now come to the main result of this section: showing that fibrations are closed under glueing. Proving this requires that the function f is an *equivalence*:

Definition 5.4.6 (Equivalences [62, Section 4.4]). Given types $A, B : \mathcal{U}$, a function $f : A \rightarrow B$ is an *equivalence* if the type $\text{Equiv } f$ is inhabited, where

$$\text{Equiv } f \triangleq (b : B) \rightarrow \text{Contr}((a : A) \times f \ a \sim b)$$

Again, this lifts to families in the obvious way: given $\Gamma : \mathcal{U}$, $A, B : \Gamma \rightarrow \mathcal{U}$ and $f : (x : \Gamma) \rightarrow Ax \rightarrow Bx$, define

$$\text{Equiv } f \triangleq (x : \Gamma) \rightarrow \text{Equiv } (f \ x)$$

Theorem 5.4.7 (Composition for glueing). *Let Φ, A, B and f be as in Definition 5.4.2. Then $\mathbf{Glue} \Phi A B f$ has a fibration structure if A and B both have one and f has the structure of an equivalence. In other words there is a function*

$$\begin{aligned} \mathbf{isFib}_{\mathbf{Glue}} : \{ \Gamma : \mathcal{U} \} \{ \Phi : \Gamma \rightarrow \mathbf{Cof} \} \{ A : \Gamma | \Phi \rightarrow \mathcal{U} \} \{ B : \Gamma \rightarrow \mathcal{U} \} \\ (f : (x : \Gamma)(u : [\Phi x]) \rightarrow A(x, u) \rightarrow B x) \rightarrow \\ ((x : \Gamma)(v : [\Phi x]) \rightarrow \mathbf{Equiv}(f x v)) \rightarrow \\ \mathbf{isFib} A \rightarrow \mathbf{isFib} B \rightarrow \mathbf{isFib}(\mathbf{Glue} \Phi A B f) \end{aligned} \quad (5.45)$$

Proof. Given $\Gamma : \mathcal{U}$, $\Phi : \Gamma \rightarrow \mathbf{Cof}$, $A : \Gamma | \Phi \rightarrow \mathcal{U}$, $B : \Gamma \rightarrow \mathcal{U}$, $f : (x : \Gamma)(u : [\Phi x]) \rightarrow A(x, u) \rightarrow B x$, $eq : (x : \Gamma)(u : [\Phi x]) \rightarrow \mathbf{Equiv}(f x v)$, $\alpha : \mathbf{isFib} A$, $\beta : \mathbf{isFib} B$, we wish to define an element of type $\mathbf{isFib}(\mathbf{Glue} \Phi A B f)$. Therefore, taking

$$\begin{aligned} e : \{0, 1\}, p : \mathbf{I} \rightarrow \Gamma, \psi : \mathbf{Cof}, q : [\psi] \rightarrow \Pi_{\mathbf{I}}(\mathbf{Glue} \Phi A B f) \\ (a_0, b_0) : \{ (a_0, b_0) : (\mathbf{Glue} \Phi A B f)(p e) \mid (\psi, q) @ e \nearrow (a_0, b_0) \} \end{aligned}$$

our goal is to define $(a_1, b_1) : (\mathbf{Glue} \Phi A B f) x$ such that $(\psi, \tilde{a}_1) \nearrow a_1$ and $(\psi, \tilde{b}_1) \nearrow b_1$, where $x : \Gamma$, $\tilde{a}_1 : [\psi] \rightarrow ((u : [\Phi x]) \rightarrow A(x, u))$ and $\tilde{b}_1 : [\psi] \rightarrow B x$ are defined by

$$\begin{aligned} x &\triangleq p \bar{e} \\ \tilde{a}_1 v &\triangleq \mathbf{fst}(q v \bar{e}) \\ \tilde{b}_1 v &\triangleq \mathbf{snd}(q v \bar{e}) \end{aligned}$$

and satisfy $\forall (v : [\psi])(u : [\Phi x]). f x u (\tilde{a}_1 v u) = \tilde{b}_1 v$ by the definition of $\mathbf{Glue} \Phi A B f$.

We start by composing over p in B to get $b'_1 : B x$ which can be thought of as a first approximation to b_1 :

$$b'_1 \triangleq \beta e p \psi (\lambda(v : [\psi])(i : \mathbf{I}) \rightarrow \mathbf{snd}(q v i)) b_0$$

Recall that a_1 will have type $(u : [\Phi x]) \rightarrow A(x, u)$ and so we assume $u : [\Phi x]$ in order to define an element of type $A(x, u)$. Note that we cannot simply compose over p in A because we do not know that $\Phi(p i)$ holds for all $i : \mathbf{I}$. Instead we will use the equivalence structure to define a_1 .

Let $C \triangleq (a : A(x, u)) \times f x u a \sim b'_1$ be the fibre of $f x u$ at b'_1 . Using Theorems 5.3.11 and 5.3.15 and the fact that both A and B are fibrations (as witnessed by α and β respectively) we can deduce that C is a fibrant object. Combined with the fact that $eq x u b'_1 : \mathbf{Contr} C$ we can use Lemma 5.4.5 to define $\varepsilon : \mathbf{Ext} C$. We can then define

$$(\psi, (\lambda(v : [\psi]) \rightarrow \tilde{a}_1 v u, \mathbf{refl} \circ \tilde{b}_1)) : \square C$$

This is well-defined because, as mentioned above, we have $\forall(v : [\psi]). f x u (\tilde{a}_1 v u) = \tilde{b}_1 v$ and, by the type of composition structures, we have $(\psi, \tilde{b}_1) \nearrow b'_1$ and so given $v : [\psi]$ we have $\mathbf{refl}(\tilde{b}_1 v) : f x u (\tilde{a}_1 v u) \sim b'_1$. Now we can define

$$\varepsilon(\psi, (\lambda(v : [\psi]) \rightarrow \tilde{a}_1 v u, \mathbf{refl} \circ \tilde{b}_1)) : C$$

Discharging our assumption $u : [\Phi x]$ and taking first and second projections of the pair defined above we get:

$$a_1 : (u : [\Phi x]) \rightarrow A(x, u) \quad p_b : (u : [\Phi x]) \rightarrow f x u (a_1 u) \sim b'_1$$

We now have a_1 and b'_1 such that $\tilde{a}_1 \nearrow a_1$ and $\tilde{b}_1 \nearrow b'_1$. However, we cannot simply take b_1 to be b'_1 because we do not know that $\forall(u : \Phi x). f x u (a_1 u) = b'_1$ and therefore cannot conclude that $(a_1, b'_1) : (\mathbf{Glue} \Phi A B f) x$. In order to solve this problem we perform one final composition in $B x$ in order to “correct” b'_1 to achieve this property. Consider the following join

$$p_b \cup (\mathbf{refl} \circ \tilde{b}_1) : [\Phi x \vee \psi] \rightarrow \Pi_{\mathbf{I}}(B x)$$

This is well defined because p_b is defined by extending $\mathbf{refl} \circ \tilde{b}_1$ and so they must be equal where they are both defined. We use this to perform one final composition in $B x$:

$$b_1 \triangleq \beta \mathbf{1} (\lambda(- : \mathbf{I}) \rightarrow x) (\Phi x \vee \psi) (p_b \cup (\mathbf{refl} \circ \tilde{b}_1)) b'_1$$

We now have $(a_1, b_1) : (\mathbf{Glue} \Phi A B f) x$ such that $(\psi, \tilde{a}_1) \nearrow a_1$ and $(\psi, \tilde{b}_1) \nearrow b_1$, as required. \square

We now have a way to interpret the glueing operation from [18] that meets some of the necessary requirements; see [18, Figure 4]. However, the current construction fails the requirement that $\mathbf{Glue} \Phi A B f$ should be equal to A when reindexing along the inclusion $\iota : \Gamma | \Phi \hookrightarrow \Gamma$. In fact, this equality should hold in the CwF of CCHM fibrations. This means that not only should $A = (\mathbf{Glue} \Phi A B f) \circ \iota : \Gamma | \Phi \rightarrow \mathcal{U}$, but also that reindexing the fibration structure derived in Theorem 5.4.7 should result in the same fibration structure with which we started. To be precise, what we require is:

$$(A, \alpha) = (\mathbf{Glue} \Phi A B f, \mathbf{isFib}_{\mathbf{Glue}} f \text{ eq } \alpha \beta)[\iota] \tag{5.46}$$

What we have at present is that the families A and $(\mathbf{Glue} \Phi A B f) \circ \iota$ are isomorphic in the sense of Definition 5.1.6.

We get to (5.46) in two steps. First we use Axiom \mathbf{ax}_9 in order to *strictify* the glueing construction to get a new, strict form of glueing, \mathbf{SGLue} , such that $\mathbf{Glue} \Phi A B f \cong \mathbf{SGLue} \Phi A B f$ but where $A = (\mathbf{SGLue} \Phi A B f) \circ \iota$. We then use Axiom \mathbf{ax}_8 to adapt the

fibration structure on $\mathbf{S}\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e}$ so that under reindexing along ι it is equal to the fibration structure on A . The order of these steps does not seem to be important; we could equally have first adapted the fibration structure on $\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e}$ and then strictified $\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e}$ with this new fibration structure.

Remark 5.4.8. *Apart from the use of the internal language of a topos, our approach to getting a glueing operation with good properties diverges from that taken by Cohen et al. [18], where glueing is defined directly with all the required properties. However it is possible to see where our final two steps occur in the original work. The strictification can be seen in the use of the case split on $\varphi\rho = 1_{\mathbb{F}}$ in [18, Definition 15]; see Section 5.6.2 for more details. Rather than defining an initial composition structure for glueing and then modifying it to get the required reindexing property, Cohen et al. define the composition structure directly. Removing all uses of the \forall operator from [18, Section 6.2] would yield our initial composition structure, and we then use the closure of \mathbf{Cof} under $\forall(i : \mathbf{I})$ (axiom \mathbf{ax}_8) in a separate step to modify this composition. We prefer this approach because it simplifies the core composition structure for glueing and makes more explicit what role \mathbf{ax}_8 plays in the construction of a model of cubical type theory.*

We now recall axiom \mathbf{ax}_9 from Figure 5.4:

$$\mathbf{ax}_9 : \{\varphi : \mathbf{Cof}\}(A : [\varphi] \rightarrow \mathcal{U})(B : \mathcal{U})(s : (u : [\varphi]) \rightarrow (A u \cong B)) \rightarrow (B' : \mathcal{U}) \times \{s' : B' \cong B \mid \forall(u : [\varphi]). A u = B' \wedge s u = s'\}$$

This states that any partial type A , which is isomorphic to a total type B everywhere that it is defined, can be extended to a total type B' that is isomorphic to B . We investigate why the cubical presheaf topos [18] satisfies this axiom in section 5.6.2. Given \mathbf{ax}_8 , it is straightforward to define a strict form of glueing.

Definition 5.4.9 (Strict glueing). Given $\Gamma : \mathcal{U}$, $\Phi : \Gamma \rightarrow \mathbf{Cof}$, $A : \Gamma | \Phi \rightarrow \mathcal{U}$, $B : \Gamma \rightarrow \mathcal{U}$ and $f : (x : \Gamma)(u : [\Phi x]) \rightarrow A(x, u) \rightarrow B x$, define $\mathbf{S}\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e} \Phi A B f : \Gamma \rightarrow \mathcal{U}$ by

$$\mathbf{S}\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e} \Phi A B f x \triangleq \mathbf{fst}(\mathbf{ax}_9 (\lambda u : [\Phi x] \rightarrow A(x, u)) (\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e} \Phi A B f x) (\lambda u : [\Phi x] \rightarrow \mathbf{g}\mathbf{l}\mathbf{u}\mathbf{e} (x, u))) \quad (5.47)$$

where $\mathbf{g}\mathbf{l}\mathbf{u}\mathbf{e} (x, u) : A(x, u) \cong \mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e} \Phi A B f x$ is the isomorphism alluded to in Definition 5.4.1 given by

$$\mathbf{g}\mathbf{l}\mathbf{u}\mathbf{e} (x, u) a \triangleq (\lambda_- \rightarrow a, f x u a)$$

Note that $\mathbf{S}\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e}$ has the desired strictness property: given any $(x, u) : \Gamma | \Phi$, by \mathbf{ax}_9 we have $A(x, u) = \mathbf{fst}(\mathbf{ax}_9 (\lambda u : [\Phi x] \rightarrow A(x, u)) (\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e} \Phi A B f x) (\lambda u : [\Phi x] \rightarrow \mathbf{g}\mathbf{l}\mathbf{u}\mathbf{e} (x, u)))$ and hence

$$\forall(x : \Gamma)(u : [\Phi x]). \mathbf{S}\mathbf{G}\mathbf{l}\mathbf{u}\mathbf{e} \Phi A B f x = A(x, u) \quad (5.48)$$

Theorem 5.4.10 (Composition structure for strict glueing). *Given Γ, Φ, A, B , and f as in Definition 5.4.9, $\mathbf{SGLue} \Phi A B f : \Gamma \rightarrow \mathcal{U}$ has a fibration structure if A and B have one and f has the structure of an equivalence.*

Proof. By Lemma 5.3.8 we have that (fibrewise) isomorphisms preserve fibration structures. Hence we obtain a fibration structure on \mathbf{SGLue} by transporting the structure obtained from $\mathbf{isFib}_{\mathbf{GLue}}$ (Theorem 5.4.7) along the isomorphism from \mathbf{ax}_9 . \square

The final step in this section in this section is to use axiom \mathbf{ax}_8 to adapt the composition structure for \mathbf{SGLue} so that we recover the original composition structure on A .

Corollary 5.4.11. *Given $\Gamma : \mathcal{U}$, $\Phi : \Gamma \rightarrow \mathbf{Cof}$, $(A, \alpha) : \mathbf{Fib}(\Gamma|\Phi)$, $(B, \beta) : \mathbf{Fib} \Gamma$ and $f : (x : \Gamma)(v : [\Phi x]) \rightarrow A(x, v) \rightarrow B x$, there exists $(G, \gamma) : \mathbf{Fib} \Gamma$ such that $(A, \alpha) = (G, \gamma)[\iota]$.*

Proof. Simply take $G = \mathbf{SGLue} \Phi A B f$; by Theorem 5.4.10 we get a composition structure for G , which we then adapt using Lemma 5.3.10 to get a new composition structure γ satisfying the required equality. \square

5.5 Univalence

Voevodsky's *univalence axiom* (Section 2.1.3) for a universe \mathcal{V} in a \mathbf{CwF} (with at least Σ -, Π - and \mathbf{Id} -types) states that for every $A, B : \mathcal{V}$ the canonical function from $\mathbf{Id} \mathcal{V} A B$ to $(f : A \rightarrow B) \times \mathbf{Equiv} f$ is an equivalence. Cohen et al. construct a universe in the (\mathbf{CwF} associated to the) presheaf topos of cubical sets whose family of types is generic for CCHM fibrations with small fibres (for a suitable notion of smallness in the metatheory) and prove that it satisfies the univalence axiom. They do so by adapting the Hofmann-Streicher universe construction for presheaf categories [31], see Section 3.3.5 for details. It is not possible to express their universe construction just using the internal type theory of a general topos, for reasons that we discuss in Theorem 7.1.1. In Chapter 7 we explain why the universe construction cannot be axiomatised naively and then show how to resolve this issue by working in a modal extension of the internal type theory. For now, we just prove a version of univalence without reference to a universe of fibrations.

To understand what this might mean, consider the following: were there to be a universe \mathcal{V} whose elements are codes for CCHM fibrations, then given fibrations $(A, \alpha), (B, \beta) : \mathbf{Fib} \Gamma$ named by functions $a, b : \Gamma \rightarrow \mathcal{V}$ into the universe, a path-equality between a and b gives (by Currying) a function $p : \Gamma \times \mathbf{I} \rightarrow \mathcal{V}$ such that $p(x, 0) = a x$ and $p(x, 1) = b x$ for all $x : \Gamma$. Then p names a fibration $(P, \rho) : \mathbf{Fib}(\Gamma \times \mathbf{I})$ such that $(P, \rho)[\langle id, 0 \rangle] = (A, \alpha)$ and $(P, \rho)[\langle id, 1 \rangle] = (B, \beta)$. The latter gives a notion of path-equality between type-families whether or not there is such a \mathcal{V} , which we study in this section in relation to equivalences between fibrations.

To give the definitions more formally, we expand our running assumption that the ambient topos \mathcal{E} comes with a universe (internal full subtopos) \mathcal{U} to the case where there is a second universe \mathcal{U}_1 with $\mathcal{U} : \mathcal{U}_1$. We sometimes refer to objects of type \mathcal{U} as *small* types and objects of type \mathcal{U}_1 as *large* types.

Definition 5.5.1 (Path equality between fibrations). Define the type of paths between CCHM fibrations $_{-} \sim_{\mathcal{U}} _{-} : \{\Gamma : \mathcal{U}\} \rightarrow \mathbf{Fib} \Gamma \rightarrow \mathbf{Fib} \Gamma \rightarrow \mathcal{U}_1$ by

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

Previously, we indicated why such a notion of path should correspond to the usual notion were there to exist a universe of CCHM fibrations. Here we explain why this notion is indeed equivalent in the concrete model given in [18]. Recall that the universe construction there is given by a variant of the usual Hofmann-Streicher universe construction for presheaf categories, as described in Section 3.3.5. This means that, in the external metatheory, there exists a type $\mathcal{V} \in \mathbf{Ty}(\Gamma)$ for all Γ given by $\mathcal{V}(I, \rho) \triangleq \mathbf{FTy}_0(yI)$ where yI denotes the Yoneda embedding of I . Every (small) fibrant type $A \in \mathbf{FTy}_0(\Gamma)$ has a code $\ulcorner A \urcorner \in \mathbf{Ter}(\Gamma \vdash \mathcal{V})$ and every $a \in \mathbf{Ter}(\Gamma \vdash \mathcal{V})$ encodes a type $El a \in \mathbf{FTy}_0(\Gamma)$ 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 $P \in \mathbf{FTy}_0(\Gamma.\mathbb{I})$ such that $P[\langle id, 0 \rangle] = A$ and $P[\langle id, 1 \rangle] = B$ for some $\Gamma \in \widehat{\square}$ and $A, B \in \mathbf{FTy}_0(\Gamma)$. From this data we can construct $p \in \mathbf{Ter}(\Gamma \vdash \mathbf{Path}_{\mathcal{V}} \ulcorner A \urcorner \ulcorner B \urcorner)$ like so:

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

for $I \in \square, \rho \in \Gamma(I)$, where $s_i : I, i \rightarrow I$ 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:

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

Therefore $(\ulcorner P \urcorner(\rho s_i, i))(i0) = \ulcorner A \urcorner(\rho)$. Similarly when $i = 1$ we have $(\ulcorner P \urcorner(\rho s_i, i))(i1) = \ulcorner B \urcorner(\rho)$, and so $\langle i \rangle \ulcorner P \urcorner(\rho s_i, i) \in \mathbf{Path}_{\mathcal{V}} \ulcorner A \urcorner \ulcorner B \urcorner$ as required.

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

$$P \triangleq El(p') \quad \text{where} \quad p'(\rho, i) \triangleq p \rho i$$

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

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

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.

We now show that given such a path $(P, \rho) : (A, \alpha) \sim_{\mathcal{U}} (B, \beta)$ it is always possible to construct an equivalence $f : (x : \Gamma) \rightarrow Ax \rightarrow Bx$. Conversely, given an equivalence $f : (x : \Gamma) \rightarrow Ax \rightarrow Bx$ between fibrations (A, α) and (B, β) , it is always possible to construct such a (P, ρ) .

Theorem 5.5.2 (Converting paths to equivalences). *There is a function*

$$\begin{aligned} \text{pathToEquiv} : \{ \Gamma : \mathcal{U} \} \{ A B : \text{Fib } \Gamma \} & & (5.49) \\ (P : A \sim_{\mathcal{U}} B) \rightarrow (f : (x : \Gamma) \rightarrow \text{fst } Ax \rightarrow \text{fst } Bx) \times \text{Equiv } f \end{aligned}$$

Proof. Given $\Gamma : \mathcal{U}, (A, \alpha), (B, \beta) : \text{Fib } \Gamma$ and $(P, \rho) : A \sim_{\mathcal{U}} B$ we define maps $f : (x : \Gamma) \rightarrow Ax \rightarrow Bx$ and $g : (x : \Gamma) \rightarrow Bx \rightarrow Ax$. First, given $x : \Gamma$ write $\langle x, id \rangle : (x, 0) \sim (x, 1)$ for the path given by $\langle x, id \rangle i \triangleq (x, i)$. Now define f and g as follows:

$$f x a \triangleq \rho 0 \langle x, id \rangle \perp \text{elim}_{\emptyset} a \qquad g x b \triangleq \rho 1 \langle x, id \rangle \perp \text{elim}_{\emptyset} b$$

This definition is well-typed since $P(x, 0) = Ax$ and $P(x, 1) = Bx$. Since both functions are defined using composition structure we can use filling (Lemma 5.3.7) to find dependently typed paths:

$$\begin{aligned} p : (x : \Gamma)(a : Ax) \rightarrow \Pi_{\Gamma} P & \text{ defined by } p x a \triangleq \text{fill } 0 \rho \langle x, id \rangle \perp \text{elim}_{\emptyset} a \\ q : (x : \Gamma)(b : Bx) \rightarrow \Pi_{\Gamma} P & \text{ defined by } q x b \triangleq \text{fill } 1 \rho \langle x, id \rangle \perp \text{elim}_{\emptyset} b \end{aligned}$$

Note that for all $x : \Gamma$ and $a : Ax$ we have $p x a 0 = a$ and $p x a 1 = f x a$. Similarly, for all $b : Bx$ we have $q x b 0 = g x b$ and $q x b 1 = b$. Now we define:

$$\begin{aligned} r : (x : \Gamma)(a : Ax) \rightarrow a \sim g x (f x a) \\ r x a i \triangleq \rho 1 \langle x, id \rangle (i = 0 \vee i = 1) ((\lambda _ \rightarrow p x a) \cup (\lambda _ \rightarrow q x (f x a))) (f x a) \\ s : (x : \Gamma)(b : Bx) \rightarrow b \sim f x (g x b) \\ s x b i \triangleq \rho 0 \langle x, id \rangle (i = 0 \vee i = 1) ((\lambda _ \rightarrow q x b) \cup (\lambda _ \rightarrow p x (g x b))) (g x b) \end{aligned}$$

Hence f and g are quasi-inverses; from which we can construct an equivalence structure [62, Chapter 4]. \square

We now wish to show that, conversely, one can convert equivalences to paths between fibrations. To do so we use the glueing construction given in Section 5.4.

Theorem 5.5.3 (Converting equivalences to paths). *There is a function*

$$\begin{aligned} \text{equivToPath} : \{\Gamma : \mathcal{U}\}\{A B : \text{Fib } \Gamma\} & & (5.50) \\ (f : (x : \Gamma) \rightarrow \text{fst } Ax \rightarrow \text{fst } Bx) \rightarrow (\text{Equiv } f) \rightarrow A \sim_{\mathcal{U}} B \end{aligned}$$

Proof. Given $\Gamma : \mathcal{U}, (A, \alpha), (B, \beta) : \text{Fib } \Gamma, f : (x : \Gamma) \rightarrow \text{fst } Ax \rightarrow \text{fst } Bx$ and $eq : \text{Equiv } f$, define the following:

$$\begin{aligned} \Phi & : \Gamma \times \mathbf{I} \rightarrow \text{Cof} \\ \Phi(x, i) & \triangleq (i = 0) \vee (i = 1) \\ C & : (\Gamma \times \mathbf{I}) | \Phi \rightarrow \mathcal{U} \\ C((x, i), u) & \triangleq ((\lambda_- : [i = 0] \rightarrow Ax) \cup (\lambda_- : [i = 1] \rightarrow Bx)) u \\ f' & : ((x, i) : \Gamma \times \mathbf{I})(u : [\Phi(x, i)]) \rightarrow C((x, i), u) \rightarrow Bx \\ f'(x, i) & \triangleq (\lambda_- : [i = 0] \rightarrow fx) \cup (\lambda_- : [i = 1] \rightarrow id) \end{aligned}$$

Now let $P \triangleq \text{SGLue } \Phi C (\lambda(x, -) \rightarrow Bx) f'$ and observe that $P(x, 0) = Ax$ and $P(x, 1) = Bx$ by the strictness property of SGLue .

Now we show that P has a fibration structure. First, we observe that C has a fibration structure, using axiom ax_1 . In order to define $\gamma : \text{isFib } C$ we take:

$$e : \{0, 1\}, \quad p : \mathbf{I} \rightarrow (\Gamma \times \mathbf{I}) | \Phi, \quad \varphi : \text{Cof}, \quad f : [\varphi] \rightarrow \Pi(C \circ p), \quad c : C(p e)$$

and aim to define $\gamma e p \varphi f c : C(p \bar{e})$. First, define the predicate $\text{pZero} : \mathbf{I} \rightarrow \Omega$ by:

$$\text{pZero } i \triangleq (\text{snd}(\text{fst}(p i)) = 0)$$

Observe that since $p : \mathbf{I} \rightarrow (\Gamma \times \mathbf{I}) | \Phi$ we know, from $\text{snd} \circ p$, that:

$$(\forall i : \mathbf{I}) \text{snd}(\text{fst}(p i)) = 0 \vee \text{snd}(\text{fst}(p i)) = 1 \tag{5.51}$$

and so, using ax_2 , we have $(\forall i : \mathbf{I}) (\text{pZero } i \vee \neg(\text{pZero } i))$ and so using ax_1 we get

$$((\forall i : \mathbf{I}) \text{pZero } i) \vee ((\forall i : \mathbf{I}) \neg(\text{pZero } i))$$

In case $(\forall i : \mathbf{I}) \text{pZero } i$, we deduce that $C \circ p = A \circ \text{fst} \circ \text{fst} \circ p$ and define:

$$\gamma e p \varphi f c \triangleq \alpha e (\text{fst} \circ \text{fst} \circ p) \varphi f c$$

Otherwise, in case $(\forall i : \mathbf{I}) \neg(\text{pZero } i)$, we use ax_2 to deduce $(\forall i : \mathbf{I}) \text{snd}(\text{fst}(p i)) = 1$,

hence $C \circ p = B \circ \text{fst} \circ \text{fst} \circ p$, and so define:

$$\gamma e p \varphi f c \triangleq \beta e (\text{fst} \circ \text{fst} \circ p) \varphi f c$$

Therefore C has a fibration structure. Next we show that f' is an equivalence for every $x : \Gamma$, $i : \mathbf{I}$ and $u : [\Phi i]$. First note that the identity function $id : Bx \rightarrow Bx$ is always an equivalence; let $idEq : \text{Equiv } id$ be a proof of this fact. Define $eq' : (x : \Gamma)(i : \mathbf{I})(u : [\Phi i]) \rightarrow \text{Equiv } (f' x i u)$ by:

$$eq' x i \triangleq (\lambda u : [i = 0] \rightarrow eq) \cup (\lambda u : [i = 1] \rightarrow idEq)$$

Hence, by Corollary 5.4.11, we get a fibration structure, $\rho : \text{isFib } P$, such that

$$(P, \rho)[\iota : (\Gamma \times \mathbf{I})|\Phi \mapsto \Gamma \times \mathbf{I}] = (C, \gamma)$$

and hence $(P, \rho)[\langle id, 0 \rangle] = (P, \rho)[\iota \circ \langle id, 0, * \rangle] = (C, \gamma)[\langle id, 0, * \rangle] = (A, \alpha)$ and similarly $(P, \rho)[\langle id, 1 \rangle] = (B, \beta)$. Therefore we define $\text{equivToPath } f e \triangleq (P, \rho)$. \square

The univalence axiom (Definition 2.1.4) becomes here the property that the map pathToEquiv is itself an equivalence. Theorem 2.1.9 tells us that this is actually the same as having a map from equivalences to paths, which we have in equivToPath , such that coercion along $\text{equivToPath } f e$ is path equal to f . This property does indeed hold:

Theorem 5.5.4 (Univalence for \sim_u). *Define*

$$\begin{aligned} \text{coerce} &: \{\Gamma : \mathcal{U}\} \{A B : \text{Fib } \Gamma\} (P : A \sim_u B) (x : \Gamma) \rightarrow \text{fst } Ax \rightarrow \text{fst } Bx \\ \text{coerce}(P, \rho) &\triangleq \text{fst}(\text{pathToEquiv}(P, \rho)) \end{aligned}$$

Given $\Gamma : \mathcal{U}$, $(A, \alpha), (B, \beta) : \text{Fib } \Gamma$, $f : (x : \Gamma) \rightarrow Ax \rightarrow Bx$ and $eq : \text{Equiv } f$, there exists a path $f \sim \text{coerce}(\text{equivToPath } f e)$.

Proof. Let $(P, \rho) \triangleq \text{equivToPath } f e$. Unfolding the definition of coerce we have

$$\text{coerce}(P, \rho) x a = \rho 0 \langle x, id \rangle \perp \text{elim}_0 a$$

Recalling that ρ is the composition structure for SGLue we can calculate $\rho 0 \langle x, id \rangle \perp \text{elim}_0 a$. We first move across the isomorphism with Glue so that a becomes $(\lambda_- \rightarrow a, f x a)$. We now trace the algorithm for composition in Glue : we begin by composing in Bx to get $b'_1 \triangleq \beta 0 (\lambda_- \rightarrow x) \perp \text{elim}_0 (f x a)$. Next, we use the equivalence structure to derive $a_1 : (u : [1 = 0 \vee 1 = 1]) \rightarrow Bx$ and $p_b : (u : [1 = 0 \vee 1 = 1]) \rightarrow id (a_1 u) \sim b'_1$, where in particular, a_1 is given by $a_1 u \triangleq \beta 1 (\lambda_- \rightarrow x) \perp \text{elim}_0 b'_1$. We then perform the final step of the composition, which is to compose from b'_1 in Bx to get b_1 . This leaves us with the

result of composing in \mathbf{Glue} as (a_1, b_1) . When transferring back across the isomorphism we simply take the first component of the pair, namely a_1 , but now regarded as a total element to get the final result of composing in \mathbf{SGlue} as $\beta \mathbf{1}(\lambda_- \rightarrow x) \perp \mathbf{elim}_\emptyset b'_1$. So, in summary we have:

$$\begin{aligned} \mathbf{coerce}(P, \rho) x a &= \rho \mathbf{0} \langle x, id \rangle \perp \mathbf{elim}_\emptyset a \\ &= \beta \mathbf{1}(\lambda_- \rightarrow x) \perp \mathbf{elim}_\emptyset b'_1 \\ &= \beta \mathbf{1}(\lambda_- \rightarrow x) \perp \mathbf{elim}_\emptyset (\beta \mathbf{0}(\lambda_- \rightarrow x) \perp \mathbf{elim}_\emptyset (f x a)) \end{aligned}$$

Since this is simply two trivial compositions applied to $f x a$ we can use filling to derive a path $f x a \sim \mathbf{coerce}(P, \rho) x a$. Now, two applications of function extensionality (Remark 5.3.16) yields a path $f \sim \mathbf{coerce}(P, \rho)$ as required. \square

5.6 Satisfying the axioms

Working informally in a constructive set theory, the authors of [18] give a model of their type theory using the topos $\widehat{\square} = \mathbf{Set}^{\square^{\text{op}}}$ of contravariant set-valued functors on a particular small category \square that they call the *category of cubes*, as described in Section 3.3. In this section we present sufficient conditions on an arbitrary small category \mathbf{C} for the topos $\widehat{\mathbf{C}} = \mathbf{Set}^{\mathbf{C}^{\text{op}}}$ of set-valued presheaves (within Intuitionistic ZF set theory [1, Section 3.2], say) to have an interval object and subobject of cofibrant propositions satisfying the axioms in Figure 5.4. We show that the category of cubes is an instance of such a \mathbf{C} (in more than one way); and we also show that in the presence of the Law of Excluded Middle, so is the simplex category $\mathbf{\Delta}$.

5.6.1 The interval object (\mathbf{ax}_1 – \mathbf{ax}_4)

Returning to the general case of a small category \mathbf{C} with its associated presheaf topos $\widehat{\mathbf{C}}$, if we take the interval object $\mathbf{I} \in \widehat{\mathbf{C}}$ to be a representable functor $y_i \triangleq \mathbf{C}(-, i)$ for some object $i \in \mathbf{C}$, then the following theorem gives a useful criterion for such an interval object to satisfy axiom \mathbf{ax}_1 .

Theorem 5.6.1. *In a presheaf topos $\widehat{\mathbf{C}}$, a representable functor $\mathbf{I} = y_i$ satisfies axiom \mathbf{ax}_1 if \mathbf{C} is a cosifted category, that is, if finite products in \mathbf{Set} commute with colimits over \mathbf{C}^{op} [26].*

Proof. \mathbf{C} is cosifted if the colimit functor $\text{colim}_{\mathbf{C}^{\text{op}}} : \widehat{\mathbf{C}} \rightarrow \mathbf{Set}$ preserves finite products. Recall that $\text{colim}_{\mathbf{C}^{\text{op}}} : \widehat{\mathbf{C}} \rightarrow \mathbf{Set}$ is left adjoint to the constant presheaf functor $\Delta : \mathbf{Set} \rightarrow \widehat{\mathbf{C}}$ and (hence) that for any $c \in \mathbf{C}$ it is the case that $\text{colim}_{\mathbf{C}^{\text{op}}} y_c \cong 1$. So when \mathbf{C} is cosifted

we have for any $c \in \mathbf{C}$

$$\begin{aligned} \widehat{\mathbf{C}}(y_c \times y_i, \Delta\{0, 1\}) &\cong \mathbf{Set}(\operatorname{colim}_{\mathbf{C}^{\text{op}}} (y_c \times y_i), \{0, 1\}) \cong \\ &\mathbf{Set}(\operatorname{colim}_{\mathbf{C}^{\text{op}}} y_c \times \operatorname{colim}_{\mathbf{C}^{\text{op}}} y_i, \{0, 1\}) \cong \mathbf{Set}(1 \times 1, \{0, 1\}) \cong \{0, 1\} \end{aligned}$$

Since decidable subobjects in $\widehat{\mathbf{C}}$ are classified by $1 + 1 = \Delta\{0, 1\}$, this means that the only two decidable subobjects of $y_c \times y_i$ are the smallest and the greatest subobjects. Since this is so for all $c \in \mathbf{C}$, it follows that $\mathbf{I} = y_i$ satisfies \mathbf{ax}_1 . \square

A more elementary characterisation of cosiftedness is that \mathbf{C} is inhabited and for every pair of objects $c, c' \in \mathbf{C}$ the category of spans $c \leftarrow \cdot \rightarrow c'$ is a connected category [2, Theorem 2.15]. Any category with finite products trivially has this property. This is the case for the category \square of cubes defined above and thus the interval in the model of [18] (where $\mathbf{C} = \square$ and \mathbf{i} is the generic De Morgan algebra) satisfies \mathbf{ax}_1 . A relevant example of a category that does not have finite products, but which is nevertheless cosifted is $\mathbf{\Delta}$, the category of inhabited finite linearly ordered sets $[0 < 1 < \dots < n]$, for which $\widehat{\mathbf{C}}$ is the category of *simplicial sets*, widely used in homotopy theory [28]. Thus the natural candidate for an interval in $\widehat{\mathbf{\Delta}}$, namely y_i when \mathbf{i} is the 1-simplex $[0 < 1]$, satisfies \mathbf{ax}_1 .

In addition to \mathbf{ax}_1 , the other axioms in Figure 5.4 concerning the interval say that \mathbf{I} is a non-trivial (\mathbf{ax}_2) model of the algebraic theory given by \mathbf{ax}_3 and \mathbf{ax}_4 , which we call *connection algebra*. (See also Definition 1.7 of [27], which considers a similar notion in a more abstract setting.) For cubical sets, the Yoneda embedding $y : \square \rightarrow \widehat{\square}$ sends the generic De Morgan algebra in \square to a De Morgan algebra in $\widehat{\square}$. This is a non-trivial connection algebra: the constants are the least and greatest elements and the binary operations are meet and join. An obvious variation on the theme of [18] would be to replace \square by the Lawvere theory for connection algebras. Note also that the 1-simplex in $\widehat{\mathbf{\Delta}}$ is a non-trivial connection algebra, the constants being its two end points and the binary operations being induced by the order-preserving binary operations of minimum and maximum on $[0 < 1]$.

5.6.2 Cofibrant propositions (\mathbf{ax}_5 – \mathbf{ax}_8) and the strictness axiom (\mathbf{ax}_9)

In a topos with an interval object, there are many candidates for a subobject $\mathbf{Cof} \hookrightarrow \Omega$ satisfying axioms \mathbf{ax}_5 – \mathbf{ax}_8 in Figure 5.4. At one extreme, one could just take \mathbf{Cof} to be the whole of Ω . At the opposite extreme, one could take the subobject (internally) inductively defined by the requirements that it contains the propositions $i = 0$ and $i = 1$ (for all $i : \mathbf{I}$) and is closed under binary disjunction, dependent and \mathbf{I} -indexed conjunction, thereby obtaining the smallest \mathbf{Cof} satisfying \mathbf{ax}_5 – \mathbf{ax}_8 . However, cofibrant propositions also have

to satisfy the strictness axiom \mathbf{ax}_9 and we consider that next.

Given a presheaf topos $\widehat{\mathbf{C}}$, we work in the CwF associated with $\widehat{\mathbf{C}}$ as in Definition 3.2.1. In particular, families over a presheaf $\Gamma \in \widehat{\mathbf{C}}$ are given by functors $(\int \Gamma)^{\text{op}} \rightarrow \mathbf{Set}$, where $\int \Gamma$ is the usual category of elements of Γ , with $\text{obj}(\int \Gamma) = (c \in \text{obj } \mathbf{C}) \times \Gamma c$ and $(\int \Gamma)((c, x), (d, y)) = \{f \in \mathbf{C}(c, d) \mid \Gamma f y = x\}$. If \mathcal{S} is a Grothendieck universe in the ambient set theory, then its Hofmann-Streicher lifting [31] to a universe \mathcal{U} in that CwF satisfies that the morphisms $\Gamma \rightarrow \mathcal{U}$ in $\widehat{\mathbf{C}}$ name the families $(\int \Gamma)^{\text{op}} \rightarrow \mathcal{S}$ taking values in $\mathcal{S} \subseteq \mathbf{Set}$.

Definition 5.6.2 (Ω_{dec}). The subobject classifier Ω in a presheaf topos $\widehat{\mathbf{C}}$ maps each $c \in \mathbf{C}$ to the set $\Omega(c)$ of *sieves* on c , that is, pre-composition closed subsets $S \subseteq \text{obj}(\mathbf{C}/c)$. Let $\Omega_{\text{dec}} \rightarrow \Omega$ be the subpresheaf whose value at each $c \in \text{obj } \mathbf{C}$ is the subset of $\Omega(c)$ consisting of those sieves S that are decidable subsets of $\text{obj}(\mathbf{C}/c)$.

Of course if the ambient set theory satisfies the Law of Excluded Middle, then $\Omega_{\text{dec}} = \Omega$. In general Ω_{dec} classifies monomorphisms $\alpha : F \rightarrow G$ in $\widehat{\mathbf{C}}$ such that for all $c \in \text{obj } \mathbf{C}$ the (injective) function $\alpha_c : F c \rightarrow G c$ has decidable image.

Theorem 5.6.3. *Interpreting the universe \mathcal{U} as the Hofmann-Streicher lifting [31] of a Grothendieck universe in \mathbf{Set} , a subobject $\mathbf{Cof} \rightarrow \Omega$ in a presheaf topos $\widehat{\mathbf{C}}$ satisfies the strictness axiom \mathbf{ax}_9 if it is contained in $\Omega_{\text{dec}} \rightarrow \Omega$.*

Proof. For each $c \in \text{obj } \mathbf{C}$, suppose we are given $S \in \Omega_{\text{dec}}(c)$. Thus S is a sieve on c and for each $c' \in \text{obj } \mathbf{C}$ and \mathbf{C} -morphism $f : c' \rightarrow c$, it is decidable whether or not $f \in S$. We can also regard S as a subpresheaf $S \hookrightarrow y_c$.

Suppose that we have families $A : (\int S)^{\text{op}} \rightarrow \mathcal{S}$, $B : (\int y_c)^{\text{op}} \rightarrow \mathcal{S}$ and a natural isomorphism s between A and the restriction of B along $S \hookrightarrow y_c$. For each \mathbf{C} -morphism $\cdot \xrightarrow{f} c$, using the decidability of S , we can define bijections $s'(f) : B'(f) \cong B(f)$ given by

$$B'(f) \triangleq \begin{cases} A(f) & \text{if } f \in S \\ B(f) & \text{if } \neg(f \in S) \end{cases} \quad \text{and} \quad s'(f) \triangleq \begin{cases} s(f) & \text{if } f \in S \\ f & \text{if } \neg(f \in S) \end{cases}$$

(compare this with Definition 15 in [18]). We make B' into a functor $(\int y_c)^{\text{op}} \rightarrow \mathcal{S}$ by transferring the functorial action of B across these bijections. Having done that, s' becomes a natural isomorphism $B' \cong B$; and by definition its restriction along $S \hookrightarrow y_c$ is s . \square

Corollary 5.6.4. *Let \mathbf{C} be a small category with finite products containing an object \mathbf{i} with the structure of a non-trivial connection algebra $0, 1 : 1 \rightarrow \mathbf{i}$, $\sqcap, \sqcup : \mathbf{i} \times \mathbf{i} \rightarrow \mathbf{i}$ (cf. Figure 5.4). Suppose that for each object $c \in \mathbf{C}$, the set $\mathbf{C}(c, \mathbf{i})$ has decidable equality. Then the topos of presheaves $\widehat{\mathbf{C}}$ satisfies all the axioms in Figure 5.4 if we take the interval \mathbf{I} to be $y_{\mathbf{i}}$ and \mathbf{Cof} to be Ω_{dec} .*

Proof. We already noted in section 5.6.1 that axioms \mathbf{ax}_1 – \mathbf{ax}_4 are satisfied by this choice of \mathbf{I} . The decidability of each set $\mathbf{C}(c, \mathbf{i})$ implies that the subobjects $\{0\} \rightarrow \mathbf{I}$ and $\{1\} \rightarrow \mathbf{I}$ in $\widehat{\mathbf{C}}$ factor through $\Omega_{\text{dec}} \rightarrow \Omega$ and hence that axiom \mathbf{ax}_5 is satisfied when $\mathbf{Cof} = \Omega_{\text{dec}}$. Note that this choice of \mathbf{Cof} automatically satisfies \mathbf{ax}_6 and \mathbf{ax}_7 ; and it satisfies axiom \mathbf{ax}_9 by Theorem 5.6.3. So it just remains to check that axiom \mathbf{ax}_8 is satisfied. We saw in Lemma 5.1.5(ii) that this axiom is equivalent to requiring cofibrations to be closed under exponentiation by \mathbf{I} . In this case cofibrations are the monomorphisms $\alpha : F \rightarrow G$ classified by Ω_{dec} and we noted after Definition 5.6.2 that they are characterized by the fact that each function $\alpha_c \in \mathbf{Set}(F c, G c)$ has decidable image. Closure of these monomorphisms under exponentiating by \mathbf{I} follows from the fact that \mathbf{C} has finite products and that $\mathbf{I} = y_1$ is representable; for then $\mathbf{I} \rightarrow (-)$ is isomorphic to the functor $\widehat{\mathbf{C}} \rightarrow \widehat{\mathbf{C}}$ induced by precomposition with $(-) \times \mathbf{i} : \mathbf{C} \rightarrow \mathbf{C}$, which clearly preserves the componentwise decidable image property. \square

The above argument for axiom \mathbf{ax}_8 does not apply to $\mathbf{\Delta}$, since it does not have finite products; we do not know whether that axiom is satisfied by constructive simplicial sets. However, in the presence of the Law of Excluded Middle (LEM), $\Omega_{\text{dec}} = \Omega$ and we have:

Corollary 5.6.5 (Classical simplicial sets). *Assuming LEM holds in the set-theoretic metatheory, then the presheaf topos of simplicial sets $\widehat{\mathbf{\Delta}}$ satisfies the axioms in Figure 5.4 if we take \mathbf{I} to be the representable presheaf on the 1-simplex and \mathbf{Cof} to be the whole of Ω .*

Proof. We already noted in section 5.6.1 that axioms \mathbf{ax}_1 – \mathbf{ax}_4 are satisfied by this choice of \mathbf{I} . If $\mathbf{Cof} = \Omega$ (that is, $\mathbf{cof} = \lambda_- \rightarrow \top$), then axioms \mathbf{ax}_5 – \mathbf{ax}_8 hold trivially. Furthermore, if LEM holds, then $\Omega_{\text{dec}} = \Omega$ and so axiom \mathbf{ax}_9 holds by Theorem 5.6.3. \square

Remark 5.6.6. *As a partial converse of Theorem 5.6.3, we have that if \mathbf{ax}_9 is satisfied by the Hofmann-Streicher universe in the CwF associated with $\widehat{\mathbf{C}}$, then each cofibrant mono $\alpha : F \rightarrow G$ has component functions $\alpha_c \in \mathbf{Set}(F c, G c)$ ($c \in \text{obj } \mathbf{C}$) whose images are $\neg\neg$ -closed subsets of $G c$. To see this we can apply an argument due to Andrew Swan [private communication] that relies upon the fact that in the ambient set theory one has*

$$(X = \emptyset) = \forall x \in X. \perp = \neg\neg(\forall x \in X. \perp) = \neg\neg(X = \emptyset) \quad (5.52)$$

For suppose given $c \in \text{obj } \mathbf{C}$ and $S \in \mathbf{Cof}(c)$. We have to use axiom \mathbf{ax}_9 to show that S is a $\neg\neg$ -closed subset of $\text{obj}(\mathbf{C}/c)$. Let $A : (\int S)^{\text{op}} \rightarrow \mathcal{S}$ be the constant functor mapping each (c', f) to $\{\emptyset\}$; and let $B : (\int y_c)^{\text{op}} \rightarrow \mathcal{S}$ map each (c', f) to $\{\{\emptyset\}, \{\emptyset \mid f \in S\}\}$ (which does extend to a functor, because S is a sieve). The restriction of B along $S \hookrightarrow y_c$ is isomorphic to A and so by \mathbf{ax}_9 there is some $B' : (\int y_c)^{\text{op}} \rightarrow \mathcal{S}$ whose restriction along $S \hookrightarrow y_c$ is equal to A and some isomorphism $s' : B' \cong B$. For any $(c', f) \in \text{obj}(\int y_c)$, suppose $X \in B'(c', f)$; then $f \in S \Rightarrow X = \emptyset$, hence $\neg\neg(f \in S) \Rightarrow \neg\neg(X = \emptyset)$ and

therefore by (5.52), $\neg\neg(f \in S) \Rightarrow (X = \emptyset)$. Therefore $\neg\neg(f \in S) \Rightarrow B'(c', f) = \{\emptyset\} \Rightarrow B(c', f) \cong \{\emptyset\} \Rightarrow f \in S$. So S is indeed a $\neg\neg$ -closed subset of $\text{obj}(\mathbf{C}/c)$.

Note that this result implies that it is not possible to take \mathbf{Cof} to be the whole of Ω and satisfy \mathbf{ax}_9 unless the ambient set theory satisfies LEM.

Since in a constructive setting equality in free De Morgan algebras is decidable, it follows that Corollary 5.6.4 gives a model of our axioms when $\mathbf{C} = \square$, the category of cubes (Definition 3.3.1). However, this uses a different choice of cofibrancy from the one in [18, Section 4.1]. In the remainder of this section we check that the CCHM notion of fibrancy satisfies our axioms.

Definition 5.6.7 (Cofibrant propositions in [18]). For each object of the category of cubes $I \in \text{obj}\square$, Cohen et al. define the *face lattice* $\mathbb{F}(I)$ to be the distributive lattice generated by symbols $(i = 0)$ and $(i = 1)$ for each $i \in I$, subject to the equations $(i = 0) \wedge (i = 1) = \perp$.

Since the free De Morgan algebra $\mathbf{dM}(I)$ is freely generated as a distributive lattice by symbols i and $1-i$ (as i ranges over the finite set I), we can regard $\mathbb{F}(I)$ as a quotient lattice of $\mathbf{dM}(I)$ via the function mapping i to $(i = 1)$ and $1-i$ to $(i = 0)$; we write $q_I : \mathbf{dM}(I) \rightarrow \mathbb{F}(I)$ for the quotient function.² It is not hard to see that for each $f \in \mathbf{C}(J, I)$, the corresponding De Morgan algebra homomorphism $\mathbf{dM}(I) \rightarrow \mathbf{dM}(J)$ (which we also write as f) induces a lattice homomorphism between the face lattices:

$$\begin{array}{ccc} \mathbf{dM}(I) & \xrightarrow{f} & \mathbf{dM}(J) \\ q_I \downarrow & & \downarrow q_J \\ \mathbb{F}(I) & \xrightarrow{\mathbb{F}f} & \mathbb{F}(J) \end{array}$$

This makes \mathbb{F} into an object of $\widehat{\square}$ and there is a monomorphism $m : \mathbb{F} \rightarrow \Omega$ whose component at $I \in \text{obj}\square$ sends each $\varphi \in \mathbb{F}(I)$ to the sieve

$$m_I(\varphi) = \{\cdot \xrightarrow{f} I \mid \mathbb{F}f \varphi = \top\} \quad (5.53)$$

We now take $\mathbf{Cof} \rightarrow \Omega$ in $\widehat{\square}$ to be the subobject given by this monomorphism $m : \mathbb{F} \rightarrow \Omega$. Axioms \mathbf{ax}_1 – \mathbf{ax}_4 hold without change from Corollary 5.6.4; and axioms \mathbf{ax}_5 – \mathbf{ax}_7 follow from the definition of the face lattices $\mathbb{F}(I)$. So it just remains to check axioms \mathbf{ax}_8 and \mathbf{ax}_9 .

Recall that the interval object in $\widehat{\square}$ is the representable presheaf $\mathbf{I} = y_{\{i\}}$ on a one-element subset $\{i\} \in \text{obj}\square$. For an arbitrary object $I \in \text{obj}\square$, with n distinct elements $i_1, \dots, i_n \in \mathbb{D}$ say, the representable y_I is isomorphic to an n -fold product \mathbf{I}^n in $\widehat{\square}$. Thus

²If $r \in \mathbf{dM}(I)$, then [18] uses the notation $(r = 1)$ for $q_I(r)$.

the sieve (5.53) corresponds to a subobject of the n -cube \mathbf{I}^n . Indeed, each φ in the face lattice $\mathbb{F}(I)$ is a finite join of irreducible elements; and each of those irreducibles is a finite conjunction of atomic conditions of the form $i = 0$ or $j = 1$. The corresponding subobject of \mathbf{I}^n is a finite union of *faces*, that is, subobjects of \mathbf{I}^n obtained by setting some co-ordinates to either 0 or 1. This disjunctive normal form for elements of $\mathbb{F}(I)$ entails that its equality is decidable and hence this \mathbf{Cof} is contained in Ω_{dec} and axiom \mathbf{ax}_9 is satisfied (Theorem 5.6.3). Finally, axiom \mathbf{ax}_8 follows from the fact that pullback of cofibrations along a projection $\pi_1 : \mathbf{I}^n \times \mathbf{I} \rightarrow \mathbf{I}^n$ has a (stable) right adjoint, or equivalently that the lattice morphisms $\mathbb{F} \pi_1 : \mathbb{F}(I) \rightarrow \mathbb{F}(I \cup \{i\})$ (for any $I \in \text{obj } \square$ and $i \in \mathbb{D} - I$) have (stable) right adjoints. This is the quantifier elimination result for face lattices; see [18, Lemma 2].

5.7 Related work

The work presented in this chapter was inspired by the suggestion of Coquand [19] that some of the constructions developed in [18] might be better understood using the internal logic of a topos. We have shown how to express Cohen, Coquand, Huber and Mörtberg’s notion of fibration in the internal type theory of a topos. The use of internal language permits an appealingly simple description (Definition 5.3.3) compared, for example, with the more abstract category-theoretic methods of weak factorization systems and model categories, which have been used for the same purpose by Gambino and Sattler [27, Section 3]. Birkedal et al. [12] develop guarded cubical type theory with a semantics based on an axiomatic version of [18] within the internal logic of a presheaf topos.

Within the framework of a topos equipped with an interval-like object, we found that quite a simple collection of axioms (Figure 5.4) suffices for this to model Martin-Löf type theory with intensional identity types satisfying a weak form of univalence. In particular, only a simple connection algebra, rather than a De Morgan algebra structure, is needed on the interval. Furthermore, the collection of propositions suitable for uniform Kan filling is not tightly constrained and can be chosen in various ways. In Section 5.6 we only considered how presheaf categories can satisfy our axioms. It might be interesting to consider models in general Grothendieck toposes (where presheaves are restricted to be sheaves for a given notion of covering), particularly *gros toposes* such as Johnstone’s topological topos [36]; this allows the interval object to be (a representable sheaf corresponding to) the usual topological interval and hence for the model of type theory to have a rather direct connection with classical homotopy types of spaces. However, although the Hofmann-Streicher [31] universe construction (the basis for the construction in Section 8.2 of [18] of a fibrant universe satisfying the full univalence axiom) can be extended from presheaf to sheaf toposes via the use of sheafification [59, Section 3], it seems that sheafification does not interact well with the CCHM notion of fibration. In another direction, recent work of

Frumin and Van Den Berg [25] makes use of our elementary, axiomatic approach using a non-Grothendieck topos, namely the effective topos [35]. Finally, Uemura has also applied this approach to show the independence of the propositional resizing axiom in cubical type theory [61].

Chapter 6

Decomposing the univalence axiom

In the previous chapter we gave an axiomatic presentation of the model of Cohen et al. [18] using the internal language of an elementary topos. While the approach of using the internal language is new, the structure of many of the proofs, such as the fact that fibrations are closed under type formers or the use of glueing to develop univalence, followed the same structure as in the original work of Cohen et al. In this chapter we will use the internal language to present new work, namely: an alternative proof of the univalence axiom in any topos satisfying our axioms.

We do so by first decomposing univalence into a collection of simpler properties. We explain why all but one of these new properties should be satisfied, not only in a topos satisfying our axioms, but in fact in any presheaf topos with a CwF constructed in the usual way, as per Section 3.2. We then explain how the remaining property is easily validated in any topos satisfying our original axioms. In fact this construction, which we call contraction, is a specialised case of the general glueing construction. However, the definition and the subsequent proofs are a lot simpler than those used for glueing.

6.1 The decomposition

Recall the decomposition of the proper univalence axiom into a naive form along with a computation rule presented in Theorem 2.1.9. In this section we further decompose the univalence axiom into even simpler properties, working in MLTT with function extensionality, but without either UIP or univalence assumed. Note that the proofs given in this section are to be interpreted in the source language which we eventually wish to model, rather than in the internal language of a topos. We show that the univalence axiom for a universe U_i is equivalent to axioms (1) to (5) given in Table 6.1 for the universe U_i .

We begin by decomposing naive univalence, \mathbf{UA}_i , into axioms (1)-(3). These axioms also follow from \mathbf{UA}_i . Recall that we are taking function extensionality as an ambient assumption. First, we recall the definition of singleton types.

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

Table 6.1: ($A, B : U_i, C : A \rightarrow B \rightarrow U_i, a : A, b : B$ and $c : C a b$, for some universe U_i)

Definition 6.1.1 (Singletons). Given a type A and element $a : A$, we can define

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

to be the type of elements of A which are equal to a . It is easily shown by path induction that the type $\mathbf{sing}(a)$ is always contractible in the sense of Definition 2.1.1.

Theorem 6.1.2. *Axioms (1)-(3) for a universe U_i are together logically equivalent to \mathbf{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 $ua : \mathbf{UA}_i$. Given arbitrary types $A, B : U_i$ and an equivalence $(f, e) : A \simeq B$ then for all $a : A$ we know that $\mathbf{sing}(a)$ is contractible, and hence by `contract` we have:

$$1 = \mathbf{sing}(f a) = \sum_{b:B} f a = b$$

Further, for any $b : B$, we know that $\mathbf{fib}_f(b)$ is contractible by $e(b) : \mathbf{isContr}(\mathbf{fib}_f(b))$. Hence, by `contract`, we have:

$$1 = \mathbf{fib}_f(b) = \sum_{a:A} f a = b$$

Therefore, we define $ua(f, e) : A = B$ as follows:

$$\begin{aligned}
A &= \sum_{a:A} 1 && \text{by } \mathbf{unit} \\
&= \sum_{a:A} \sum_{b:B} f a = b && \text{by } \mathbf{funext} \text{ and } \mathbf{contract} \text{ on } \mathbf{sing}(f a) \\
&= \sum_{b:B} \sum_{a:A} f a = b && \text{by } \mathbf{flip} \\
&= \sum_{b:B} 1 && \text{by } \mathbf{funext} \text{ and } \mathbf{contract} \text{ on } \mathbf{fib}_f(b) \text{ (contractible by } e(b)) \\
&= B && \text{by } \mathbf{unit}
\end{aligned}$$

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

$$\begin{array}{ccc} A \rightarrow \sum_{a:A} 1 & \sum_{a:A} \sum_{b:B} C \ a \ b \rightarrow \sum_{b:B} \sum_{a:A} C \ a \ b & \\ a \mapsto (a, *) & (a, b, c) \mapsto (b, a, c) & \end{array}$$

are both easily shown to be bi-invertible and hence equivalences, and from the fact that any contractible type is equivalent to 1 [62, Lemma 3.11.3.]. Therefore given $ua : \mathbf{UA}_i$ we simply apply it to these equivalences to get the required equalities (1)-(3). \square

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

Lemma 6.1.3. *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 $\mathbf{coerce}(p \cdot q) = \mathbf{coerce}(q) \circ \mathbf{coerce}(p)$ where $p \cdot q$ is the transitive proof that $A = C$.*

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

Theorem 6.1.4. *Axioms (1)-(5) for a universe U_i are together logically equivalent to $\sum_{ua:\mathbf{UA}_i} \mathbf{UA}\beta_i(ua)$.*

Proof. For the forwards direction we know from Theorem 6.1.2 that axioms (1) to (3) allow us to construct a term $ua : \mathbf{UA}_i$. If, in addition, we assume axioms (4) and (5) then we can show that for all $(f, e) : A \simeq B$ we have $\mathbf{coerce}(ua(f, e)) = f$ as follows.

Since ua was constructed as the concatenation of five equalities then, in light of Lemma 6.1.3, we have that coercing along $ua(f, e)$ is equal to the result of coercing along each stage of the composite equality $ua(f, e)$. We know the result of coercing along `unit` and `flip` from `unit` β and `flip` β respectively. In the case of `contract` we are coercing between two contractible types and can therefore take the coercion to be any function that we like, since all such functions are propositionally equal. Therefore, starting with an arbitrary $a : A$, we can track what happens at each stage of this process like so:

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

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

For the reverse direction we assume that we are given $ua : \mathbf{UA}_i$ and $ua\beta : \mathbf{UA}\beta_i(ua)$. We can now apply Theorem 6.1.2 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` β and `flip` β satisfying axioms (4) and (5) respectively. \square

Corollary 6.1.5. *Axioms (1)-(5) for a universe U_i are together logically equivalent to the proper univalence axiom for U_i .*

Proof. By combining Theorem 2.1.9 and Theorem 6.1.4. \square

6.2 Applications in models of type theory

In this section we discuss one reason why the result given in Corollary 6.1.5 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 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 a topos with the structure described in Section 5.1. As discussed there, this result will therefore apply to the cubical sets model presented in [18], but it should also apply to many similar models of type theory [11, 6, 8, 54, 12].

Note that we are assuming function extensionality. Every model of univalence must satisfy function extensionality [62, 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 object, cf. [62, 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, using the notation of Categories with Families, an equivalence in the type theory will give rise to terms,

$$f \in \mathbf{Ter}(\Gamma \vdash A \rightarrow B) \quad \text{and} \quad g \in \mathbf{Ter}(\Gamma \vdash B \rightarrow A)$$

which are not exact inverses, but rather are inverses modulo the interpretation of proposi-

tional equality, that is, the existence of terms,

$$\alpha \in \mathbf{Ter}(\Gamma \vdash \mathbf{Id}_{A \rightarrow B}(g \circ f, id)) \quad \text{and} \quad \beta \in \mathbf{Ter}(\Gamma \vdash \mathbf{Id}_{B \rightarrow A}(f \circ g, id)).$$

The existence of these do not imply that $g \circ f = id$ or $f \circ g = id$, for example in the cubical sets model where α and β would be paths between the composites and the identity, as described in Section 3.3.3. 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 3.2, and should be true more generally in any model which validates eta-rules for 1 and Σ , since then the maps back and forth will be inverses up to judgemental equality in the type theory and hence their interpretations in the model will be strict inverses.

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 instances 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 a topos with the structure that we axiomatised in Section 5.1. For ease of reference we will refer to this topos as \mathcal{E} . For the rest of this section we return to working in the internal type theory of \mathcal{E} .

6.2.1 Strictification

Recall from Section 5.1.3 that we postulate the existence of a term:

$$\mathbf{ax}_9 : \{\varphi : \mathbf{Cof}\}(A : [\varphi] \rightarrow \mathcal{U})(B : \mathcal{U})(s : (u : [\varphi]) \rightarrow (A u \cong B)) \rightarrow (B' : \mathcal{U}) \times \{s' : B' \cong B \mid \forall (u : [\varphi]). A u = B' \wedge s u = s'\}$$

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

We now lift this strictification property from objects to fibrations.

Theorem 6.2.1. *Given $\Gamma : \mathcal{U}$ and $\Phi : \Gamma \rightarrow \mathbf{Cof}$, a cofibrant-partial fibration $A : \mathbf{Fib}(\Gamma|\Phi)$ and a total fibration $B : \mathbf{Fib} \Gamma$ with $iso : A \cong B[\iota]$, we can construct a new type and*

isomorphism:

$$A' : \mathbf{Fib}(\Gamma) \quad \text{and} \quad iso' : A' \cong B$$

such that

$$A'[\iota] = A \quad \text{and} \quad iso' \circ \iota = iso$$

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

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

$$A' x \triangleq \mathbf{fst}(\mathbf{ax}_9(A(x, -), B x, iso(x, -)))$$

$$iso' x \triangleq \mathbf{snd}(\mathbf{ax}_9(A(x, -), B x, iso(x, -)))$$

Now consider the equalities that are required to hold. From the properties of \mathbf{ax}_9 we already have that $A' \circ \iota = A$ and $iso' \circ \iota = iso$. Therefore we just need to define a composition structure $\alpha' : \mathbf{isFib} A'$ such that $\alpha'[\iota] = \alpha$.

Since $A' \cong B$ and $\beta : \mathbf{isFib} B$ we can use Lemma 5.3.8 to deduce that A' has a composition structure, which we call α'_{pre} . We then define $\alpha' \triangleq \mathbf{realign}(\Phi, \alpha, \alpha'_{pre})$ using Lemma 5.3.10 to ensure that $\alpha'[\iota] = \alpha$.

□

Note that, since $\mathbf{realign}$ is stable under reindexing, in the sense that $\mathbf{realign}(\Phi, \beta, \alpha)[\gamma] = \mathbf{realign}(\Phi \circ \gamma, \beta[\gamma \times id], \alpha[\gamma])$ for any $\gamma : \Delta \rightarrow \Gamma$, and all other constructions used above are performed fibrewise, we can deduce that the construction given in Theorem 6.2.1 is also stable under reindexing.

6.2.2 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 5.5.1, 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 6.2.2 (Misaligned path equality between fibrations). Define the type of misaligned paths between CCHM fibrations $- \sim_{\cong} - : \{\Gamma : \mathcal{U}\} \rightarrow \mathbf{Fib} \Gamma \rightarrow \mathbf{Fib} \Gamma \rightarrow \mathcal{U}_1$ by

$$A \sim_{\cong} B \triangleq (P : \mathbf{Fib}(\Gamma \times \mathbf{I})) \times (A \cong P[\langle id, 0 \rangle]) \times (B \cong P[\langle id, 1 \rangle])$$

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 6.2.3. Given fibrations $A, B : \mathbf{Fib} \Gamma$ we define a new fibration

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

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

$$\begin{aligned} C &: (\Gamma \times \mathbf{I})|\Phi \rightarrow \mathcal{U} \\ C((x, i), u) &\triangleq ((\lambda_- : [i = 0] \rightarrow Ax) \cup (\lambda_- : [i = 1] \rightarrow Bx)) u \end{aligned}$$

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

To see that C is fibrant we observe that the interval \mathbf{I} is internally connected in the sense of \mathbf{ax}_1 in Figure 5.4. This means that any path $p : \mathbf{I} \rightarrow (\Gamma \times \mathbf{I})|\Phi$ must either factor as $p = \langle p', 0, * \rangle$ or as $p = \langle p', 1, * \rangle$. 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 Theorem 5.5.3 where the family C occurs as an intermediate construction, and where we defined this fibration structure in more detail. Note that this construction is stable under reindexing:

$$A[\gamma] \vee B[\gamma] = (A \vee B)[\gamma \times id \times id]$$

for any $\gamma : \Delta \rightarrow \Gamma$. Finally, observe that for all $A, B : \mathbf{Fib} \Gamma$ we have:

$$(A \vee B)[\langle id, 0, * \rangle] = A \quad \text{and} \quad (A \vee B)[\langle id, 1, * \rangle] = B$$

Definition 6.2.4. Given $\Gamma : \mathcal{U}$, fibrations $A, B : \mathbf{Fib} \Gamma$ and a misaligned path $(D, f_0, f_1) : A \sim_{\cong} B$ we define

$$f_0 \vee f_1 : A \vee B \cong D[\iota]$$

as follows:

$$\begin{aligned} (f_0 \vee f_1)(x, i, u) &: (\mathbf{fst}(A \vee B))(x, i, u) \rightarrow (\mathbf{fst} D)(x, i) \\ (f_0 \vee f_1)(x, i, u) &\triangleq \begin{cases} f_0 x & \text{when } u : [i = 0] \\ f_1 x & \text{when } u : [i = 1] \end{cases} \end{aligned}$$

where $(x, i, u) : (\Gamma \times \mathbf{I})|\Phi$. Observe that for all $\gamma : \Delta \rightarrow \Gamma$ we have,

$$(f_0 \circ \gamma) \vee (f_1 \circ \gamma) = (f_0 \vee f_1) \circ (\gamma \times id \times id)$$

and also that:

$$(f_0 \vee f_1) \circ \langle id, 0, * \rangle = f_0 \quad \text{and} \quad (f_0 \vee f_1) \circ \langle id, 1, * \rangle = f_1$$

We now use these two constructions to show the following result:

Lemma 6.2.5. *There exists a function*

$$\text{improve} : \{\Gamma : \mathcal{U}\} \{A B : \text{Fib } \Gamma\} \rightarrow A \sim_{\cong} B \rightarrow A \sim_{\mathcal{U}} B$$

Proof. Take $\Gamma : \mathcal{U}$, $A, B : \text{Fib } \Gamma$ and $(P, f_0, f_1) : A \sim_{\cong} B$ and observe that $f_0 \vee f_1 : A \vee B \cong P[\iota]$. Therefore we can use Theorem 6.2.1 to strictify P in order to get $P' : \text{Fib}(\Gamma \times \mathbf{I})$ such that $P'[\iota] = A \vee B$, where ι is the restriction $(\Gamma \times \mathbf{I})|_{\Phi} \rightarrow \Gamma \times \mathbf{I}$. Now consider reindexing P' along $\langle id, 0 \rangle : \Gamma \rightarrow \Gamma \times \mathbf{I}$ we get:

$$P'[\langle id, 0 \rangle] = P'[\iota \circ \langle id, 0, * \rangle] = P'[\iota][\langle id, 0, * \rangle] = (A \vee B)[\langle id, 0, * \rangle] = A$$

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

$$\text{improve}(P[\gamma \times id], f_0 \circ \gamma, f_1 \circ \gamma) = \text{improve}(P, f_0, f_1)[\gamma \times id]$$

for any $\gamma : \Delta \rightarrow \Gamma$.

□

6.2.3 Function extensionality

As discussed previously, function extensionality holds straightforwardly in any type theory which includes an interval object/type with certain computational properties, cf. [62, Lemma 6.3.2]. See Remark 5.3.16 or [18, Section 3.2] for a proof in the case of cubical type theory.

6.2.4 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 $A, B : \text{Fib } \Gamma$.

Theorem 6.2.6 (Isovalence). *Given fibrations $A, B : \text{Fib } \Gamma$ with $f : A \cong B$ we can construct a path $\text{isopath}(f) : A \sim_{\mathcal{U}} B$.*

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

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

as required. Note that, in this case, `improve` will in fact only improve $B[\mathbf{fst}]$ at 0, since at 1 we improve along the identity, which does nothing. Note that, for any $\gamma : \Delta \rightarrow \Gamma$ we have:

$$\begin{aligned} \mathbf{isopath} (f \circ \gamma) &= \mathbf{improve}(B[\gamma][\mathbf{fst}], f \circ \gamma, id_{B[\gamma]}) \\ &= \mathbf{improve}(B[\mathbf{fst}][\gamma \times id], f \circ \gamma, id_B \circ \gamma) \\ &= \mathbf{improve}(B[\mathbf{fst}], f, id_B)[\gamma \times id] \\ &= (\mathbf{isopath} f)[\gamma \times id] \end{aligned}$$

and hence `isopath` is stable under substitution. \square

Corollary 6.2.7. *Axioms (1) and (2) hold in the CwF of fibrant types in \mathcal{E} .*

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 5.1.6. 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. \square

We have seen that we can easily satisfy properties (1) and (2) using our axioms. 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 6.2.8. *Given fibrations $(A, \alpha), (B, \beta) : \mathbf{Fib} \Gamma$ with $f : A \cong B$, coercing along `isopath` f is (propositionally) equal to applying f .*

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

$$iso(x, i) = \mathbf{snd}(\mathbf{ax}_9((A, \beta) \vee (B, \beta)(x, i, -), B x, (f \vee id)(x, i, -))) : P x \cong B x$$

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

$$iso(x, 0) = (f \vee id)(x, 0) = f x$$

and

$$iso(x, 1) = (f \vee id)(x, 1) = id$$

Also, observe that,

$$\begin{aligned}
\forall i.(i = 0 \vee i = 1) &\Rightarrow (\forall i.i = 0) \vee (\forall i.i = 1) && \text{by } \mathbf{ax}_1 \\
&\Rightarrow 1 = 0 \vee 0 = 1 && \text{by instantiating the } \forall\text{s} \\
&\Rightarrow \perp && \text{by } \mathbf{ax}_2
\end{aligned}$$

and hence $\forall i.(i = 0 \vee i = 1) = \perp$. Now calculate:

$$\begin{aligned}
&\mathbf{coerce} (\mathbf{isopath} f) x a_0 \\
&= \rho_0 \langle x, id \rangle \perp \mathbf{elim}_\emptyset a_0 && \text{by unfolding definitions}^1 \\
&= \rho_{pre} 0 \langle x, id \rangle (\forall i.(i = 0 \vee i = 1)) q a_0 && \text{by Lemma 5.3.10 (for some } q\text{)} \\
&= \rho_{pre} 0 \langle x, id \rangle \perp \mathbf{elim}_\emptyset a_0 && \text{by the argument above} \\
&= iso^{-1}(x, 1) (\beta 0 \langle x, id \rangle \perp \mathbf{elim}_\emptyset (iso(x, 0) a_0)) && \text{by Lemma 5.3.8} \\
&= \beta 0 \langle x, id \rangle \perp \mathbf{elim}_\emptyset (iso(x, 0) a_0) && \text{since } iso(x, 1) = id \\
&= \beta 0 \langle x, id \rangle \perp \mathbf{elim}_\emptyset (f x a_0) && \text{since } iso(x, 0) = f x
\end{aligned}$$

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 $\mathbf{coerce} (\mathbf{isopath} f) x a_0$ like so:

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

Therefore, coercing along $\mathbf{isopath} f$ is always propositionally equal to applying f . \square

Corollary 6.2.9. *Axioms (4) and (5) hold in the CwF of fibrant types in \mathcal{E} (for the terms constructed in Corollary 6.2.7).*

Proof. By Theorem 6.2.8. \square

6.2.5 Axiom (3)

In light of the previous section, the only axiom remaining is axiom (3). Our goal here is, given a contractible fibration $A : \mathbf{Fib} \Gamma$, 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 $(\lambda_- \rightarrow 1, !_1) : \mathbf{Fib}(\Gamma)$. Therefore we will ambiguously write $1 : \mathbf{Fib}(\Gamma)$ for the pair $(\lambda_- \rightarrow 1, !_1)$.

¹Note that there are different ways to interpret \mathbf{coerce} in the model. This interpretation is not in general the same as the one obtained by directly interpreting Definition 2.1.6. 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.

Definition 6.2.10 (The contraction of a family). Given a family $A : \Gamma \rightarrow \mathcal{U}$ we define the contraction of A as

$$\begin{aligned} C_A &: \Gamma \times \mathbf{I} \rightarrow \mathcal{U} \\ C_A(x, i) &\triangleq [i = 0] \rightarrow A(x) \end{aligned}$$

Note that, for any $\gamma : \Delta \rightarrow \Gamma$, we have $C_{A \circ \gamma} = C_A \circ (\gamma \times id)$.

We now need to show that C_A is fibrant whenever A is both fibrant and contractible (Definition 5.4.4). Recall from Lemma 5.4.5 that a family being both fibrant and contractible is equivalent to having an extension structure (Definition 5.4.3) for that family.

We therefore construct a fibrancy structure for C_A as follows:

Theorem 6.2.11. *If $(A, \alpha) : \mathbf{Fib} \Gamma$ is contractible then we can construct a composition structure for C_A . That is, there is a function*

$$\mathbf{isFib}_C : \{\Gamma : \mathcal{U}\} \{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \mathbf{isFib} A \rightarrow \mathbf{Contr} A \rightarrow \mathbf{isFib} C_A$$

Proof. Take $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$, $\alpha : \mathbf{isFib} A$ and $ctr : \mathbf{Contr} A$. Since A is both fibrant and contractible then, by Lemma 5.4.5, we can construct an extension structure $\varepsilon \triangleq \mathbf{toExt} \alpha ctr : \mathbf{Ext} A$. We can then define a composition structure $\mathbf{isFib}_C \alpha ctr : \mathbf{isFib}(C_A)$ like so:

$$\mathbf{isFib}_C \alpha ctr e p \varphi f c_0 u \triangleq \varepsilon (\mathbf{fst}(p\bar{e})) (\varphi, (\lambda v \rightarrow f v \bar{e} u))$$

for $u : [\mathbf{snd}(p\bar{e}) = 0]$. Since ε is an extension structure we have that, for any $v : [\varphi]$,

$$\begin{aligned} \mathbf{isFib}_C \alpha ctr e p \varphi f c_0 u &= \lambda u \rightarrow \varepsilon (\mathbf{fst}(p\bar{e})) (\varphi, (\lambda v \rightarrow f v \bar{e} u)) \\ &= \lambda u \rightarrow f v \bar{e} u \\ &= f v \bar{e} \end{aligned}$$

and hence $(\varphi, f) @ \bar{e} \nearrow \mathbf{isFib}_C \alpha ctr e p \varphi f c_0$ as required. Therefore we have a defined a valid composition operation for C_A . Further, note that for any $\gamma : \Delta \rightarrow \Gamma$ we have:

$$\begin{aligned} \mathbf{isFib}_C (\alpha[\gamma]) (ctr \circ \gamma) e p \varphi f c_0 u & \\ &= \mathbf{toExt} (\alpha[\gamma]) (ctr \circ \gamma) (\mathbf{fst}(p\bar{e})) (\varphi, (\lambda v \rightarrow f v \bar{e} u)) \\ &= ((\mathbf{toExt} \alpha ctr) \circ \gamma) (\mathbf{fst}(p\bar{e})) (\varphi, (\lambda v \rightarrow f v \bar{e} u)) \\ &= (\mathbf{toExt} \alpha ctr) (\gamma (\mathbf{fst}(p\bar{e}))) (\varphi, (\lambda v \rightarrow f v \bar{e} u)) \\ &= (\mathbf{toExt} \alpha ctr) (\mathbf{fst}((\gamma \times id)(p\bar{e}))) (\varphi, (\lambda v \rightarrow f v \bar{e} u)) \\ &= (\mathbf{isFib}_C (\alpha) (ctr)) e ((\gamma \times id) \circ p) \varphi f c_0 u \\ &= (\mathbf{isFib}_C \alpha ctr)[\gamma \times id] e p \varphi f c_0 u \end{aligned}$$

Therefore $\text{isFib}_C (\alpha[\gamma]) (ctr \circ \gamma) = (\text{isFib}_C \alpha ctr)[\gamma \times id]$ and hence isFib_C is stable under reindexing. \square

Theorem 6.2.12. *There exists a function*

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

Proof. Given $\Gamma : \mathcal{U}$, $(A, \alpha) : \text{Fib } \Gamma$ and $ctr : \text{Contr } A$, we observe that

$$C_A[\langle id, 0 \rangle](x) = C_A(x, 0) = ([0 = 0] \rightarrow A(x)) \cong (1 \rightarrow A(x)) \cong A(x)$$

and

$$C_A[\langle id, 1 \rangle](x) = C_A(x, 1) = ([1 = 0] \rightarrow A(x)) \cong (\emptyset \rightarrow A(x)) \cong 1$$

Therefore we have

$$((C_A, \text{isFib}_C \alpha ctr), f_A, g_A) : (A, \alpha) \sim_{\cong} 1$$

where $f_A : A \cong C_A[\langle id, 0 \rangle]$ and $g_A : 1 \cong C_A[\langle id, 1 \rangle]$ are the obvious isomorphisms indicated above. Hence we can define

$$\text{contract}((A, \alpha), ctr) \triangleq \text{improve}((C_A, \text{isFib}_C \alpha ctr), f_A, g_A) : (A, \alpha) \sim_{\mathcal{U}} 1$$

as required. Further, for any $\gamma : \Delta \rightarrow \Gamma$, we have:

$$\begin{aligned} \text{contract} ((A, \alpha)[\gamma]) (ctr \circ \gamma) &= \text{improve}((C_{A \circ \gamma}, \text{isFib}_C (\alpha[\gamma]) (ctr \circ \gamma)), f_{A \circ \gamma}, g_{A \circ \gamma}) \\ &= \text{improve}((C_A \circ (\gamma \times id), (\text{isFib}_C \alpha ctr)[\gamma \times id]), f_A \circ \gamma, g_A \circ \gamma) \\ &= \text{improve}((C_A, \text{isFib}_C \alpha ctr)[\gamma \times id], f_A \circ \gamma, g_A \circ \gamma) \\ &= \text{improve}((C_A, \text{isFib}_C \alpha ctr), f_A, g_A)[\gamma \times id] \\ &= \text{contract}((A, \alpha), ctr)[\gamma \times id] \end{aligned}$$

and hence contract is stable under reindexing. \square

Corollary 6.2.13. *The CwF of fibrant types in \mathcal{E} models axiom (3).*

6.3 Conclusion

In this chapter we have shown how the univalence axiom can be decomposed into an alternative set of properties. This decomposition happens inside of MLTT, without reference to any particular model. We then showed how these new properties are satisfied by any topos satisfying our axioms from Chapter 5. Most of the new properties follow fairly

straightforwardly from the strictness axiom, \mathbf{ax}_9 , and from the properties of fibrations, such as the realignment lemma (Lemma 5.3.10) and the fact that fibrations are closed under isomorphism. The final property, number (3), is shown to hold by defining the contraction of a family (Definition 6.2.10). This can be seen as a significantly simplified case of the glueing construction presented in Section 5.4.

In Chapter 5 we showed how to repeat many of the constructions of Cohen et al. [18] using the internal language of a topos. Although we were unable to construct a univalent universe using the internal language, we did show some results relevant to univalence. The idea was that those results should imply that, were there to be a universe containing all small fibrations, then that universe would satisfy the univalence axiom. In this chapter we proved the same result but with, we believe, simpler constructions. Therefore, when taken together, the results in this chapter prove the following:

Theorem 6.3.1. *Given any topos \mathcal{E} which is a model of our axioms (Figure 5.4). If \mathcal{E} also models a type-theoretic universe of \mathcal{U} -small fibrations then that universe satisfies the univalence axiom.*

In fairness, it should be noted that while this proof of the univalence axiom avoids the use of the glueing construction from Section 5.4, that does not imply that we have entirely eliminated the need for glueing. In particular, the (external) proof that the universe of \mathcal{U} -small fibrations is a fibrant object (a fibration over 1) still requires the use of the glueing construction. The fact that the universe is fibrant is essential, not only to ensure that it is contained in the CwF of fibrant types, but also to allow us to compose paths in the universe as we do in Theorem 6.1.2.

Therefore the only piece missing in order to give a full internal account of the cubical sets model of univalent type theory is the construction of such a universe of fibrant types. This problem is resolved in Chapter 7.

Chapter 7

Internal universes

In Chapter 5 we saw how models of univalent type theory could be axiomatised using the internal type theory of an elementary topos. However, while we were able to replicate many of the constructions of Cohen et al. [18] using this approach, we were unable to axiomatise the necessary properties of a universe of (small) fibrations (Section 3.3.5) using the internal language.

In this chapter we propose an extension to the usual internal language which will allow us to axiomatise the properties we want. We will then use this extended language in order to show how a universe can be constructed from a simpler requirement that the interval be *tiny*.

7.1 The “no-go” theorem for internal universes

In this section we show why there can be no universe that weakly classifies CCHM fibrations in an internal sense. Such a weak classifier would be given by the following data

$$\begin{array}{ll} \mathcal{V} : \mathcal{U}_1 & \text{code} : \{\Gamma : \mathcal{U}\}(\Phi : \mathbf{Fib} \Gamma) \rightarrow \Gamma \rightarrow \mathcal{V} \\ \mathcal{E} : \mathbf{Fib} \mathcal{V} & \text{Elcode} : \{\Gamma : \mathcal{U}\}(\Phi : \mathbf{Fib} \Gamma) \rightarrow [\mathcal{E}[\text{code } \Phi] = \Phi] \end{array} \quad (7.1)$$

where $\mathbf{Fib} \Gamma$ is, as in Definition 5.3.5¹, the collection of small fibrations over Γ . That is, those fibrations whose fibres are in \mathcal{U} . Here \mathcal{V} is the universe and \mathcal{E} is a CCHM fibration over it which is a weak classifier in the sense that any (small) fibration $\Phi : \mathbf{Fib} \Gamma$ can be obtained from it (up to equality) by re-indexing along some function $\text{code } \Phi : \Gamma \rightarrow \mathcal{V}$. (The word “weak” refers to the fact that we do not require there to be a *unique* function $\gamma : \Gamma \rightarrow \mathcal{V}$ with $\mathcal{E}[\gamma] = \Phi$.)

We will show that the data in (7.1) implies that the interval must be trivial ($0 = 1$),

¹Technically we need to generalise the definition given there to allow Γ to range over types of any size, specifically over \mathcal{U}_1 so that $\mathcal{E} : \mathbf{Fib} \mathcal{V}$ is well-formed. Otherwise the definition remains unchanged. In particular, the fibres of the fibration still lie in \mathcal{U} .

contradicting the assumption \mathbf{ax}_2 in Figure 5.4. This is because (7.1) allows one to deduce that if a family of types $A : \Gamma \rightarrow \mathcal{U}$ has the property that each $A x$ has a fibration structure when regarded as a family over the unit type 1 , then there is a fibration structure for the whole family A ; and yet there are families where this cannot be the case.

This was first pointed out in Remark 5.3.6, but we repeat the construction here for readability. Consider the family $P : \mathbb{I} \rightarrow \mathcal{U}$ with $P i \triangleq [0 = i]$. For each $i : \mathbb{I}$, the type $P i$ has a fibration structure $\pi i : \mathbf{isFib} \{1\} (\lambda _ \rightarrow P i)$, because the internal equality satisfies the principle of uniqueness of identity proofs. Specifically, given $e : \{0, 1\}$, $p : \mathbb{I} \rightarrow 1$, $\varphi : \mathbf{Cof}$, $f : ([\varphi] \rightarrow \Pi_{\mathbb{I}}(\lambda _ \rightarrow P i)) = ([\varphi] \rightarrow \mathbb{I} \rightarrow [0 = i])$ and $a_0 : \{a_0 : [0 = i] \mid (\varphi, f) @ e \nearrow a_0\}$ then we can simply define

$$\pi i e p \varphi f a_0 \triangleq a_0 \quad : \quad \{a_1 : [0 = i] \mid (\varphi, f) @ \bar{e} \nearrow a_1\}$$

where the condition that $(\varphi, f) @ \bar{e} \nearrow a_0$ is satisfied by the fact that the type of proofs $[0 = i]$ is a subterminal.

However, the family P as a whole cannot have a fibration structure. That is, we have $\mathbf{isFib} P \rightarrow \perp$. This is because if we had a fibration structure, $\alpha : \mathbf{isFib} P$, then we could define a substitution function as in (5.35):

$$\mathbf{subst} : i \sim j \rightarrow P i \rightarrow P j$$

Since we have $id : 0 \sim 1$ and $*$: $P 0$ we could therefore apply these arguments to \mathbf{subst} in order to get:

$$\mathbf{subst} id * : P 1$$

However, we know that $P 1 = [0 = 1]$; therefore by combining this with \mathbf{ax}_2 from Figure 5.4 we get a contradiction.

From this we deduce the following “no-go”² theorem for internal universes of CCHM fibrations.

Theorem 7.1.1 (The no-go theorem). *The existence of types and functions as in (7.1) for CCHM fibrations is contradictory. More precisely, if $\mathbf{IntUniv} : \mathcal{U}_2$ is the dependent record type with fields \mathcal{V} , $\mathcal{E}\ell$, \mathbf{code} and \mathbf{Elcode} as in (7.1), then there is a term of type $\mathbf{IntUniv} \rightarrow \perp$.*

*Proof.*³ Suppose we have an element of $\mathbf{IntUniv}$ and hence functions as in (7.1). Given any family $A : \Gamma \rightarrow \mathcal{U}$ such that each fibre is a fibrant object, $\pi : (x : \Gamma) \rightarrow \mathbf{isFib} \{1\} (\lambda _ \rightarrow A x)$,

²We are stealing Shulman’s terminology [58, section 4.1].

³See <https://doi.org/10.17863/CAM.35681> for an Agda version of this proof.

then, for each $x : \Gamma$ we can define:

$$\text{code}((\lambda _ \rightarrow Ax), \pi x) : 1 \rightarrow \mathcal{V}$$

and hence by applying $*$: 1 and then abstracting over x we get a map $a : \Gamma \rightarrow \mathcal{V}$ defined like so:

$$ax = \text{code}((\lambda _ \rightarrow Ax), \pi x) * \quad (7.2)$$

We can then reindex $\mathcal{E}\ell$ to get a fibration over Γ :

$$\Phi : \text{Fib } \Gamma \quad \Phi = \mathcal{E}\ell[a] \quad (7.3)$$

Using **Elcode** we can show that $\text{fst } \Phi = A$. Specifically, given any $x : \Gamma$ we have

$$\begin{aligned} \text{fst } \Phi x &= \text{fst } (\mathcal{E}\ell[a]) x && \text{by definition of } \Phi \\ &= \text{fst } \mathcal{E}\ell(ax) && \text{by definition of } _[-] \\ &= \text{fst } \mathcal{E}\ell(\text{code}((\lambda _ \rightarrow Ax), \pi x) *) && \text{by definition of } a \\ &= \text{fst } (\mathcal{E}\ell[\text{code}((\lambda _ \rightarrow Ax), \pi x)]) * && \text{by definition of } _[-] \\ &= \text{fst } ((\lambda _ \rightarrow Ax), \pi x) * && \text{by Elcode} \\ &= Ax && \text{by evaluation} \end{aligned}$$

Hence by function extensionality we have $\text{fst } \Phi = A$.

Therefore we can deduce that $\text{snd } \Phi : \text{isFib } A$. However, we saw above how to transform such an element into a proof of \perp by taking $\Gamma \triangleq \mathbf{I}$ and $A \triangleq \lambda i \rightarrow [0 = i]$. So altogether we have a proof of $\text{IntUniv} \rightarrow \perp$. \square

Remark 7.1.2. This counterexample generalises to other notions of fibration: it is not usually the case that any type family $A : \Gamma \rightarrow \mathcal{U}$ for which Ax is fibrant over 1 for all $x : \Gamma$, is fibrant over Γ . The above proof should be compared with the proof that there is no “fibrant replacement” type-former in *Homotopy Type System* (HTS); see https://ncatlab.org/homotopytypetheory/show/Homotopy+Type+System#fibrant_replacement. Theorem 7.3.2 below provides a further example of a global construct that does not internalize.

7.2 Crisp type theory

The proof of Theorem 7.1.1 depends upon the fact that in the internal language the **code** function can be applied to elements with free variables. In this case it is the variable $x : \Gamma$ in $\text{code}((\lambda _ \rightarrow Ax), \pi x) *$; by abstracting over it we get a function $\Gamma \rightarrow \mathcal{V}$ and re-indexing $\mathcal{E}\ell$ along this function gives the offending fibration (7.3). Nevertheless, the cubical sets presheaf

topos does contain a (univalent) universe which is a CCHM fibration classifier, but only in an *external* sense. Thus there is an object \mathcal{V} in $\widehat{\square}$ and a global section $\mathcal{E}l : 1 \rightarrow \mathbf{Fib} \mathcal{V}$ with the property that for any object Γ and morphism $\Phi : 1 \rightarrow \mathbf{Fib} \Gamma$, there is a morphism $\mathbf{code} \Phi : \Gamma \rightarrow \mathcal{V}$ so that Φ is equal to the composition $\mathbf{Fib}(\mathbf{code} \Phi) \circ \mathcal{E}l : 1 \rightarrow \mathbf{Fib} \Gamma$; see Section 3.3.5 for a concrete description of \mathcal{V} . The internalisation of this property replaces the use of global elements $1 \rightarrow \Gamma$ of an object by local elements, that is, morphisms $X \rightarrow \Gamma$ where X ranges over a suitable collection of generating objects (for example, the representable objects in a presheaf topos); and we have seen that such an internalised version cannot exist.

Nevertheless, we would like to explain the construction of universes like $\mathcal{V} \in \widehat{\square}$ using some kind of type-theoretic language that builds on Chapter 5. So we seek a way of manipulating global elements of an object Γ , within the internal language. One cannot do so simply by quantifying over elements of the type $1 \rightarrow \Gamma$, because of the (internal) isomorphism $\Gamma \cong (1 \rightarrow \Gamma)$. Instead, we pass to a modal type theory that can speak about global elements, which we call *crisp type theory*. Its judgements, such as $\Delta \mid \Gamma \vdash a : A$, have two context zones—where Δ represents global elements and Γ the usual, local ones. The context structure is that used for an S4 necessitation modality [52, 22, 58], because a global element from Δ can be used locally, but global elements cannot depend on local variables from Γ . Following [58], we say that the left-hand context Δ contains *crisp* hypotheses about the types of variables, written $x :: A$.

The interpretation of crisp type theory in cubical sets makes use of the comonad $\flat : \widehat{\square} \rightarrow \widehat{\square}$ that sends a presheaf A to the constant presheaf on the set of global sections of A ; thus $\flat A(X) \cong A(1)$ for all $X \in \square$ (where $1 \in \square$ is terminal). Then a judgement $\Delta \mid \Gamma \vdash a : A$ describes the situation where Δ is a presheaf, Γ is a family of presheaves over $\flat\Delta$, A is a family over $\Sigma(\flat\Delta)\Gamma$ and a is an element of that family. The rules of crisp type theory are designed to be sound for this interpretation. Compared with ordinary type theory, the key constraint is that *types in the crisp context and terms substituted for crisp variables depend only on crisp variables*. The crisp variable and (admissible) substitution rules are:

$$\frac{}{\Delta, x :: A, \Delta' \mid \Gamma \vdash x : A} \quad \frac{\Delta \mid \diamond \vdash a : A \quad \Delta, x :: A, \Delta' \mid \Gamma \vdash b : B}{\Delta, \Delta'[a/x] \mid \Gamma[a/x] \vdash b[a/x] : B[a/x]} \quad (7.4)$$

The semantics of the variable rule, which says that global elements can be used locally, uses the counit $\varepsilon A : \flat A \rightarrow A$ of the comonad \flat mentioned above. In the substitution rule, \diamond stands for the empty context, so a and A may only depend upon the crisp variables from Δ . The other rules of crisp type theory (those for Π types, Σ types, etc.) carry the crisp context along. For our application we do not need a type-former for \flat , but instead make use of crisp Π types (see, e.g. [22, 51]), that is, Π types whose domain is crisp with

$$\frac{\Delta \mid \diamond \vdash A : \mathcal{U}_m \quad \Delta, x :: A \mid \Gamma \vdash B : \mathcal{U}_n}{\Delta \mid \Gamma \vdash (x :: A) \rightarrow B : \mathcal{U}_{m \sqcup n}}$$

$$\frac{\Delta, x :: A \mid \Gamma \vdash b : B}{\Delta \mid \Gamma \vdash \lambda x :: A. b : (x :: A) \rightarrow B}$$

$$\frac{\Delta \mid \Gamma \vdash f : (x :: A) \rightarrow B \quad \Delta \mid \diamond \vdash a : A}{\Delta \mid \Gamma \vdash f a : B[a/x]}$$

Figure 7.1: Formation, introduction and elimination rules for crisp Π types

$\beta\eta$ judgemental equalities. The formation, introduction and elimination rules are shown in Figure 7.1. In these rules, because the argument variable x is crisp, its type A , and the term a to which the function f is applied, must also be crisp.

We also use *crisp induction* [58] for crisp type theory's identity type, $x \equiv y$. Specifically, this is identity elimination with a family $y :: A, p :: x \equiv y \vdash C(y, p)$ whose parameters are crisp variables, so that for every such $A :: \mathcal{U}_n, x :: A$ and $C : (y :: A)(p :: x \equiv y) \rightarrow \mathcal{U}_n$ we get a map

$$(y :: A)(p :: x \equiv y)(z : C x \mathbf{refl}) \rightarrow C y p \quad (7.5)$$

together with a β judgemental equality. In this thesis we are working with the topos' internal identity type defined by $x \equiv y \triangleq [x = y]$ with $\mathbf{refl} \triangleq *$, and therefore the above induction principle follows immediately from the fact that this identity type is extensional. However, we mention this fact to clarify what we require when working in a setting without an extensional identity type, such as in the Agda development which accompanies this thesis.

Remark 7.2.1 (Presheaf models of crisp type theory). Crisp type theory is motivated by the specific presheaf topos $\widehat{\square}$. However, very little is required of a category \mathbf{C} for the presheaf topos $\widehat{\mathbf{C}}$ to soundly interpret it using the comonad $\flat = p^* \circ p_*$, where $p_* : \widehat{\mathbf{C}} \rightarrow \mathbf{Set}$ takes the global sections of a presheaf and its left adjoint $p^* : \mathbf{Set} \rightarrow \widehat{\mathbf{C}}$ sends sets to constant presheaves. Explicitly:

$$p_*(A) \triangleq \widehat{\mathbf{C}}(1, A) \quad p^*(X)(-) \triangleq X$$

This \flat preserves finite limits (because p^* has a left adjoint given by left Kan extension along $\mathbf{C} \rightarrow 1$). Although the details remain to be fully worked out, it appears that to model crisp type theory with crisp Π types satisfying the rules in Figure 7.1 and crisp identity induction (7.5) (and moreover a \flat modality with crisp \flat induction, which we do not use here), the only additional condition needed is that this comonad is idempotent

(meaning that the comultiplication $\delta : \flat \rightarrow \flat \circ \flat$ is an isomorphism).

This idempotence holds if $\widehat{\mathbf{C}}$ is a connected topos, which is the case if \mathbf{C} is a connected category—for example, when \mathbf{C} has a terminal object. In this case, when \mathbf{C} is connected, any natural transformation $f : 1 \rightarrow p^*(X)$ is entirely determined by its action at any object, since $f_I(*) = f_J(*) \in X$ whenever I and J are connected by a morphism in \mathbf{C} . Therefore $p_*(p^*(X)) = \widehat{\mathbf{C}}(1, p^*(X)) \cong X$ for any set X and hence $p_* \circ p^* \cong id$. From this we deduce $\flat \circ \flat = p^* \circ p_* \circ p^* \circ p_* \cong p^* \circ p_* = \flat$ with $\delta : \flat \rightarrow \flat \circ \flat$ being one side of this isomorphism. Note that this argument assumes that \mathbf{C} is inhabited, but when \mathbf{C} is empty then δ is trivially an isomorphism.

If $\widehat{\mathbf{C}}$ has a terminal object, then it is in fact a local topos [37, Sect. C3.6] and \flat has a right adjoint; in which case, conjecturally [58, Remark 7.5], one gets a model of the whole of Shulman’s spatial type theory, of which crisp type theory is a part. In fact \square does not just have a terminal object, it has all finite products (as does any Lawvere theory) and from this it follows that $\widehat{\square}$ is not just local, but also cohesive [43].

Remark 7.2.2 (Agda-flat). Vezzosi has created a fork of Agda, called *Agda-flat* [3], which allows us to explore crisp type theory. It adds the ability to use crisp variables⁴ $x :: A$ in places where ordinary variables $x : A$ may occur in Agda, and checks the modal restrictions in the above rules. For example, Agda-flat quite correctly rejects the following attempted application of a crisp- Π function to an ordinary argument

$$\text{wrong} : (A :: \mathcal{U}_n)(B : \mathcal{U}_m)(f : (_ :: A) \rightarrow B)(x : A) \rightarrow B \quad \text{wrong } A B f x = f(x)$$

while the variant with $x :: A$ succeeds. This is a simple example of keeping to the modal discipline that crisp type theory imposes; for more complicated cases, such as occur in the proof of Theorem 7.3.3 below, we have found Agda-flat indispensable for avoiding errors. However, Agda-flat implements a superset of crisp type theory and more work is needed to understand their precise relationship. For example, Agda’s ability to define inductive types leads to new types in Agda-flat, such as the \flat modality itself; and its pattern-matching facilities allow one to prove properties of \flat that go beyond crisp type theory. Agda allows one to switch off pattern-matching in a module; to be safe we do that as far as possible in our development. Installation instructions for Agda-flat are at <https://doi.org/10.17863/CAM.35681>.

7.3 Universes from tiny intervals

In crisp type theory, to avoid the inconsistency in the “no-go” theorem (Theorem 7.1.1), we can weaken the definition of a universe in (7.1) by taking `code` and `Elcode` to be crisp

⁴The Agda-flat concrete syntax for “ $x :: A$ ” is “ $x : \{b\} A$ ”.

functions of fibrations Φ (and implicitly, of the base type Γ of the fibration). For if `code` has type $\{\Gamma :: \mathcal{U}\}(\Phi :: \mathbf{Fib} \Gamma)(x : \Gamma) \rightarrow \mathcal{V}$, then the proof of a contradiction is blocked when in (7.2) we try to apply `code` to $\Phi = ((\lambda _ \rightarrow Ax), \pi x)$, which depends upon the local variable $x : \Gamma$. Indeed we show in this section that given an extra assumption about the interval \mathbf{I} , that holds for cubical sets, it is possible to define a universe with such crisp coding functions which moreover are unique, so that one gets a classifying fibration, rather than just a weakly classifying one.

It should be noted that none of results in this chapter depend on the exact definition of fibration as given in Definition 5.3.3. In fact, the universe construction and other results generalise directly to other notions of fibration, such as the one used in cartesian cubical type theory [6]. Therefore, for the remainder of this chapter we will work more generally, by abstracting the notion of fibration to something parametrised by an arbitrary \mathbf{Comp} structure.

Definition 7.3.1 (Comp-fibrations). Given some structure $\mathbf{Comp} : \mathcal{U}^{\mathbf{I}} \rightarrow \mathcal{U}$, we define the type of \mathbf{Comp} -fibration structures for a family $A : \Gamma \rightarrow \mathcal{U}$ to be:

$$\mathbf{isFib}_{\mathbf{Comp}} A \triangleq (p : \Gamma^{\mathbf{I}}) \rightarrow \mathbf{Comp}(A \circ p)$$

Where \mathbf{Comp} is obvious from the context we will simply write \mathbf{isFib} for $\mathbf{isFib}_{\mathbf{Comp}}$. Note that we recover the previous notion of fibration, up-to a reordering of arguments, by taking

$$\mathbf{Comp} A \triangleq (e : \{0, 1\}) \rightarrow \mathbf{Comp}' e A$$

where \mathbf{Comp}' is the composition operation previously defined in Definition 5.3.3. As before we write $\mathbf{Fib} \Gamma$ for the type $(A : \Gamma \rightarrow \mathcal{U}) \times \mathbf{isFib}(A)$.

Recall from Section 3.3.1 that in the cubical sets model, the type \mathbf{I} denotes the representable presheaf $y\{i\} \in \widehat{\square}$ on the object $\{i\} \in \square$. Since \square has finite products, there is a functor $_ \times \{i\} : \square \rightarrow \square$. Pre-composition with this functor induces an endofunctor on presheaves $(_ \times \{i\})^* : \widehat{\square} \rightarrow \widehat{\square}$ which has left and right adjoints, given by left and right Kan extension [47, Chap. X] along $_ \times \{i\}$. Hence by the Yoneda Lemma, for any $F \in \widehat{\square}$ and $X \in \square$

$$\begin{aligned} (F^{\mathbf{I}}) X &= \widehat{\square}(yX \times \mathbf{I}, F) \\ &= \widehat{\square}(yX \times y\{i\}, F) \\ &\cong \widehat{\square}(y(X \times \{i\}), F) \\ &\cong F(X \times \{i\}) \\ &= ((_ \times \{i\})^* F) X \end{aligned}$$

naturally in both X and F . It follows that the exponential functor $(-)^{\mathbb{I}} : \widehat{\square} \rightarrow \widehat{\square}$ is naturally isomorphic to $(- \times \{i\})^*$ and hence not only has a left adjoint (corresponding to product with \mathbb{I}) but also a right adjoint. The significance of objects in a category with finite products that are not only exponentiable (product with them has a right adjoint), but also whose exponential functor has a right adjoint was first pointed out by Lawvere in the context of synthetic differential geometry [42]. He called such objects “atomic”, but we will follow later usage [65] and call them *tiny*.⁵ Thus the interval in cubical sets is tiny and we have a right adjoint to the path functor $(-)^{\mathbb{I}}$ that we denote by $\sqrt{} : \widehat{\square} \rightarrow \widehat{\square}$. So for each $B \in \widehat{\square}$, the functor $\widehat{\square}(-^{\mathbb{I}}, B) : \widehat{\square} \rightarrow \mathbf{Set}$ is representable by \sqrt{B} , that is, there are bijections $\widehat{\square}(A^{\mathbb{I}}, B) \cong \widehat{\square}(A, \sqrt{B})$, natural in A .

Given Γ and $A : \Gamma \rightarrow \mathcal{U}$ in $\widehat{\square}$, from Definition 5.3.3 we have that fibration structures $1 \rightarrow \mathbf{isFib} A$ correspond to sections of $\mathbf{fst} : ((p : \Gamma^{\mathbb{I}}) \times \mathbf{Comp}(A \circ p)) \rightarrow \Gamma^{\mathbb{I}}$ as indicated by α in the following diagram:

$$\begin{array}{ccc} \Gamma^{\mathbb{I}} & \xrightarrow{\alpha} & (p : \Gamma^{\mathbb{I}}) \times \mathbf{Comp}(A \circ p) \\ & \searrow id & \downarrow \mathbf{fst} \\ & & \Gamma^{\mathbb{I}} \end{array}$$

Transposing across the adjunction $(-)^{\mathbb{I}} \dashv \sqrt{}$, we get morphisms making the following diagram commute:

$$\begin{array}{ccc} \Gamma & \xrightarrow{\bar{\alpha}} & \sqrt{((p : \Gamma^{\mathbb{I}}) \times \mathbf{Comp}(A \circ p))} \\ & \searrow id & \downarrow \sqrt{\mathbf{fst}} \\ \Gamma & \xrightarrow{\eta_{\Gamma}} & \sqrt{(\Gamma^{\mathbb{I}})} \end{array}$$

We therefore have that fibration structures for A correspond to sections of the pullback $\pi_1 : R_{\Gamma}A \rightarrow \Gamma$ of $\sqrt{\mathbf{fst}}$ along the unit $\eta_{\Gamma} : \Gamma \rightarrow \sqrt{(\Gamma^{\mathbb{I}})}$ of the adjunction at Γ (which is the adjoint transpose of $id : \Gamma^{\mathbb{I}} \rightarrow \Gamma^{\mathbb{I}}$), as indicated below:

$$\begin{array}{ccccc} & & \bar{\alpha} & & \\ & & \curvearrowright & & \\ \Gamma & \cdots \rightarrow & R_{\Gamma}A & \xrightarrow{\pi_2} & \sqrt{((p : \Gamma^{\mathbb{I}}) \times \mathbf{Comp}(A \circ p))} \\ & \searrow id & \downarrow \pi_1 & \lrcorner & \downarrow \sqrt{\mathbf{fst}} \\ & & \Gamma & \xrightarrow{\eta_{\Gamma}} & \sqrt{(\Gamma^{\mathbb{I}})} \end{array}$$

⁵Warning: the adjective “tiny” is sometimes used to describe an object X of a \mathcal{V} -enriched cocomplete category \mathcal{C} for which the hom \mathcal{V} -functor $\mathcal{C}(X, -) : \mathcal{C} \rightarrow \mathcal{V}$ preserves colimits; see [57] for example. We prefer Kelly’s term *small-projective object* for this property. In the special case that $\mathcal{V} = \mathcal{C}$ and \mathcal{C} is cartesian closed and has sufficient properties for there to be an adjoint functor theorem, then a small-projective object is in particular a tiny one in the sense we use here.

$$\begin{aligned}
\sqrt{} &: (A :: \mathcal{U}_n) \rightarrow \mathcal{U}_n \\
\mathbf{R} &: \{A :: \mathcal{U}_n\}\{B :: \mathcal{U}_m\}(f :: A^{\mathbf{I}} \rightarrow B) \rightarrow A \rightarrow \sqrt{B} \\
\mathbf{L} &: \{A :: \mathcal{U}_n\}\{B :: \mathcal{U}_m\}(g :: A \rightarrow \sqrt{B}) \rightarrow A^{\mathbf{I}} \rightarrow B \\
\mathbf{LR} &: \{A :: \mathcal{U}_n\}\{B :: \mathcal{U}_m\}\{f :: A^{\mathbf{I}} \rightarrow B\} \rightarrow \mathbf{L}(\mathbf{R} f) \equiv f \\
\mathbf{RL} &: \{A :: \mathcal{U}_n\}\{B :: \mathcal{U}_m\}\{g :: A \rightarrow \sqrt{B}\} \rightarrow \mathbf{R}(\mathbf{L} g) \equiv g \\
\mathbf{R}_{nat} &: \{A :: \mathcal{U}_n\}\{B :: \mathcal{U}_m\}\{C :: \mathcal{U}_k\}(g :: A \rightarrow B)(f :: B^{\mathbf{I}} \rightarrow C) \rightarrow \mathbf{R}(f \circ g^{\mathbf{I}}) \equiv \mathbf{R}f \circ g
\end{aligned}$$

Figure 7.2: Axioms for tinytness of the interval in crisp type theory

This characterization of fibration structure does not depend on the particular definition of \mathbf{Comp} , so should apply to many notions of fibration. We will show how it leads to the construction of a universe $\mathcal{V} = R_{\mathcal{U}}id$ and family $\pi_1 : R_{\mathcal{U}}id \rightarrow \mathcal{U}$ which is a classifier for fibrations. However, there are two problems that have to be solved in order to carry out the construction within type theory:

- First, for \mathbf{Elcode} in (7.1) to be an equality (rather than just an isomorphism), one needs the choice of $R_{\Gamma}A$ to be strictly functorial with respect to re-indexing along Γ (and hence to be a dependent right adjoint in the sense of [16]).
- Secondly, one cannot use ordinary type theory as the internal language to formulate the construction, because the right adjoint to $(-)^{\mathbf{I}}$ does not internalize, as the following theorem shows.

Theorem 7.3.2. *There is no internal right adjoint to the path functor $(-)^{\mathbf{I}} : \widehat{\square} \rightarrow \widehat{\square}$ for cubical sets. In other words, there is no family of natural isomorphisms $(-^{\mathbf{I}} \rightarrow B) \cong (- \rightarrow \sqrt{B}) : \widehat{\square} \rightarrow \widehat{\square}$ (for $B \in \widehat{\square}$).*

Proof. It is an elementary fact about adjoint functors that such a family of natural isomorphisms is also natural in B . Note that $1^{\mathbf{I}} \cong 1$. So if we had such a family, then we would also have isomorphisms $B \cong (1 \rightarrow B) \cong (1^{\mathbf{I}} \rightarrow B) \cong (1 \rightarrow \sqrt{B}) \cong \sqrt{B}$ which are natural in B . Therefore $\sqrt{}$ would be isomorphic to the identity functor and hence so would be its left adjoint $(-)^{\mathbf{I}}$. Hence $\mathbf{I} \rightarrow _$ and $1 \rightarrow _$ would be isomorphic functors $\widehat{\square} \rightarrow \widehat{\square}$, which implies (by the internal Yoneda Lemma) that \mathbf{I} is isomorphic to the terminal object 1 , contradicting the fact that \mathbf{I} has two distinct global elements, \mathbf{ax}_2 . \square

We will solve the first of the two problems mentioned above in the same way that Voevodsky [64] solves a similar strictness problem (see also [16, Section 6]): apply $\sqrt{}$ once and for all to the displayed universe and then re-index, rather than *vice versa* (as done above). The second problem is solved by using the crisp type theory of the previous section to make the right adjoint $\sqrt{}$ suitably global. The axioms we use are given in Figure 7.2. The function \mathbf{R} gives the operation for transposing (global) morphisms across

the adjunction $(-)^{\mathbb{I}} \dashv \surd$, with inverse L (the bijection being given by LR and RL); and R_{nat} is the naturality of this operation. The other properties of an adjunction follow from these, in particular its functorial action $\surd : \{A :: \mathcal{U}_n\}\{B :: \mathcal{U}_m\}(f :: A \rightarrow B) \rightarrow \surd A \rightarrow \surd B$ can be defined like so:

$$\surd f \triangleq R(\varepsilon_A \circ f)$$

where $\varepsilon_A :: (\surd A)^{\mathbb{I}} \rightarrow A$ is the counit of the adjunction, which in turn can be defined as:

$$\varepsilon_A = L(id_{\surd A})$$

Note that Figure 7.2 assumes that the right adjoint to $\mathbb{I} \rightarrow (-)$ preserves universe levels. The soundness of this for $\widehat{\square}$ relies on the fact that this adjoint is given by right Kan extension [47, Chap. X] along $_ \times \mathbb{I} : \square \rightarrow \square$ and hence sends a presheaf valued in the n th Grothendieck universe to another such.

Theorem 7.3.3 (Universe construction⁶). *Consider the notion of fibration in Definition 7.3.1 with any definition of composition structure \mathbf{Comp} (e.g. the CCHM one used in Chapter 5). Given the axioms for a tiny interval (Figure 7.2), there is a universe $\mathcal{V} : \mathcal{U}_1$ with a classifying fibration $\mathcal{E}l : \mathbf{Fib} \mathcal{V}$ equipped with the following data:*

$$\begin{aligned} \text{code} &: \{\Gamma :: \mathcal{U}\}(\Phi :: \mathbf{Fib} \Gamma) \rightarrow \Gamma \rightarrow \mathcal{V} \\ \text{Elcode} &: \{\Gamma :: \mathcal{U}\}(\Phi :: \mathbf{Fib} \Gamma) \rightarrow [\mathcal{E}l[\text{code} \Phi] = \Phi] \\ \text{codeEl} &: \{\Gamma :: \mathcal{U}\}(\gamma :: \Gamma \rightarrow U) \rightarrow [\text{code}(\mathcal{E}l[\gamma]) = \gamma] \end{aligned} \tag{7.6}$$

Proof. Consider the display function associated with the first universe:

$$\begin{aligned} \text{Elt} &: \mathcal{U}_1 & \text{pr} &: \text{Elt} \rightarrow \mathcal{U} \\ \text{Elt} = (A : \mathcal{U}) \times A & & \text{pr}(A, x) &= A \end{aligned} \tag{7.7}$$

We have $\mathbf{Comp} : \mathcal{U}^{\mathbb{I}} \rightarrow \mathcal{U}$ and hence using the transpose operation from Figure 7.2, $R \mathbf{Comp} : \mathcal{U} \rightarrow \surd \mathcal{U}$. We define $\mathcal{V} : \mathcal{U}_1$ by taking a pullback:

$$\begin{array}{ccc} \mathcal{V} & \xrightarrow{\pi_2} & \surd \text{Elt} \\ \pi_1 \downarrow & \lrcorner & \downarrow \surd \text{pr} \\ \mathcal{U} & \xrightarrow{R \mathbf{Comp}} & \surd \mathcal{U} \end{array}$$

⁶We just construct a universe for fibrations with fibers in \mathcal{U}_0 . However, similar universes $\mathcal{V}_n : \mathcal{U}_{n+1}$ can be constructed for fibrations with fibers in \mathcal{U}_n , for each n .

Explicitly, we construct the pullback like so:

$$\mathcal{V} = \{(A, B) : \mathcal{U} \times \sqrt{\text{Elt}} \mid \sqrt{\text{pr}} B = \text{R Comp } A\} \quad (7.8)$$

with π_1 and π_2 the obvious first and second projections. Transposing this square across the adjunction $(-)^{\text{I}} \dashv \sqrt{}$ we get $\text{pr} \circ \text{L } \pi_2 = \text{Comp} \circ \pi_1^{\text{I}} : \mathcal{V}^{\text{I}} \rightarrow \mathcal{U}$ since:

$$\begin{aligned} \text{pr} \circ \text{L } \pi_2 &= \text{L}(\sqrt{\text{pr}} \circ \pi_2) && \text{by naturality of L} \\ &= \text{L}(\text{R Comp} \circ \pi_1) && \text{by the previous diagram} \\ &= \text{L}(\text{R}(\text{Comp} \circ \pi_1^{\text{I}})) && \text{by naturality of R} \\ &= \text{Comp} \circ \pi_1^{\text{I}} && \text{by LR} \end{aligned}$$

Therefore the following diagram commutes:

$$\begin{array}{ccc} \mathcal{V}^{\text{I}} & \xrightarrow{\text{L } \pi_2} & \text{Elt} \\ \pi_1^{\text{I}} \downarrow & & \downarrow \text{pr} \\ \mathcal{U}^{\text{I}} & \xrightarrow{\text{Comp}} & \mathcal{U} \end{array}$$

Considering the first and second components of $\text{L } \pi_2$, we have $\text{L } \pi_2 = \langle \text{Comp} \circ \pi_1^{\text{I}}, v \rangle$ for some $v : (p : \mathcal{V}^{\text{I}}) \rightarrow \text{Comp}(\pi_1 \circ p)$; hence v is an element of $\text{isFib}\{\mathcal{V}\}_{\pi_1}$ and so we can define

$$\mathcal{E}l : \text{Fib } \mathcal{V} \quad \mathcal{E}l = (\pi_1, v) \quad (7.9)$$

So it just remains to construct the functions in (7.6). Given $\Gamma :: \mathcal{U}$ and $\Phi = (A, \alpha) :: \text{Fib } \Gamma$, we have $\alpha :: \text{isFib } A = (p : \Gamma^{\text{I}}) \rightarrow \text{Comp}(A \circ p)$. So the outer square in the diagram below commutes:

$$\begin{array}{ccc} & & \langle \text{Comp} \circ A^{\text{I}}, \alpha \rangle \\ & \curvearrowright & \\ \Gamma^{\text{I}} & & \text{Elt} \\ & \searrow & \downarrow \text{pr} \\ & & \mathcal{U} \\ & \nearrow & \downarrow \text{pr} \\ \Gamma^{\text{I}} & \xrightarrow{\text{L } \pi_2} & \text{Elt} \\ & \searrow & \downarrow \text{pr} \\ & & \mathcal{U} \\ & \nearrow & \downarrow \text{pr} \\ \Gamma^{\text{I}} & \xrightarrow{\text{L } \pi_2} & \text{Elt} \\ & \searrow & \downarrow \text{pr} \\ & & \mathcal{U} \\ & \nearrow & \downarrow \text{pr} \\ \Gamma^{\text{I}} & \xrightarrow{\text{L } \pi_2} & \text{Elt} \\ & \searrow & \downarrow \text{pr} \\ & & \mathcal{U} \\ & \nearrow & \downarrow \text{pr} \\ \Gamma^{\text{I}} & \xrightarrow{\text{L } \pi_2} & \text{Elt} \\ & \searrow & \downarrow \text{pr} \\ & & \mathcal{U} \end{array} \quad (7.10)$$

Transposing across the adjunction $(-)^{\text{I}} \dashv \sqrt{}$, this means that the outer square in the following diagram also commutes and therefore induces a function $\text{code } \Phi : \Gamma \rightarrow \mathcal{V}$ to the

pullback.

$$\begin{array}{ccc}
 & & \text{R}\langle \text{Comp} \circ A^{\mathbb{I}}, \alpha \rangle \\
 & \curvearrowright & \\
 \Gamma & \xrightarrow{\text{code } \Phi} & \mathcal{V} \xrightarrow{\pi_2} \sqrt{\text{Elt}} \\
 & \searrow A & \downarrow \pi_1 \quad \lrcorner \quad \downarrow \sqrt{\text{pr}} \\
 & & \mathcal{U} \xrightarrow{\text{R Comp}} \sqrt{\mathcal{U}}
 \end{array} \tag{7.11}$$

So we have that $\pi_1 \circ \text{code } \Phi = A$ and $\pi_2 \circ \text{code } \Phi = \text{R}\langle \text{Comp} \circ A^{\mathbb{I}}, \alpha \rangle$. Transposing the latter back across the adjunction and recalling that $\text{L}\pi_2 = \langle \text{Comp} \circ \pi_1^{\mathbb{I}}, v \rangle$ gives:

$$\begin{aligned}
 \langle \text{Comp} \circ A^{\mathbb{I}}, \alpha \rangle &= \text{L}(\text{R}\langle \text{Comp} \circ A^{\mathbb{I}}, \alpha \rangle) \\
 &= \text{L}(\pi_2 \circ \text{code } \Phi) \\
 &= \text{L}\pi_2 \circ (\text{code } \Phi)^{\mathbb{I}} \\
 &= \langle \text{Comp} \circ \pi_1^{\mathbb{I}}, v \rangle \circ (\text{code } \Phi)^{\mathbb{I}} \\
 &= \langle \text{Comp} \circ \pi_1^{\mathbb{I}} \circ (\text{code } \Phi)^{\mathbb{I}}, v \circ (\text{code } \Phi)^{\mathbb{I}} \rangle \\
 &= \langle \text{Comp} \circ (\pi_1 \circ \text{code } \Phi)^{\mathbb{I}}, v \circ (\text{code } \Phi)^{\mathbb{I}} \rangle \\
 &= \langle \text{Comp} \circ A^{\mathbb{I}}, v \circ (\text{code } \Phi)^{\mathbb{I}} \rangle
 \end{aligned}$$

From this, and the injectivity of the pairing map, we get $v \circ (\text{code } \Phi)^{\mathbb{I}} = \alpha$. Combining this with the proof of $\pi_1 \circ \text{code } \Phi = A$, we get the desired proof of $\text{Elcode } \Phi$ since:

$$\begin{aligned}
 \mathcal{E}\ell[\text{code } \Phi] &= (\pi_1, v)[\text{code } \Phi] \\
 &= (\pi_1 \circ \text{code } \Phi, v \circ (\text{code } \Phi)^{\mathbb{I}}) \\
 &= (A, \alpha) \\
 &= \Phi
 \end{aligned}$$

Finally, taking $\Gamma = \mathcal{V}$ and $\Phi = \mathcal{E}\ell$ in (7.11), the uniqueness property of the pullback implies that $\text{code } \mathcal{E}\ell = id$; and similarly, for any $\gamma :: \Delta \rightarrow \Gamma$ we have that $(\text{code } \Phi) \circ \gamma = \text{code}(\Phi[\gamma])$. Together these properties give us the desired proof $\text{codeEl } \gamma$ that

$$\text{code}(\mathcal{E}\ell[\gamma]) = (\text{code } \mathcal{E}\ell) \circ \gamma = id \circ \gamma = \gamma$$

as required. □

Remark 7.3.4. The above theorem can be generalised by replacing the particular universe $id : \mathcal{U} \rightarrow \mathcal{U}$ by an arbitrary one $E_0 : \mathcal{U}_0 \rightarrow \mathcal{U}$. So long as the composition structure Comp lands in \mathcal{U}_0 , one can use the above method to construct a universe of fibrant types from

among the U_0 types. The application of this generalisation we have in mind is to directed type theory; for example one can first construct the universe of fibrant types in the CCHM sense and then make a universe of covariant discrete fibrations in the Riehl-Shulman [55] sense from the fibrant types (repeating the construction with a different interval object).

Remark 7.3.5. The results in this section only make use of the fact that the functor $\sqrt{} : \widehat{\square} \rightarrow \widehat{\square}$ is right adjoint to the exponential $\mathbf{I} \rightarrow (-)$ and we saw at the beginning of this section why such a right adjoint exists. It is possible to give an explicit description of presheaves of the form $\sqrt{\Gamma}$, but so far we have not found such a description useful.

7.4 Applications

Models. Theorem 7.3.3 is the missing piece that allows a completely internal development of a model of univalent foundations based upon the CCHM notion of fibration, albeit internal to crisp type theory rather than ordinary type theory. One can define a CwF in crisp type theory whose objects are crisp types $\Gamma :: \mathcal{U}_1$, whose morphisms are crisp functions $\gamma :: \Gamma' \rightarrow \Gamma$, whose families are crisp CCHM fibrations $\Phi = (A, \alpha) :: \mathbf{Fib} \Gamma$ and whose elements are crisp dependent functions $f :: (x : \Gamma) \rightarrow Ax$. To see that this gives a model of univalent foundations one needs to prove:

- (a) The CwF is a model of intensional type theory with Π -types and inductive types (Σ -types, identity types, booleans, W -types, ...).
- (b) The type $\mathcal{V} :: \mathcal{U}_1$ constructed in Theorem 7.3.3 is fibrant (as a family over the unit type).
- (c) The classifying fibration $\mathcal{E}l :: \mathbf{Fib} \mathcal{V}$ satisfies the univalence axiom in this CwF.

These steps follow from previous results and the axioms given in Figure 7.2, together with the assumptions about the interval object and cofibrant types listed in Figure 5.4 from Chapter 5.

Part (a) was shown in Section 5.3 of Chapter 5. There we were not working in crisp type theory, however, the proofs still hold since the internal type theory used there is simply crisp type theory with the crisp context always taken to be empty. All of the rules and judgements are then the same as before.

Part (b) can be proved using the (strict) *glueing* operation defined in Section 5.4. The proof is essentially the same as the one given in [18].

Part (c) can be proved either using the method in Section 5.5, which mirrors the approach taken by Cohen et al. [18], or the simplified approach presented in Chapter 6.

Remark 7.4.1 (The interval is connected). Recall that we previously assumed that the interval was connected, \mathbf{ax}_1 in Figure 5.4. In fact, this axiom becomes redundant when assuming that the interval is tiny since \mathbf{ax}_1 follows from the axioms in Figure 7.2. The proof in Theorem 5.6.1, that the interval in cubical sets is connected, essentially uses the fact that $\widehat{\square}$ is a cohesive topos (Remark 7.2.1). However it also follows directly from the tiny property: connectedness holds iff $(\mathbf{I} \rightarrow \mathbb{B}) \cong \mathbb{B}$, where $\mathbb{B} = 1 + 1$ is the type of Booleans; since we postulate that $\mathbf{I} \rightarrow _$ has a right adjoint, it therefore preserves this coproduct and hence $(\mathbf{I} \rightarrow \mathbb{B}) \cong (\mathbf{I} \rightarrow 1) + (\mathbf{I} \rightarrow 1) \cong 1 + 1 = \mathbb{B}$.

Remark 7.4.2 (Alternative models). We have focussed on axioms satisfied by $\widehat{\square}$ and the CCHM notion of fibration in that presheaf topos. However, the universe construction in Theorem 7.3.3 also applies to the cartesian cubical sets model [6], and we expect it is possible to give proofs in crisp type theory of its fibrancy and univalence as well.

In this chapter we only consider “cartesian” path-based models of type theory, in which a path is an arbitrary function out of an interval object, or in other words, the path functor is given by an exponential. The models in [39] and [11] are not cartesian in that sense—the path functors they use are right adjoint to certain functorial cylinders [27] not given by cartesian product.⁷ However, those path functors do have right adjoints and universes in these models can be constructed using the method of Theorem 7.3.3. This is because the proof does not actually depend on the fact that the path functor is of the form $(_)^{\mathbf{I}}$. In fact the theorem works exactly the same if we replace $(_)^{\mathbf{I}}$ with some arbitrary functor \wp , even if \wp is not an internal functor. A proof in crisp type theory that those universes are fibrant (as families over 1) and univalent may require a modification of our axiomatic treatment of cofibrancy; we leave this for future work.

Universe hierarchies. Given that there are many notions of fibration that one may be interested in, it is natural to ask how relationships between them induce relationships between universes of fibrant types. As motivating examples of this, we might want a cubical type theory with a universe of fibrations with *regularity*, an extra strictness corresponding to the computation rule for identity types in intensional type theory; or a three-level directed type theory with non-fibrant, fibrant, and co/contravariant universes. Towards building such hierarchies, it is possible to show in crisp type theory that universes are functorial in the notion of fibration they encapsulate—when one notion of fibrancy implies another, the first universe includes the second.

Lemma 7.4.3. *Let $\mathbf{Comp}^1, \mathbf{Comp}^2 : \mathcal{U}^{\mathbf{I}} \rightarrow \mathcal{U}$ be two notions of composition, \mathbf{isFib}^1 and \mathbf{isFib}^2 the corresponding fibration structures, and \mathcal{V}^1 and \mathcal{V}^2 the corresponding classifying*

⁷Furthermore, obvious candidates for an interval object are not necessarily tiny in those models—for example, for the 1-simplex $\Delta[1]$ the exponential $\Delta[1] \rightarrow (_)$ in the topos $\widehat{\Delta}$ of simplicial sets does not have a right adjoint.

universes. A morphism of fibration structures is a function $f_{\Gamma,A} :: \mathbf{isFib}^1 A \rightarrow \mathbf{isFib}^2 A$ for all Γ and $A : \Gamma \rightarrow \mathcal{U}$, such that f is stable under reindexing: given $\gamma : \Delta \rightarrow \Gamma$, and $\alpha : \mathbf{isFib}^1 \Gamma A$ then

$$f_{\Gamma,A}(\alpha)[\gamma] = f_{\Delta,A \circ h}(\alpha[\gamma])$$

Such a morphism of fibrations f induces a function $\bar{f} : \mathcal{V}^1 \rightarrow \mathcal{V}^2$, and this preserves identity and composition in the sense that $\overline{id} = id$ and $\overline{g \circ f} = \bar{g} \circ \bar{f}$.

7.5 Conclusion

In this chapter we addressed the shortcomings of the work in Section 5.5. Specifically, the fact that the material on univalence does not apply to an actual universe object. We recalled why there can be no internal description of the univalent universe itself if one uses ordinary type theory as the internal language. Instead we extended ordinary type theory with a suitable modality and then gave a universe construction that hinges upon the tinyness property enjoyed by the interval in cubical sets. We call this language *crisp type theory* and our work inside it has been carried out and checked using an experimental version of Agda provided by Vezzosi [3].

Chapter 8

Conclusion

This thesis examined the extent to which the Kan cubical sets model of homotopy type theory can be explained using the internal language of the topos of cubical sets. Chapter 5 showed that most of the constructions can be carried out starting from the nine axioms given in Figure 5.4. This axiomatic approach means that the constructions carried out in the internal language either apply directly, or are easily adapted, to related models of type theory such as the cartesian cubical model [6]. We also believe that this presentation is cleaner than the external presentation, especially for readers familiar with type theory.

In Chapter 6 we showed how this internal approach can be used for a new purpose, namely, for presenting a simpler proof of univalence. In the process we proved intermediate results about the model which may be useful for other purposes, such as the principle of isovalence (Theorem 6.2.6).

The final step missing from the work described above was the construction of a univalent universe using this internal approach. This problem was addressed in Chapter 7 where we extended the internal language with a modality for manipulating global elements of a type, which allowed us to axiomatise the property needed in order to construct a univalent universe, namely, the fact that the interval is tiny (Section 7.3).

Therefore, in conclusion, this thesis presents a complete internal presentation of the model given by Cohen et al. [18] including the construction of univalent universes. It generalises the model to other settings by axiomatising, in the internal language, sufficient properties to carry out the model construction. Finally, it also presents an alternative proof of the univalence axiom by decomposing it into sub-axioms which it then shows are validated, fairly straightforwardly, in any topos satisfying the axioms.

This thesis is accompanied by an extensive Agda development, available at <https://doi.org/10.17863/CAM.35681>, which formalises all of the internal language arguments, as well as the decomposition of univalence given in Section 6.1. This Agda development was invaluable for verifying the work contained in this thesis; for identifying mistakes and missing steps in early versions of many proofs. The Agda was not only useful for checking

work done on paper, but was also extremely useful for experimenting with new ideas and constructions.

8.1 Future work (higher inductive types)

The obvious direction for future work would be to investigate the extent to which we can carry out the construction of higher inductive types (HITs) [62, Chapter 6] using the internal language approach. Recently, a new paper by Coquand, Huber and Mörtberg [21] was published which presents the semantics for certain HITs in the cubical sets model, and takes a first step towards defining a general schema for HITs. Initial experiments suggest that it is possible to repeat some of these constructions in the internal language by using quotient inductive types (QITs) [5]. Although further investigation is required to see how much can be done using this approach, as well as fully justifying the semantics of QITs, the initial results seem promising.

The key insight in the recent work [21] is that the fibration structure described in Definition 5.3.3 can be decomposed into two new operations called *homogeneous-fibration structure* and *transport structure*. A family has a homogeneous-fibration structure whenever each of its fibres is a fibrant object (a fibration over 1). In Remark 5.3.6 we observed that every fibration satisfies this property, but that not every family satisfying this property is a fibration. A transport structure captures the extra (non-fibrewise) structure that differentiates fibrations and families of fibrant objects. We now formally define both of these concepts in the internal type theory.

Definition 8.1.1 (Homogeneous-fibration structure). A *homogeneous-fibration structure* for a family $A : \Gamma \rightarrow \mathcal{U}$ over a type $\Gamma : \mathcal{U}$ is an element of the type $\mathbf{isHFib} A$ where $\mathbf{isHFib} : \{\Gamma : \mathcal{U}\}(A : \Gamma \rightarrow \mathcal{U}) \rightarrow \mathcal{U}$ is defined by:

$$\mathbf{isHFib} \{\Gamma\} A \triangleq (e : \{0, 1\})(x : \Gamma)(\varphi : \mathbf{Cof})(f : [\varphi] \rightarrow \mathbf{I} \rightarrow Ax) \rightarrow \{a_0 : Ax \mid (\varphi, f) @ e \nearrow a_0\} \rightarrow \{a_1 : Ax \mid (\varphi, f) @ \bar{e} \nearrow a_1\} \quad (8.1)$$

This definition is equivalent to asking that every fibre is a fibrant object. We have simply eliminated the redundant $p : \mathbf{I} \rightarrow 1$ that would appear in the direction use of that definition.

Definition 8.1.2 (Transport structure). A *transport structure* for a family $A : \Gamma \rightarrow \mathcal{U}$ is an element of the type $\mathbf{isTransp} A$ where $\mathbf{isTransp} : \{\Gamma : \mathcal{U}\}(A : \Gamma \rightarrow \mathcal{U}) \rightarrow \mathcal{U}$ is defined by:

$$\mathbf{isTransp} \{\Gamma\} A \triangleq (e : \{0, 1\})(p : \mathbf{I} \rightarrow \Gamma)\{\varphi : \mathbf{Cof} \mid \varphi \Rightarrow \forall(i : \mathbf{I}). pi = pe\} \rightarrow \{a_0 : A(pe)\} \rightarrow \{a_1 : A(p\bar{e}) \mid (\varphi, \bar{a}_0) \nearrow a_1\} \quad (8.2)$$

where $\bar{a}_0 : [\varphi] \rightarrow A(p\bar{e})$ is a_0 regarded as a partial element of $A(p\bar{e})$.

We can then show the decomposition of fibration structure into homogeneous-fibration structure and transport structure that we mentioned above.

Theorem 8.1.3. *There are functions*

$$\mathbf{fromFib} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \mathbf{isFib} A \rightarrow \mathbf{isHFib} A \times \mathbf{isTransp} A \quad (8.3)$$

$$\mathbf{toFib} : \{\Gamma : \mathcal{U}\}\{A : \Gamma \rightarrow \mathcal{U}\} \rightarrow \mathbf{isHFib} A \rightarrow \mathbf{isTransp} A \rightarrow \mathbf{isFib} A \quad (8.4)$$

Proof. Given $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$ and $\alpha : \mathbf{isFib} A$ we define $\mathbf{fromFib} \alpha = (h_A, t_A)$ where $h_A : \mathbf{isHFib} A$ and $t_A : \mathbf{isTransp} A$ are defined by

$$\begin{aligned} h_A e x \varphi f a_0 &\triangleq \alpha e (\lambda_- \rightarrow x) \varphi f a_0 \\ t_A e p \varphi a_0 &\triangleq \alpha e p \varphi (\lambda_{--} \rightarrow a_0) a_0 \end{aligned}$$

For the reverse direction, take $\Gamma : \mathcal{U}$, $A : \Gamma \rightarrow \mathcal{U}$, $h_A : \mathbf{isHFib} A$ and $t_A : \mathbf{isTransp} A$ and define a composition structure $\mathbf{toFib} h_A t_A : \mathbf{isFib} A$ like so:

$$\mathbf{toFib} h_A t_A e p \varphi f a_0 \triangleq h_A e (p\bar{e}) \varphi f' (t_A e p \perp a_0)$$

where $f' : [\varphi] \rightarrow \mathbf{I} \rightarrow A(p\bar{e})$ is defined by

$$f' u i \triangleq t_A e (\lambda(j : \mathbf{I}) \rightarrow p(i \sqcup_e j)) (i = \bar{e}) (f u i)$$

and where \sqcup_e is given by $\sqcup_0 \triangleq \sqcup$ and $\sqcup_1 \triangleq \sqcap$. □

To understand the advantage of this decomposition, consider how we might attempt to construct higher inductive types. As an example, take the suspension [62, Section 6.5]. This is a parameterised-HIT defined, in the language of the external cubical sets \mathbf{CwF} (Section 3.3), like so: given any type $A \in \mathbf{FTy}(\Gamma)$ we have a new type $\Sigma A \in \mathbf{FTy}(\Gamma)$, called the suspension of A , with terms

$$\begin{aligned} \mathbf{north} &\in \mathbf{Ter}(\Gamma \vdash \Sigma A) \\ \mathbf{south} &\in \mathbf{Ter}(\Gamma \vdash \Sigma A) \\ \mathbf{merid}(a) &\in \mathbf{Ter}(\Gamma \vdash \mathbf{Path}_A(\mathbf{north}, \mathbf{south})) \end{aligned}$$

for every $a \in \mathbf{Ter}(\Gamma \vdash A)$, which is initial (in a suitable sense) amongst types with this structure. Moreover, these constructions must be stable under reindexing in the usual sense.

Internally we might try to define such a construction by first defining a map $\Sigma : \mathcal{U} \rightarrow \mathcal{U}$, taking ΣA to be the type inductively generated by the constructors,

$$\begin{aligned} \text{north} &: \Sigma A \\ \text{south} &: \Sigma A \\ \text{merid} &: A \rightarrow \mathbb{I} \rightarrow \Sigma A \end{aligned}$$

quotiented by the relation that makes $\text{merid}(a, 0) = \text{north}$ and $\text{merid}(a, 1) = \text{south}$ for all $a : A$. This would then lift to $\Sigma : \{\Gamma : \mathcal{U}\}(A : \Gamma \rightarrow \mathcal{U}) \rightarrow \Gamma \rightarrow \mathcal{U}$ in a fibrewise way:

$$\Sigma A x \triangleq \Sigma (A x)$$

The problem comes when trying to lift this further to a map $\Sigma : \{\Gamma : \mathcal{U}\} \rightarrow \mathbf{Fib} \Gamma \rightarrow \mathbf{Fib} \Gamma$ because, given that A is fibrant, there is no obvious reason why ΣA would also be fibrant.

One idea would be to try to freely add the results of any unsolved composition problems in ΣA to construct the “free fibrant family” on ΣA . This process is known as *fibrant replacement* and presents two major issues. The first is that the operation of fibrant replacement is not, in general, stable under substitution, but our interpretation of HITs must be stable under substitution. The second problem is that, without worrying about stability, the obvious definition of fibrant replacement does not preserve smallness of fibres. For example, given $\Gamma : \mathcal{U}_1$ and $A : \Gamma \rightarrow \mathcal{U}$ then we currently have $\Sigma A : \Gamma \rightarrow \mathcal{U}$, but if we were to fibrantly replace ΣA in the obvious way then we would get a map $\Gamma \rightarrow \mathcal{U}_1$, i.e. one universe level up (equal to the universe level of the base, Γ). These issues, and the semantics of HITs more generally, are discussed in [45].

However, since having a *homogeneous* composition structure is a fibrewise property, it does seem to be possible to construct the “homogeneous-fibrant replacement” of a family, and this process is stable under reindexing and preserves smallness. Moreover, it is possible to show that if a family A has a transport structure then its homogeneous-fibrant replacement does too.

Now, note that whenever $A : \Gamma \rightarrow \mathcal{U}$ has a transport structure $t_A : \mathbf{isTransp} A$ then we can define a transport structure $t_{\Sigma A} : \mathbf{isTransp}(\Sigma A)$ like so:

$$\begin{aligned} t_{\Sigma A} \text{ e } p \ \varphi \ \text{north} &\triangleq \text{north} \\ t_{\Sigma A} \text{ e } p \ \varphi \ \text{south} &\triangleq \text{south} \\ t_{\Sigma A} \text{ e } p \ \varphi \ (\text{merid } a \ i) &\triangleq \text{merid } (t_A \text{ e } p \ \varphi \ a) \ i \end{aligned}$$

This respects the quotient on ΣA and satisfies the requirements of a transport structure.

Therefore we can construct the suspension as follows. Take a fibration $(A, \alpha) : \mathbf{Fib} \Gamma$ and let $\Sigma' A$ be the homogeneous-fibrant replacement of ΣA . From above, we know that

whenever A is fibrant, and hence has a transport structure, then ΣA has a transport structure as well, and hence so does its homogeneous-fibrant replacement. By definition $\Sigma' A$ is always homogeneous-fibrant. Therefore, by (8.4), $\Sigma' A$ is fibrant. Moreover, the “free” part of the homogeneous-fibrant replacement ensures that $\Sigma' A$ will still satisfy the correct elimination principle amongst fibrant types. Therefore $\Sigma' A$ correctly models the suspension of A .

The work described here is still very much in the initial stages. A formalisation of the argument given above can be found in the Agda development which accompanies this thesis at <https://doi.org/10.17863/CAM.35681>. As alluded to at the start of the chapter, the construction of the homogeneous-fibrant replacement requires the existence of quotient inductive types (QITs) [5]. At present, these are being simulated in Agda using some potentially unsound extensions and any future work in this direction would need to establish the soundness of these construction in general topoi, or specifically in the topos of cubical sets.

A similar approach in this direction can also be seen in Simon Boulier’s thesis [13, Section 5.2.5], where he constructs homotopy pushouts in the internal language (which he calls $\mathbb{I}TT$) by using a quotient type and then taking a fibrant replacement.

Bibliography

- [1] P. Aczel and M. Rathjen. Notes on constructive set theory. Book draft, 2010.
- [2] J. Adámek, J. Rosický, and E. M. Vitale. *Algebraic Theories: a Categorical Introduction to General Algebra*, volume 184 of *Cambridge Tracts in Mathematics*. Cambridge University Press, 2010.
- [3] Agda-flat. <https://github.com/agda/agda/tree/flat>.
- [4] Agda Project. wiki.portal.chalmers.se/agda.
- [5] T. Altenkirch and A. Kaposi. Type theory in type theory using quotient inductive types. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '16, pages 18–29, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3549-2. doi: 10.1145/2837614.2837638. URL <http://doi.acm.org/10.1145/2837614.2837638>.
- [6] C. Angiuli, G. Brunerie, T. Coquand, K.-B. Hou (Favonia), R. Harper, and D. R. Licata. Cartesian cubical type theory. <https://github.com/dlicata335/cart-cube/blob/master/cart-cube.pdf>, December 2017.
- [7] S. Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2016.
- [8] S. Awodey. A cubical model of homotopy type theory. *arXiv preprint arXiv:1607.06413*, 2016. <https://arxiv.org/abs/1607.06413>.
- [9] R. Balbes and P. Dwinger. *Distributive Lattices*. University of Missouri Press, 1975.
- [10] M. Bezem and T. Coquand. A Kripke model for simplicial sets. *Theoretical Computer Science*, 574:86 – 91, 2015.
- [11] M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets. In R. Matthes and A. Schubert, editors, *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, volume 26 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 107–128. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014.

- [12] L. Birkedal, A. Bizjak, R. Clouston, H. B. Grathwohl, B. Spitters, and A. Vezzosi. Guarded Cubical Type Theory: Path Equality for Guarded Recursion. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:17, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-022-4. doi: 10.4230/LIPIcs.CSL.2016.23. URL <http://drops.dagstuhl.de/opus/volltexte/2016/6563>.
- [13] S. Boulier. Extending type theory with syntactical models. 2018. PhD thesis.
- [14] N. Bourbaki. Univers. Théorie des topos et cohomologie étale des schémas. Tome 1: Théorie des topos. In *Séminaire de Géométrie Algébrique du Bois-Marie (SGA 4)*, volume 269 of *Lecture Notes in Computer Science*, pages 185–217. Springer-Verlag Berlin, 1972.
- [15] R. Brown and G. H. Mosa. Double categories, 2-categories, thin structures and connections. *Theory and Applications of Categories*, 5(7):163–175, 1999.
- [16] R. Clouston, B. Manna, R. E. Møgelberg, A. M. Pitts, and B. Spitters. Modal dependent type theory and dependent right adjoints. arXiv:1804.05236, April 2018.
- [17] J. Cockx and A. Abel. Sprinkles of extensionality for your vanilla type theory. Abstract of a talk at TYPES 2016, Novi Sad, Serbia, 2016.
- [18] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In Tarmo Uustalu, editor, *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, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-030-9. doi: 10.4230/LIPIcs.TYPES.2015.5. URL <http://drops.dagstuhl.de/opus/volltexte/2018/8475>.
- [19] T. Coquand. A cubical type theory, August 2015. Slides of an invited talk at Domains XII, University of Cork, Ireland, www.cse.chalmers.se/~coquand/cork.pdf.
- [20] T. Coquand and N. A. Danielsson. Isomorphism is equality. *Indagationes Mathematicae*, 24(4):1105–1120, 2013.
- [21] T. Coquand, S. Huber, and A. Mörtberg. On higher inductive types in cubical type theory. 2018. URL <http://doi.acm.org/10.1145/3209108.3209197>.
- [22] V. de Paiva and E. Ritter. Fibrational modal type theory. *Electronic Notes in Theoretical Computer Science*, 323:143–161, 2016. Proceedings of the Tenth Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2015).

- [23] 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. ISBN 978-3-540-61780-8.
- [24] M. Escardo. A small type of propositions *à la* Voevodsky in Agda, August 2015. <http://www.cs.bham.ac.uk/~mhe/impredicativity/>.
- [25] D. Frumin and B. van den Berg. A homotopy-theoretic model of function extensionality in the effective topos. *ArXiv e-prints*, arXiv:1701.08369, 2018.
- [26] P. Gabriel and F. Ulmer. *Lokal Präsentierbare Kategorien*, volume 221 of *Lecture Notes in Mathematics*. Springer-Verlag, 1971.
- [27] N. Gambino and C. Sattler. The Frobenius condition, right properness, and uniform fibrations. *Journal of Pure and Applied Algebra*, 221:3027–3068, 2017.
- [28] P. G. Goerss and J. F. Jardine. *Simplicial Homotopy Theory*. Modern Birkhäuser Classics. Birkhäuser Basel, 2009.
- [29] M. Hofmann. Extensional concepts in intensional type theory. 1995. PhD thesis. University of Edinburgh.
- [30] 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.
- [31] M. Hofmann and T. Streicher. Lifting Grothendieck universes. Unpublished note, <https://www2.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf>, 1997.
- [32] M. Hofmann and T. Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36, pages 83–111. Oxford Logic Guides, Oxford University Press, 1998.
- [33] S. Huber. Canonicity for cubical type theory. *CoRR*, abs/1607.04156, 2016. URL <http://arxiv.org/abs/1607.04156>.
- [34] S. Huber. *Cubical Interpretations of Type Theory*. Phd thesis, University of Gothenburg, 2016.
- [35] J. M. E. Hyland. The effective topos. In A. S. Troelstra and D. van Dalen, editors, *The L. E. J. Brouwer Centenary Symposium*, pages 165–216. North-Holland, Amsterdam, 1982.

- [36] P. T. Johnstone. On a topological topos. *Proceedings of the London Mathematical Society*, s3-38(2):237–271, 1979.
- [37] P. T. Johnstone. *Sketches of an Elephant, A Topos Theory Compendium, Volumes 1 and 2*. Number 43–44 in Oxford Logic Guides. Oxford University Press, 2002.
- [38] D. M. Kan. Abstract homotopy. i. *Proceedings of the National Academy of Sciences*, 41(12):1092–1096, 1955.
- [39] C. Kapulkin and P. L. Lumsdaine. The simplicial model of univalent foundations (after Voedodsky). *ArXiv e-prints*, arXiv:1211.2851v4 [math.LO], June 2016.
- [40] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, 1986.
- [41] F. W. Lawvere. Functorial semantics of algebraic theories. *Proceedings of the National Academy of Sciences of the United States of America*, 50(1):869–872, 1963.
- [42] F. W. Lawvere. Toward the description in a smooth topos of the dynamically possible motions and deformations of a continuous body. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 21(4):377–392, 1980.
- [43] F. W. Lawvere. Axiomatic cohesion. *Theory and Applications of Categories*, 19(3):41–49, 2007.
- [44] D. R. Licata, I. Orton, A. M. Pitts, and B. Spitters. Internal universes in models of homotopy type theory. In H. Kirchner, editor, *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [45] P. L. Lumsdaine and M. Shulman. Semantics of higher inductive types. May 2017. arXiv:1705.07088.
- [46] P. L. Lumsdaine and M. A. Warren. The local universes model: An overlooked coherence construction for dependent type theories. *ACM Trans. Comput. Logic*, 16(3):23:1–23:31, July 2015.
- [47] S. MacLane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics 5. Springer, 1971.
- [48] M. E. Maietti. Modular correspondence between dependent type theories and categories including pretopoi and topoi. *Mathematical Structures in Computer Science*, 15:1089–1149, 2005.

- [49] P. J. May. *A Concise Course in Algebraic Topology*. Chicago Lectures in Mathematics. University of Chicago Press, 1999.
- [50] I. Moerdijk and E. Palmgren. Type theories, toposes and constructive set theory: predicative aspects of ast. *Annals of Pure and Applied Logic*, 114(1):155 – 201, 2002. ISSN 0168-0072. doi: [https://doi.org/10.1016/S0168-0072\(01\)00079-3](https://doi.org/10.1016/S0168-0072(01)00079-3). URL <http://www.sciencedirect.com/science/article/pii/S0168007201000793>. Troelstra Festschrift.
- [51] A. Nanevski, F. Pfenning, and B. Pientka. Contextual modal type theory. *ACM Trans. Comput. Logic*, 9(3):23:1–23:49, 2008.
- [52] F. Pfenning and R. Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11:511–540, 2001.
- [53] A. M. Pitts. Categorical logic. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures*, pages 40–128. Oxford University Press, 2000. ISBN 0-19-853781-6.
- [54] 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.
- [55] E. Riehl and M. Shulman. A type theory for synthetic ∞ -categories. arXiv:1705.07442, December 2017.
- [56] G. Rosolini. *Continuity and Effectiveness in Topoi*. PhD thesis, University of Oxford, 1986.
- [57] C. Sattler. The equivalence extension property and model structures. arXiv:1704.06911, August 2017.
- [58] M. Shulman. Brouwer’s fixed-point theorem in real-cohesive homotopy type theory. *Mathematical Structures in Computer Science*, pages 1–86, 2017.
- [59] T. Streicher. Universes in toposes. In L. Crosilla and P. Schuster, editors, *From Sets and Types to Topology and Analysis, Towards Practicable Foundations for Constructive Mathematics*, volume 48 of *Oxford Logic Guides*, chapter 4, pages 78–90. Oxford University Press, 2005.
- [60] A. Swan. An algebraic weak factorisation system on 01-substitution sets: A constructive proof. *Journal of Logic & Analysis*, 8(1):1–35, 2016.

- [61] T. Uemura. Cubical assemblies and independence of the propositional resizing axiom. arXiv:1803.06649, April 2018.
- [62] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations for Mathematics*. <http://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.
- [63] B. van den Berg. Path categories and propositional identity types. *ArXiv e-prints*, arXiv:1604.06001 [math.CT], April 2016.
- [64] V. Voevodsky. C-systems defined by universe categories: Presheaves. arXiv:1706.03620, June 2017.
- [65] D. Yetter. On right adjoints to exponentiable functors. *Journal of Pure and Applied Algebra*, 45:287–304, 1987. Corrections in volume 58:103–105, 1989.