Material Set Theory in Homotopy Type Theory

Elisabeth Stenholm

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Acknowledgements

There are many people who have supported me during my PhD studies and I will try my best to express my gratitude to them here. First and foremost, I would like to thank my two main supervisors: Håkon Gylterud and Anders Mörtberg.

Håkon, I am beyond grateful for your endless patience and support, which has been constant since day one. Your unwavering optimism is a source of comfort and joy in my life, and it would certainly have been harder for me to get to this point were it not for you cheering me on. Although things did not quite turn out as expected, I wouldn't change it for the world. I am glad to have met such a wonderful person as you.

To Anders, I am forever grateful that you took me on as a PhD student during the final part of my studies, outside your other responsibilities and while navigating life with a newborn. It has been equally as enjoyable working with you this time as it was during my master's thesis. My thanks also go out to Mikhail Barash for agreeing to be my co-supervisor at the same time. You always give good advice.

I want to thank my co-authors, Daniel Gratzer and Niccolò Veltri, for a great experience writing collaborative papers. More broadly, I am grateful to the HoTT community for being open and welcoming to me. I have met so many incredibly talented and wonderful people at the conferences I attended during my studies, and I would be lucky to meet these people again.

To my fellow PhD students at the PUT-group and the other friends I have made since coming to Bergen: life is nothing without a good group of friends, and you have certainly given me this. Thanks for all the lunches with good company and good conversation. We definitely created the best hangout spot at the department!

My thanks also go to my family, and especially to my sister, Alice, for always being there for me whenever I needed someone to talk to. I am grateful to have a sibling such as you.

To the people who are no longer in my life, but who supported me through large parts of my studies, I am very grateful.

Abstract

This thesis investigates models of material set theory in Homotopy Type Theory (HoTT), that is, interpretations of the language of set theory into HoTT such that the interpretations of the set theoretic axioms can be shown to hold.

An important feature of a given model is how the equality relation is interpreted. In HoTT, the type for equality between two elements is called the *identity type*. The central theme of this thesis is that all its models interpret equality as the identity type.

The thesis is based on three papers. The first being a closer investigation of one specific model of constructive set theory, called the *iterative hierarchy of sets* (its construction predating this thesis) which is shown to have many desirable properties. In particular, it is shown that, as well as giving a model of material set theory in HoTT, it also gives a model of *extensional type theory* inside HoTT.

The second paper utilises that HoTT is a proof relevant framework which has higher structure, to give a higher level generalisation of the interpretations of the set theoretic axioms given in the first paper. It also contains the construction of a higher level generalisation of the model in that paper. At the first level of generalisation, this becomes a model of multisets, rather than sets. The construction is also made into an *internal universe* (this can be thought of as a "type of types") of n-types, which is itself an n-type.

The final paper investigates models of non-wellfounded set theory in HoTT. Non-wellfounded sets are sets which may contain an infinite membership chain: $\mathrm{a}_0 \ni \mathrm{a}_1 \ni \mathrm{a}_2 \ni \cdots$, as opposed to wellfounded sets where every such chain is required to be finite. The non-wellfounded sets are useful for modeling circular phenomena such as state machines or streams in computer science. The models in this paper interpret non-wellfounded sets as non-wellfounded trees. One family of models is obtained by dualising the construction of the iterative hierarchy of sets investigated in the first paper. A surprising result is that this dualisation does not yield a model of the usual kind of non-wellfounded sets (there are several), but rather of a different, less well known, notion of non-wellfounded sets.

Sammendrag

Denne avhandlingen undersøker modeller av materiell mengdelære i homotopi typeteori (HoTT), det vil si, tolkninger av språket for mengdelære inn i HoTT slik at tolkningene av aksiomene i mengdelære kan vises å holde.

En viktig egenskap av en gitt modell er hvordan likhetsrelasjonen blir tolket. I HoTT er typen for likhet mellom to elementer den såkalte *identitetstypen*. Det sentrale temaet for denne avhandlingen er at alle modellene tolker likhet som identitetstypen.

Avhandlingen består av tre artikler. Den første er en nærmere undersøkning av en spesifikk modell av konstruktiv mengdelære som kalles for det *iterative hierarkiet av mengder* (denne konstruksjonen ble laget før denne avhandlingen), som blir vist å ha mange ønskelige egenskaper. Det vises spesielt at, i tillegg til å gi en modell for materiell mengdelære i HoTT, så gir den også en modell for *ekstensionell typeteori* i HoTT.

Den andre artikkelen bruker at HoTT er et bevisrelevant rammeverk med høyere struktur, til å gi en høyereordens generalisering av tolkningene av mengdelæreaksiomene som ble gitt i den første artikkelen. Den inneholder også konstruksjonen av en høyereordens generalisering av modellen i den artikkelen. På det første nivået av generalisering blir dette en modell av multimengder istedet for mengder. Konstruksjonen blir også gjort til et *internt univers* (dette kan tenkes på som en "type av typer") av n-typer, som i seg selv er en n-type.

Den siste artikkelen undersøker modeller for ikke-velfundert mengdelære i HoTT. Ikke-velfunderte mengder er mengder som kan inneholde en uendelig lang rekke med medlemskap: $\mathrm{a}_0 \ni \mathrm{a}_1 \ni \mathrm{a}_2 \ni \cdots$, i motsettning til velfunderte mengder hvor hver slik rekke må være endelig. De ikke-velfunderte mengdene kan brukes til å modellere sirkulære fenomen som for eksempel tilstandsmaskiner eller strømmer i informatikk. Modellene i denne artikkelen tolker ikke-velfunderte mengder som ikke-velfunderte trær. En familie av modeller blir laget ved å dualisere konstruksjonen av det iterative hierarkiet av mengder som ble undersøkt i den første artikkelen. Et overraskende resultat er at denne dualiseringen ikke gir en modell av den vanlige typen ikke-velfunderte mengder (det finnes flere), uten istedet en annen, mindre velkjent versjon av ikke-velfunderte mengder.

List of publications

This thesis is based on the following three papers. As is customary in theoretical computer science, the authors are listed in alphabetical order.

1. Daniel Gratzer, Håkon Gylterud, Anders Mörtberg, Elisabeth Stenholm, (2024), *The Category of Iterative Sets in Homotopy Type Theory and Univalent Foundations*, accepted for publication in Mathematical Structures in Computer Science. Preprint available at: `https://arxiv.org/abs/2402.04893`

2. Håkon Gylterud, Elisabeth Stenholm, (2023), *Univalent Material Set Theory*. Preprint available at: `https://arxiv.org/abs/2312.13024`

3. Håkon Gylterud, Elisabeth Stenholm, Niccolò Veltri, (2024), *Terminal Coalgebras and Non-wellfounded Sets in Homotopy Type Theory*. Preprint available at: `https://arxiv.org/abs/2001.06696`

Introduction

Material set theory in the form of Zermelo–Fraenkel set theory (ZFC), is the established theoretical foundation for mathematics today. With this as the meta-theoretical framework, all constructions reduce to sets and all mathematical statements to questions of set equality and set membership. One might be doubtful if such a reduction accurately captures how a mathematician thinks about their work. For instance, in material set theory one can ask the question "Is 3 a member of 17?" as 3 and 17 are sets. If one encodes the natural numbers as the von Neumann ordinals (taking the successor of x to be $x \cup \{x\}$), the answer is "Yes", but if one encodes them as the Zermelo ordinals (taking the successor of x to be $\{x\}$) the answer is "No". How should we think about this fact? One approach is to say that the question is nonsensical or irrelevant, so it does not matter what the answer is. But, might it not be desirable to work within a framework in which one cannot make such nonsensical statements? Moreover, do working mathematicians really think about the natural number 3 as a particular set? Or do they rather think about the number in relationship to other mathematical entities? These questions have been debated among philosophers of mathematics and a proposed alternative is to have a *structuralist* foundation (Benacerraf, 1965)).

In a structuralist framework, objects are opaque, in the sense that one cannot inspect the objects themselves to see what they "are" and then derive properties about them. For example, one cannot say that the number 3 "is" the set $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$ and then conclude that 2, "being" the set $\{\emptyset, \{\emptyset\}\}$, is a member of 3. Instead, objects are defined by how they relate to each other and to the structures in which they live. For instance, in a structuralist foundation, one would define the number 3 as the natural number which is the successor to the natural number 2. This affords us all the information we need about the number 3 in order to successfully use it in our everyday mathematical endeavors. Taking the structuralist approach might thus be argued to be closer to how ordinary mathematics is conducted in the minds of mathematicians.

There are several structuralist frameworks for mathematics. Using category theory, the *Elementary Theory of the Category of Sets* (ETCS), as proposed by Lawvere (1964), is a straightforward structuralisation of ZFC. The mathematics in ETCS is classical.

Another structuralist framework is *Homotopy Type Theory* (HoTT) (The Univalent Foundations Program, 2013), which is a specific flavor of *Type Theory*: a framework for computations and constructive mathematics. This thesis investigates some particular models of material set theory in HoTT. The motivation for doing so is two-fold. First, constructing models of set theory in HoTT shows that any construction in set theory can also be carried out in HoTT. That is, we do not lose anything by using HoTT as our foundation of mathematics.* Second, viewing set theory from the lens of HoTT may give us new perspectives and insight. For instance, the entities in HoTT come equipped with higher level structure, as opposed to the entities in classical mathematics. This is used in one of the papers in this thesis to give a generalisation of material set theory based on this higher level structure. This generalisation gives an interesting connection between multisets and groupoids. As an added bonus, for those of us for whom material set theory never quite sat right as the foundation of mathematics and who did not particularly enjoy working within it: Doing set theory inside HoTT actually makes it fun!

1 Homotopy Type Theory

As this is a thesis on Homotopy Type Theory, let us start with a brief introduction to the topic. This section is not intended as a complete introduction, only as a quick overview so as to make the content of the thesis slightly more accessible. There are suggestions for further reading for those interested in HoTT at the end of the section.

The main concepts of type theory (Martin-Löf, 1984) are those of *types* and *terms*. Every term is associated with exactly one type. For example, there is a type of natural numbers, usually denoted $\mathbb{N}$, and the term 3 is a term of the type $\mathbb{N}$, which is denoted as $3 : \mathbb{N}$. Types and terms are defined by inference rules, which, in the structuralist spirit, one may think of as saying how these objects relate to other objects, without taking a stance on what the objects “are”. For instance, there are two rules for constructing terms of the type $\mathbb{N}$:

- There is a term $0 : \mathbb{N}$.
- Whenever $x : \mathbb{N}$, then there is a term $\mathrm{S}\, x : \mathbb{N}$.

*The flavor of set theory given by the models is by default constructive, since HoTT is constructive, but it is consistent to add classical axioms, such as AC to the ambient type theory, in which case one gets a model of classical set theory.

The type $\mathbb{N}$ also comes equipped with a recursion/induction principle, which gives us a way of constructing functions out of, and proving properties about, $\mathbb{N}$. Again, this recursion/induction principle is part of the definition of the natural numbers, rather than a derived concept.

The fact that type theory can be used as a logical framework for mathematics comes from the use of *dependent types*. These are types which might vary depending on the value of some term. As an example, consider the type $\mathrm{Vec}_{\mathbb{N}}\ n$ of vectors of natural numbers having length n. The type depends on the value of the variable n. For instance, $\mathrm{Vec}_{\mathbb{N}}\ 2$ is the type of vectors of length 2, which is not the same as the type $\mathrm{Vec}_{\mathbb{N}}\ 3$ of vectors of length 3. This might not seem groundbreaking, but the power of dependent types for expressing mathematical statements comes from the *Curry–Howard correspondence*, or the *propositions as types* interpretation of type theory. With this interpretation one considers a mathematical proposition to be a type and a term of this type to be a proof of the corresponding proposition. The type formers correspond to the logical connectives and the introduction and elimination rules for terms correspond to inference rules for proofs.

Propositional logic can be expressed without dependent types, but for universal and existential quantification, dependent types are needed. To see this, suppose you wish to show that the statement $\mathscr{P}$ holds for all natural numbers. What you have is one statement for each natural number, and you wish to construct a proof of each of your (countably infinitely many) statements. With the propositions as types interpretation, your statement then corresponds to some type $P(n)$, which depends on a variable $n : \mathbb{N}$, i.e. you have a type for each natural number n.

In order to express universal and existential quantification, we use Π-types and Σ-types, respectively. Starting with Π-types, given a type A and some type $B(a)$ dependent on $a : A$, we can construct the type

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

This type can both be seen as the type of dependent functions from A to B, that is, functions where the value of $a : A$ lies in the type $B(a)$, and as universal quantification of (the statement corresponding to) B. Terms are constructed by λ-abstraction. Given a term $b(a) : B(a)$ for each $a : A$, we can construct the following term:

$$\lambda\, a.b : \prod_{a:A} B(a),$$

which is the function sending a to $b(a)$. In the special case when B does not depend on A, the type is denoted by $A \to B$.

For Σ-types, given A and B as before, we can construct the type

$$\sum_{a:A} B(a).$$

This type can both be seen as the type of dependent pairs of terms from A and B, that is, pairs of terms $a : A$ and $b : B(a)$, and as existential quantification[†] of (the statement corresponding to) B. Given a term $a : A$ and $b : B(a)$ we can construct the following term:

$$(a, b) : \sum_{a:A} B(a),$$

which is the pair of the two given terms. In the special case when B does not depend on A, the type is denoted by $A \times B$.

The mathematics one gets by the propositions as types interpretation is by nature constructive, i.e. one cannot prove the axiom of choice or other non-constructive axioms (although it is consistent to add these as axioms). This essentially comes from the fact that type theory is also a formalism for computations.

The "homotopy" part of Homotopy Type Theory comes from the view of the *identity type*. With the propositions as types interpretation, the statement "x equals y" should be represented by some type. Martin-Löf (1975) introduced, for every type A and pair of terms $x, y : A$, the identity type, denoted $x = y$, which represents the statement that x equals y. This type might be empty, if x does not equal y, or contain some term $p : x = y$, which is then a proof that x equals y.

Since we can construct the identity type between any two terms of the same type, we can in particular consider the identity type $p = q$ between two terms $p, q : x = y$. How should we interpret this type? The terms p and q are proofs that x equals y. The type $p = q$ then represents the statement that the proofs p and q are equal. If one does not consider proofs to be objects available for manipulation, then such a statement might not make sense. In this case, we might want to say that any two proofs of equality (between the same two terms) are equal. This statement is not provable in type theory (Hofmann and Streicher, 1998), but it is consistent to add this as an axiom, which is usually called *Uniqueness of Identity Proofs* (UIP). The resulting theory is then proof irrelevant and has no higher structure.

Homotopy type theory takes another approach to the type $p = q$. In HoTT one instead allows for the terms of this type to be distinct, which

[†]There are two possible interpretations of existential quantification in HoTT, a proof relevant and a proof irrelevant one, but we will not go into this distinction here. The interested reader is directed to the resources at the end of the section.

corresponds to an interpretation of types as *spaces* and terms as *points in the space*. The type $x = y$ then represents the type of paths between the points x and y, the type $p = q$ is the type of homotopies between the paths p and q, and so on. This can be iterated indefinitely, creating a hierarchy of higher identity types. For a given type, this hierarchy might become trivial at some point, or it might contain non-trivial structure at every level. The point at which the hierarchy becomes trivial, if it does, is called the *type level* of the given type and a type of level n is called an *n-type*. Propositions, i.e. statements that are either true or false, but contain no more information, are interpreted in HoTT as those types A such that $x = y$ for all $x, y : A$. We will call such types *h-propositions*. Sets are interpreted as types at the next type level, i.e. such that for any two points $x, y : A$ and paths $p, q : x = y$ we have $p = q$. That is, types which can contain distinct points, but for which any two points can be identified in at most one way, i.e. types which satisfy UIP. Such types will be called *h-sets*.

The view of types as spaces also suggests an interpretation of the type $A = B$, for two types A and B, namely as (homotopy) equivalences of the spaces. The type of equivalences between A and B is usually denoted $A \simeq B$ and the interpretation of the identity type $A = B$ as homotopy equivalences is captured by adding to type theory the *Univalence Axiom* (UA), which gives an equivalence between the types $A = B$ and $A \simeq B$. In HoTT, one usually also adds *higher inductive types* (HITs) to the theory, which are types defined not only by their terms but also by adding non-trivial higher paths between terms.

We will not go into more detail about HoTT here, but there is much material out there for the interested reader. For an excellent introduction to HoTT for people new to the subject but with a background in mathematics, see Rijke (2022). For the main book about HoTT, usually referred to as "The HoTT Book", written as a collaborative effort by the HoTT community, see (The Univalent Foundations Program, 2013).

2 Equality in models of set theory in Homotopy Type Theory

Material set theory and HoTT can both be modeled in each other. In one direction, there is an interpretation of HoTT in set theory which formalises the idea of types as spaces by interpreting types as Kan complexes (Kapulkin and Lumsdaine, 2021). In the other direction, one wants to construct some type V together with a binary relation $\_ \in \_ : V \to V \to V$ and an interpretation of the language of set theory such that each type corresponding to an axiom is inhabited.

As part of the model, one has to give an interpretation of equality. That is, for each pair of terms $x, y : V$, one has to give a type $R(x, y)$ which represents the proposition that the sets represented by x and y are equal. There are two ways of doing this, either taking the type $R(x, y)$ to be the identity type $x = y$, or taking it to be some other type. Models taking the latter approach are called *setoid models*, where the word setoid refers to using some binary relation other than the identity type as equality. Previous work in this direction include the original model by Aczel (1978), the generalisation of Aczel's model by Gallozzi (2019) and the setoid model of non-wellfounded sets by Lindström (1989). The work by Aczel and Gambino (2006) constructs a model of set theory by creating an extension of type theory.

In this thesis we take the former approach, namely constructing models which interpret equality as the identity type. Let us here call such a model an *identity type model*, to distinguish it from setoid models. An example of an identity type model is the one in the HoTT Book, where the type of sets is constructed as a higher inductive type (Section 10.5). The model which this thesis builds upon is the one by Gylterud (2018), using the *type of iterative sets*. This model is equivalent to the HoTT Book model, but does not use higher inductive types for its construction.

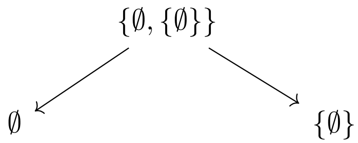
Identity type models are in a sense the ones which "fit" best inside HoTT. To see this, note that the principle of *indescernibility of identicals* holds for the identity type in HoTT. This is a philosophical principle about equality which states that if two objects are equal, then any property holds for one if and only if it holds for the other. In other words, there is no property which will allow you to *discern* a difference between two identical objects. This is instantiated in HoTT by the transport function, which, given a type family P and a term $p : x = y$, constructs an equivalence from $P\,x$ to $P\,y$. This means that any statement you can express in HoTT is invariant under the identity type. Thus, a model with the identity type for equality will fit well into HoTT in that any statement you can express will be invariant for two sets which are equal. This means in particular that any internalisation of a statement in first order logic (the language used for material set theory) will be invariant under set equality. If you have a setoid model, you do not get indescernibility of identicals. For each specific statement, you would need to manually show that it is invariant under the relation you have chosen. If the relation is distinct from the identity type there will be statements which are not invariant under the relation. Such a model is still a valid model and is useful for establishing consistency results, but it will not fit as seamlessly into HoTT as an identity type model, which is why we have chosen to focus on such models here.

The argument above could actually be generalised: given two logical theories which both have a notion of equality, a model of one inside the other, with equality in the first interpreted as equality in the second, would be the most seamless model of the first in the second (supposing that the second model has indescernibility of identicals). One can also apply this argument more generally when internalising mathematical concepts in HoTT. There might be several different ways of defining a concept, but the one which will be most ergonomic to use is one where the identity type is equivalent to the usual notion of equality for that concept. This is known as the *Structure Identity Principle* (Buss, Kohlenbach, and Rathjen, 2012) (the HoTT Book, Section 9.8).

3 Sets as trees

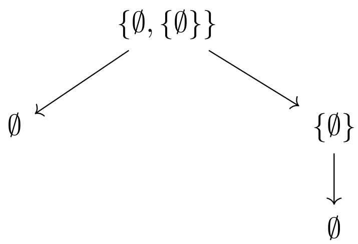
The earliest model of material set theory in type theory (HoTT had not entered the scene at this point in time) is the model by Aczel (1978), which is a setoid model. All models in this thesis build on the idea of Aczel's model. The intuition behind his model is to view material sets as trees. Let us take a moment to see how this works.

Consider the set $x = \{\emptyset, \{\emptyset\}\}$. We can "unfold" this set into a tree. Starting at the top level, x has two elements. Thus, we construct a tree which has x as its root and the two elements $\emptyset$ and $\{\emptyset\}$ as children to the root:



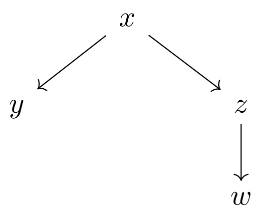
(1)

In the next step, we unfold the elements $\emptyset$ and $\{\emptyset\}$. The empty set has no elements, so as a (sub)tree, it has no children. The set $\{\emptyset\}$ has one element, and thus we unfold it into a (sub)tree which has one child. This child is the tree representing the empty set, which has no children. At this point, we are done, and we have the following complete tree:



(2)

In this way, we can go from a material set to a tree. But we can also go the other way—from trees to material sets. To see this, consider the following tree:



(3)

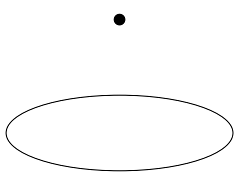
The root of the tree, x, has two children, so it represents a set with two elements: we write this as $\{\_,\_\}$. Continuing downwards in the tree, y represents the empty set since it has no children and z represents a set with one element, which gives us $\{\emptyset, \{\_\}\}$. Finally, w represents the empty set, which means that the whole tree represents the set $\{\emptyset, \{\emptyset\}\}$.

This idea of sets as trees is captured in Aczel's model by taking the type $\mathrm{W}_{A:U}\, A$, relative to a given a type universe U, as the type of sets. This type can be thought of as a type of trees. It has one constructor:

$$\mathrm{sup} : \prod_{A:U} (A \to \mathrm{W}_{A:U}\, A) \to \mathrm{W}_{A:U}\, A.$$

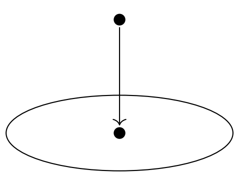
This constructor, together with an elimination principle, makes $\mathrm{W}_{A:U}\, A$ the initial algebra of the functor $X \mapsto (\sum_{A:U} A \to X)$.

Thinking of the terms of $\mathrm{W}_{A:U}\, A$ as trees, the constructor sup is a function such that whenever you have some indexed collection of trees, sup adds a root node and an edge from the root to the root of each of the given trees. As an example, consider again the tree above. How do we construct this as a term of $\mathrm{W}_{A:U}\, A$? Let us start from the bottom this time. The nodes y and w can be constructed by adding a root to the empty collection of trees. Specifically, let ex-falso : $\mathbf{0} \to \mathrm{W}_{A:U}\, A$ be the unique map from the empty type into $\mathrm{W}_{A:U}\, A$. Then the term sup $\mathbf{0}$ ex-falso can be visualized as:



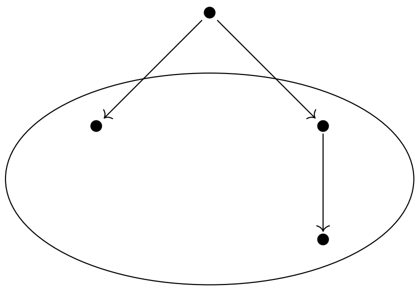
(4)

The subtree at the node z is then the one obtained by adding a root to the tree we just constructed, visualised as:



(5)

This is thus the term $\sup \mathbf{1}\,(\lambda\, 0.\, \sup \mathbf{0}\ \text{ex-falso})$. Finally, the full tree is the one obtained by adding a root to the collection of the trees corresponding to the nodes y and z. This can be visualised as:



(6)

This is then the term

$$\sup \mathbf{2}\ (\lambda\, 0.\, \sup \mathbf{0}\ \text{ex-falso}; \lambda\, 1.\, \sup \mathbf{1}\, (\lambda\, 0.\, \sup \mathbf{0}\ \text{ex-falso})) : \mathrm{W}_{A:U}\, A.$$

Note that we can construct sets with multiple copies of the same element. For instance, the term $\sup \mathbf{2}\ (\lambda\, x.\, \sup \mathbf{0}\ \text{ex-falso})$ represents the set $\{\emptyset, \emptyset\}$. We thus need to interpret equality in such a way that this term is identified with $\sup \mathbf{1}\, (\lambda\, 0.\, \sup \mathbf{0}\ \text{ex-falso})$, which represents the set $\{\emptyset\}$.

In Aczel's model, equality is therefore taken to be bisimulation. That is, he defines the binary relation $\sim$ by induction as:

$$(\sup\ A\, f) \sim (\sup\ B\, g) := \left(\prod_{a:A} \sum_{b:B} f\, a \sim g\, b \right) \times \left(\prod_{b:B} \sum_{a:A} f\, a \sim g\, b \right)$$

and interprets equality as $\sim$. Under this relation we have

$$\sup \mathbf{2}\ (\lambda\, \_.\, \sup \mathbf{0}\ \text{ex-falso}) \sim \sup \mathbf{1}\, (\lambda\, 0.\, \sup \mathbf{0}\ \text{ex-falso}),$$

as desired.

Aczel then uses $\sim$ to define the $\in$-relation. Viewing the terms of $\mathrm{W}_{A:U}\, A$ as trees, the idea is that a tree is a member of another tree if the set it represents is equal to the set represented by a subtree at one of the children of the root of the other tree. Thus, Aczel says that $x : \mathrm{W}_{A:U}\, A$ is an element of $\sup\ A\, f : \mathrm{W}_{A:U}\, A$ if there is some index $a : A$ such that $f\, a \sim x$. More precisely, he defines the relation (which we will here subscript with $\sim$)

$$\begin{aligned} \_ \in' \_ &: \mathrm{W}_{A:U}\, A \to \mathrm{W}_{A:U}\, A \to \mathrm{Type} \\ x \in' (\sup\ A\, f) &:= \sum_{a:A} f\, a \sim x. \end{aligned}$$

He then proceeds to show that his axioms for constructive set theory, defined in the same paper, hold in this model.

Gylterud (2019) considers the same type, $\mathrm{W}_{A:U}\, A$, but uses instead the identity type for equality and for the $\in$-relation. That is, the membership relation is defined as:

$$\begin{aligned} \_ \in \_ &: \mathrm{W}_{A:U}\, A \to \mathrm{W}_{A:U}\, A \to \mathrm{Type} \\ x \in (\sup\ A\, f) &:= \sum_{a:A} f\, a = x. \end{aligned}$$

The identity type on $\mathrm{W}_{A:U}\, A$ is tree isomorphism. Therefore, in this model, the two terms representing $\{\emptyset, \emptyset\}$ and $\{\emptyset\}$ are distinct, as the two corresponding trees are not tree isomorphic. So what one gets with this definition can be seen as a model of (higher dimensional) multisets.

The idea of the type of iterative sets (Gylterud, 2018) is to restrict to the terms of $\mathrm{W}_{A:U}\, A$ which represent proper sets, as opposed to multisets. This is achieved by taking the subtype V^0 of trees where the children of each node are distinct. This subtype thus contains the term representing $\{\emptyset\}$, but not the one representing $\{\emptyset, \emptyset\}$, as the two children of the root are equal in this tree. The definition of the membership relation is the same as in the multiset model, just restricted to the subtype. This gives us a model of (non-multi)set theory with the identity type for equality.

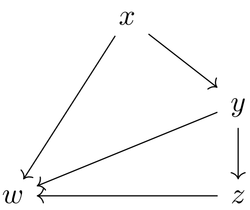
4 Non-wellfounded sets

So far, we have only considered wellfounded sets. These are sets for which there is no infinite descending membership chain:

$$a_0 \ni a_1 \ni a_2 \ni \cdots$$

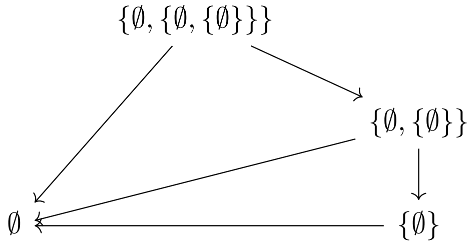
For instance, the sets represented by the terms of $\mathrm{W}_{A:U}\, A$ are all wellfounded, as the terms are wellfounded as trees. Many constructions in mathematics can be achieved by using such sets. However, there are some notions one might want to model mathematically, for which it is helpful to also allow sets which contain an infinite descending membership chain. Such sets are called non-wellfounded sets. There are many examples in both computer science and mathematics of non-wellfounded or circular phenomena: state machines, streams, non-Noetherian rings, etc. It is possible to model these using only wellfounded sets, but it then needs to be done in a roundabout way. For such constructions, it can be more convenient to work in a framework which allows non-wellfounded constructions from the start, giving a motivation for studying non-wellfounded set theory.

In order to understand the intuition behind non-wellfounded sets, let us revisit the representation of sets as wellfounded trees introduced in the previous section. However, let us here make a change to our previous representation of sets as trees, and instead represent them as wellfounded graphs[‡] (i.e. graphs with no infinite paths). This is acheived by having just one node representing each distinct set, otherwise the idea is the same. For instance, the set $\{\emptyset, \{\emptyset, \{\emptyset\}\}\}$ corresponds to the node x in the graph



(7)

Each of the nodes represent a (unique) set and the edges represent the $\in$-relation. Or, in the terminology of non-wellfounded set theory, this graph has a *decoration* of sets: each node u can be labeled with a set $d\,u$ such that $s \in d\,u$ if and only if there exists some child v of u in the graph such that $d\,v = s$. In other words, d is a decoration if $d\,u = \{d\,v \,|\, u \to v\}$ for all nodes u, where $u \to v$ means that there is an edge from u to v in the graph. For instance, w is decorated with the empty set, as it has no children in the graph, and thus the decoration is a set which has no elements. Consequently, z is decorated with the set $\{\emptyset\}$ as it has one child, w, which is decorated with $\emptyset$. Labeling the nodes in the graph with their decoration we get the following, decorated graph:



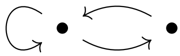
(8)

The axiom of foundation, which implies that all sets are wellfounded, can then be seen as the statement that there is no infinite path in the graph corresponding to a set. The idea of non-wellfounded set theory is to remove this restriction on infinite paths, and say that (some class of) non-wellfounded graphs have decorations of sets.

It turns out though, that for non-wellfounded sets, the axiom of extensionality is no longer enough to uniquely determine the equality relation. Recall that the axiom of extensionality states that two sets are equal if and

[‡]Throughout the thesis the word *graph* will mean *directed* graph.

only if they have the same elements. Now consider the following graph:


(9)

A decoration of this graph would be two sets x and y satisfying the following equations:

$$x = \{x, y\} \tag{10}$$
$$y = \{x\} \tag{11}$$

We may now ask whether x and y are equal. It turns out that both answers are compatible with the axiom of extensionality.

On one hand, suppose that $x \neq y$. Then there is some set, namely y, such that $y \in x$ but $y \notin y$, and hence the extensionality axiom holds. On the other hand, suppose that $x = y$. Then, for any set z it is the case that $z \in x$ if and only if $z \in y$, and so the extensionality axiom is again intact. Therefore, when allowing for non-wellfounded sets, we need to include extra information on when sets are equal, in such a way that the extensionality axiom still holds. This is done by replacing the axiom of foundation with some other axiom, which we call an *anti-foundation axiom.* As seen by the discussion above, this axiom needs to accomplish two things:

1. state which non-wellfounded sets are allowed and
2. uniquely determine the equality relation.

There are several possible anti-foundation axioms. For an introduction to non-wellfounded set theory and a comparison of the different anti-foundation axioms, see (Aczel, 1988). In the following sections we will consider the two of these which will be modeled in this thesis.

4.1 Aczel's anti-foundation axiom

Probably the most well-known anti-foundation axiom is what is usually referred to as *Aczel's anti-foundation axiom* (AFA) (Aczel, 1988). The idea behind AFA is to allow all graphs to represent sets. Equality is then uniquely determined by saying that any two decorations of a graph must be equal. Concretely, Aczel's anti-foundation axiom is the statement

AFA: *Every graph has a unique decoration.*

With AFA, we can define sets which contain themselves as elements. The simplest example of such a set is the Quine atom, q, which is the set represented by the following graph:

(12)

An alternative way to define this set is as the unique solution to the equation

$$x = \{x\}. \tag{13}$$

In general, given a finite graph, we can write down a set of equations which the decoration of the nodes satisfies. AFA states that any such set of equations has a unique solution. The uniqueness of the solution completely characterises equality. For instance, consider again equations (10) and (11). Taking $x = q$ and $y = q$ is a solution to this system of equations. Since any solution is unique, it follows that $x = y$.

4.2 Scott's anti-foundation axiom

There is another anti-foundation axiom, called *Scott's anti-foundation axiom* (SAFA), first proposed by Dana Scott.[§] It comes from his construction of non-wellfounded sets as non-wellfounded, *irredundant trees*. A tree is irredundant if there are no non-trivial tree automorphisms on it. Scott's anti-foundation axiom is then the statement that sets are irredundant trees.

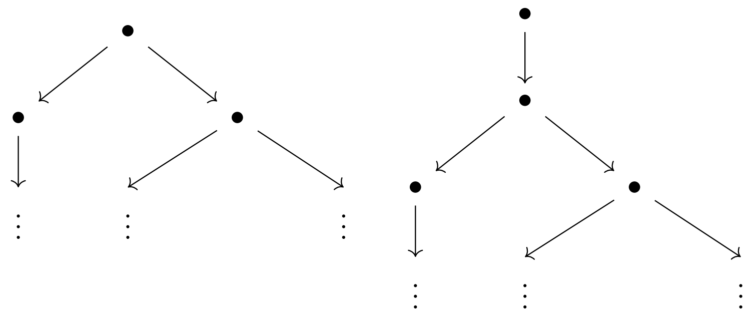
We can state this in a way that is closer to AFA, by rephrasing it as a statement about which graphs have decorations. The idea is that each node of a graph represents the set given by the unfolding tree starting at that node. In order to get the irredundant trees, we restrict to *Scott extensional* graphs (Aczel, 1988). A graph is Scott extensional if for any two nodes a and b such that their unfolding trees are tree isomorphic, we have $a = b$. Scott's anti-foundation axiom is then the statement (D'Agostino and Visser, 2002):

> **SAFA:** *Every Scott extensional graph has a unique injective decoration.*

Assuming SAFA, rather than AFA, we get a different answer to the question of whether the decorations x and y of the nodes of the graph (9) are

[§]This axiom was originally defined in an unpublished paper (Scott, 1960), which the author has not managed to find a copy of.

equal. The unfolding trees corresponding to the nodes can be (partly) visualised as:



(14)

These are not tree isomorphic, and thus the graph is Scott extensional. Therefore, assuming SAFA, the graph has a unique injective decoration. The injectivity then implies that $x \neq y$.

5 The coalgebraic view of set theory

The intuition behind the models of material set theory in HoTT contained in this thesis is closely connected to the coalgebraic view of set theory. This view comes from the relationship between *set-like* binary relations and coalgebras for the powerset functor. A binary relation R is set-like if for every x, the collection of elements y such that $(y, x) \in R$ is a set. Given such a relation R, we construct the coalgebra map which sends an element x to the set of elements y such that $(y, x) \in R$, i.e. the set of related elements. In the other direction, given a coalgebra map we construct a binary relation R by taking $(y, x) \in R$ if y belongs to the set given by applying the coalgebra map to x.

Moreover, extensionality holds precisely when the coalgebra map is injective. Given a relation R, construct the coalgebra map as stated above. Then for elements x and y, the statement that they map to the same set is precisely the statement that for any z, we have $(z, x) \in R$ if and only if $(z, y) \in R$. Thus, the statement of the extensionality axiom is precisely the statement that the coalgebra map is injective.

As observed already by Rieger (1957), any fixed point of the powerset functor gives rise to a model of set theory, except for foundation. (Note that the coalgebra map of a fixed point is an isomorphism and therefore an injective map, so the extensionality axiom holds.) Rieger worked in a classical framework, but this result also holds in HoTT. Showing that the set theoretic axioms hold amounts to reflecting the corresponding constructions in the meta-theory, via the isomorphism, into the model. Since HoTT is constructive, it follows that the model obtained in this way becomes constructive.

Two special fixed points of functors, if they exist, are the initial algebra and the terminal coalgebra. For the powerset functor, the initial algebra is a model of set theory with foundation, while the terminal coalgebra becomes a model of set theory with AFA instead of foundation.

6 Higher level generalisation of set theory

In material set theory, the elementhood relation is proposition-valued. That is, for any two sets x and y, the statement $y \in x$ only contains information on whether it holds, and no further information or structure. Thus, a model of set theory where the $\in$-relation is interpreted as a family of h-propositions is in some sense the “correct” way of interpreting material set theory. For the type of iterative sets, V^0, the membership relation becomes proposition-valued.

However, in HoTT we have higher structure in the form of higher type levels. One can therefore ask what kind of structure we get if the membership relation is valued in n-types instead. In the second paper of the thesis we investigate this question and also construct a model at each type level. For instance, a model such that the $\in$-relation is valued in h-sets can be seen as the “correct” way to model multisets. That is, the type $y \in x$ may contain several distinct terms, but no higher structure above that. The type $y \in x$ may then be thought of as stating how many times y occurs in x. For instance, for the multiset $\{\emptyset, \emptyset\}$, the type $\emptyset \in \{\emptyset, \emptyset\}$ is equivalent to the finite type with two terms, which is an h-set. The investigation into h-set valued models gives an interesting connection between groupoids and multisets.

7 Formalisation

The contents of this thesis have been formalised in the proof assistant `Agda` (The Agda development team, 2024). The formalisation for all three papers builds on the `agda-unimath` library of univalent mathematics (Rijke et al., 2024). In the formalisation, the flag `--without-K`, which disables UIP, has been used and the univalence axiom postulated. The repository containing the formalisation of everything in the thesis, except Section 7 of Paper III, can be found at: `https://git.app.uib.no/hott/hott-set-theory`. The formalisation of Section 7 of Paper III can be found at: `https://github.com/niccoloveltri/aczel-mendler`. Throughout the thesis, the Agda logo will be placed next to results which have been formalised in Agda.

8 Organisation of the thesis

The first paper contains a closer study of the type of iterative sets, V^0. We show that it forms a Tarski style, internal universe of h-sets, which is itself an h-set, and which is closed under all the usual types and type formers. One desirable property of this universe is that the decoding holds up to definitional equality, which makes it very easy to work with.

We also investigate the category structure induced by the universe structure, and compare it to the category of h-sets. In particular, we exploit the fact that the type of iterative sets is an h-set to construct a Category with Families structure on it, which fails for the category of h-sets.

Paper II then generalises the construction of the type of iterative sets to construct the type of iterative n-types. It also defines the concept of an $\in$-structure and investigates such structures where the $\in$-relation is not valued in propositions, but rather in n-types. This gives rise to a higher level generalisation of material set theory. In particular, it gives an interesting connection between multisets and groupoids, and shows that multisets can be seen as the first level generalisation of sets.

Finally, Paper III investigates models of non-wellfounded sets in HoTT. It contains two different models, one of SAFA and one of AFA. The first is obtained by dualising the construction of the type of iterative n-types, found in Paper II. The second model is obtained by adapting the Aczel–Mendler construction of the terminal coalgebra for the powerset functor (Aczel and Mendler, 1989). This construction relies crucially on propositional resizing, and is due to Niccolò Veltri, one of the co-authors of the paper. Both models are such that equality is interpreted as the identity type.

References

Aczel, Peter (1978). “The Type Theoretic Interpretation of Constructive Set Theory”. In: *Logic Colloquium ’77*. Ed. by A. MacIntyre, L. Pacholski, and J. Paris. North–Holland, Amsterdam-New York, pp. 55–66.

— (1988). *Non-well-founded sets*. Vol. 14. CSLI Lecture Notes. With a foreword by Jon Barwise [K. Jon Barwise]. Stanford University, Center for the Study of Language and Information, Stanford, CA, pp. xx+137. ISBN: 0-937073-22-9.

Aczel, Peter and Nicola Gambino (Mar. 2006). “The generalised type-theoretic interpretation of constructive set theory”. In: *Journal of Symbolic Logic* 71.1, pp. 67–103. DOI: `10.2178/jsl/1140641163`. URL: `http://dx.doi.org/10.2178/jsl/1140641163`.

Aczel, Peter and Nax Mendler (1989). "A final coalgebra theorem". In: *Category Theory and Computer Science*. Ed. by David H. Pitt et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 357–365.

Benacerraf, Paul (1965). "What Numbers Could not Be". In: *The Philosophical Review* 74.1. Publisher: [Duke University Press, Philosophical Review], pp. 47–73. ISSN: 0031-8108. DOI: `10.2307/2183530`.

Buss, Samuel R., Ulrich Kohlenbach, and Michael Rathjen (July 25, 2012). "Mathematical Logic: Proof Theory, Constructive Mathematics". In: *Oberwolfach Reports* 8.4, pp. 2963–3002. ISSN: 1660-8933, 1660-8941. DOI: `10.4171/owr/2011/52`. URL: `https://ems.press/doi/10.4171/owr/2011/52`.

D'Agostino, Giovanna and Albert Visser (2002). "Finality regained: a coalgebraic study of Scott-sets and multisets". In: *Archive for Mathematical Logic* 41.3, pp. 267–298. ISSN: 0933-5846. DOI: `10.1007/s001530100110`. URL: `https://doi.org/10.1007/s001530100110`.

Gallozzi, Cesare (2019). "Homotopy type-theoretic interpretations of constructive set theories". In: *Mathematical Structures in Computer Science*, pp. 1–32. DOI: `10.1017/S0960129519000148`.

Gylterud, Håkon Robbestad (2018). "From multisets to sets in homotopy type theory". In: *The Journal of Symbolic Logic* 83.3, pp. 1132–1146. DOI: `10.1017/jsl.2017.84`.

— (2019). "Multisets in type theory". In: *Mathematical Proceedings of the Cambridge Philosophical Society*, pp. 1–18. DOI: `10.1017/S0305004119000045`.

Hofmann, Martin and Thomas Streicher (1998). "The groupoid interpretation of type theory". In: *Twenty-five years of constructive type theory (Venice, 1995)*. Vol. 36. Oxford Logic Guides. New York: Oxford Univ. Press, pp. 83–111.

Kapulkin, Krzysztof and Peter Lumsdaine (Mar. 2021). "The simplicial model of Univalent Foundations (after Voevodsky)". In: *Journal of the European Mathematical Society* 23.6, pp. 2071–2126. DOI: `10.4171/JEMS/1050`.

Lawvere, F. William (Dec. 1964). "An Elementary Theory of the Category of Sets". In: *Proceedings of the National Academy of Sciences of the United States of America* 52.6, pp. 1506–1511. ISSN: 0027-8424. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC300477/`.

Lindström, Ingrid (1989). "A construction of non-well-founded sets within Martin-Löf's type theory". In: *The Journal of Symbolic Logic* 54.1, pp. 57–64. ISSN: 0022-4812. DOI: `10.2307/2275015`. URL: `https://doi.org/10.2307/2275015`.

Martin-Löf, Per (Jan. 1, 1975). "An Intuitionistic Theory of Types: Predicative Part". In: *Studies in Logic and the Foundations of Mathematics*.

Ed. by H. E. Rose and J. C. Shepherdson. Vol. 80. Logic Colloquium ’73. Elsevier, pp. 73–118. DOI: 10.1016/S0049-237X(08)71945-1.

Martin-Löf, Per (1984). *Intuitionistic type theory. Notes by Giovanni Sambin.* Vol. 1. Studies in Proof Theory. Bibliopolis, Naples, pp. iv+91. ISBN: 88-7088-105-9.

Rieger, Ladislav (1957). “A contribution to Gödel’s axiomatic set theory, I”. eng. In: *Czechoslovak Mathematical Journal* 07.3. Publisher: Institute of Mathematics, Academy of Sciences of the Czech Republic, pp. 323–357. ISSN: 0011-4642. URL: https://eudml.org/doc/11888.

Rijke, Egbert (Dec. 21, 2022). *Introduction to Homotopy Type Theory.* arXiv: 2212.11082[math]. URL: http://arxiv.org/abs/2212.11082.

Rijke, Egbert et al. (2024). *The agda-unimath library.* URL: https://github.com/UniMath/agda-unimath/.

Scott, Dana (1960). “A Different Kind of Model for Set Theory”. Unpublished paper given at the 1960 Stanford Congress of Logic, Methodology and Philosophy of Science.

The Agda development team (2024). *Agda.* URL: https://github.com/agda/agda.

The Univalent Foundations Program (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics.* Institute for Advanced Study: homotopytypetheory.org. URL: http://homotopytypetheory.org/book.

Contents

Paper I

The Category of Iterative Sets in Homotopy Type Theory and Univalent Foundations

Daniel Gratzer, Håkon Robbestad Gylterud, Anders Mörtberg and *Elisabeth Stenholm*

Abstract

When working in Homotopy Type Theory and Univalent Foundations, the traditional role of the category of sets, $\mathcal{S}et$, is replaced by the category $h\mathcal{S}et$ of homotopy sets (h-sets); types with h-propositional identity types. Many of the properties of $\mathcal{S}et$ hold for $h\mathcal{S}et$ ((co)completeness, exactness, local cartesian closure, etc.). Notably, however, the univalence axiom implies that $\mathsf{Ob}\, h\mathcal{S}et$ is not itself an h-set, but an h-groupoid. This is expected in univalent foundations, but it is sometimes useful to also have a stricter universe of sets, for example when constructing internal models of type theory. In this work, we equip the type of iterative sets V^0, due to Gylterud (2018) as a refinement of the pioneering work of Aczel (1978) on universes of sets in type theory, with the structure of a Tarski universe and show that it satisfies many of the good properties of h-sets. In particular, we organize V^0 into a (non-univalent strict) category and prove that it is locally cartesian closed. This enables us to organize it into a category with families with the structure necessary to model extensional type theory internally in HoTT/UF. We do this in a rather minimal univalent type theory with W-types, in particular we do not rely on any HITs, or other complex extensions of type theory. Furthermore, the construction of V^0 and the model is fully constructive and predicative, while still being very convenient to work with as the decoding from V^0 into h-sets commutes definitionally for all type constructors. Almost all of the paper has been formalized in `Agda` using the `agda-unimath` library of univalent mathematics.

Acknowledgments: Anders would like to thank the late Vladimir Voevodsky for interesting him in the problem of set universes of sets in Univalent Foundations. In the last email Vladimir sent to Anders (on August 17, 2017) he asked about progress on this problem and mentioned that he had had a new idea for its solution. Unfortunately Vladimir never replied with any details about the solution, but we hope that he would have been pleased with the solution in this paper.

This work is also a testament to influence of the late Peter Aczel, who already in the late 1970s laid the foundations of this work. We are also indebted to the late Erik Palmgren, whose research, teaching and encouragement shaped the constructions herein.

The authors would like to thank Peter LeFanu Lumsdaine for valuable comments and discussions, and for stating our main result before we had done so ourselves. We are also very grateful to Henrik Forssell for helping us get a copy of the relevant chapter of the 1984 master thesis of Salvesen (1984), which only seems to exist in hard copy at the University of Oslo library.

Anders Mörtberg has been supported by the Swedish Research Council (Vetenskapsrådet) under Grant No. 2019-04545.

1 Introduction

Foundational theories of mathematics are concerned with collections of mathematical objects. Depending on the specific foundation, these collections might be called sets, classes or types. Among the many schisms of foundational theories, we find the one between *material* and *structural*. In a material foundational theory, the objects within a collection have an identity independent of the collection, and it is a sensible question to compare elements of different collections by equality. On the other hand, in a structural theory, the elements of a collection have no identity separate from the collection, and the important aspects of a collection are how its structure interacts with the other collections, for instance through functional relations.

Traditional set theories, such as Zermelo–Fraenkel set theory (ZF), are material foundational theories: there is a global elementhood relation and a global identity relation, meaning that all objects of the theory are possible elements of any set and can be compared to any other elements. This gives each set an inherent structure of membership relations between its elements, the elements of its elements, and so on. On the other hand, intensional Martin-Löf type theory (MLTT) (Martin-Löf, 1975) is a structural theory where the identity type compares only elements of the same type. Furthermore, in *Homotopy Type Theory and Univalent Foundations*[*] (HoTT/UF) (The Univalent Foundations Program, 2013) the Univalence Axiom (UA) (Voevodsky, 2010) can be seen in structural terms as saying that *structural equivalence is identity* (Awodey, 2013). HoTT/UF also distinguishes its types into *h-levels*/*n-types*: contractible types, h-proposition, h-sets, h-groupoids, and so on (Voevodsky, 2015). The h-sets correspond to sets as realized by other structural set theories, while types of higher h-levels are (higher-dimensional) groupoids which are not primitive objects in other foundational theories.

In type theory, the types are organized into universes, and UA is formulated relative to a specific universe. Thus, one can have both *univalent* and *non-univalent* universes living side by side. Univalence of a universe is

[*]We will refer to the book *Homotopy Type Theory: Univalent Foundations of Mathematics* (The Univalent Foundations Program, 2013) as the “the HoTT Book” throughout the rest of the paper.

mostly a positive feature: since every definable operation respects equality, structures can be transported along equivalences using univalence. One immediate observation is that in a univalent universe containing at least the booleans, the subuniverse of all h-sets in that universe cannot itself be an h-set. However, there are situations where it would be useful to have *a family of h-sets which itself is an h-set.* One such situation is when constructing the set model of type theory, as for example a category with families (CwF) (Dybjer, 1996), within HoTT/UF. The natural way of doing this would be to start with a univalent universe $\mathcal{U}$ and attempt to equip the corresponding category of h-sets ($\mathcal{hSet}_{\mathcal{U}}$) with a CwF structure. Part of the structure of a CwF is a presheaf Ty, which is usually formalized in HoTT/UF as a contravariant functor from the category into h-sets. The objects of the source category are thought of as contexts and the Ty-functor specifies what the types are in a given context. The natural choice when organizing $\mathcal{hSet}_{\mathcal{U}}$ into a CwF would be to let $\mathsf{Ty}(\Gamma) := \Gamma \to \mathsf{hSet}_{\mathcal{U}}$. Informally, the types in context Γ are simply families of h-sets (in $\mathcal{U}$) over Γ. However, since $\mathsf{hSet}_{\mathcal{U}}$ is not an h-set, this is ill-typed.

The agenda of this paper is to explore how one specific choice of a cumulative hierarchy of h-sets, namely the hierarchy V^0 as defined by Gylterud (2018), can be used as a (non-univalent) universe in HoTT/UF. In particular, we will study the structural and categorical properties of this inherently material structure and use it as the basis for a CwF structure. V^0 is a good starting point for our investigation into internal models of type theory since its construction uses only elementary type-formers: Π-types, Σ-types, W-types and identity types. In particular, neither the type V^0 itself nor the $\in$-relation defined on it require higher-inductive types, truncations or quotients. Since V^0 is an h-set, and it is closed under the usual type formers, it assembles into a model of MLTT with uniqueness of identity proofs and function extensionality, constructed within MLTT+UA. In this work, UA plays an essential role. We use it to, for instance, characterize the identity type of V^0. Using UA can sometimes result in constructions which lack the nice computational properties one has in bare MLTT. In our case however, since V^0 itself is built from elementary type-formers, many of the crucial equations, such as the ones for decoding type formers in V^0, hold definitionally. This makes V^0 extremely ergonomic from a formalization perspective.

Indeed, almost all of this paper has been formalized in the proof assistant `Agda` (The Agda development team, 2024)—a dependently typed programming language where one can construct both programs and proofs using the same syntax. Throughout the paper the `Agda` logo, , next to a result is a clickable link to the `Agda` code for that result. For basic results and constructions in HoTT/UF, we have used the `agda-unimath` library (Rijke, Stenholm, et al., 2024)—a large `Agda` library of formalized mathematics from the univalent point of view. Our formalization is in many places more general than the results presented in this paper as many constructions used here have a generalization to higher h-levels, and it is these generalized constructions that have been formalized. They are used for the Univalent Material Set Theory developed in Paper II.

1.1 Formal meta-theory and assumptions

While our formalization has been carried out in `Agda` on top of `agda-unimath`, the results in this paper can be obtained in a more modest type theory, and are modular in the sense that if you strengthen the underlying type theory with more types, such as quotients or more universes, these will be reflected in the internal model. The majority of the results assume that one works in MLTT extended with UA. By "MLTT" we take an intensional version of MLTT with the same types and type formers as in (Martin-Löf, 1982, Table 2), namely:

- Π-types, denoted $\prod_{x:A} B(x)$ with application denoted by juxtaposition and λ-abstraction by $\lambda(x : A).b(x)$.
- Σ-types, denoted $\Sigma_{x:A}B(x)$ with projections pr_1 and pr_2.
- W-types, denoted $\mathsf{W}_{x:A}\, B(x)$ with canonical elements $\mathsf{sup}\, A\, f$.
- Identity types, denoted $a = a'$, sometimes subscripted $a =_A a'$ for clarity, with reflective elements $\mathsf{refl}_a : a = a$.
- Binary sum types, denoted $A + B$ with injections inl and inr.
- Base types: empty, unit, bool, $\mathbb{N}$, with tt being the canonical element of unit, $\mathsf{true}, \mathsf{false}$ the elements of bool, and elements of $\mathbb{N}$ denoted by $\mathsf{0}$ and $\mathsf{s}\, n$. We also let $\mathsf{Fin}\, n$ denote the type with n elements.
- Universes, denoted $\mathcal{U}$, closed under the aforementioned type formers. For constructions needing more than one universe level, we will subscript them $\mathcal{U}_0, \mathcal{U}_1, \cdots, \mathcal{U}_\ell, \cdots$.

One important difference to (Martin-Löf, 1982) though is that we of course do not assume equality reflection and instead have intensional identity types as in (Martin-Löf, 1975). Another difference is that we, for convenience, assume definitional η for Σ-types. Our system is hence very similar to Martín Escardó's spartan MLTT (M. H. Escardó, 2019) and the basic system used in `UniMath` (Voevodsky, Ahrens, Grayson, et al., 2020), but with the addition of W-types. The only construction going beyond this is the construction of set quotients in Section 3.3, which assumes that the universe has set quotients. The construction of subuniverses in Section 3.5 also naturally assumes that the starting universe has subuniverses as well. But even with these extensions, the development is completely constructive and predicative, in particular we do not rely on LEM, AC, or any resizing principles (Voevodsky, 2011).

For convenience, we also rely on definitions and notational conventions from the the HoTT Book. Among these are:

- Definitional/judgmental equality is denoted by $\equiv$.
- Homotopy of functions is denoted $f \sim g$, with $\mathsf{refl\text{-}htpy}$ denoting $\lambda a.\mathsf{refl}$
- Type equivalence is denoted $A \simeq B$ with identity equivalence $\mathsf{id\text{-}equiv} : A \simeq A$.
- h-levels/n-types, in this paper we mainly work with types in $\mathsf{hProp}_{\mathcal{U}}$ and $\mathsf{hSet}_{\mathcal{U}}$, i.e. the h-propositions and h-sets in a given universe.
- We use pattern-matching freely in definitions and proofs, instead of explicit eliminators.

1.2 Contributions of the paper

While Section 2 sets the stage by recounting the definition of V^0, the foundation of this paper's contributions is built in Section 3: we show that V^0 forms a Tarski-style universe closed under Π-types, Σ-types, identity types, coproducts, set quotients, and that it contains basic types like empty, unit, bool, $\mathbb{N}$ and a hierarchy of subuniverses. Proposition 1 is central to this, as it characterizes the small types representable in V^0 as those which can be embedded into it. Since the decoding of all type formers is definitional, this gives an ergonomic universe of h-sets which itself is an h-set, which can be used in HoTT/UF. In Section 4 we shed light on the categorical properties of V^0. In particular, we show that it is a locally cartesian closed category, with finite limits and colimits, and that there is a full and faithful functor back to $\mathcal{hSet}_{\mathcal{U}}$ which preserves this structure. The final technical contribution is the construction of an extensional model of MLTT internal in MLTT+UA, based on V^0. This is done by giving CwF structure to V^0. The formalization of this includes contributions to `agda-unimath`, in particular the definition of a CwF with associated structure. A bibliographic contribution can be found in Section 6, where we compare our constructions to existing developments on the relationship between set theory and type theory. This relationship has taken many forms over the years, and goes back to the 1970s.

2 Definition of V^0 and its basic properties

The ideas behind V^0 trace back to *The type theoretic interpretation of constructive set theory* by Aczel (1978). In *op. cit.*, Aczel constructed a model of set theory in dependent type theory relying upon a non-trivial defined equality relation on the underlying type of the model in order to (hereditarily) force set-extensionality. This underlying type of the model is what we in modern parlance would call a W-type. To construct V^0, we opt to carve out a subtype of a W-type rather than take such a quotient. Instead of defining an equivalence relation which identifies the elements of the W-type which represent the same set, we shall identify a subtype of the W-type which contains only the canonical representations of each (iterative) set. Thus, we get a model of set theory in type theory where the equality *is* interpreted as the identity type and no further non-trivial identifications are required.

In this section we will review the definition of V^0 and prove some properties about it. In particular, we will show that V^0 is an h-set. In order to define V^0, we start by recalling the W-type Aczel used: "the unrestricted iterative hierarchy". It is the type of well-founded trees with branching types chosen freely from a fixed universe $\mathcal{U}_\ell$.

Definition 1 ( ). Given a universe $\mathcal{U}_\ell$, we define the type V^∞_ℓ as

$$\mathsf{V}^\infty_\ell := \mathsf{W}_{A:\mathcal{U}_\ell} A$$

We will usually omit the universe level ℓ for $\mathcal{U}_\ell$ and V^∞_ℓ, and write simply $\mathcal{U}$ and V^∞. When seeing V^∞ as a type of sets, an element $\mathsf{sup}\, A\, f : \mathsf{V}^\infty$ represents a set whose elements are indexed by the type $A : \mathcal{U}$. The function $f : A \to \mathsf{V}^\infty$ picks out the element at each index. Since the function f need not be injective, the same element can be picked out several times. Indeed, the role of Aczel's equivalence relation on this type was to erase this multiplicity. If we instead omit this further identification V^∞ can be seen as a type of multisets (Gylterud, 2019).

Notation 1. Given $x : \mathsf{V}^\infty$, we follow Aczel (1978) and define a pair of operations $\overline{x} : \mathcal{U}$ and $\widetilde{x} : \overline{x} \to \mathsf{V}^\infty$, as follows:

$$\overline{\mathsf{sup}\, A\, f} := A \qquad\qquad \widetilde{\mathsf{sup}\, A\, f} := f$$

We present two characterizations of equality in V^∞ as both are useful in different contexts. We note that both characterizations rely on univalence.

The first is an instance of a more general characterization of equality in W-types (Gylterud, 2019, Lemma 1). It states that two elements are equal if they have equivalent underlying indexing types and this equivalence is coherent with respect to the functions picking out the elements.

Theorem 1 ((Gylterud, 2019, Theorem 1),). *For two elements* $x, y : \mathsf{V}^\infty$ *the canonical map*

$$(x = y) \to \left(\sum_{e \colon \overline{x} \simeq \overline{y}} \widetilde{x} \sim \widetilde{y} \circ e \right)$$

which sends refl *to* $(\mathsf{id\text{-}equiv}, \mathsf{refl\text{-}htpy})$*, is an equivalence.*

The second characterization of equality in V^∞ states that two elements in V^∞ are equal when the functions picking out the elements are fiberwise equivalent. Intuitively, this means that they pick out the same elements the same number of times. One can think of this characterization of equality as a higher level generalization of the axiom of extensionality.

Theorem 2 ((Gylterud, 2019, Theorem 2),). *For two elements* $x, y : \mathsf{V}^\infty$ *the canonical map*

$$(x = y) \to \prod_{z : \mathsf{V}^\infty} \mathsf{fib}\, \widetilde{x}\, z \simeq \mathsf{fib}\, \widetilde{y}\, z$$

which sends refl *to* $\lambda z.\mathsf{id\text{-}equiv}$*, is an equivalence.*

Proof. We reproduce the proof here for convenience. We have the following chain of equivalences:

$$(x = y) \simeq \left(\sum_{e \colon \overline{x} \simeq \overline{y}} \widetilde{x} \sim \widetilde{y} \circ e \right) \simeq \left(\prod_{z : \mathsf{V}^\infty} \mathsf{fib}\, \widetilde{x}\, z \simeq \mathsf{fib}\, \widetilde{y}\, z \right)$$

The first equivalence is the one constructed in Theorem 1. The second equivalence is proven by Gylterud (2019, Lemma 5). One directly checks that the constructed equivalence computes as desired for refl. □

We will not dwell much on our structures being models of material set theory, but rather focus on their structural properties in this paper. However, we will define the elementhood relation on V^∞ following Gylterud (2019). This elementhood relation, and its well-foundedness, will be used in later constructions.

Definition 2 (Elementhood,). We define $\in \colon \mathsf{V}^\infty \to \mathsf{V}^\infty \to \mathcal{U}$ by

$$x \in y := \mathsf{fib}\, \widetilde{y}\, x$$

In particular, for canonical elements we get

$$(x \in \mathsf{sup}\, A\, f) \equiv \left(\sum_{a : A} f\, a = x \right)$$

The exensionality property of Theorem 2 can now be reformulated as an equivalence

$$(x = y) \simeq \left(\prod_{z : \mathsf{V}^\infty} z \in x \simeq z \in y \right)$$

By virtue of univalence, we obtain this extensionality result without taking quotients by set extensionality or bisimulation like Aczel does. In particular, we are able to avoid working with quotients or setoids while still achieving the equivalence above.

Note that $x \in y$ need not be an h-proposition, i.e., y could contain several instances of x. This is because, as discussed above, there is no restriction on the function $\widetilde{y}$ and its fibers. We will soon focus our attention to a subtype of V^∞ where these fibers are h-propositions, i.e., where they have at most one inhabitant. But first, we will look at how some familiar sets can be represented in V^∞.

We define the empty set as follows:

$$\emptyset := \mathsf{sup}\,\mathsf{empty}\,\mathsf{empty\text{-}elim}$$

This represents the empty set since for any $x : \mathsf{V}^\infty$, the type $x \in \emptyset$ is empty.

Given $x : \mathsf{V}^\infty$ we can construct the singleton containing x as follows:

$$\{x\} := \mathsf{sup}\,\mathsf{unit}\,(\lambda\_.x)$$

The type $x \in \{x\}$ is inhabited by $(\mathsf{tt}, \mathsf{refl})$. Indeed, for any $y : \mathsf{V}^\infty$, there is an equivalence $(y \in \{x\}) \simeq (y = x)$.

We can also construct the unordered pair of two elements $x, y : \mathsf{V}^\infty$:

$$\{x, y\} := \mathsf{sup}\,\mathsf{bool}\,(\lambda b.\mathsf{if}\; b\; \mathsf{then}\; x\; \mathsf{else}\; y)$$

For any $z : \mathsf{V}^\infty$, the type $z \in \{x, y\}$ is equivalent to $(z = x) + (z = y)$. Note in particular that the type $x \in \{x, x\}$ is equivalent to $(x = x) + (x = x)$, which contains at least two distinct elements. Thus, $\{x, x\}$ is a multiset which contains two copies of x. Using images one can whittle this down to an iterative set, see Paper II for details on the various types of pairing in higher h-levels.

In order to construct a universe of sets we need to ensure that the $\in$-relation is h-proposition valued, i.e., that any element occurs at most once in a set. As the type $x \in y$ is the type of homotopy fibers of $\widetilde{y}$ over x, this type would be an h-proposition if $\widetilde{y}$ was an embedding:

Definition 3 ((the HoTT Book, Definition 4.6.1),). A function $f : A \to B$ is an *embedding* if $\mathsf{ap}\, f\, x\, y : x = y \to f\,x = f\,y$ is an equivalence for all $x\,y : A$.

We write $\mathsf{is\text{-}emb}\, f$ for the type of proofs that f is an embedding and $f : A \hookrightarrow B$ for $\sum_{f:A\to B} \mathsf{is\text{-}emb}\, f$. A key observation about embeddings is:

Lemma 1 ((the HoTT Book, Lemma 7.6.2),). *A function $f : A \to B$ is an embedding if and only if it has h-propositional fibers.*

This motivates Gylterud's definition of iterative sets in HoTT/UF (Gylterud, 2018):

Definition 4 (Iterative sets,). We define $\mathsf{is\text{-}iterative\text{-}set} : \mathsf{V}^\infty \to \mathcal{U}$ as

$$\mathsf{is\text{-}iterative\text{-}set}\,(\mathsf{sup}\, A\, f) := (\mathsf{is\text{-}emb}\, f) \times \left(\prod_{a:A} \mathsf{is\text{-}iterative\text{-}set}\,(f\, a)\right)$$

The idea is to pick out those elements $x : \mathsf{V}^\infty$ for which the function that selects elements is an embedding and such that the elements of x satisfy the same criterion, recursively. This means that any $y : \mathsf{V}^\infty$ element is a member of x at most once and, consequently, x encode a set rather than a multiset. For these sets the $\in$-relation becomes h-proposition valued by Lemma 1, as desired.

Not all the elements in V^∞ are iterative sets. For example, the unordered pair $\{\emptyset, \emptyset\}$ from above is *not* an iterative set as the function in the definition is not an embedding.[†] On the other hand, the empty set, $\emptyset$, is an iterative set, since $\mathsf{empty\text{-}elim}$ is always an embedding, regardless of the codomain. Moreover, for any iterative set $x : \mathsf{V}^0$, the singleton $\{x\}$ is an iterative set since any map from an h-proposition into an h-set is an embedding (we will see that V^0 is an h-set in Theorem 3). Furthermore, if x and y are distinct iterative sets then $\{x, y\}$ is also an iterative set. To see this, it suffices to verify that the below map $\phi : \mathsf{bool} \to \mathsf{V}^0$ is an embedding if it is injective:

$$\phi\, b := \mathsf{if}\, b\, \mathsf{then}\, x\, \mathsf{else}\, y$$

Given $b_1, b_2 : \mathsf{bool}$, either $b_1 = b_2$, in which case we are done, or $b_1 \neq b_2$, in which case we get a path between x and y, from which the result follows by assumption.

Definition 5 (Type of iterative sets,). We define the type of *iterative sets* as follows:

$$\mathsf{V}^0 := \sum_{x:\mathsf{V}^\infty} \mathsf{is\text{-}iterative\text{-}set}\, x$$

[†]There is a different way to construct pairs which does yield an iterative set when applied to iterative sets. For details, see the proof of the the axioms of Myhill's constructive set theory given by Gylterud (2018).

We will extend the previously introduced notation to apply to iterative sets:

$$\overline{(\mathsf{sup}\, A\, f, p)} := A \qquad \widetilde{(\mathsf{sup}\, A\, f, p)} := f$$

Moreover, the elementhood relation $\in$ defined on V^∞ gives rise to an elementhood relation for V^0 given by projecting out the underlying elements in V^∞ and applying $\in$: for any $x, y : \mathsf{V}^0$ we let $x \in y := \mathsf{pr}_1\, x \in \mathsf{pr}_1\, y$. We use the same notation for both relations, as it will be clear from context which one is meant.

Lemma 2 ( ). *For all $x : \mathsf{V}^\infty$, the type* $\mathsf{is\text{-}iterative\text{-}set}\, x$ *is an h-proposition.*

Proof. This follows by induction on $x : \mathsf{V}^\infty$, together with the fact that being an embedding is an h-proposition. □

Corollary 1 ( ). *The projection $\mathsf{pr}_1 : \mathsf{V}^0 \to \mathsf{V}^\infty$ is an embedding, i.e. V^0 is a subtype of V^∞.*

Proof. This is an instance of the fact that for any type A and family P of h-propositions over A, the first projection $\mathsf{pr}_1 : \sum_{a:A} P\, a \to A$ is an embedding. □

Having an embedding $\mathsf{V}^0 \hookrightarrow \mathsf{V}^\infty$ means that equality in V^0 is exactly equality of the corresponding elements in V^∞. Since we have already characterized equality in V^∞, we can use this characterization to show that V^0 is an h-set.

Theorem 3 ( ). *V^0 is an h-set.*

Proof. For $(x, p), (y, q) : \mathsf{V}^0$ we have a chain of equivalences:

$$((x, p) =_{\mathsf{V}^0} (y, q)) \simeq (x =_{\mathsf{V}^\infty} y) \simeq \left(\prod_{z : \mathsf{V}^\infty} z \in x \simeq z \in y \right)$$

The first equivalence is the characterization of equality in subtypes. The second is Theorem 2. Note that $z \in x \equiv \mathsf{fib}\, \widetilde{x}\, z$, and $\widetilde{x}$ is an embedding by p. Thus $z \in x$ is an h-proposition. The same holds for $z \in y$. Thus, the rightmost type in the chain of equivalences above is a family of equivalences between h-propositions, and is thus an h-proposition. It then follows that the type $(x, p) =_{\mathsf{V}^0} (y, q)$ is an h-proposition. □

Given a type $A : \mathcal{U}$ and an embedding $f : A \hookrightarrow \mathsf{V}^0$, we can construct an element of V^0. This function is the counterpart to sup for V^∞, and while it is not formally a constructor it behaves like one in that the recursion and elimination principles, with fitting computation rules, are provable for it (Paper II).

Remark 1. The underscores in the constructions below denote proof terms for the h-propositions involved. We omit these for readability, and refer the interested reader to the formalization.

Definition 6 ( ). We define the following function:

$$\mathsf{sup}^0 : \left(\sum_{A:\mathcal{U}} A \hookrightarrow \mathsf{V}^0\right) \to \mathsf{V}^0$$
$$\mathsf{sup}^0\,(A, f) := (\mathsf{sup}\,A\,(\pi_0 \circ f), \_)$$

Similarly, given an element of V^0, we can extract the underlying type and embedding.

Definition 7 ( ). We define the following function:

$$\mathsf{desup}^0 : \mathsf{V}^0 \to \left(\sum_{A:\mathcal{U}} A \hookrightarrow \mathsf{V}^0\right)$$
$$\mathsf{desup}^0\,(\mathsf{sup}\,A\,f, \_) := (A, (f, \_))$$

By virtue of being a W-type, V^∞ is the initial algebra to the polynomial functor

$$X \mapsto \left(\sum_{A:\mathcal{U}} A \to X\right)$$

Similarly, V^0 is the initial algebra for the functor $X \mapsto (\sum_{A:\mathcal{U}} A \hookrightarrow X)$, even though this functor is not polynomial. The initiality induces an equivalence $\mathsf{V}^0 \simeq \left(\sum_{A:\mathcal{U}} A \hookrightarrow \mathsf{V}^0\right)$, realized by the maps sup^0 and desup^0 above. These results are due to Paper II, which extends this construction to a whole hierarchy of functors $X \mapsto (\sum_{A:\mathcal{U}} A \hookrightarrow_n X)$, for $n : \mathbb{N}_{-1}$. Each of these have an initial algebra, given by a higher level generalization of V^0.

3 V^0 as a universe à la Tarski

The type V^0 can be thought of as a type of material sets, in the sense that V^0 together with the binary relation $\in$ is a model of constructive set theory (Gylterud, 2018). This section demonstrates that, more type-theoretically, V^0 can be organized into a universe à la Tarski. In this way, V^0 becomes a universe of h-sets which is itself an h-set. Furthermore, V^0 is a strict universe in the sense that the decoding from codes to types is definitional. For instance, the decoding of the code for the natural numbers is definitionally equal to the type of natural numbers, and the decoding of a Π- or Σ-type of a family is the actual Π- or Σ-type of the decoding of the family.

We begin by defining the decoding family for our universe, V^0, as the underlying index type for each of its elements.

Definition 8 (Decoding,). We define the decoding function $\mathsf{El}^0 : \mathsf{V}^0 \to \mathcal{U}$ by

$$\mathsf{El}^0\, x := \overline{x}$$

It is easy to prove that the decoding of each code in V^0 is also an h-set:

Theorem 4 ( ). *For every $x : \mathsf{V}^0$ the type $\mathsf{El}^0\, x$ is an h-set.*

Proof. Recall that $\widetilde{x}$ embeds $\mathsf{El}^0\, x$ into V^0. By Theorem 3 V^0 is an h-set. Since any type which embeds into an h-set is an h-set, it follows that $\mathsf{El}^0\, x$ is an h-set. □

Note that for any $A : \mathcal{U}$ and embedding $f : A \hookrightarrow \mathsf{V}^0$ we have the definitional equality $\mathsf{El}^0\,(\mathsf{sup}^0\ (A, f)) \equiv A$. That is, if we construct a code for a type in $\mathcal{U}$ using sup^0 (which is what we usually do), then the decoding of this code is definitionally equal to the type we started with. This is very convenient when working with the universe V^0, especially for formalization.

As a universe, V^0 contains codes of all the traditional type formers as long as they are present in the underlying universe, $\mathcal{U}$. Using sup^0, one can construct a code for a given type $A : \mathcal{U}$ in V^0 if there is an embedding $A \hookrightarrow \mathsf{V}^0$. In fact, there is a code for A in V^0 precisely when it can be embedded into V^0.

Proposition 1 ( ). *For any $A : \mathcal{U}$ there is an equivalence*

$$\left(A \hookrightarrow \mathsf{V}^0\right) \simeq \left(\sum_{a:\mathsf{V}^0} \mathsf{El}^0\, a = A\right)$$

Proof. The maps back and forth are

$$\alpha : \left(A \hookrightarrow \mathsf{V}^0\right) \to \sum_{a:\mathsf{V}^0} \mathsf{El}^0\, a = A$$
$$\alpha\, f := \left(\mathsf{sup}^0\ (A, f)\,, \mathsf{refl}\right)$$

$$\beta : \left(\sum_{a:\mathsf{V}^0} \mathsf{El}^0\, a = A\right) \to \left(A \hookrightarrow \mathsf{V}^0\right)$$
$$\beta\,(a, \mathsf{refl}) := \widetilde{a}$$

We compute as follows:

$$\alpha(\beta(a, \mathsf{refl})) \equiv \alpha(\widetilde{a}) \equiv \left(\mathsf{sup}^0\ (\overline{a}, \widetilde{a})\,, \mathsf{refl}\right) = (a, \mathsf{refl})$$
$$\mathsf{pr}_1\,(\beta(\alpha\, f)) \equiv \mathsf{pr}_1\,\left(\beta(\mathsf{sup}^0\ (A, f)\,, \mathsf{refl})\right) \equiv \mathsf{pr}_1\left(\widetilde{\mathsf{sup}^0\ (A, f)}\right) \equiv \mathsf{pr}_1\, f$$

Thus, α is a quasi-equivalence and therefore an equivalence. □

We emphasize that the definitional equation $\mathsf{El}^0\,(\mathsf{sup}^0\ (A, f)) \equiv A$ simplifies the definition of α as we may then use refl for the second argument. Moreover, $\beta \circ \alpha$ definitionally preserves the function underlying the embedding. The same is not true of the *witness* that this function is an embedding, but such witnesses belong to a contractible type and can safely be ignored.

3.1 Basic types

We now construct codes for some basic types in V^0.

Proposition 2 ( ). V^0 *contains the empty type, unit type and booleans.*

Proof. We define the elements $\mathsf{empty}^0, \mathsf{unit}^0, \mathsf{bool}^0 : \mathsf{V}^0$ as follows:

$$\begin{aligned}
\mathsf{empty}^0 &:= \emptyset \\
\mathsf{unit}^0 &:= \{\emptyset\} \\
\mathsf{bool}^0 &:= \{\emptyset, \{\emptyset\}\}
\end{aligned}$$

There were all verified to be iterative sets in Section 2. □

We note that the expected equations hold up to definitional equality:

$$\mathsf{El}^0\,\mathsf{empty}^0 \equiv \mathsf{empty}, \qquad \mathsf{El}^0\,\mathsf{unit}^0 \equiv \mathsf{unit} \qquad \text{and} \qquad \mathsf{El}^0\,\mathsf{bool}^0 \equiv \mathsf{bool}.$$

Proposition 3 ( ). V^0 *contains the natural numbers.*

Proof. By Proposition 1 it is enough to construct an embedding $\mathbb{N} \hookrightarrow \mathsf{V}^0$. Here there is a choice of encoding of the naturals in V^0 and several encodings are possible. We will use the von Neumann encoding and show that this is an embedding.

First we define the successor function in V^0:

$$\begin{aligned}
&\mathsf{suc}^0 : \mathsf{V}^0 \to \mathsf{V}^0 \\
&\mathsf{suc}^0\, x := \mathsf{sup}^0\,(\overline{x} + \mathsf{unit}, \varphi_x)
\end{aligned}$$

In the above, $\varphi_x : \overline{x} + \mathsf{unit} \to \mathsf{V}^0$ is defined as follows:

$$\begin{aligned}
\varphi_x\,(\mathsf{inl}\,a) &:= \widetilde{x}\,a \\
\varphi_x\,(\mathsf{inr}\,b) &:= x
\end{aligned}$$

To see that the map φ_x is an embedding, note that for any $z : \mathsf{V}^0$ the fiber $\mathsf{fib}\,\varphi_x\,z$ is equivalent to $(z \in x) + (x = z)$. Both summands are h-propositions and they are disjoint: if they were both inhabited we could derive $x \in x$ which contradicts the well-foundedness of $\in$ (Paper II).

The von Neumann encoding of the natural numbers is then the function:

$$\begin{aligned} &f : \mathbb{N} \to \mathsf{V}^0 \\ &f\,0 := \emptyset \\ &f\,(\mathsf{s}\,n) := \mathsf{suc}^0\,(f\,n) \end{aligned}$$

It remains to show that f is an embedding. As $\mathbb{N}$ and V^0 are both h-sets it suffices that f is injective. Observe that $\overline{f\,x} \simeq \mathsf{Fin}\,x$, so if $f\,n = f\,m$ then $\mathsf{Fin}\,n \simeq \mathsf{Fin}\,m$ from which $n = m$ follows.

Having shown that $f : \mathbb{N} \to \mathsf{V}^0$ is an embedding, we define the (code for the) natural numbers in V^0 as follows:

$$\mathbb{N}^0 := \mathsf{sup}^0\ (\mathbb{N}, f)$$ □

Note, again, that the decoding holds up to definitional equality:

$$\mathsf{El}^0\,\mathbb{N}^0 \equiv \mathbb{N}$$

3.2 Type formers

We now turn to closing V^0 under the standard type formers. For these constructions we will need ordered pairs.

Lemma 3 ( ). *There is an ordered pairing operation* $\langle \_, \_ \rangle : \mathsf{V}^0 \times \mathsf{V}^0 \hookrightarrow \mathsf{V}^0$.

Proof. Ordered pairs are constructed using the Norbert Wiener encoding. The details of this construction can be found in the proof of Theorem 7 in Paper II. □

Proposition 4 ( ). V^0 *is closed under* Π*-types.*

Proof. Let $x : \mathsf{V}^0$ and $y : \mathsf{El}^0\,x \to \mathsf{V}^0$. By Lemma 12 in Paper II there is an embedding:

$$\mathsf{graph}_{x,y} : \left(\prod_{a:\mathsf{El}^0\,x} \mathsf{El}^0\,(y\,a) \right) \hookrightarrow \mathsf{V}^0$$

This map sends $\varphi : \prod_{a:\mathsf{El}^0\,x} \mathsf{El}^0\,(y\,a)$ to the element

$$\mathsf{sup}^0\,\left(\mathsf{El}^0\,x, \lambda a.\langle \widetilde{x}\,a, \widetilde{(y\,a)}\,(\varphi\,a)\rangle\right).$$

The Π-type is then defined as follows:

$$\Pi^0\,x\,y := \mathsf{sup}^0\,\left(\prod_{a:\mathsf{El}^0\,x} \mathsf{El}^0\,(y\,a), \mathsf{graph}_{x,y} \right)$$ □

The decoding holds up to definitional equality:

$$\mathsf{El}^0\ \left(\Pi^0\, x\, y\right) \equiv \prod_{a:\mathsf{El}^0\, x} \mathsf{El}^0\, (y\, a)$$

Corollary 2 ( ). V^0 *is closed under (non-dependent) function types. Let* $x \to^0 y$ *denote the code for the type* $\mathsf{El}^0\, x \to \mathsf{El}^0\, y$.

Proposition 5 ( ). V^0 *is closed under* Σ*-types.*

Proof. Let $x : \mathsf{V}^0$ and $y : \mathsf{El}^0\, x \to \mathsf{V}^0$. Define a putative embedding as follows:

$$f : \left(\sum_{a:\mathsf{El}^0\, x} \mathsf{El}^0\, (y\, a)\right) \to \mathsf{V}^0$$
$$f\, (a, b) := \langle \widetilde{x}\, a, \widetilde{(y\, a)}\, b \rangle$$

This is the composition of two embeddings: $\langle \_, \_ \rangle$ and $\lambda(a, b).(\widetilde{x}\, a, \widetilde{(y\, a)}\, b)$ and therefore an embedding. The last function is an embedding because $\widetilde{x}$ is an embedding and as is $\widetilde{(y\, a)}$ for every $a : \mathsf{El}^0\, x$. We may now define the code for Σ-types:

$$\Sigma^0\, x\, y := \mathsf{sup}^0 \left(\sum_{a:\mathsf{El}^0\, x} \mathsf{El}^0\, (y\, a)\; f\right) \qquad \square$$

The decoding holds up to definitional equality:

$$\mathsf{El}^0\ \left(\Sigma^0\, x\, y\right) \equiv \sum_{a:\mathsf{El}^0\, x} \mathsf{El}^0\, (y\, a)$$

Corollary 3 ( ). V^0 *is closed under cartesian products. Let* $x \times^0 y$ *be the code for* $\mathsf{El}^0\, x \times \mathsf{El}^0\, y$.

In order to construct coproducts in V^0 we need two lemmas about embeddings.

Lemma 4 ( ). *Given types* Y, Z *and h-set* X *with a point* $x_0 : X$, *any embedding* $f : X \times Y \hookrightarrow Z$ *gives rise to an embedding by fixing the first coordinate:* $f\, (x_0, \_) : Y \hookrightarrow Z$.

Proof. We need to show that for any $z : Z$, the fiber of $f\, (x_0, \_)$ over z is an h-proposition. But the following chain of equivalences holds:

$$\left(\sum_{y:Y} f\, (x_0, y) = z\right) \simeq \left(\sum_{y:Y} \sum_{\sum_{x:X} (x = x_0)} f\, (x, y) = z\right)$$
$$\simeq \left(\sum_{((x,y),p):\mathsf{fib}\, f\, z} x = x_0\right)$$

The last type is an h-proposition since $\mathsf{fib}\, f\, z$ is an h-proposition by Lemma 1 and for each $((x, y), p) : \mathsf{fib}\, f\, z$, the type $x = x_0$ is an h-proposition. ☐

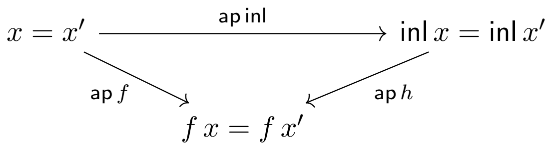
Lemma 5 ( ). *Given types X, Y, and Z together with embeddings $f : X \hookrightarrow Z$ and $g : Y \hookrightarrow Z$. If $f\, x \neq g\, y$ for all $x : X$ and $y : Y$ then the following map is an embedding:*

$$
\begin{aligned}
&h : X + Y \to Z \\
&h\, (\mathsf{inl}\, x) := f\, x \\
&h\, (\mathsf{inr}\, y) := g\, y
\end{aligned}
$$

Proof. Let $s, t : X + Y$. We need to show that $\mathsf{ap}\, h : s = t \to h\, s = h\, t$ is an equivalence. Using induction on coproducts, there are two kinds of cases to consider: when s and t lie in different summands, and when they lie in the same one.

First, suppose without loss of generality that $s \equiv \mathsf{inl}\, x$ and $t \equiv \mathsf{inr}\, y$. In this case we need to show that $\mathsf{ap}\, h : \mathsf{inl}\, x = \mathsf{inr}\, y \to f\, x = g\, y$ is an equivalence. But both types are empty, so any map between them is an equivalence.

Now, suppose without loss of generality that $s \equiv \mathsf{inl}\, x$ and $t \equiv \mathsf{inl}\, x'$. We need to show that $\mathsf{ap}\, h : \mathsf{inl}\, x = \mathsf{inl}\, x' \to f\, x = f\, x'$ is an equivalence. But note that the following diagram commutes:



Since both $\mathsf{ap}\, f$ and $\mathsf{ap}\,\mathsf{inl}$ are equivalences it follows that $\mathsf{ap}\, h$ is an equivalence. ☐

Proposition 6 ( ). *V^0 is closed under coproducts.*

Proof. Let $x, y : \mathsf{V}^0$. Define the map

$$
\begin{aligned}
&f : \mathsf{El}^0\, x + \mathsf{El}^0\, y \to \mathsf{V}^0 \\
&f\, (\mathsf{inl}\, a) := \langle \mathsf{empty}^0, \widetilde{x}\, a \rangle \\
&f\, (\mathsf{inr}\, b) := \langle \mathsf{unit}^0, \widetilde{y}\, b \rangle
\end{aligned}
$$

By Lemma 4 both $\lambda a. \langle \mathsf{empty}^0, \widetilde{x}\, a \rangle$ and $\lambda b. \langle \mathsf{unit}^0, \widetilde{y}\, b \rangle$ are embeddings. Moreover, suppose $\langle \mathsf{empty}^0, \widetilde{x}\, a \rangle = \langle \mathsf{unit}^0, \widetilde{y}\, b \rangle$ for some $a : \mathsf{El}^0\, x$ and $b : \mathsf{El}^0\, y$. It then follows that $\mathsf{empty}^0 = \mathsf{unit}^0$, which is a contradiction. Therefore, by Lemma 5 we conclude that f is an embedding.

We now define the coproduct:

$$x +^0 y := \mathsf{sup}^0\, (\mathsf{El}^0\, x + \mathsf{El}^0\, y, f)$$ ☐

Note that the decoding holds up to definitional equality:

$$\mathsf{El}^0\,(x +^0 y) \equiv \mathsf{El}^0\,x + \mathsf{El}^0\,y$$

Proposition 7 ( ). V^0 *is closed under identity types.*

Proof. Let $x : \mathsf{V}^0$ and $a, a' : \mathsf{El}^0\,x$. Define the following map:

$$\begin{aligned} &f : a = a' \to \mathsf{V}^0 \\ &f\,p :\equiv \emptyset \end{aligned}$$

This is an embedding as it is a map from an h-proposition into an h-set. The identity type in V^0 is then defined as follows:

$$\mathsf{Id}^0\,x\,a\,a' :\equiv \mathsf{sup}^0\,(a = a', f)$$

□

The decoding holds up to definitional equality:

$$\mathsf{El}^0\,(\mathsf{Id}^0\,x\,a\,a') \equiv (a = a')$$

We emphasize that $\mathsf{El}^0\,x$ is an h-set for any for any $x : \mathsf{V}^0$. Accordingly, $\mathsf{Id}^0\,x\,a\,a'$ is necessarily a proposition for any $a, a' : \mathsf{El}^0\,x$. In particular, $\mathsf{Id}^0\,x\,a\,a'$ satisfies UIP. As this identity type represents internalizes the ambient identity type, other expected properties of the identity type (such as function extensionality) also hold.

3.3 Set quotients

In order to define set quotients in V^0, we must assume that these quotients exist in our starting universe $\mathcal{U}$. More specifically, we first assume that there is a function of the following type

$$-/- : \prod_{A:\mathcal{U}} (A \to A \to \mathcal{U}) \to \mathsf{hSet}_{\mathcal{U}}$$

We then ensure that A/R realizes the quotient of A by the relation R by requiring a map $[-]_R : A \to A/R$ such that $R\,a\,b \to [a]_R = [b]_R$ for all $a, b : A$. We also assume a suitable elimination principle: given a family of h-sets $P : A/R \to \mathsf{hSet}_{\mathcal{U}}$, we can construct a function $\prod_{x:A/R} P\,x$ from a function $q : \prod_{x:A/R} P\,[a]_R$ which coheres with the map $R\,a\,b \to [a]_R = [b]_R$. The require that function we get satisfies a coherence condition, and if we precompose it with the quotient map $[-]_R$ we get back q. (For the exact assumptions, see the formalization .)

While we don't need to assume that $R : A \to A \to \mathcal{U}$ is an equivalence relation (h-propositional, symmetric, reflexive, transitive), the constructions below will use the fact any R induces an equivalence relation $|R| : A \to A \to \mathsf{hProp}_{\mathcal{U}}$ defined by $|R|\,a\,b := ([a]_R = [b]_R)$.

To streamline the process, we will use an interesting formulation of equivalence relations:

Lemma 6. *Given a relation $R : A \to A \to \mathsf{hProp}_{\mathcal{U}}$, the following are equivalent:*

- *R is an equivalence relation*
- *$R\,a\,b \simeq \prod_{c:A} (R\,a\,c \simeq R\,b\,c)$ for all $a, b : A$*
- *$R\,a\,b \simeq (R\,a =_{A \to \mathcal{U}} R\,b)$ for all $a, b : A$*

Proof. Since R is an (h-propositional) binary relation, the above statements are all h-propositions. The last two are equivalent by function extensionality and univalence. It thus remains to show that being an equivalence relation is equivalent to one of the last two–we choose the middle one.

Assume that R is an equivalence relation. Everything in sight is an h-proposition, so the equivalences are logical equivalences. Thus, assume that $R\,a\,b$. Then we get a map $\prod_{c:A} R\,a\,c \leftrightarrow R\,b\,c$ by transitivity and symmetry. In the other direction, if $\prod_{c:A} R\,a\,c \leftrightarrow R\,b\,c$, choose $c = a$ in order to obtain $R\,a\,a \leftrightarrow R\,a\,b$. Since R is reflexive, we get $R\,a\,b$.

Conversely, assume $R\,a\,b \simeq \prod_{c:A} (R\,a\,c \simeq R\,b\,c)$ for all $a, b : A$. To show reflexivity, let $b = a$ and notice that $\prod_{c:A} (R\,a\,c \simeq R\,a\,c)$ has a canonical element, from which we obtain $R\,a\,a$. Symmetry is obtained by observing that $\prod_{c:A} (R\,a\,c \simeq R\,b\,c) \simeq \prod_{c:A} (R\,b\,c \simeq R\,a\,c)$ and hence $R\,a\,b \simeq R\,b\,a$. For transitivity, remember that $R\,a\,b$ gives $\prod_{c:A} (R\,a\,c \simeq R\,b\,c)$, thus if we have $R\,b\,c$ we get $R\,a\,c$ by following the backwards direction of the equivalence. □

The property $R\,a\,b \simeq \prod_{c:A} (R\,a\,c \simeq R\,b\,c)$ for all $a, b : A$ essentially states that the equivalence classes of R behave well. Tangentially, we note that this requirement make sense even when R is a general binary family, not only of h-propositions. Thus, this property might make for interesting future study.

Proposition 8 ( ). *V^0 is closed under set quotients. That is, given $a : \mathsf{V}^0$ and $R : \mathsf{El}^0\,a \to \mathsf{El}^0\,a \to \mathcal{U}$ there is $a/^0R : \mathsf{V}^0$ such that $\mathsf{El}^0\,(a/^0R) \equiv (\mathsf{El}^0\,a)/R$.*

Proof. By Proposition 1 it suffices to construct an embedding $\mathsf{El}^0\,a/R \hookrightarrow \mathsf{V}^0$. We define $f : \mathsf{El}^0\,a \to \mathsf{V}^0$ and prove that for any $x, x' : \mathsf{El}^0\,a$ we have $([x]_R = [x']_R) \simeq (f\,x = f\,x')$. By the elimination principle for set quotients, this will induce an embedding $\mathsf{El}^0\,a/R \hookrightarrow \mathsf{V}^0$.

Thus, let $f\,x = \mathsf{sup}^0\left(\sum_{y:\mathsf{El}^0\,a} |R|\,x\,y, \widetilde{a} \circ \mathsf{pr}_1\right)$, and observe the chain of equivalences:

$$
\begin{aligned}
&(f\,x = f\,x') \\
&\quad \equiv \left(\mathsf{sup}^0 \left(\sum_{y:\mathsf{El}^0\,a} |R|\,x\,y, \widetilde{a} \circ \mathsf{pr}_1\right) = \mathsf{sup}^0 \left(\sum_{y:\mathsf{El}^0\,a} |R|\,x'\,y, \widetilde{a} \circ \mathsf{pr}_1\right)\right) \\
&\quad \simeq \sum_{\alpha:\left(\sum_{y:\mathsf{El}^0\,a} |R|\,x\,y\right) \simeq \left(\sum_{y:\mathsf{El}^0\,a} |R|\,x'\,y\right)} \widetilde{a} \circ \mathsf{pr}_1 = \widetilde{a} \circ \alpha \circ \mathsf{pr}_1 \\
&\quad \simeq \sum_{\alpha:\left(\sum_{y:\mathsf{El}^0\,a} |R|\,x\,y\right) \simeq \left(\sum_{y:\mathsf{El}^0\,a} |R|\,x'\,y\right)} \mathsf{pr}_1 = \alpha \circ \mathsf{pr}_1 \\
&\quad \simeq \prod_{y:\mathsf{El}^0\,a} |R|\,x\,y \simeq |R|\,x'\,y \\
&\quad \simeq |R|\,x\,x' \\
&\quad \equiv \left([x]_R = [x']_R\right)
\end{aligned}
$$

Note that we have used the characterization of equivalence relations given by Lemma 6 in the next to last step. □

Remark 2. We note that our choice of embedding to define $a/^0R : \mathsf{V}^0$ is similar to the construction of set quotients using type-theoretic replacement due to Rijke (2017).

3.4 Using the type formers

Using the types and type formers in V^0 we can construct new types. The decoding of these composite types will then hold up to definitional equality. For example, given elements $x, y : \mathsf{V}^0$, a map $f : \mathsf{El}^0\,x \to \mathsf{El}^0\,y$ and $b : \mathsf{El}^0\,y$ we can define the code for the fiber of f over b as

$$\mathsf{fib}^0\,f\,b := \Sigma^0\,x\,(\lambda a.\mathsf{Id}^0\,y\,(f\,a)\,b)$$

After applying the decoding function, we get obtain the following definitional equality:

$$\mathsf{El}^0\,\left(\mathsf{fib}^0\,f\,b\right) \equiv \mathsf{fib}\,f\,b$$

3.5 Universes

The flexible handling of hierarchies of universes is a key feature of dependent type theory. It makes it easy to formalize higher order concepts, and mathematical structures. Our universe construction retains this ability, and in this subsection we demonstrate that by observing that the types

$\mathsf{V}^0_0, \mathsf{V}^0_1, \cdots, \mathsf{V}^0_\ell, \cdots$ form a hierarchy of universes, where each universe occurs as a type with a code in the next.

Proposition 9 **( ).** *For any universe level ℓ there is a code $\mathsf{V}^0_\ell\text{-}\mathsf{code} : \mathsf{V}^0_{\ell+1}$ for V^0_ℓ with the definitional equality $\mathsf{El}^0\,\mathsf{V}^0_\ell\text{-}\mathsf{code} \equiv \mathsf{V}^0_\ell$*

Proof. Given a universe level ℓ, we need to construct an embedding $\mathsf{V}^0_\ell \hookrightarrow \mathsf{V}^0_{\ell+}$. For this, we start by constructing an embedding $\mathsf{V}^\infty_\ell \hookrightarrow \mathsf{V}^\infty_{\ell+}$. Thus define the map

$$\begin{aligned} &\varphi : \mathsf{V}^\infty_\ell \to \mathsf{V}^\infty_{\ell+} \\ &\varphi\,(\mathsf{sup}\,A\,f) := \mathsf{sup}\,A\,(\varphi \circ f) \end{aligned}$$

(Note that we are using cumulative universes in the ambient type theory, so $A : \mathcal{U}_{\ell+}$ whenever $A : \mathcal{U}_\ell$.) To show that φ is an embedding, let $\mathsf{sup}\,A\,f, \mathsf{sup}\,B\,g : \mathsf{V}^\infty_\ell$ be arbitrary elements. We need to show that

$$\mathsf{ap}\,\varphi : \mathsf{sup}\,A\,f = \mathsf{sup}\,B\,g \to \mathsf{sup}\,A\,(\varphi \circ f) = \mathsf{sup}\,B\,(\varphi \circ g)$$

is an equivalence. First, note that the following diagram commutes:

$$\begin{array}{ccc} \sum_{X:\mathcal{U}_\ell} X \to \mathsf{V}^\infty_\ell & \xrightarrow{\lambda(X,h).(X,\varphi\circ h)} & \sum_{X:\mathcal{U}_{\ell+}} X \to \mathsf{V}^\infty_{\ell+} \\ {\scriptstyle \mathsf{sup}}\downarrow & & \downarrow{\scriptstyle \mathsf{sup}} \\ \mathsf{V}^\infty_\ell & \xrightarrow[\varphi]{} & \mathsf{V}^\infty_{\ell+} \end{array}$$

The map sup is an equivalence, so φ is an embedding if and only if the top map is an embedding. Thus we need to show the following to be an equivalence:

$$\mathsf{ap}\,(\lambda(X,h).(X,\varphi \circ h)) : (A,f) = (B,g) \to (A,\varphi \circ f) = (B,\varphi \circ g)$$

While we might hope to argue that this is a fiberwise embedding and therefore the total map is an embedding as well. Unfortunately, our induction hypothesis does not state that φ is an embedding, i.e., that ap is an equivalence for *all* elements. It only ensures that it is an equivalence for *some* elements. We must then take a different path and instead note that the diagram below commutes.

$$\begin{array}{ccc} (A,f) = (B,g) & \xrightarrow{\mathsf{ap}\,(\lambda(X,h).(X,\varphi\circ h))} & (A,\varphi \circ f) = (B,\varphi \circ g) \\ {\scriptstyle \simeq}\downarrow & & \downarrow{\scriptstyle \simeq} \\ \sum_{e:A\simeq B} f \sim g \circ e & \xrightarrow[\lambda(e,H).(e,\mathsf{ap}\,\varphi\circ H)]{} & \sum_{e:A\simeq B} \varphi \circ f \sim \varphi \circ g \circ e \end{array}$$

The vertical maps are provided by Theorem 1. Using 3-for-2, the top map is an equivalence if and only if the bottom one is an equivalence. We now note that it suffices to check this property on fibers, so we need to show that given $e : A \simeq B$, the following map is an equivalence:

$$(\lambda H.\mathsf{ap}\,\varphi \circ H) : f \sim g \circ e \to \varphi \circ f \sim \varphi \circ g \circ e$$

We now recall that postcomposition by a family of maps is an equivalence if it is a family of equivalences. Finally, we must argue that $\mathsf{ap}\,\varphi : f\,a = g\,(e\,a) \to \varphi\,(f\,a) = \varphi\,(g\,(e\,a))$ is an equivalence for all $a : A$. This follows from the induction hypothesis. Thus, we conclude that $\varphi : \mathsf{V}^\infty_\ell \to \mathsf{V}^\infty_{\ell+}$ is an embedding.

To argue that this equivalence restricts to V^0, we must show this equivalence sends iterative sets to iterative sets. Thus let $\mathsf{sup}\,A\,f : \mathsf{V}^\infty_\ell$ be such that f is an embedding and $f\,a$ is an iterative set for all $a : A$. We must argue that $\mathsf{sup}\,A\,(\varphi \circ f)$ is an iterative set. But the map $\varphi \circ f$ is an embedding as the composition of two embeddings. Moreover, by the induction hypothesis, for any $a : A$, $\varphi\,(f\,a)$ is an iterative set since $f\,a$ is an iterative set.

Therefore, φ is an embedding from V^0_ℓ into $\mathsf{V}^0_{\ell+}$. The code for V^0_ℓ in $\mathsf{V}^0_{\ell+}$ is thus defined as

$$\mathsf{V}^0_\ell\text{-}\mathsf{code} :\equiv \mathsf{sup}^0\,(\mathsf{V}^0_\ell, \varphi)$$ □

Note that the decoding holds up to definitional equality:

$$\mathsf{El}^0\,\mathsf{V}^0_\ell\text{-}\mathsf{code} \equiv \mathsf{V}^0_\ell$$

Proposition 10. V^0 *is* not *a univalent universe.*

Proof. For $x, y : \mathsf{V}^0$, $x = y$ is an h-proposition as V^0 is an h-set, but $\mathsf{El}^0\,x \simeq \mathsf{El}^0\,y$ is in general a proper h-set. □

4 V^0 as a category

The universe structure on V^0 induces a category with a full and faithful functor into $\mathcal{hSet}_\mathcal{U}$. In this section, we define this category and show that it is closed under many essential constructions (finite limits and colimits, exponentials, and more). This category provides another concrete way for us to *measure* the adequacy of V^0 as a replacement for $\mathsf{hSet}_\mathcal{U}$; the former induces a closely related category to the latter, sharing many similar properties.

Definition 9 ( ). Let $\mathcal{V}$ be the category with

- $\mathsf{Ob}\,\mathcal{V} :\equiv \mathsf{V}^0$

- $\mathsf{Hom}_{\mathcal{V}}(x, y) :\equiv \mathsf{El}^0\, x \to \mathsf{El}^0\, y$
- id and $\circ$ are simply the identity function and function composition.

All laws hold by refl as id and $\circ$ are the identity and composition from the ambient type theory. For $x, y : \mathsf{V}^0$, the type $\mathsf{Hom}(x, y)$ is an h-set as it consists of functions into an h-set.

Note that we will take *category* to denote what the the HoTT Book calls "precategory", *univalent category* to denote what the book calls "category", and *strict category* to denote a category where the objects form an h-set. Hence, $\mathcal{V}$ is a strict category as V^0 is an h-set.

The following holds more-or-less by construction:

Lemma 7 ( ). *The map* El^0 *induces a full and faithful functor from* $\mathcal{V}$ *to* $h\mathcal{S}et_{\mathcal{U}}$.

Clearly, $\mathcal{V}$ is not a univalent category since it possesses objects with non-trivial automorphisms, but the type of objects in $\mathcal{V}$ is an h-set. Still, one might ask if El^0 is an equivalence of categories. This does not appear to be true in general, but can be implied by further axioms. For instance, the axiom of choice implies that El^0 is an equivalence. The core of this is whether every type in $\mathcal{U}$ can be equipped with an iterative set-structure—a property known as well-founded materialization. We discuss this further in Section 6.4.

Fortunately, even without additional axioms we are able to show that $\mathcal{V}$ retains much of the essential structure of $h\mathcal{S}et_{\mathcal{U}}$ and that El^0 preserves many important categorical constructions. In order to show that $\mathcal{V}$ is closed under some categorical structure, it therefore suffices to break the process into two stages:

1. Show that $h\mathcal{S}et_{\mathcal{U}}$ is closed under *e.g.,* finite limits, exponentials, *etc.*
2. Show that the objects involved land in the image of El^0.

Better still, $h\mathcal{S}et_{\mathcal{U}}$ is well-studied and known to be closed under all the categorical structures we consider (Rijke and Spitters, 2015). Our task is therefore reduced only to showing that various objects of $h\mathcal{S}et_{\mathcal{U}}$ land in the image of El^0. For this, we repeatedly capitalize on the fact that the decoding El^0 holds up to definitional equality; it ensures that the final step can be rephrased as follows: show that there exists an iterative structure on the objects involved. This pattern is used repeatedly to prove the following result:

Theorem 5 ( ). $\mathcal{V}$ *is closed under and* El^0 *preserves the following:*

1. *initial object,*

2. *terminal object,*
3. *finite coproducts,*
4. *pushouts,*
5. *finite products,*
6. *pullbacks, and*
7. *exponentials.*

Proof. As $h\mathcal{S}et_{\mathcal{U}}$ supports these structures it suffices to show that each of the representing objects land in the image of El^0. This clearly follows from the results in Section 3, for instance, the existence of the initial and terminal object follows from Proposition 2, and e.g., pullbacks can be constructed through Σ^0 and fib^0 just like in $h\mathcal{S}et_{\mathcal{U}}$. □

As $\mathcal{V}$ has pullbacks/pushouts and terminal/initial object we directly get:

Corollary 4. *$\mathcal{V}$ has finite limits and colimits. These are preserved by El^0.*

We defer further categorical considerations of $\mathcal{V}$ to future work and instead turn our attention to slice categories of $\mathcal{V}$, which play an important role in the study of it as a model of type theory.

4.1 Slice categories of $\mathcal{V}$

Similar methods to the ones above also apply when showing that the slice categories $\mathcal{V}/a$ are well-behaved. In particular, El^0 induces a full and faithful functor $\mathcal{V}/a \to h\mathcal{S}et_{\mathcal{U}}/\mathsf{El}^0\, a$. We can use this fact to deduce, e.g., that $\mathcal{V}/a$ is cartesian closed.

Proposition 11 ( ). *For any $a : \mathsf{V}^0$, $\mathcal{V}/a$ has finite limits.*

Proof. This can be proved using the standard argument: products in a slice category are realized by pullbacks in the underlying category and connected limits are realized by limits of the underlying diagram. We could also argue by noting that $h\mathcal{S}et_{\mathcal{U}}/\mathsf{El}^0\, a$ is finitely complete and that limits of diagrams in the image of El^0 remain in the image of El^0. □

We give a bit more details in the following proof as it showcases the usefulness of being able to encode things directly in V^0, combined with the fact that El^0 strictly decodes to the expected thing in $\mathcal{U}$.

Proposition 12 ( ). *For any $a : \mathsf{V}^0$, $\mathcal{V}/a$ has exponentials.*

Proof. Given $(x, f), (y, g) : \mathsf{Ob}\,(\mathcal{V}/a)$, define their exponential as the element

$$\mathsf{exp}\,(x, f)\,(y, g) := \Sigma^0\, a\,(\lambda\, i.\, \mathsf{fib}^0\, g\, i \to^0 \mathsf{fib}^0\, f\, i)$$

Note that we have the following definitional equality:

$$\mathsf{El}^0\,(\mathsf{exp}\,(x, f)\,(y, g)) \equiv \sum_{i : \mathsf{El}^0\, a} \mathsf{fib}\, g\, i \to \mathsf{fib}\, f\, i$$

This is the exponential in $h\mathcal{S}et_{\mathcal{U}}/\mathsf{El}^0\, a$, so $\mathsf{exp}\,(x, f)\,(y, g)$ is an exponential object in $\mathcal{V}/a$. □

Corollary 5. *$\mathcal{V}$ is locally cartesian closed and El^0 is a locally cartesian closed functor.*

Finally, the following proposition foreshadows the next section where we build a model of type theory on $\mathcal{V}$. In that section, we wish to interpret types in context a as elements of $\mathcal{V}/a$ and to realize substitution as pullback. It is well-known, however, that the result is merely pseudofunctorial and therefore insufficient to form a (strict) model of type theory (Seely, 1984; Curien, Garner, and Hofmann, 2014). In the specific case of $h\mathcal{S}et_{\mathcal{U}}$, there is a well-known pseudo-natural equivalence between the slice category $h\mathcal{S}et_{\mathcal{U}}/a$ and the functor category $[a, h\mathcal{S}et_{\mathcal{U}}]$ which remedies the coherence issues. This equivalence restricts to the full subcategory determined by $\mathcal{V}$:

Proposition 13 ( ). *Given $a : \mathsf{V}^0$ and writing a for the corresponding discrete category associated with $\mathsf{El}^0\, a$, there is a canonical equivalence $\mathcal{V}/a \simeq [a, \mathcal{V}]$.*

Proof. The equivalence is constructed in the standard way. The functor from $[a, \mathcal{V}]$ to $\mathcal{V}/a$ sends $F : \mathsf{Ob}\,[a, \mathcal{V}]$ to the element $\Sigma^0\, a\, F$ together with the first projection. In the other direction, an element $(x, f) : \mathsf{Ob}\,(\mathcal{V}/a)$ is mapped to the functor $\lambda i.\mathsf{fib}^0\, f\, i$. □

5 V^0 as a category with families

Having established that V^0 organizes into a well-behaved category, we now take this a step further by showing that V^0 supports a model of extensional type theory. Since our goal is to do this very formally, our task is threefold: first, we must specify what we mean by a model of type theory. To this end, we have formalized a particular presentation of a *category with families (CwF)* (Dybjer, 1996). This extends a category with the additional structure required to interpret dependent type theory. Next, we show that $\mathcal{V}$ can be equipped with this additional structure. Finally, since our definition of a

CwF does not prescribe closure under any connectives, we detail how to extend a CwF with various connectives and show that the CwF structure on $\mathcal{V}$ supports these extensions.

Remark 3. Of these three steps, only the first two are fully formalized. The obstruction to formalizing closure under all relevant extensions is, surprisingly, completely independent of V^0. Rather, it stems from the fact that the equations governing substitution hold only up to propositional equality, leading to complicated path and transport computations when defining the substitution properties of said structure.

5.1 The definition of categories with families

In the paper and formalization we rely on the following formulation of categories with families.

Definition 10 (Category with families,). A category with families (CwF) consists of:

- A category $\mathcal{C}$ with a terminal object,
- a presheaf $\mathsf{Ty}_{\mathcal{C}} : \mathcal{C}^{\mathsf{op}} \to h\mathcal{S}et_{\mathcal{U}}$,
- a presheaf $\mathsf{Tm}_{\mathcal{C}} : \left(\int \mathsf{Ty}_{\mathcal{C}}\right)^{\mathsf{op}} \to h\mathcal{S}et_{\mathcal{U}}$,
- a functor $-.- : \int \mathsf{Ty}_{\mathcal{C}} \to \mathcal{C}$, and
- for each $\Delta : \mathsf{Ob}\, \mathcal{C}$ and $(\Gamma, A) : \int \mathsf{Ty}_{\mathcal{C}}$ a natural equivalence

$$\mathsf{Hom}\,(\Delta, \Gamma.\, A) \simeq \sum_{\gamma\, :\mathsf{Hom}\,(\Delta,\Gamma)} \mathsf{Tm}_{\mathcal{C}}\,(\Delta, A \cdot \gamma)$$

Here we have written $A \cdot \gamma$ for $\mathsf{Ty}_{\mathcal{C}}\,(\gamma\, A)$, and $\int \mathsf{Ty}_{\mathcal{C}}$ for the *category of elements* of $\mathsf{Ty}_{\mathcal{C}}$, i.e., the total space of the right fibration induced by $\mathsf{Ty}_{\mathcal{C}}$. Intuitively, objects in $\mathcal{C}$ interpret the contexts of our type theory, while morphisms interpret substitutions. The additional presheaves are used to interpret types and terms. Specifically, the set of semantic types in context $\Gamma : \mathsf{Ob}\, \mathcal{C}$ is given by $\mathsf{Ty}_{\mathcal{C}}\, \Gamma$ while the set of terms of type $A : \mathsf{Ty}_{\mathcal{C}}\, \Gamma$ is given by $\mathsf{Tm}_{\mathcal{C}}\,(\Gamma, A)$. The functoriality of $\mathsf{Ty}_{\mathcal{C}}$ and $\mathsf{Tm}_{\mathcal{C}}$ is precisely the structure required to interpret the application of substitutions to types and terms.

The terminal object interprets the empty context and the functor from $\int \mathsf{Ty}_{\mathcal{C}}$ to $\mathcal{C}$ interprets context extension. A context $\Gamma : \mathsf{Ob}\, \mathcal{C}$ can be extended by a type $A : \mathsf{Ty}_{\mathcal{C}}\, \Gamma$ in that context, to produce a new context $\Gamma.\, A : \mathsf{Ob}\, \mathcal{C}$.

Finally, the natural equivalence ties together substitutions and elements of $\mathsf{Tm}_{\mathcal{C}}$. In particular, the inverse encodes the ability to extend a substitution with a term. Following this last observation, we write $\Gamma.\, a$ for the element $\mathsf{Hom}\,(\Delta, \Gamma.\, A)$ induced by the element $(\Gamma, a) : \sum_{\gamma\, :\mathsf{Hom}\,(\Delta,\Gamma)} \mathsf{Tm}_{\mathcal{C}}\,(\Delta, A \cdot\ \gamma)$.

5.2 Equipping $\mathcal{V}$ with a CwF structure

We now turn to equipping $\mathcal{V}$ with a CwF structure. We begin by defining $\mathsf{Ty}_{\mathcal{V}}$ as follows:

$$\mathsf{Ty}_{\mathcal{V}}\, X := \mathsf{El}^0\, X \to \mathsf{V}^0$$

Intuitively, a type in context X is precisely an X-indexed family of sets. There is, however, a major subtlety in this definition that should be emphasized: the version of the definition where V^0 is replaced by $\mathsf{hSet}_{\mathcal{U}}$ would be *incorrect*. We have required that $\mathsf{Ty}_{\mathcal{V}}\, X$ always be an h-set as it is assumed to be an $\mathsf{hSet}_{\mathcal{U}}$ valued presheaf. Therefore, it is only after finding an adequate "h-set of h-sets" that we can define the set model of type theory in this manner.

The definition of the presheaf of terms is also reasonably direct:

$$\mathsf{Tm}_{\mathcal{V}}\,(X, A) = \prod_{x : \mathsf{El}^0\, X} \mathsf{El}^0\,(A\, x)$$

We now show that, along with $\mathcal{V}$, these two definitions assemble into a CwF.

Proposition 14 ( ). *$\mathcal{V}$ can be equipped with a CwF structure.*

Proof. We have given the putative definitions of $\mathsf{Ty}_{\mathcal{V}}$ and $\mathsf{Tm}_{\mathcal{V}}$. We note that it is straightforward to ensure that both are suitably functorial. The functorial action is given by precomposition and all the required equations hold on-the-nose.

It remains to show that these three pieces of data satisfy the required properties of a CwF. We have already shown that $\mathcal{V}$ has a terminal object, so it remains to discuss the interpretation of context extension. Fix $X : \mathsf{V}^0$ and $A : \mathsf{Ty}_{\mathcal{V}}\, X$. We define $X.\, A : \mathsf{V}^0$ as $\Sigma^0\, X\, A$. The natural equivalence then follows from the η principle of dependent sums. □

By virtue of Proposition 13, we further note that types A in context X in this model are realized up to equivalence by families $\mathsf{El}^0\, A \to \mathsf{El}^0\, X$ and terms are likewise determined by sections. By presenting $\mathsf{Ty}_{\mathcal{V}}$ and $\mathsf{Tm}_{\mathcal{V}}$ in terms of (dependent) products rather than families and sections, we are able to equip both with strictly functorial actions.

We emphasize that the accomplishment here is not in the definition itself; it mirrors the naïve definition of the set model of type theory as presented by e.g., Hofmann (1997). What is crucial is that V^0 retains enough of the good behavior of $\mathsf{hSet}_{\mathcal{U}}$ to support such a straightforward definition of the CwF structure while still managing to be an h-set itself.

Remark 4. We note that there are many closely related presentations of models of type theory (categories with attributes (Cartmell, 1986), contextual categories (Cartmell, 1986; Streicher, 1991), comprehension categories (Jacobs, 1993; Jacobs, 1999), natural models (Fiore, 2012; Awodey, 2018) and so on). We have opted for CwFs because the CwF structure on V^0 is particularly simple and enjoys an exceptional number of definitional equalities. In particular, as opposed to other models which recover terms indirectly as sections to display maps, CwFs require the presheaf of terms as part of their data. This allows us to choose a particular definitional representative for the type of terms in our model and we are then able to explicitly select dependent functions. We shall see that this makes closing $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{V}})$ under various constructions particularly straightforward, as most naturality conditions hold definitionally.

5.3 Further structure on V^0 as a CwF

While we have constructed a CwF structure on $\mathcal{V}$, we have thus far only shown that the model interprets the basic structural rules of type theory, but not that it is closed under any connectives. The process of extending the model with new connectives is essentially modular: for each connective, we specify the relevant structure on top of a CwF necessary to interpret it and then show that the CwF $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{V}})$ supports this additional structure.

We illustrate the process with Π-types. First, we must define a Π-structure on a CwF.

Definition 11 (Π-structure,). A Π-structure on a CwF $(\mathcal{C}, \mathsf{Ty}_{\mathcal{C}}, \mathsf{Tm}_{\mathcal{C}})$ is defined by the following:

- An operation $\mathsf{pi}_{\mathcal{C}} : \prod_{\Gamma : \mathsf{Ob}\,\mathcal{C}} \mathsf{Ty}_{\mathcal{C}}\,\Gamma \to \mathsf{Ty}_{\mathcal{C}}\,(\Gamma.\,A) \to \mathsf{Ty}_{\mathcal{C}}\,\Gamma$, natural in Γ.
- For any $\Gamma : \mathsf{Ob}\,\mathcal{C}$, $A : \mathsf{Ty}_{\mathcal{C}}\,\Gamma$, and $B : \mathsf{Ty}_{\mathcal{C}}\,(\Gamma.\,A)$ an isomorphism $\alpha_{\mathsf{pi}_{\mathcal{C}}}$ between $\mathsf{Tm}_{\mathcal{C}}\,(\Gamma, \mathsf{pi}_{\mathcal{C}}\,\Gamma\,A\,B)$ and $\mathsf{Tm}_{\mathcal{C}}\,(\Gamma.A, B)$, natural in Γ.

Remark 5. One may unpack the content of $\alpha_{\mathsf{pi}_{\mathcal{C}}}$ to see that it includes the introduction, elimination, β-, and η-rules for Π-types. The additional requirement of naturality enforces the stability of the introduction and elimination rules under substitution.

Lemma 8 ( ). *The CwF* $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{V}})$ *supports a* Π*-structure.*

Proof. We begin by defining $\mathsf{pi}_{\mathcal{V}}$ as follows:

$$\mathsf{pi}_{\mathcal{V}}\,\Gamma\,A\,B := \lambda(\gamma : \mathsf{El}^0\,(\Gamma)).\,\Pi^0\,(A\,\gamma)\,(\lambda a.B\,(\gamma, a))$$

Naturality in Γ is a straightforward computation. The definition of $\alpha_{\mathsf{pi}_{\mathcal{V}}}$, after unfolding, reduces to the manifestly natural equivalence induced by currying:

$$\prod_{\gamma:\Gamma}\prod_{y:Y(\gamma)} Z(\gamma,y) \simeq \prod_{p:\sum_{\gamma:\Gamma} Y(\gamma)} Z(p) \qquad \square$$

We have formalized both the definition of Π-structures and the particular Π-structure on $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{V}})$ in `Agda`. However, already some small inconveniences emerge. For instance, in the definition of naturality for $\alpha\mathsf{pi}_{\mathcal{V}}$, we must specify an equality dependent on the proof witnessing naturality of $\mathsf{pi}_{\mathcal{V}}$. The dependence is straightforward in this case, but becomes more complex for the later structures. Accordingly, we present only paper proofs for them.

Furthermore, $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{C}})$ also supports dependent sums.

Definition 12 (Σ-structure)**.** A Σ-structure on a CwF $(\mathcal{C}, \mathsf{Ty}_{\mathcal{C}}, \mathsf{Tm}_{\mathcal{C}})$ consists of the following two pieces of data:

- An operation $\mathsf{sig}_{\mathcal{C}} : \prod_{\Gamma:\mathsf{Ob}\,\mathcal{C}} \mathsf{Ty}_{\mathcal{C}}\,\Gamma \to \mathsf{Ty}_{\mathcal{C}}\,(\Gamma.\,A) \to \mathsf{Ty}_{\mathcal{C}}\,\Gamma$, natural in Γ.
- For any $\Gamma : \mathsf{Ob}\,\mathcal{C}$, $A : \mathsf{Ty}_{\mathcal{C}}\,\Gamma$, and $B : \mathsf{Ty}_{\mathcal{C}}\,(\Gamma.\,A)$ a natural isomorphism $\alpha_{\mathsf{sig}_{\mathcal{C}}}$ between $\mathsf{Tm}_{\mathcal{C}}\,(\Gamma, \mathsf{sig}_{\mathcal{C}}\,\Gamma\,A\,B)$ and pairs $\sum_{a:\mathsf{Tm}_{\mathcal{C}}(\Gamma,A)} \mathsf{Tm}_{\mathcal{C}}\,(\Gamma, B \cdot (\mathsf{id}.a))$.

Lemma 9. *The CwF* $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{V}})$ *supports a* Σ*-structure.*

Proof. We define $\mathsf{sig}_{\mathcal{V}}$ as follows:

$$\mathsf{sig}_{\mathcal{V}}\,\Gamma\,A\,B := \lambda(x : \mathsf{El}^0\,\Gamma).\,\Sigma^0\,(A\,x)\,(\lambda a.B\,(x,a))$$

The remaining structure follows directly. In particular, even though the naturality requires complex path algebra to state properly in the specific CwF on $\mathcal{V}$ all these paths are given by reflexivity. $\square$

We next consider a representative inductive type: booleans.

Definition 13 (bool-structure)**.** A bool-structure on a CwF $(\mathcal{C}, \mathsf{Ty}_{\mathcal{C}}, \mathsf{Tm}_{\mathcal{C}})$ consists of the following two pieces of data:

- An operation $\mathsf{bool}_{\mathcal{C}} : \prod_{\Gamma:\mathsf{Ob}\,\mathcal{C}} \mathsf{Ty}_{\mathcal{C}}\,\Gamma$ natural in Γ.
- A pair of operations $\mathsf{true}_{\mathcal{C}}, \mathsf{false}_{\mathcal{C}} : \prod_{\Gamma:\mathsf{Ob}\,\mathcal{C}} \mathsf{Tm}_{\mathcal{C}}\,(\Gamma, \mathsf{bool}_{\mathcal{C}})$ also natural in Γ.

Furthermore, the following canonical map induced by substitution must be an isomorphism for each $\Gamma : \mathsf{Ob}\,\mathcal{C}$ and $A : \mathsf{Ty}_{\mathcal{C}}\,(\Gamma.\,\mathsf{bool}_{\mathcal{C}})$:

$$\mathsf{Tm}_{\mathcal{C}}\,(\Gamma.\,\mathsf{bool}_{\mathcal{C}}, A) \to \mathsf{Tm}_{\mathcal{C}}\,(\Gamma, \lambda\gamma.\,A(\gamma, \mathsf{true}_{\mathcal{C}})) \times \mathsf{Tm}_{\mathcal{C}}\,(\Gamma, \lambda\gamma.\,A(\gamma, \mathsf{false}_{\mathcal{C}}))$$

Lemma 10. *The CwF* $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{V}})$ *supports a* bool*-structure.*

Proof. We define $\mathsf{bool}_{\mathcal{V}}$ as follows:

$$\mathsf{bool}_{\mathcal{V}}\,\Gamma = \lambda x.\mathsf{bool}^0$$

By construction, $\mathsf{Tm}_{\mathcal{V}}\,(\Gamma, \mathsf{bool}_{\mathcal{V}})$ is definitionally equal to $\mathsf{El}^0\,\Gamma \to \mathsf{bool}$ and so we define $\mathsf{true}_{\mathcal{V}}$ as $\lambda\gamma.\,\mathsf{true}$ and $\mathsf{false}_{\mathcal{V}}$ as $\lambda\gamma.\,\mathsf{false}$. The calculations that these definitions are natural and that the required map is an isomorphism are routine. □

By similar considerations, we may define and close $(\mathcal{V}, \mathsf{Ty}_{\mathcal{V}}, \mathsf{Tm}_{\mathcal{V}})$ under many other connectives already constructed in Section 3: extensional identity types, natural numbers, and universes among others. Putting all of this together, we conclude the following:

Theorem 6. $\mathcal{V}$ *supports a model of extensional type theory with the standard connectives.*

We note that this result, combined with Proposition 13 and the series of results about El^0 preserving various categorical connectives can be summarized by the informal slogan: $\mathcal{V}$ supports a model of type theory which internalizes the set-level fragment of the ambient type theory.

6 Relationship to set models of type theory and other set universes in HoTT/UF

The idea of set theoretic semantics of type theory is of course an old and natural one. An early reference where this is written down more formally is the master thesis of Salvesen (1984, Chapter 5). As discussed in the introduction the work presented in this paper goes back to the model of CZF in type theory of Aczel (1978). Aczel also interpreted extensional type theory with universes in an extension of CZF with a hierarchy of inaccessible sets (Aczel, 1999). In fact, Aczel's V occurs already in the PhD thesis of Leversha (1976) where it was used to represent ordinals constructively. Various earlier work has also relied on Aczel's V to model type theory. For instance, Werner (1997) modeled the core system of `Coq` in ZFC and vice versa, using Aczel's encoding of sets. A refinement by Barras (2010) and Barras (2012) models the core system of `Coq` system in intuitionistic ZF, and formalizes the model in `Coq` (Coq). More recently, Palmgren (2019) presented an interpretation of extensional Martin-Löf type theory (Martin-Löf, 1982) into intensional Martin-Löf type theory via setoids, also relying on Aczel's V. Palmgren's work was also formalized in `Agda`.

Aczel's V was revisited in HoTT/UF by Gylterud (2018) and Gylterud (2019) who observed that this gives a universe of multisets, but that one can restrict it, as in Definition 5, to get a universe of h-sets. These universes of (multi)sets has recently also been further studied by Escardó and de Jong who has their own `Agda` formalization as part of the `TypeTopology` project (M. Escardó and Tom de Jong, 2023). Among many other things, they have two more proofs of Theorem 3 formalized. Various HITs for representing *finite* multisets have also been considered in HoTT/UF (BGW17; Frumin et al., 2018; Choudhury and Fiore, 2019; Angiuli et al., 2021; Veltri, 2021; Joram and Veltri, 2023), however these are of course not sufficient to model full type theory.

We will now discuss other approaches to constructing strict categories of sets in HoTT/UF that could also serve as internal models of type theory. These often require various extensions of the quite minimal univalent type theory that we have relied on in this paper.

6.1 The cumulative hierarchy in the HoTT Book

The HoTT Book postulates a universe of sets as a higher inductive type called the *cumulative hierarchy* (the HoTT Book, Definition 10.5.1). Gylterud (2018, Section 8) establishes an equivalence between the HoTT Book V and V^0, which makes it possible to transfer all of our results over to V. One remark about the HoTT Book V is that it is h-set truncated, while V^0 is not. This means that the eliminator one gets for the HoTT Book V only lets one directly eliminate into h-sets, while V^0 can be directly eliminated into types of arbitrary homotopy level. Similarly many basic constructions, like $\in : V \to V \to \mathcal{U}$, is a bit more complicated to define for the HoTT Book V as it is not sufficient to only define them for point constructors, but one has to check that the definitions are compatible with the higher constructors as well. A practical and appealing aspect of V^0 is hence that it is easy to define operations by pattern-matching on it. Another is that it is not postulated, but simply constructed from W-types.

6.2 Inductive-recursive universes

An alternative approach to modeling type theory in type theory is to rely on quotient inductive-inductive types (QIITs) as considered by Altenkirch and Kaposi (2016). However, they run into the same problem as discussed above when working in HoTT and trying to eliminate their QIIT into $\mathsf{hSet}_{\mathcal{U}}$. In particular, as the QIIT representation of type theory is h-set truncated they cannot eliminate directly into $\mathsf{hSet}_{\mathcal{U}}$ as it is a 1-type (the same issue also applies to the HoTT Book V). The authors resolve this by considering an inductive-recursive universe closed under the relevant structure, which can be shown to be a set without any need to set truncate. This enjoys many of the nice properties of V^0, like El decoding type constructors definitionally, but induction-recursion is proof theoretically quite strong and it is again interesting to emphasize that we can construct V^0 using only W-types.

6.3 Covered Marked Extensional Well-founded Orders

In their recent paper, T. de Jong et al., 2023 show that the HoTT Book V is equivalent to the type of covered marked extensional well-founded orders ($\mathsf{MEWO}_{\mathsf{cov}}$), and hence to V^0. The results in this paper thus imply that the type $\mathsf{MEWO}_{\mathsf{cov}}$ can be equipped with a universe structure. A strength of the universe V^0 is the computational aspect of the decoding function El^0. Unfortunately, the two underlying maps of the equivalence between V^0 and $\mathsf{MEWO}_{\mathsf{cov}}$ do not compose definitionally to the identity when going from V^0 to $\mathsf{MEWO}_{\mathsf{cov}}$ and back again. This means that the induced decoding for $\mathsf{MEWO}_{\mathsf{cov}}$ given by going to V^0 and then applying El^0 is not as computationally well-behaved as El^0 on V^0, as the decoding will only hold up to propositional equality.

6.4 Relationship to $h\mathcal{S}et_{\mathcal{U}}$

One reason to consider the category of iterative sets is to regard it as a replacement for $h\mathcal{S}et_{\mathcal{U}}$. As noted in Section 4, El^0 induces a fully-faithful functor, but it may fail to be essentially surjective. The statement that El^0 is essentially surjective corresponds to Shulman's *axiom of well-founded materialization* (Shulman, 2010) and which is, in turn, implied by the axiom of choice.

If the functor is essentially surjective, it forms a categorical equivalence between $\mathcal{V}$ and a univalent category and thus describes $h\mathcal{S}et_{\mathcal{U}}$ as the *Rezk completion* (Ahrens, Krzystof Kapulkin, and Shulman, 2015) of $\mathcal{V}$. Informally, this shows, modulo classical axioms, that $\mathcal{V}$ is a more rigid presentation of $h\mathcal{S}et_{\mathcal{U}}$. Moreover, even without additional axioms $\mathcal{V}$ and V^0 are closed under essentially every construction of interest.

6.5 Well-ordered sets

Another approach to defining a strict universe of sets, inspired by Voevodsky's simplicial set model (Krzysztof Kapulkin and Lumsdaine, 2021), is to consider well-ordered sets. By relying heavily on Zermelo's well-ordering principle, and hence choice, one can obtain a strict category of well-ordered sets with the relevant structure, also as a subcategory of $h\mathcal{S}et_{\mathcal{U}}$. This was experimented with in `UniMath` (Voevodsky, Ahrens, Grayson, et al., 2020) by

Mörtberg (2018). However, this turned out to be harder to work with formally than expected because of all the propositional truncations and hence not completed. Furthermore, if completed this would only *merely* give us the existence of an internal model and hence lead to a weaker result than Theorem 6.

7 Conclusion and future work

We have constructed a universe of h-sets that is itself an h-set and structured it into an internal model of extensional type theory. This main result can perhaps also be proven for other h-set universes of h-sets, such as the ones mentioned in Section 6, but certain properties of our construction makes it very convenient to work with, also formally. First and foremost, the definitional decoding of type formers means that one avoids complex transports. Secondly, the construction is carried out using basic type-formers, and has a (provable) elimination principle which directly allows elimination into general types. This development works in a fairly minimalist univalent type theory, as long as it has W-types. These W-types can be large, and need only be small if one wants to reflect a hierarchy of universes, as in Section 3.5. The results should thus have a broad applicability in models of HoTT/UF.

In the formalization, we stopped short of adding additional structure to the CwF on $\mathcal{V}$ after Π-types. The obstacles are in fact not in providing the structure for our model, such as Σ-structure, but the general formulation of what that extra structure constitutes on CwFs based on categories (see Remark 3 in Section 5). To the best of our knowledge there are no other formalizations of CwFs with Σ-structure out there that do not assume UIP or other axioms and which do not use setoids or a more extensional equality. It would be interesting to attempt formalizing this in cubical type theory (CCHM18) where equality of Σ-types is easier to work with because of the primitive path-over types in the form of PathP-types. An experiment along these lines was performed by Vezzosi (2017) in `Cubical Agda` (Vezzosi, Mörtberg, and Abel, 2021). In this small formalization Vezzosi considered the CwF structure on h-set valued presheaves. It would of course not have been possible to fully complete this for the same reason as discussed in this paper, but it turned out that some of the constructions and equations that one has to check were easier than in a corresponding formalization in `UniMath` by Mörtberg (2017). This also suggests a further direction to explore: $\mathcal{V}$ valued presheaves. These should enjoy the same nice properties as $\mathcal{hSet}_{\mathcal{U}}$ valued presheaves, but it should be possible to organize also them into a model of type theory internally in HoTT/UF.

Another avenue of further study is to take a closer look at Shulman's axiom of well-founded materialization. Just like univalence, it makes sense to formulate this axiom relative to a given universe of types. The construction of V^0 can be carried out on any universe, so a reasonable reformulation of well-founded materialization in type theory could be: a universe $\mathcal{U}$ has well-founded materialization if $\mathsf{El}^0 : \mathsf{V}^0_{\mathcal{U}} \to h\mathcal{S}et_{\mathcal{U}}$ is essentially surjective. As mentioned, this follows from AC, but does not seem to be inherently non-constructive. For instance, V^0 itself has well-founded materialization for trivial reasons. The most pertinent question is perhaps whether well-founded materialization and univalence can constructively coexist. If we start with a univalent $\mathcal{U}$, one could take the image of El^0 in $h\mathcal{S}et_{\mathcal{U}}$ to obtain a univalent universe which also somewhat trivially has well-founded materialization. However, it is not immediate that this is closed under Π-types and Σ-types as a naïve attempt quickly runs into choice problems.

References

Aczel, Peter (1978). "The Type Theoretic Interpretation of Constructive Set Theory". In: *Logic Colloquium '77*. Ed. by A. MacIntyre, L. Pacholski, and J. Paris. North–Holland, Amsterdam-New York, pp. 55–66.

— (1999). "On Relating Type Theories and Set Theories". In: *Types for Proofs and Programs*. Ed. by Thorsten Altenkirch, Bernhard Reus, and Wolfgang Naraschewski. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–18. ISBN: 978-3-540-48167-6.

Ahrens, Benedikt, Krzystof Kapulkin, and Michael Shulman (2015). "Univalent categories and the Rezk completion". In: *Mathematical Structures in Computer Science* 25.5, pp. 1010–1039. DOI: `10.1017/S0960129514000486`.

Altenkirch, Thorsten and Ambrus Kaposi (2016). "Type theory in type theory using quotient inductive types". In: *SIGPLAN Not.* 51.1, pp. 18–29. ISSN: 0362-1340. DOI: `10.1145/2914770.2837638`. URL: `https://doi.org/10.1145/2914770.2837638`.

Angiuli, Carlo et al. (2021). "Internalizing representation independence with univalence". In: *Proc. ACM Program. Lang.* 5.POPL. DOI: `10.1145/3434293`. URL: `https://doi.org/10.1145/3434293`.

Awodey, Steve (Oct. 2013). "Structuralism, Invariance, and Univalence". In: *Philosophia Mathematica* 22.1, pp. 1–11. ISSN: 0031-8019. DOI: `10.1093/philmat/nkt030`.

— (2018). "Natural models of homotopy type theory". In: *Mathematical Structures in Computer Science* 28.2, pp. 241–286. DOI: `10.1017/S0960129516000268`.

Barras, Bruno (2010). "Sets in Coq, Coq in Sets". In: *Journal of Formalized Reasoning* 3 (1), pp. 29–48.

Barras, Bruno (2012). "Semantical Investigations in Intuitionistic Set Theory and Type Theories with Inductive Families". Available at `http://www.lsv.fr/~barras/habilitation/`.

Basold, Henning, Herman Geuvers, and Niels van der Weide (2017). "Higher Inductive Types in Programming". In: *Journal of Universal Computer Science* 23.1, pp. 63–88. DOI: `10.3217/jucs-023-01-0063`.

Cartmell, John (1986). "Generalised algebraic theories and contextual categories". In: *Annals of Pure and Applied Logic* 32, pp. 209–243. ISSN: 0168-0072. DOI: `10.1016/0168-0072(86)90053-9`.

Choudhury, Vikraman and Marcelo Fiore (2019). *The finite-multiset construction in HoTT.* Talk at the International Conference on Homotopy Type Theory (HoTT 2019). URL: `https://hott.github.io/HoTT-2019/conf-slides/Choudhury.pdf`.

Cohen, Cyril et al. (2018). "Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom". In: *21st International Conference on Types for Proofs and Programs (TYPES 2015)*. Ed. by Tarmo Uustalu. Vol. 69. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 5:1–5:34. ISBN: 978-3-95977-030-9. DOI: `10.4230/LIPIcs.TYPES.2015.5`.

Curien, Pierre-Louis, Richard Garner, and Martin Hofmann (2014). "Revisiting the categorical interpretation of dependent type theory". In: *Theoretical Computer Science* 546. Models of Interaction: Essays in Honour of Glynn Winskel, pp. 99–119. ISSN: 0304-3975. DOI: `10.1016/j.tcs.2014.03.003`.

Dybjer, Peter (1996). "Internal type theory". In: *Types for Proofs and Programs (TYPES 1995).* Ed. by Stefano Berardi and Mario Coppo. Vol. 1158. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 120–134. ISBN: 978-3-540-70722-6. DOI: `10.1007/3-540-61780-9_66`.

Escardó, Martín and Tom de Jong (2023). *Iterative multisets, iterative sets, and iterative ordinals in the TypeTopology library.* Available at `https://www.cs.bham.ac.uk/~mhe/TypeTopology/Iterative.index.html`.

Escardó, Martín Hötzel (2019). *Introduction to univalent foundations of mathematics with Agda.* Available at `https://www.cs.bham.ac.uk/~mhe/HoTT-UF-in-Agda-Lecture-Notes/index.html`.

Fiore, Marcelo (2012). *Discrete generalised polynomial functors.* Slides from talk given at ICALP 2012. URL: `https://www.cl.cam.ac.uk/~mpf23/talks/ICALP2012.pdf`.

Frumin, Dan et al. (2018). "Finite Sets in Homotopy Type Theory". In: *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs.* CPP 2018. Los Angeles, CA, USA: Association for Computing Machinery, pp. 201–214. ISBN: 9781450355865. DOI: 10.1145/3167085.

Gylterud, Håkon Robbestad (2018). "From multisets to sets in homotopy type theory". In: *The Journal of Symbolic Logic* 83.3, pp. 1132–1146. DOI: 10.1017/jsl.2017.84.

— (2019). "Multisets in type theory". In: *Mathematical Proceedings of the Cambridge Philosophical Society*, pp. 1–18. DOI: 10.1017/S0305004119000045.

Hofmann, Martin (1997). "Syntax and Semantics of Dependent Types". In: *Semantics and Logics of Computation*. Ed. by Andrew M. Pitts and Peter Dybjer. Publications of the Newton Institute. Cambridge University Press, pp. 79–130.

Jacobs, Bart (1993). "Comprehension categories and the semantics of type dependency". In: *Theoretical Computer Science* 107.2, pp. 169–207. ISSN: 0304-3975. DOI: 10.1016/0304-3975(93)90169-T.

— (1999). *Categorical Logic and Type Theory*. Studies in Logic and the Foundations of Mathematics 141. Amsterdam: North Holland.

Jong, T. de et al. (2023). "Set-Theoretic and Type-Theoretic Ordinals Coincide". In: *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1–13. DOI: 10.1109/LICS56636.2023.10175762. URL: https://doi.ieeecomputersociety.org/10.1109/LICS56636.2023.10175762.

Joram, Philipp and Niccolò Veltri (2023). "Constructive Final Semantics of Finite Bags". In: *14th International Conference on Interactive Theorem Proving (ITP 2023)*. Ed. by Adam Naumowicz and René Thiemann. Vol. 268. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 20:1–20:19. ISBN: 978-3-95977-284-6. DOI: 10.4230/LIPIcs.ITP.2023.20. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITP.2023.20.

Kapulkin, Krzysztof and Peter Lumsdaine (Mar. 2021). "The simplicial model of Univalent Foundations (after Voevodsky)". In: *Journal of the European Mathematical Society* 23.6, pp. 2071–2126. DOI: 10.4171/JEMS/1050.

Leversha, Gerald (1976). "Formal Systems for Constructive Mathematics". PhD thesis. University of Manchester.

Martin-Löf, Per (Jan. 1, 1975). "An Intuitionistic Theory of Types: Predicative Part". In: *Studies in Logic and the Foundations of Mathematics*. Ed. by H. E. Rose and J. C. Shepherdson. Vol. 80. Logic Colloquium '73. Elsevier, pp. 73–118. DOI: 10.1016/S0049-237X(08)71945-1.

— (1982). "Constructive mathematics and computer programming". In: *Logic, Methodology and Philosophy of Science VI, Proceedings of the Sixth International Congress of Logic, Methodology and Philosophy of Science,* *Hannover 1979*. Ed. by L. Jonathan Cohen et al. Vol. 104. Studies in Logic and the Foundations of Mathematics. North-Holland, pp. 153–175. DOI: 10.1016/S0049-237X(09)70189-2.

Mörtberg, Anders (2017). *Type category structure on h-set valued presheaves in UniMath*. Available at https://github.com/UniMath/TypeTheory/blob/master/TypeTheory/Instances/Presheaves.v.

— (Feb. 2018). *Category structures on well-ordered sets in UniMath*. Available at https://github.com/UniMath/UniMath/blob/master/UniMath/CategoryTheory/categories/wosets.v.

Palmgren, Erik (2019). *From type theory to setoids and back*. arXiv:1909.01414 [math]. DOI: 10.48550/arXiv.1909.01414. URL: https://arxiv.org/abs/1909.01414.

Rijke, Egbert (2017). "The join construction". In: *ArXiv e-prints*. arXiv: 1701.07538 [math.CT].

Rijke, Egbert and Bas Spitters (2015). "Sets in homotopy type theory". In: *Mathematical Structures in Computer Science* 25.5, pp. 1172–1202. DOI: 10.1017/S0960129514000553.

Rijke, Egbert, Elisabeth Stenholm, et al. (2024). *The agda-unimath library*. URL: https://github.com/UniMath/agda-unimath/.

Salvesen, Anne B. (1984). "Typeteori - en studie". Master's thesis. University of Oslo.

Seely, R. A. G. (1984). "Locally cartesian closed categories and type theory". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 95.1, pp. 33–48. DOI: 10.1017/S0305004100061284.

Shulman, Michael (Apr. 2010). "Stack semantics and the comparison of material and structural set theories". In: DOI: 10.48550/arXiv.1004.3802. URL: https://arxiv.org/abs/1004.3802.

Streicher, Thomas (1991). *Semantics of Type Theory*. Birkhäuser Boston. ISBN: 9781461204336. DOI: 10.1007/978-1-4612-0433-6. URL: http://dx.doi.org/10.1007/978-1-4612-0433-6.

The Agda development team (2024). *Agda*. URL: https://github.com/agda/agda.

The Coq Development Team (2021). *The Coq Proof Assistant*. URL: https://www.coq.inria.fr.

The Univalent Foundations Program (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: homotopytypetheory.org. URL: http://homotopytypetheory.org/book.

Veltri, Niccolò (2021). "Type-Theoretic Constructions of the Final Coalgebra of the Finite Powerset Functor". In: *6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)*. Ed. by Naoki Kobayashi. Vol. 195. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-

Zentrum für Informatik, 22:1–22:18. ISBN: 978-3-95977-191-7. DOI: 10.4230/LIPIcs.FSCD.2021.22. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.FSCD.2021.22.

Vezzosi, Andrea (2017). *Parts of a CwF structure on h-set valued presheaves in Cubical Agda*. Available at ttps://github.com/Saizan/cubical-demo/blob/master/examples/Cubical/Examples/FunctorCWF.agda.

Vezzosi, Andrea, Anders Mörtberg, and Andreas Abel (2021). "Cubical Agda: A dependently typed programming language with univalence and higher inductive types". In: *Journal of Functional Programming* 31, e8. DOI: 10.1017/S0956796821000034.

Voevodsky, Vladimir (2010). *The equivalence axiom and univalent models of type theory*. Notes from a talk at Carnegie Mellon University, available at http://www.math.ias.edu/vladimir/files/CMU_talk.pdf.

— (2011). *Resizing rules*. Talk at TYPES workshop 2011, Bergen, Norway. Available at https://www.math.ias.edu/vladimir/sites/math.ias.edu.vladimir/files/2011_Bergen.pdf.

— (2015). "An experimental library of formalized Mathematics based on the univalent foundations". In: *Mathematical Structures in Computer Science* 25.5, pp. 1278–1294. DOI: 10.1017/S0960129514000577.

Voevodsky, Vladimir, Benedikt Ahrens, Daniel Grayson, et al. (2020). *UniMath — a computer-checked library of univalent mathematics*. https://github.com/UniMath/UniMath.

Werner, Benjamin (1997). "Sets in Types, Types in Sets". In: *Proceedings of the Third International Symposium on Theoretical Aspects of Computer Software*. TACS '97. Berlin, Heidelberg: Springer-Verlag, pp. 530–346. ISBN: 354063388X.

Paper II

Univalent Material Set Theory

Håkon Robbestad Gylterud and *Elisabeth Stenholm*

Abstract

Homotopy type theory (HoTT) can be seen as a generalisation of structural set theory, in the sense that 0-types represent structural sets within the more general notion of types. For material set theory, we also have concrete models as 0-types in HoTT, but this does not currently have any generalisation to higher types. The aim of this paper is to give such a generalisation of material set theory to higher type levels within Homotopy Type Theory. This is achieved by generalising the construction of the type of iterative sets (Gylterud, 2018) to obtain an n-type universe of n-types. At level 1, this gives a connection between groupoids and multisets.

More specifically, we define the notion of an $\in$-structure as a type with an extensional binary type family and generalise the axioms of constructive set theory to higher type levels. There is a tight connection between the univalence axiom and the extensionality axiom of $\in$-structures. Once an $\in$-structure is given, its elements can be seen as representing types in the ambient type theory. A useful property of these structures is that an $\in$-structure of n-types is itself an n-type, as opposed to univalent universes, which have higher type levels than the types in the universe.

The theory has an alternative, coalgebraic formulation, in terms of coalgebras for a certain hierarchy of functors, P^n, which generalises the powerset functor from sub-types to covering spaces and n-connected maps in general. The coalgebras which furthermore are fixed-points of their respective functors in the hierarchy are shown to model the axioms given in the first part.

As concrete examples of models for the theory developed we construct the initial algebras of the P^n functors. In addition to being an example of initial algebras of non-polynomial functors, this construction allows one to start with a univalent universe and get a hierarchy of $\in$-structures which gives a stratified $\in$-structure representation of that universe. These types are moreover n-type universes of n-types which contain all the usual types an type formers. The universes are cumulative both with respect to universe levels and with respect to type levels.

All the results are formalised in the proof-assistant Agda.

1 Introduction

Material set theories are set theories which emphasise the notion of sets as collections of elements (often themselves sets) and where the identity of individual elements is tracked across sets, usually with a global, binary membership relation ($\in$). This category includes the traditional Zermelo–Fraenkel (ZF) set theory. ZF is a theory in the language of first-order logic, intended as a foundation for mathematics. In what follows, we work within the framework of Homotopy Type Theory (HoTT), fully formalized in Agda. HoTT is a structural framework, with the Univalence Axiom in particular allowing identification of types which are structurally the same, i.e. equivalent – deemphasising the individual elements and their identity outside the structure of the type. Taking a step back, one can see HoTT as a generalisation of structural set theory to higher type levels. The high-level question we attempt to answer in this paper is: What is the corresponding generalisation of material set theory to higher type levels? From this vantage point we will regard material set theory in the same way that a classical mathematician regards group theory. Namely, we study a certain type of mathematical structures, and interest ourselves in their properties and relationships. The structures we study are not groups, but what we call $\in$-structures: types with an extensional elementhood relation.

Since we are working in HoTT, we can consider the type level of the underlying type of sets and of the relation of an $\in$-structure. In classical set theory, the statement "$x \in y$" is a proposition. But in our setting, we can consider $\in$-structures where $x \in y$ is a type of any type level. Similarly, we can generalise the axioms of ZF to higher type levels. An example of such a $\in$-structure, where $x \in y$ is allowed to be a type of any level, was considered in *Multisets in type theory* (Gylterud, 2019) by one of the authors.

We aim to give a higher level generalisation of material set theory, by considering $\in$-structures where $x \in y$ is an n-type. With care, the usual properties, which we know and love from set-theory, can be reformulated and proven to hold in our models. But sometimes what used to be a single property generalises to several properties when taking higher type levels into consideration. Let us, for the sake of building some intuition, take a closer look at level 1 of this generalisation.

In a level 1 $\in$-structure, elements are related by a set-valued $\in$-relation: given two elements, $x, y : V$ the type $x \in y$ is a set. One way of understanding this to think of y as a multiset where x may occur more than once. For instance, if $x \in y$ is a finite type with n elements, then we can think of this as saying that x occurs n times in y. The generalised properties support this interpretation: For instance, level 1 (unordered) tupling allows the formation of multisets of the form $\{x_0, \cdots, x_n\}_1$ where repetitions will be counted separately. But a level 1 $\in$-structure may also support level 0 (unordered) tupling, with a separate operation, $\{x_0, \cdots, x_n\}_0$, which becomes a set: $x \in \{x_0, \cdots, x_n\}_0$ being proposition for any x.

The connection between level 1 $\in$-structures and level 1 types, i.e. groupoids, is the (perhaps surprising) observation that these multisets represent groupoids. First of all, a level 1 $\in$-structure is itself a groupoid: The identifications between multisets are free to permute the occurrences of a given element, giving rise to non-trivial automorphisms. For instance, the type $\{\emptyset, \emptyset\}_1 = \{\emptyset, \emptyset\}_1$ has two distinct elements. A consequence of this is that if we look at the total type of elements of a multiset, $\mathrm{El}\ x := \sum_{y:V} y \in x$, we get a groupoid – the groupoid represented by x. At first glance, it might seem as though $\mathrm{El}\ x$ might always be a set. For instance, $\mathrm{El}\,\{\emptyset, \emptyset\}_1$ is a set with two elements. But, by nesting multisets, we can represent other groupoids. For instance, the cyclic group with two elements (as a groupoid) is represented by $\{\{\emptyset, \emptyset\}_1\}_0$, the set which contains the multiset $\{\emptyset, \emptyset\}_1$ exactly once. The reason why $\mathrm{El}\,\{\{\emptyset, \emptyset\}_1\}_0 = B(\mathbb{Z}_2)$ is a bit subtle. Notice, the alternation of subscripts on the tuplings. Had we instead chosen $\{\{\emptyset, \emptyset\}_1\}_1$, we would have two occurrences of $\{\emptyset, \emptyset\}_1$, because of its two automorphisms, while (perhaps counter-intuitively) $\mathrm{El}\,\{\{\emptyset, \emptyset\}_1\}_1$ is the unit type. When we do a 0-singleton of a multiset, say x, the total type is in general the connected component, because $y \in \{x\}_0 \simeq \|y = x\|_{-1}$ and hence $\mathrm{El}\,\{x\}_0 \simeq \sum_{y:V} \|y = x\|_{-1}$. So, if x, as $\{\emptyset, \emptyset\}_1$ does, has non-trivial automorphisms, these will show up in $\mathrm{El}\,\{x\}_0$. In a strong enough level 1 $\in$-structure, any (small) group can be represented.

There is an immediate connection between univalent set theory and homotopy type theory, whereby there is an equivalence between $\in$-structures and coalgebras for the n-truncated maps functors $\mathrm{P}_U^{n+1} : \mathrm{Type} \to \mathrm{Type}$, which associates to each type X the type of n-truncated maps into X. Thus, $\mathrm{P}_U^0\, X$ is the type of subtypes of X, while $\mathrm{P}_U^1\, X$ is the type of covering spaces of X and so on. We show that these functors have initial algebras, V^n, and determine the univalent set theory axioms satisfied by the initial algebras and other fixed-points of these functors. These initial algebras hence form a family of models of univalent material set theory, motivating the axioms and interpolating between the standard iterative hierarchy and the generalised multisets.

One way in which univalent material set theory distinguishes itself in HoTT is that type levels are *off by one*. What this means is that models form n-type based families of n-types: if $A, B : V$ are n-types in $(V, \in)$ (a notion made precise in Definition 4) then $A = B$ is an $n-1$-type. This means that V is an n-type. This contrasts the situation for univalent universes, where a well-known result (Kraus, 2015) states that if U contains strict n-types the type level of U is itself at least $n+1$.

The model given by the initial algebra V^n is of level n, and thus the type V^n is an n-type. The type can be equipped with the structure of a Tarski style universe. The decoding of an element in V^n is an n-type, so V^n is an n-type universe of n-types. Moreover, the decoding holds up to definitional equality, making it very ergonomic to use.

Type levels being off by one might seem strange at first, but it is caused by the $\in$-relation imposing extra structure and thereby killing automorphisms. This observation generalises what is known about the cumulative hierarchy in models of (usual) set theory in HoTT, where V is a set of sets. Especially in category theory, this can be useful to strictify structures. For instance, as explored in Paper I, when recreating the category with family structure on sets in HoTT, one finds oneself blocked by the fact that the types in a context forms a strict groupoid, not a set. By using an $\in$-structure as the category of contexts, the off-by-one property sidesteps this block, yielding a good category with families.

Another perspective we explore is extracting types from $\in$-structures. A given element of an $\in$-structure has a type of elements, and considering the whole $\in$-structure we can ask what types can be represented as types of elements within it. Some insights into how replacement affects representations of types, such as $\mathbb{N}$ for the axiom of infinity, has been collected in Section 3. In particular, we show that the replacement property in set theory says that the $\in$-structure supports all choices of representations of a type equally (Proposition 14).

1.1 Contributions

The following are the main contributions of the paper.

- Construction of initial algebras for the non-polynomial functors P^n_U, generalising the construction of the type of iterative of sets as the initial algebra for the powerset functor to higher type levels (Theorem 15).
- Proof that these initial algebras are n-type universes of n-types, with definitional decoding (Section 7).
- Generalisation of the axioms of set theory to properties of $\in$-structures of any type level (Section 2).
- A framework for representations of types in $\in$-structures. This is applied to give a new formulation of the axiom of infinity, which does not fix a specific encoding of the natural numbers (Definition 5.7).
- Equivalence of P^n_U-coalgebras and U-like $\in$-structures, generalising the well-known connection between coalgebra and set theory (Theorem 3).
- Proof that any fixed-point of P^n_U is a model of the generalisations of the axioms of set theory, except foundation, both generalising and proving

in HoTT the result by Rieger (Rieger, 1957) (Section 5).
- New and short proof of the fiberwise equivalence lemmas: equivalence of families of maps (resp. equivalences) and maps (resp. equivalences) of total spaces respecting the first coordinate (Lemma 5 and Corollary 2).

Some of the ideas and definitions of this article were present in an unpublished preprint, titled “Non-wellfounded sets in HoTT” (Gylterud and Bonnevier, 2020). This preprint however, had a flawed argument in its fourth section and the main construction of that preprint cannot be carried out as described there. The results from Sections 2 and 3 of the preprint, which were correct, have been generalised to higher type levels. These generalisations can now be found in Section 2 and 5 of the current paper.

1.2 Formalisation

Everything in this paper has been formalised in the `Agda` proof assistant (The Agda development team, 2024). Our formalisation builds on the `agda-unimath` library (Rijke et al., 2024), which is an extensive library of formalised mathematics from the univalent point of view.

The formalisation for this paper can be found at: `https://git.app.uib.no/hott/hott-set-theory`. Throughout the paper there will also be clickable links to specific lines of Agda code corresponding to a given result. These will be shown as the Agda logo .

1.3 Notation and universes

A lot of the basic constructions within HoTT have an established notation at this point in time. Nevertheless, to avoid confusion, we include here a list of some of the, perhaps less obvious, notation we will use in this paper. The notation we do not include in this list will usually follow the conventions of the HoTT Book (The Univalent Foundations Program, 2013).

- $\mathbf{0}$ denotes the empty type, with eliminator ex-falso.
- $\mathbf{1}$ denotes the unit type.
- $\mathbf{2}$ denotes the type with two elements.
- $\mathbb{N}$ denotes the type of natural numbers, with constructors 0 and s.
- id-equiv denotes the identity equivalence, on a given type.
- $f \sim g$ denotes the type of homotopies from f to g: $\prod_{x:X} f\,x = g\,x$.
- refl-htpy denotes the homotopy $f \sim f$ given by the map $\lambda\,x.$ refl.
- fiber $f\,y$ denotes the homotopy fiber: $\sum_{x:X} f\,x = y$.
- π_0 and π_1 denote the first, respectively second, projection out of a Σ-type.

- Given a path $p : x = y$, p^{-1} denotes the inverse path $y = x$.
- Given a family P of types over X and a path $p : x = y$, tr_p^P denotes the transport function from $P\,x$ to $P\,y$ over p.
- Given types A and B and a path $p : A = B$, $\mathrm{coe}\ p : A \to B$ is the map defined by path induction, taking the identity map for refl.
- Given an invertible function f (usually an equivalence), f^{-1} denotes the inverse.
- Given a family of maps $f : \prod_{x:X} P\,x \to Q\,x$, $\mathrm{tot}\ f : \sum_{x:X} P\,x \to \sum_{x:X} Q\,x$ is the function: $\lambda\,(x,p).\,(x, f\,x\,p)$.
- $\text{is-}n\text{-trunc-map}\ f$ is the proposition that f is an n-truncated map:

$$\prod_{y:Y} \text{is-}n\text{-type}\,(\text{fiber}\ f\,y).$$

- $X \hookrightarrow Y$ is the type of propositionally truncated maps:

$$\sum_{f:X\to Y} \text{is-}(-1)\text{-trunc-map}\ f.$$

- $X \hookrightarrow_n Y$ is the type of n-truncated maps: $\sum_{f:X\to Y} \text{is-}n\text{-trunc-map}\ f$.
- $X \twoheadrightarrow Y$ is the type of (-1)-connected maps:

$$\sum_{f:X\to Y} \prod_{y:Y} \text{is-contr}\ \|\,\text{fiber}\ f\,y\|_{-1}.$$

- $X \twoheadrightarrow_n Y$ is the type of n-connected maps:

$$\sum_{f:X\to Y} \prod_{y:Y} \text{is-contr}\ \|\,\text{fiber}\ f\,y\|_{n}.$$

- $\exists!_{x:X} P\,x$ denotes the type: $\text{is-contr}\ (\sum_{x:X} P\,x)$.
- funext is the function $f \sim g \to f = g$ given by function extensionality.
- ua is the function $X \simeq Y \to X = Y$ given by univalence.
- Prop_U is the type of all propositions in the universe U, i.e. the type $\sum_{X:U} \text{is-prop}\ X$.
- Set_U is the type of all sets in the universe U, i.e. the type $\sum_{X:U} \text{is-set}\ X$.
- More generally, $n\text{-Type}_U$ is the type of all n-types in the universe U, i.e. the type $\sum_{X:U} \text{is-}n\text{-type}\ X$.

We will use the same terminology as the HoTT Book regarding type levels. But we will also define a notion of *level* for $\in$-structures and elements in (the carrier of) an $\in$-structure. This overloading of terminology should be fine however, since it should be clear from the context what kind of level we are referring to. The notions of *mere proposition* and *mere set* are used to denote types of level -1 and 0 respectively, when there is need for clarity.

In this paper we will assume two type universes[*], a large univalent universe, denoted Type, and a small univalent universe, denoted U. We use cumulative universes, i.e. U : Type and X : Type, for all $X : U$. It is assumed that both U and Type are closed under the usual type formers, such as Π-types, Σ-types, and identity types. The constructions below also use the empty type and the type of natural numbers. We will use function extensionality freely.

From Section 4 and onwards, we will also assume that we can construct small images in certain situations. This assumption is informed by Rijke's modified join construction (Rijke, 2017), which can be used to construct such small images. One can alternatively assume that U is closed under homotopy colimits, from which the smallness of (certain) images follows by the join construction.

2 ∈-structures

In this section we give the definition of ∈-structures[†] and formulate properties of these inspired by set theory. Most of the properties are indexed by a type level, from 0 to ∞. The level 0 version of the property is equivalent to the usual set theoretic concept for ∈-structures of level 0, while the ∞ version was explored in Gylterud (2019).

Definition 1 ( ). An **∈-structure** is a pair $(V, \in)$ where V : Type and $\in : V \to V \to \text{Type}$, which is **extensional**: for each $x, y : V$, the canonical map $x = y \to \prod_{z:V} z \in x \simeq z \in y$ is an equivalence of types.

Extensionality states that we can distinguish sets by their elements. It is expressed in first order logic using logical equivalence, but since we are working in the framework of HoTT and want to allow for elementhood relations of higher type level, we use instead equivalence of types. This of course reduces to logical equivalence in the case when the ∈-relation is propositional.

Many times we will want to talk about all members of a given element in V. We introduce a notation for this.

Definition 2 ( ). Given an ∈-structure, $(V, \in)$, we define the family $\text{El} : V \to \text{Type}$ by $\text{El}\ a :\equiv \sum_{x:V} x \in a$.

[*]The formalisation differs in this respect as it uses a hierarchy of universes, but the relationship between U and Type in the article, is the same as the relationship between `UU i` and `UU (lsuc i)` in the formalisation.

[†]This section defines the notion of ∈-structure slightly differently than ∈-structures were defined in a previous article by one of the authors (Gylterud, 2018). This difference is by incorporating extensionality and generalising to higher type levels.

The usual notion of (extensional) model of set theory corresponds to $\in$-structures, $(V, \in)$, where $x \in y$ is a mere proposition for each $x, y : V$ (and consequently V is a mere set). However, there are examples of extensional $\in$-structures where V is not a mere set. One such example, based on Aczel's (Aczel, 1978) type $\mathrm{W}_{A:U}\, A$, was explored in an article by one of the authors (Gylterud, 2019).

We can stratify $\in$-structures based on the type level of the $\in$-relation.

Definition 3 ( ). Given $n : \mathbb{N}_{-2}$, an $\in$-structure $(V, \in)$ is said to be of **level (n+1)** if for every $x, y : V$ the type $x \in y$ is an n-type.

The following proposition explains the occurrence of $n{+}1$ in the definition above:

Proposition 1 ( ). *In an $\in$-structure, $(V, \in)$, of level n the type V is an n-type.*

Proof. Let $(V, \in)$ be an $\in$-structure of level $(n + 1)$, for $n : \mathbb{N}_{-2}$. By extensionality $x = y$ is equivalent to $\prod_{z:V} z \in x \simeq z \in y$, which is an n-type, hence V is an $(n + 1)$-type. $\square$

Remark: By definition there are no -2 level $\in$-structures, and the -1 level structures all have trivial $\in$-structure. Thus, we shall from here on focus on the $\in$-structures of level 0 or higher.

Of special interest will be the elements of an $\in$-structure which look like sets, in the sense that elementhood is a proposition. This is by definition the case for all elements in $\in$-structures of level 0, but such elements may occur in structures of all levels.

Definition 4 ( ). An element $x : V$ is a $(k + 1)$-type in $(V, \in)$ if $y \in x$ is of level $k : \mathbb{N}_{-1}$ for all $y : V$.

If $x : V$ is a 0-type in $(V, \in)$, we also say it is a *mere set* in $(V, \in)$.

Proposition 2 ( ). *If $x : V$ is a $(k + 1)$-type in $(V, \in)$ then $x = y$ is a k-type for any $y : V$.*

Proof. By extensionality, the type $x = y$ has the same type level as the type $\prod_{z:V} z \in x \simeq z \in y$. The latter is a k-type as $z \in x$ is a k-type for every $z : V$. $\square$

Note that $x : V$ being a k-type in $(V, \in)$ does not imply that $\mathrm{El}\, x$ is a k-type. However, if V is a k-type then $\mathrm{El}\, x$ is a k-type if x is a k-type.

2.1 Ordered pairing

The characteristic property of ordered pairs is that two pairs are equal if and only if the first coordinates are equal and the second coordinates are equal, i.e. it is a pair where the order of the elements matters. In our setting this means that the ordered pair of two elements $x, y : V$ should be an element $\langle x, y\rangle : V$ such that for any other ordered pair $\langle x', y'\rangle$, for some $x', y' : V$, $\langle x, y\rangle = \langle x', y'\rangle$ exactly when $x = x'$ and $y = y'$. Using the characterisation of the identity types of cartesian products, this is equivalently saying that $\langle x, y\rangle = \langle x', y'\rangle$ exactly when $(x, y) =_{V\times V} (x', y')$. In order to allow for higher level ∈-structures the "exactly when" should be replaced with type equivalence. Moreover, we want it to be the canonical one, in the sense that if $x \equiv x'$ and $y \equiv y'$ then the equivalence should send refl to refl.

In HoTT, this is the statement that ordered pairing is an embedding $V \times V \hookrightarrow V$. This neatly encapsulates and generalises the usual characterisation of equality of ordered pairs in a way that is completely independent of the level of the ∈-structure. But it does not uniquely define the encoding of ordered pairs. Indeed, there are several ways to encode ordered pairs in ordinary set theory. The usual Kuratowsky pairing will work for 0-level ∈-structures, but not for higher level structures. Luckily, Norbert Wiener's encoding, $\langle x, y\rangle := \{\{\{x\}, \emptyset\}, \{\{y\}\}\}$, will work for structures of any level.

This was originally observed in previous work by one of the authors (Gylterud, 2019). However, there the equivalence $(\langle x, y\rangle = \langle x', y'\rangle) \simeq ((x, y) =_{V\times V} (x', y'))$ was not required to be the canonical one, as we require here. Moreover, the encoding of ordered pairs used the untruncated variants of singletons and unordered pairs. But it may be the case that we can only construct truncated variants in a given ∈-structure. Therefore, we will here construct ordered pairs, based on the Wiener encoding, but for any truncation level of singletons and unordered pairs.

Since the results which follow are independent of encoding, we will not commit to any specific way of forming ordered pairs, but simply assume ordered pairing as an extra structure.

Definition 5 ()**.** Given an ∈-structure, $(V, \in)$, **an ordered pairing structure** on $(V, \in)$ is an embedding $V \times V \hookrightarrow V$.

While having ordered pairing is a structure, once the pairing structure is fixed, the notion of *being an ordered pair* is a proposition. As is clear from the proof below, this fact is just another formulation of the characterisation of equality of ordered pairs.

Proposition 3. *Being an ordered pair is a mere proposition: for a fixed ∈-structure* $(V, \in)$ *with ordered pairing structure* $\langle -, -\rangle$*, the type* $\sum_{a,b:V} \langle a, b\rangle = x$ *is a proposition, for all* $x : V$*.*

Proof. Observe that $\sum_{a,b:V} \langle a, b\rangle = x$ is the fiber of $\langle -, -\rangle$ over x, which is a proposition since $\langle -, -\rangle$ is an embedding. □

2.2 Properties of ∈-structures

In this section we explore how further set-theoretic notions, such as pairing, union, replacement, separation and exponentiation, can be expressed as mere propositions about ∈-structures. As mentioned in the introduction, these notions can be generalised in different ways by using different levels of truncation. When characterising unions, for instance, it makes a difference whether the truncated existential quantifier or dependent pair types are used. With the truncated existential quantifier we only get one copy of each element in the union, but with the dependent pair type we may get more copies.

A given ∈-structure can satisfy several versions, but a recurring theme is that n-level structures will only satisfy the k-truncated versions for $k \leq n$.

Convention: *In the rest of the paper we will consider the type of truncation levels to be the type* $\mathbb{N}^{\infty}_{-2}$*, i.e. the usual truncation levels extended by an element* ∞ *for which* $\|P\|_\infty$ *is defined as* P *and such that* $\infty - 1 = \infty = \infty + 1$*.*

The properties of ∈-structures will be parameterised by truncation level. If the truncation level is omitted, in the notation or reference to a property, we mean the variant that is labeled with 0 *(which usually involves* (-1)*-truncation in the definition).*

Interestingly, even the untruncated versions of the set theoretic properties end up being propositions. For instance, to have all ∞-unions is a mere property of ∈-structures. This is because the properties characterise the material sets they claim existence of up to equality, by extensionality.

Proposition 4 ( )**.** *Given an ∈-structure,* $(V, \in)$*, let* $\phi : V \to \mathrm{Type}$ *be a type family on* V*. Then the type* $\sum_{x:V} \prod_{z:V} z \in x \simeq \phi\, z$ *is a proposition.*

Proof. Assume $(x, \alpha) : \sum_{x:V} \prod_{z:V} z \in x \simeq \phi\, z$, then it is enough to show that the type is contractible. We have the following chain of equivalences:

$$\left(\sum_{x':V} \prod_{z:V} z \in x' \simeq \phi\, z\right) \simeq \left(\sum_{x':V} \prod_{z:V} z \in x' \simeq z \in x\right) \simeq \left(\sum_{x':V} x' = x\right)$$

The last type is contractible. □

Remark: Proposition 4 states that the generalisation of the unrestricted comprehension, $\{z | \phi\, z\}$, determines a set uniquely, when existent. Many existence statements in set theory can be seen as fleshing out for which forms ϕ this comprehension defines a set.

Unordered tuples

The usual notion of pairing naturally extends to unordered tupling of any arity. The arity can be expressed by any type. The usual pairing operation is tupling with respect to the booleans, the singleton operation is tupling for the unit type, and the empty set is tupling for the empty type. We generalise the property of having tupling to all truncation levels.

Definition 6 ( ). Given $k : \mathbb{N}_{-1}^{\infty}$ and a type I, an ∈-structure, $(V, \in)$, has $(k+1)$**-unordered I-tupling** if for every $v : I \to V$ there is $\{v\}_{k+1} : V$ such that $\prod_{z:V} z \in \{v\}_{k+1} \simeq \| \sum_{i:I} v\, i = z \|_k$

For the special cases $I = \mathrm{Fin}\ n$, we say that $(V, \in)$ has k-unordered n-tupling. If $(V, \in)$ has k-unordered n-tupling for every $n : \mathbb{N}$, we say that $(V, \in)$ has finite, k-unordered tupling.

We will use the usual notation for finite tuplings, but with a subscript for the truncation level: k-unordered n-tupling is denoted by $\{x_0, \cdots, x_{n-1}\}_k$. For k-unordered **0**-tupling we will use the notation $\emptyset$. Observe that $\{x_0, \cdots, x_{n-1}\}_k$ is a k-type in $(V, \in)$.

The set $\emptyset$ is the set with no elements. For any $z : V$ we have:

$$z \in \emptyset \simeq \left\| \sum_{i:\mathbf{0}} \text{ex-falso } i = z \right\|_{k-1} \simeq \| \ \mathbf{0} \ \|_{k-1} \simeq \mathbf{0} \tag{1}$$

Note here that the truncation level k does not matter (hence why we exclude it from the notation $\emptyset$). By extensionality, the sets corresponding to k-unordered and k'-unordered **0**-tupling, for any two k and k', are equal. Note also that $\emptyset$ is a mere set in any ∈-structure.

Remark: We say that an ∈-structure, $(V, \in)$, has **empty set** if there is an element $x : V$ such that $\prod_{z:V} z \in x \simeq \mathbf{0}$. By the previous paragraph, this is equivalent to saying that $(V, \in)$ has **0**-tupling.

However, the singletons $\{x\}_k$ may be different for different truncation levels. For any $x, z : V$ we have:

$$z \in \{x\}_k \simeq \left\| \sum_{i:\mathbf{1}} x = z \right\|_{k-1} \simeq \| \ x = z \ \|_{k-1} \tag{2}$$

In the special case when x is a mere set in $(V, \in)$, the truncation level does not matter in that $\{x\}_k = \{x\}_0$, by extensionality. As an example: $\{\emptyset\}_k = \{\emptyset\}_{k'}$ for any two k and k', since $\emptyset$ is a mere set. Starting at level 1, repetitions matter in tupling: Given $k > 0$ we have $\{\emptyset, \emptyset\}_k \neq \{\emptyset\}_k$, since $\emptyset \in \{\emptyset, \emptyset\}_k \simeq \|\emptyset = \emptyset + \emptyset = \emptyset\|_k \simeq 2$ while $\emptyset \in \{\emptyset\}_k \simeq \|\emptyset = \emptyset\|_k \simeq 1$.

An interesting usage of 0-unordered 1-tupling combined with 1-unordered n-tupling is the construction of a set s_n such that $\mathrm{El}\ s_n = B(S_n)$, the classifying type of the symmetric group on n elements. Simply let $s_n = \{\{n\emptyset\}_1\}_0$, where $n\emptyset : V^n$ is the vector with n copies of $\emptyset$.

Ordered pairs from unordered tuples

As noted above, ordered pairs can be constructed from the empty set, singletons and unordered pairs, using Norbert Wiener's encoding: $\langle x, y\rangle := \{\{\{x\}, \emptyset\}, \{\{y\}\}\}$. In $\in$-structures of arbitrary level it makes sense to ask what level of truncation $\{-\}_k$ and $\{-,-\}_{k'}$ should be used when defining ordered pairs. The construction using the (-1)-truncated variants does not work in higher level structures since $\{\{\{x\}_0, \emptyset\}_0, \{\{y\}_0\}_0\}_0 = \{\{\{x'\}_0, \emptyset\}_0, \{\{y'\}_0\}_0\}_0$ is a proposition, while $(x, y) = (x', y')$ need not be.

If $(V, \in)$ has level n, then we know that $(x, y) = (x', y')$ has type level $n-1$. For that case the $(n-1)$-truncated variants would give us the correct type level for $\langle x, y\rangle = \langle x', y'\rangle$. But, we observe that $\{-\}_n = \{-\}_\infty$ in $\in$-structures of level n since we have $z \in \{x\}_n \simeq \|z = x\|_{n-1} \simeq (z = x)$. For unordered pairs we need to distinguish between the (-1)-truncated case and all other cases since coproducts are not closed under propositions. More specifically, for $n \geq 1$ we have $\{-,-\}_n = \{-,-\}_\infty$ if $(V, \in)$ has level n, since $z \in \{x, y\}_n \simeq \|(z = x) + (z = y)\|_{n-1} \simeq ((z = x) + (z = y))$. This equivalence does not hold for $n = 0$ and arbitrary $x, y : V$. However, if $x \neq y$, then $z = x \to z \neq y$ and the equivalence holds. We use these observations to make a general construction of ordered pairs which we can instantiate for $\in$-structures of all levels.

Lemma 1 ( ). *Given an $\in$-structure, $(V, \in)$, with an operation $\alpha : V \times V \to V$ with equivalences $e : \prod_{x,y:V} x \neq y \to \prod_{z:V} z \in \alpha(x, y) \simeq ((z = x) + (z = y))$, and two disjoint embeddings, $f\ g : V \hookrightarrow V$, i.e. $\prod_{x,y:V} f\,x \neq g\,y$, then $\alpha \circ (f \times g) : V \times V \to V$ is an embedding.*

Proof. We need to show, for any $(x, y), (x', y') : V \times V$, that $\mathrm{ap}_{\alpha\circ(f\times g)} : (x, y) = (x', y') \to \alpha(f\,x, g\,y) = \alpha(f\,x', g\,y')$ is an equivalence. To this end, it is enough to construct some equivalence between the identity types that sends $\mathrm{refl} : \alpha(f\,x, g\,y) = \alpha(f\,x, g\,y)$ to $\mathrm{refl} : (x, y) = (x, y)$.

We have the following chain of equivalences:

$$\begin{aligned}
&(\alpha(f\,x, g\,y) = \alpha(f\,x', g\,y')) && (3)\\
&\quad\simeq \prod_{z:V} z \in \alpha(f\,x, g\,y) \simeq z \in \alpha(f\,x', g\,y') && (4)\\
&\quad\simeq \prod_{z:V} ((z = f\,x) + (z = g\,y)) \simeq ((z = f\,x') + (z = g\,y')) && (5)\\
&\quad\simeq \prod_{z:V} ((z = f\,x) \simeq (z = f\,x')) \times ((z = g\,y) \simeq (z = g\,y')) && (6)
\end{aligned}$$

$$
\begin{aligned}
&\simeq \left(\prod_{z:V}(z = f\,x) \simeq (z = f\,x')\right) \times \left(\prod_{z:V}(z = g\,y) \simeq (z = g\,y')\right) && (7)\\
&\simeq \big(f\,x = f\,x'\big) \times \big(g\,y = g\,y'\big) && (8)\\
&\simeq \big(x = x'\big) \times \big(y = y'\big) && (9)\\
&\simeq ((x, y) = (x', y')) && (10)
\end{aligned}
$$

In step (4) we use extensionality for $(V, \in)$. In step (5) we use the equivalences $e\,(f\,x)\,(g\,y)$ and $e\,(f\,x')\,(g\,y')$, together with the fact that $f\,x \neq g\,y$ and $f\,x' \neq g\,y'$. The equivalence (6) follows from the fact that $z = f\,x$ and $z = g\,y'$, and $z = f\,x'$ and $z = g\,y$, are, respectively, mutually exclusive. In step (9) we use the fact that f and g are embeddings.

We chase $\mathrm{refl} : \alpha(f\,x, g\,y) = \alpha(f\,x, g\,y)$ through the equivalence:

$$
\begin{aligned}
\mathrm{refl} &\mapsto \lambda\,z.\,\text{id-equiv} && (11)\\
&\mapsto \lambda\,z.\,(e\,z) \circ \text{id-equiv} \circ (e\,z)^{-1} && (12)\\
&= \lambda\,z.\,\text{id-equiv} && (13)\\
&\mapsto \lambda\,z.(\text{id-equiv}, \text{id-equiv}) && (14)\\
&\mapsto (\lambda\,z.\,\text{id-equiv}, \lambda\,z.\,\text{id-equiv}) && (15)\\
&\mapsto (\mathrm{refl}, \mathrm{refl}) && (16)\\
&= \big(\mathrm{ap}_f\ \mathrm{refl}, \mathrm{ap}_g\ \mathrm{refl}\big) && (17)\\
&\mapsto \Big(\mathrm{ap}_f^{-1}\,\big(\mathrm{ap}_f\ \mathrm{refl}\big), \mathrm{ap}_g^{-1}\,\big(\mathrm{ap}_g\ \mathrm{refl}\big)\Big) && (18)\\
&= (\mathrm{refl}, \mathrm{refl}) && (19)\\
&\mapsto \mathrm{refl} && (20)
\end{aligned}
$$

where we have used the fact that extensionality for $(V, \in)$ sends refl to id-equiv. □

For the Norbert Wiener construction of ordered pairs we thus have to show that both $\{\{-\}_n\}_n$ and $\{\{-\}_n, \emptyset\}_n$ are embeddings. Using the previous observation about the relationship between the n-truncated versions and the ∞-truncated ones, it is enough to show this for the ∞-truncated versions.

Lemma 2 ( ). *The function $\{-\}_\infty : V \to V$ is an embedding.*

Proof. We follow the same strategy as in the proof of Lemma 1. For any $x, y : V$ we have the following chain of equivalences:

$$(\{x\}_\infty = \{y\}_\infty) \simeq \prod_{z:V} z \in \{x\}_\infty \simeq z \in \{y\}_\infty \tag{21}$$
$$\simeq \prod_{z:V} (z = x) \simeq (z = y) \tag{22}$$
$$\simeq (x = y) \tag{23}$$

In step (21) we use extensionality for $(V, \in)$.

Let $e : \prod_{x:V} \prod_{z:V} z \in \{x\}_\infty \simeq (z = x)$ be the defining family of equivalences for $\{-\}_\infty$. We chase refl : $\{x\}_\infty = \{x\}_\infty$ along the chain of equivalences above:

$$\text{refl} \mapsto \lambda\, z.\, \text{id-equiv} \tag{24}$$
$$\mapsto \lambda\, z.(e\, x\, z) \circ \text{id-equiv} \circ (e\, x\, z)^{-1} \tag{25}$$
$$= \lambda\, z.\, \text{id-equiv} \tag{26}$$
$$\mapsto \text{refl} \tag{27}$$

In step (24) we use the fact that extensionality for $(V, \in)$ sends refl to id-equiv. □

Lemma 3 ( ). *Let $\alpha : V \to V \to V$ be such that $e : \prod_{x,y:V} x \neq y \to \prod_{z:V} z \in \alpha\, x\, y \simeq ((z = x) + (z = y))$. Then $\lambda\, x.\, \alpha\, \{x\}_\infty\, \emptyset : V \to V$ is an embedding.*

Proof. First, we observe that for all $x : V$, $\{x\}_\infty \neq \emptyset$ since $x \in \{x\}_\infty$ is inhabited but $x \in \emptyset$ is empty. We now follow the same strategy as in the proof of Lemma 1. For any $x, y : V$ we have the following chain of equivalences:

$$(\alpha\, \{x\}_\infty\, \emptyset = \alpha\, \{y\}_\infty\, \emptyset) \tag{28}$$
$$\simeq \prod_{z:V} (z \in \alpha\, \{x\}_\infty\, \emptyset) \simeq (z \in \alpha\, \{y\}_\infty\, \emptyset) \tag{29}$$
$$\simeq \prod_{z:V} ((z = \{x\}_\infty) + (z = \emptyset)) \simeq ((z = \{y\}_\infty) + (z = \emptyset)) \tag{30}$$
$$\simeq \prod_{z:V} ((z = \{x\}_\infty) \simeq (z = \{y\}_\infty)) \times ((z = \emptyset) \simeq (z = \emptyset)) \tag{31}$$
$$\simeq \prod_{z:V} (z = \{x\}_\infty) \simeq (z = \{y\}_\infty) \tag{32}$$
$$\simeq (\{x\}_\infty = \{y\}_\infty) \tag{33}$$
$$\simeq (x = y) \tag{34}$$

In step (29) we use extensionality for $(V, \in)$. In step (31) we use the fact that $\{x\}_\infty \neq \emptyset$ and $\{y\}_\infty \neq \emptyset$. In step (32) we use the fact that $(z = \emptyset) \simeq (z = \emptyset)$ is contractible since $\emptyset$ is a mere set, and hence $z = \emptyset$ is a proposition. In step (34) we use Lemma 2.

We chase $\mathrm{refl} : \alpha\,\{x\}_\infty\,\emptyset = \alpha\,\{x\}_\infty\,\emptyset$ along the chain of equivalences above:

$$\begin{aligned}
\mathrm{refl} &\mapsto \lambda\, z.\, \text{id-equiv} && (35)\\
&\mapsto \lambda\, z.(e\, z) \circ \text{id-equiv} \circ (e\, z)^{-1} && (36)\\
&= \lambda\, z.\, \text{id-equiv} && (37)\\
&\mapsto \lambda\, z.(\text{id-equiv}, \text{id-equiv}) && (38)\\
&\mapsto \lambda\, z.\, \text{id-equiv} && (39)\\
&\mapsto \mathrm{refl} && (40)\\
&= \mathrm{ap}_{\{-\}_\infty}\ \mathrm{refl} && (41)\\
&\mapsto \mathrm{ap}^{-1}_{\{-\}_\infty} \left(\mathrm{ap}_{\{-\}_\infty}\ \mathrm{refl}\right) && (42)\\
&= \mathrm{refl} && (43)
\end{aligned}$$

In step (35) we use the fact that extensionality for $(V, \in)$ sends refl to id-equiv. □

Theorem 1 ( ). *If $(V, \in)$ is an $\in$-structure of level n which has $\emptyset$, $\{-\}_n$ and $\{-,-\}_n$, then it has an ordered pairing structure given by $\lambda\,(x, y).\{\{\{x\}_n, \emptyset\}_n, \{\{y\}_n\}_n\}_n$.*

Proof. Corollary of Lemma 1, Lemma 2 and Lemma 3. □

Restricted separation

In constructive set theory (Aczel, 1978), restricted separation is the ability to construct sets of the form $\{z \in x \mid \Phi\, z\}$, for a formula Φ where all quantifiers are bounded (i.e. $\forall a \in b \cdots$ and $\exists a \in b \cdots$). One way to internalise this to type theory is to require that the predicate $\Phi : V \to \mathrm{Prop}_U$ is a predicate of propositions in U. We do the same here in that we require the predicate to take values in U, but as with the other properties, we generalise to higher truncation levels.

Definition 7 ( ). An $\in$-structure, $(V, \in)$, has **U-restricted** $(k+1)$**-separation**, for $k : \mathbb{N}^{\infty}_{-1}$, if for every $x : V$ and $P : \mathrm{El}\, x \to k\text{-Type}_U$ there is an element $\{x \mid P\} : V$ such that $\prod_{z:V} z \in \{x \mid P\} \simeq \sum_{e : z \in x} P\,(z, e)$.

Remark: Usually, in set theory, the predicate is defined for any x, even when taking the restricted separation. But this falls under the above: Assume

$Q : V \to k\text{-Type}_U$, then one can readily define $P(z,e) = Q\,z$ and the defining property becomes $z \in \{x \mid P\} \simeq (z \in x \times Q\,z)$. However, going in the opposite direction only works at level 0, because the predicate P may otherwise depend on the specific witness of elementhood.

Remark: At level 1, we can use 0-unordered 1-tupling, 1-unordered tupling and U-restricted 1-separation, to construct for every group G a set s_G such that El $s_G = B(G)$, the classifying type for the group.[‡] First let $|G|\emptyset : |G| \to V$ be the constant function mapping every element of the group to $\emptyset$. This is the tuple to which we will apply 1-unordered tupling, constructing $\{|G|\emptyset\}_1$. This multiset has one copy of $\emptyset$ for every element of the group, hence its automorphisms in V are the bijections on $|G|$:

$$
\begin{aligned}
(\{|G|\emptyset\}_1 = \{|G|\emptyset\}_1) &\simeq \prod_{z:V} (z \in \{|G|\emptyset\}_1) \simeq (z \in \{|G|\emptyset\}_1) && (44)\\
&\simeq \prod_{z:V} (|G| \times (z = \emptyset)) \simeq (|G| \times (z = \emptyset)) && (45)\\
&\simeq (|G| =_U |G|) && (46)
\end{aligned}
$$

If we take the 0-singleton $\{\{|G|\emptyset\}_1\}_0$, we get a set which has $B(\mathrm{Aut}_U\ |G|)$ as its type of elements:

$$
\begin{aligned}
\mathrm{El}\,\{\{|G|\emptyset\}_1\}_0 &= \sum_{z:V} z \in \{\{|G|\emptyset\}_1\}_0 && (47)\\
&\simeq \sum_{z:V} \|z = \{|G|\emptyset\}_1\|_{-1} && (48)\\
&= B(\mathrm{Aut}_V\,\{|G|\emptyset\}_1) && (49)\\
&\simeq B(\mathrm{Aut}_U\ |G|) && (50)
\end{aligned}
$$

The final ingredient is that we need a map $f : B(G) \to B(\mathrm{Aut}\ |G|)$. This map is the well-known map induced by multiplication in the group. This is a cover, i.e. its fibers are sets, hence fiber $f : B(\mathrm{Aut}\ |G|) \to 0\text{-Type}$, which we will coerce along the equivalence $B(\mathrm{Aut}_U\ |G|) \simeq \mathrm{El}\,\{\{|G|\emptyset\}_1\}_0$ to obtain $P_G : \mathrm{El}\,\{\{|G|\emptyset\}_1\}_0 \to 0\text{-Type}$. Thus, we define $s_G := \{\{\{|G|\emptyset\}_1\}_0 | P_G\}$, which then has the desired property:

[‡]For a thorough treatment of groups and classifying types, see the *Symmetry* book (Bezem et al., 2022)

$$
\begin{aligned}
\mathrm{El}\ s_G &= \sum_{z:V} z \in \{\{\{|G|\emptyset\}_1\}_0|P_G\} && (51)\\
&\simeq \sum_{z:V} \sum_{e:z\in\{\{|G|\emptyset\}_1\}_0} P_G(z,e) && (52)\\
&\simeq \sum_{q:\mathrm{El}\,\{\{|G|\emptyset\}_1\}_0} P_G\, q && (53)\\
&\simeq \sum_{q:B(\mathrm{Aut}\ |G|)} \text{fiber } f\, q && (54)\\
&\simeq B(G) && (55)
\end{aligned}
$$

Replacement

Replacement says that given a set a, and a description of how to replace its elements, we can create a new set containing exactly the replacements of elements in a. Of course, two elements in a may be replaced by the same elements, so the property must include truncation.

Definition 8 ( ). An ∈-structure, $(V, \in)$, has $(k+1)$**-replacement**, for $k : \mathbb{N}_{-1}^{\infty}$, if for every $a : V$ and $f : \mathrm{El}\ a \to V$ there is an element $\{f(x)\,|\,x \in a\} : V$ such that $\prod_{z:V} z \in \{f(x)\,|\,x \in a\} \simeq \left\| \sum_{x:\mathrm{El}\ a} f\ x = z \right\|_k$

Remark: Having k-replacement is to have k-unordered I-tupling for $I = \mathrm{El}\, a$, for any $a : V$.

In Section 3 we further explore the implications of k-replacement and how it affects representations of types within ∈-structures.

Union

Just as for tupling, there is a notion of union to consider for each type level, for ∈-structures, distinguished by the truncation of the existential quantifier.

Definition 9 ( ). An ∈-structure, $(V, \in)$, has $(k+1)$**-union**, for $k : \mathbb{N}_{-1}^{\infty}$, if for every $x : V$ there is $\bigcup_k x : V$ such that $\prod_{z:V} z \in \bigcup_k x \simeq \left\| \sum_{y:V}\ z \in y\ \times\ y \in x \right\|_k$.

As usual, from pairing and general unions, one can define binary k-union by $x \cup_k y :\equiv \bigcup_k \{x, y\}_k$. For $z : V$ we have:

$$z \in x \cup_k y \simeq \left\| \sum_{w:V} z \in w \ \times \ w \in \{x, y\}_k \right\|_{k-1} \tag{56}$$

$$\simeq \left\| \sum_{w:V} z \in w \ \times \ \|w = x + w = y\|_{k-1} \right\|_{k-1} \tag{57}$$

$$\simeq \left\| \sum_{w:V} z \in w \ \times \ (w = x + w = y) \right\|_{k-1} \tag{58}$$

$$\simeq \left\| z \in x + z \in y \right\|_{k-1} \tag{59}$$

Binary 0-union is the usual binary union in set theory, where copies are discarded. The higher level binary union keeps copies. For example, $\{\emptyset\}_1 \cup_0 \{\emptyset\}_1 = \{\emptyset\}_0$, while $\{\emptyset\}_1 \cup_k \{\emptyset\}_1 = \{\emptyset, \emptyset\}_k$, for $k \geq 1$.

Exponentiation

Exponentiation states that for any two sets a and b there is a set that contains exactly the functions from a to b. In order to express this property we need to first generalise the notion of a function internal to an $\in$-structure. As opposed to the other properties, the notion of a function can be expressed in a uniform way for $\in$-structures of any level. As in usual set theory, the notion of a function is relative to a choice of ordered pairing structure.

But what constitutes a function between two generalised sets, say $a, b : V$? The perhaps easiest answer, which is the one we will argue for here, is that it is in essence a function $\mathrm{El}\ a \to \mathrm{El}\ b$, which can then be represented as set itself by taking its graph. At level 0, in usual set theory, a function is completely determined by its graph. The general situation is, however, that if a and b are of level $n + 1$ there is an n-type of function structures which can be put on a set of pairs. We call this structure $\mathrm{operation}_{a\,b}$, and Proposition 8 constructs an equivalence between $\mathrm{El}\ a \to \mathrm{El}\ b$ and sets with $\mathrm{operation}_{a\,b}$-structure.

Remark: As usual in constructive set theory, we use the exponentiation axiom instead of the powerset axiom. One could define the powerset axiom for higher level structures by saying that $z : V$ is a subset of $x : V$ if for all $y : V$ there is an embedding $y \in z \hookrightarrow y \in x$. However, in any fixed-point model, the powerset axiom would require a small subobject classifier in the ambient type theory, hence why we do not consider it further in this paper.

Definition 10 ( ). Given three elements $a, b, f : V$ in an $\in$-structure

$(V, \in)$ with ordered pairing $\langle -, - \rangle : V \times V \hookrightarrow V$, define a type:

$$\begin{aligned} \text{operation}_{a\,b}\ f := &\left(\prod_{x:V} x \in a \simeq \sum_{y:V} \langle x, y \rangle \in f \right) \\ &\times \left(\prod_{x:V} \prod_{y:V} \langle x, y \rangle \in f \to y \in b \right) \\ &\times \left(\prod_{z:V} z \in f \to \sum_{x:V} \sum_{y:V} z = \langle x, y \rangle \right) \end{aligned}$$

The first conjunct of $\text{operation}_{a\,b}\ f$ states that the pairs in f form an operation with domain a, and the second states that its codomain is b. The third conjunct, which is always a proposition by Proposition 3, states that f only contains pairs.

Note that operation here is the same as Definition 8 in Gylterud (2019). However, in this paper we define another type which we show is equivalent to operation and which is of a more type theoretic flavor.

The type $\text{operation}_{a\,b}\ f$ is not always a proposition. Hence, an element $p : \text{operation}_{a\,b}\ f$ should be regarded as an operation structure on f with domain a and codomain b. However, in the case when a and b are mere sets, $\text{operation}_{a\,b}\ f$ is a proposition.

Proposition 5 ( ). *Given three elements $a, b, f : V$ in an ∈-structure $(V, \in)$ with ordered pairing $\langle -, - \rangle : V \times V \hookrightarrow V$, if a and b are mere sets in $(V, \in)$, then* $\text{operation}_{a\,b}\ f$ *is a proposition equivalent to the following type:*

$$\begin{aligned} &\left(\prod_{x:V} x \in a \to \underset{y:V}{\exists!}\, \langle x, y \rangle \in f \right) \\ \times &\left(\prod_{x:V} \prod_{y:V} \langle x, y \rangle \in f \ \to\ x \in a \ \times\ y \in b \right) \\ \times &\left(\prod_{z:V} z \in f \to \sum_{x:V} \sum_{y:V} z = \langle x, y \rangle \right) \end{aligned}$$

Proof. We start by proving the equivalence. First, we observe that for any $x : V$, $p : x \in a$, and function $e : \sum_{y:V} \langle x, y \rangle \in f \to x \in a$ we have an equivalence

$$\text{fiber}\ e\ p \ \simeq \sum_{y:V} \langle x, y \rangle \in f$$

since for any $q : \sum_{y:V} \langle x, y\rangle \in f$ the type $e\,q = p$ is contractible because a is a mere set. It thus follows that for any $x : V$ we have

$$\begin{aligned}\left(x \in a \simeq \sum_{y:V} \langle x, y\rangle \in f\right) \\ &\simeq \sum_{e:\sum_{y:V}\langle x,y\rangle \in f \to x \in a} \prod_{p:x\in a} \text{is-contr}\,(\text{fiber } e\,p) \\ &\simeq \left(\sum_{y:V} \langle x, y\rangle \in f \to x \in a\right) \times \left(x \in a \to \underset{y:V}{\exists!} \langle x, y\rangle \in f\right)\end{aligned}$$

The desired equivalence follows from this, after some currying and rearranging.

Since $\exists!_{y:V} \langle x, y\rangle \in f$ is a proposition it follows that

$$\left(\prod_{x:V} x \in a \to \exists!_{y:V} \langle x, y\rangle \in f\right)$$

is a proposition. When a and b are mere sets, $x \in a \times y \in b$ is a proposition, and hence

$$\left(\prod_{x:V}\prod_{y:V} \langle x, y\rangle \in f \;\to\; (x \in a) \;\times\; (y \in b)\right)$$

is a proposition. Finally, since ordered pairing is an embedding, and $\sum_{x:V} \sum_{y:V} z = \langle x, y\rangle$ is essentially the fibres of the pairing operation, it follows that

$$\prod_{z:V} z \in f \to \sum_{x:V} \sum_{y:V} z = \langle x, y\rangle$$

is a proposition. □

This equivalence was proven as Lemma 6.10 in Gylterud (2018) for the type V constructed there. Here we show the equivalence for general $\in$-structures.

Corollary 1. *If* $(V, \in)$ *is of level 0 then* $\text{operation}_{a\,b}\ f$ *is a proposition, for all* $a, b, f : V$.

We can now define exponentiation for $\in$-structures.

Definition 11 ( ). An $\in$-structure, $(V, \in)$ with an ordered pairing structure, has **exponentiation** if for every two $a, b : V$ there is an element b^a such that $\prod_{f:V}(f \in b^a) \simeq \text{operation}_{a\,b}\ f$.

While operation is a straightforward internalisation of the set theoretic definition of a function, it can be inconvenient to work with. What follows is a definition that is equivalent to operation but which is sometimes easier to use.

Definition 12 ( ). Given three elements $a, b, f : V$ in an ∈-structure $(V, \in)$ with ordered pairing $\langle -, - \rangle : V \times V \hookrightarrow V$, define a type:

$$\text{operation'}_{a\,b}\ f := \sum_{\phi:\text{El }a \to \text{El }b} \prod_{z:V} \left(z \in f \simeq \sum_{x:\text{El }a} \langle \pi_0\ x, \pi_0\,(\phi\ x) \rangle = z \right)$$

Proposition 6 ( ). *For any $a, b, f : V$ we have the following equivalence:*

$$\text{operation}_{a\,b}\ f \simeq \text{operation'}_{a\,b}\ f$$

Before we prove this equivalence we prove two general lemmas that we will need, concerning fiberwise functions and equivalences.

Lemma 4 ( ). *Let A and B be types and let $C : B \to \text{Type}$ be a type family over B. For any embedding $f : A \hookrightarrow B$ and element $\gamma : \prod_{(b,c):\sum_{b:B} C\,b} \text{fiber}\ f\ b$ we have an equivalence*

$$\sum_{a:A} C\,(f\ a) \simeq \sum_{b:B} C\,b.$$

Proof. Let $F : \sum_{a:A} C\,(f\ a) \to \sum_{b:B} C\,b$ be the function given by $F\,(a, c) := (f\,a, c)$. For $(b, c) : \sum_{b:B} C\,b$, we have the following chain of equivalences:

$$\begin{aligned} \text{fiber}\ F\,(b, c) &\simeq \sum_{a:A} \sum_{c':C\,(f\,a)} \sum_{p:f\,a=b} \text{tr}_p^C\ c' = c & (60) \\ &\simeq \sum_{a:A} \sum_{p:f\,a=b} \sum_{c':C\,(f\,a)} c' = \text{tr}_{p^{-1}}^C\ c & (61) \\ &\simeq \text{fiber}\ f\,b & (62) \end{aligned}$$

Step (62) uses the fact that $\sum_{c':C\,(f\,a)} c' = \text{tr}_{p^{-1}}^C\ c$ is contractible.

Thus, fiber $F\ (b, c)$ is a proposition, since f is an embedding, which is inhabited by γ, and therefore contractible. □

There is an equivalence between fiberwise equivalences and equivalences of the total spaces that respect the first coordinate. This will be useful several times because it gives two equivalent characterisations of equality in slices over a type.

Lemma 5 ( ). *For any type A, and any families $P, Q : A \to \text{Type}$ we have that*

$$\left(\prod_{x:A} P\,x \to Q\,x\right) \simeq \left(\sum_{\alpha : \sum_{x:A} P\,x \to \sum_{x:A} Q\,x} \pi_0 \circ \alpha = \pi_0\right)$$

Proof. We have the following chain of equivalences:

$$\begin{aligned}
\left(\prod_{x:A} P\,x \to Q\,x\right) &\simeq \prod_{(x,_):\sum_{x:A} P\,x} Q\,x && (63)\\
&\simeq \prod_{(x,_):\sum_{x:A} P\,x} \; \sum_{(x',_):\sum_{x':A} x'=x} Q\,x' && (64)\\
&\simeq \prod_{(x,_):\sum_{x:A} P\,x} \; \sum_{(x',_):\sum_{x':A} Q\,x'} x' = x && (65)\\
&\simeq \sum_{\alpha : \sum_{x:A} P\,x \to \sum_{x:A} Q\,x} \pi_0 \circ \alpha = \pi_0 && (66)
\end{aligned}$$

where (63) is currying, (64) follows from the fact that $\sum_{x':A} x' = x$ is contractible and (66) is the interchange law between Σ-types and Π-types, together with function extensionality. The equivalence sends f to $(\text{tot}\ f, \text{refl})$. □

Remark: Lemma 5 has been independently added by Egbert Rijke to the agda-unimath library (Rijke et al., 2024). The proof there is slightly different, with a direct construction of the equivalence, instead of equivalence reasoning.

Corollary 2 ( ). *For any type A, and any families $P, Q : A \to \text{Type}$ we have an equivalence*

$$\left(\prod_{x:A} P\,x \simeq Q\,x\right) \simeq \left(\sum_{\alpha : \sum_{x:A} P\,x \simeq \sum_{x:A} Q\,x} \pi_0 \circ \alpha = \pi_0\right)$$

Proof. This follows from Lemma 5 by an application of Theorem 4.7.7 in the HoTT Book (The Univalent Foundations Program, 2013, p. 185), which states that a fiberwise transformation is a fiberwise equivalence if and only if the corresponding total function is an equivalence. □

Proof of Proposition 6. We outline the key steps of the equivalence. For full details see the Agda formalisation. We have the following chain of equivalences:

$$
\begin{aligned}
\text{operation}_{a\,b}\ f \simeq{}& \left(\prod_{z:\mathrm{El}\ f} \text{fiber}\ \langle -,-\rangle\ (\pi_0\ z)\right) && (67)\\
&\times \left(\sum_{\sigma:\mathrm{El}\ a \simeq \sum_{(x,y):V\times V}\langle x,y\rangle \in f} \pi_0 \sim \pi_0 \circ \pi_0 \circ \sigma\right)\\
&\times \left(\sum_{\phi:\sum_{(x,y):V\times V}\langle x,y\rangle \in f \to \mathrm{El}\ b} \pi_0 \circ \phi \sim \pi_1 \circ \pi_0\right)\\
\simeq{}& \sum_{\gamma:\prod_{z:\mathrm{El}\ f}\text{fiber}\ \langle -,-\rangle\ (\pi_0\ z)} && (68)\\
&\quad \left(\sum_{\sigma:\mathrm{El}\ a\simeq \mathrm{El}\ f} \pi_0 \circ \sigma^{-1} \sim \pi_0 \circ \pi_0 \circ \gamma\right)\\
&\quad \times \left(\sum_{\phi:\mathrm{El}\ f \to \mathrm{El}\ b} \pi_0 \circ \phi \sim \pi_1 \circ \pi_0 \circ \gamma\right)\\
\simeq{}& \sum_{\sigma:\mathrm{El}\ a\simeq\mathrm{El}\ f}\ \sum_{\phi:\mathrm{El}\ f\to\mathrm{El}\ b}\ \prod_{z:\mathrm{El}\ f}\ \sum_{(x,y):V\times V} && (69)\\
&\quad (\langle x,y\rangle = \pi_0\, z) \times \big(\pi_0\,(\sigma^{-1}\ z) = x\big) \times (\pi_0\,(\phi\ z) = y)\\
\simeq{}& \sum_{\sigma:\mathrm{El}\ a\simeq\mathrm{El}\ f}\ \sum_{\phi:\mathrm{El}\ f\to\mathrm{El}\ b}\ \prod_{z:\mathrm{El}\ f} \langle \pi_0\,(\sigma^{-1}\ z), \pi_0\,(\phi\ z)\rangle = \pi_0\ z && (70)\\
\simeq{}& \sum_{\phi:\mathrm{El}\ a\to\mathrm{El}\ b}\ \sum_{\sigma:\mathrm{El}\ a\simeq\mathrm{El}\ f}\ \prod_{x:\mathrm{El}\ a} \langle \pi_0\ x, \pi_0\,(\phi\ x)\rangle = \pi_0\,(\sigma\ x) && (71)\\
\simeq{}& \text{operation'}_{a\,b}\ f && (72)
\end{aligned}
$$

where in (67) we apply Corollary 2 to the first conjunct and Lemma 5 to the second conjunct, and then rearrange. In (68) we use Lemma 4 to construct an equivalence $\sum_{(x,y):V\times V}\langle x,y\rangle \in f \simeq \mathrm{El}\ f$ which we apply in the second and third conjuncts. In (69) we rearrange and then use the interchange law for $\prod$-types and $\sum$-types. In (70) we use extensionality for cartesian products and the fact that $\sum_{(x,y):V\times V}(\pi_0\,(\sigma^{-1}\ z), \pi_0\,(\phi\, z)) = (x,y)$ is contractible. In (71) we swap El a for El f and then rearrange. Finally, in (72) we use Corollary 2 again. □

The notion of operation' captures type theoretic functions in the sense that the type of all operations from $a : V$ to $b : V$ is a subtype of the type $\mathrm{El}\, a \to \mathrm{El}\, b$.

Proposition 7 ( ). *Given $a, b : V$ in an $\in$-structure $(V, \in)$ with ordered pairing $\langle -, - \rangle$, there is a canonical embedding*

$$\left(\sum_{f:V} \text{operation'}_{a\,b}\ f\right) \hookrightarrow (\mathrm{El}\,a \to \mathrm{El}\,b)$$

Proof. By swapping the Σ-types we have the following equivalence

$$\begin{aligned}&\left(\sum_{f:V} \text{operation'}_{a\,b}\ f\right) \\ \simeq\ & \sum_{\phi:\mathrm{El}\,a \to \mathrm{El}\,b}\ \sum_{f:V} \prod_{z:V} \left(z \in f \simeq \sum_{x:\mathrm{El}\ a} \langle \pi_0\ x, \pi_0\,(\phi\ x) \rangle = z\right) \qquad (73)\end{aligned}$$

The type $\sum_{f:V} \prod_{z:V} (z \in f \simeq \sum_{x:\mathrm{El}\ a} \langle \pi_0\ x, \pi_0\,(\phi\ x) \rangle = z)$ is a proposition, by Proposition 4. The embedding is thus the composition of (73) with the first projection. □

An operation from $a : V$ to $b : V$ is thus a function from $\mathrm{El}\,a \to \mathrm{El}\,b$. A natural question to ask is when all functions $\mathrm{El}\,a \to \mathrm{El}\,b$ correspond to some operation.

Proposition 8 ( ). *Given $a, b : V$ in an $\in$-structure $(V, \in)$ of level $n : \mathbb{N}^\infty$, with ordered pairing $\langle -, - \rangle$, the embedding in Proposition 7 is an equivalence if $(V, \in)$ has n-replacement.*

Proof. Suppose $(V, \in)$ has n-replacement. By applying it to the map $\lambda\, x. \langle \pi_0\ x, \pi_0\,(\phi\ x) \rangle : \mathrm{El}\,a \to V$, we get an element of the following type:

$$\sum_{f:V} \prod_{z:V} \left(z \in f \simeq \left\| \sum_{x:\mathrm{El}\ a} \langle \pi_0\ x, \pi_0\,(\phi\ x) \rangle = z \right\|_{n-1}\right)$$

Claim: The type $\sum_{x:\mathrm{El}\ a} \langle \pi_0\ x, \pi_0\,(\phi\ x) \rangle = z$ is $(n-1)$-truncated.

Given that the claim is true we can drop the truncation. Thus the type

$$\sum_{f:V} \prod_{z:V} \left(z \in f \simeq \sum_{x:\mathrm{El}\ a} \langle \pi_0\ x, \pi_0\,(\phi\ x) \rangle = z\right)$$

is contractible, because it is an inhabited proposition, and hence the embedding constructed in Proposition 7 is an equivalence.

It remains to prove the claim. First, we have the following equivalence:

$$\sum_{x:\mathrm{El}\ a} (\langle \pi_0\ x, \pi_0\,(\phi\ x)\rangle = z) \simeq \sum_{(x,y):\mathrm{El}\,a\times\mathrm{El}\,b} (\langle \pi_0\ x, \pi_0\,y\rangle = z) \times (\phi\ x = y) \tag{74}$$

This follows from the fact that $\sum_{y:\mathrm{El}\,b} \phi\ x = y$ is contractible. The type $\phi\ x = y$ is $(n-1)$-truncated since V is an n-type, by Proposition 1. The type $\sum_{(x,y):\mathrm{El}\,a\times\mathrm{El}\,b} (\langle \pi_0\ x, \pi_0\ y\rangle = z)$ is the fiber of the composite map $\langle -, - \rangle \circ (\pi_0 \times \pi_0)$ over $z : V$. The map $\langle -, - \rangle$ is an embedding and thus an $(n-1)$-truncated map. The fibers of the two projection maps over z are $z \in a$ and $z \in b$ respectively. These types are $(n-1)$-truncated. So the composite map is $(n-1)$-truncated, and the claim follows. □

Accessible elements and foundation

We choose the same approach as the HoTT Book (The Univalent Foundations Program, 2013) to well-foundedness, namely accessibility predicates. The axiom of foundation is then the statement that all elements are accessible. This need not be true in a given ∈-structure, but the subtype of accessible elements inherits the ∈-structure from the base type. This new ∈-structure satisfies foundation.

Definition 13 ( ). Given an ∈-structure, $(V, \in)$, define inductively the predicate $\mathrm{Acc} : V \to \mathrm{Type}$ by

- $\mathrm{acc} : \prod_{x:V} (\prod_{y:V} y \in x \to \mathrm{Acc}\, y) \to \mathrm{Acc}\, x$

Lemma 6 ( ). *For every $x : V$ the type $\mathrm{Acc}\, x$ is a mere proposition.*

Proof. Lemma 10.3.2 in the HoTT Book (The Univalent Foundations Program, 2013, p. 454). □

Definition 14 ( ). An ∈-structure, $(V, \in)$, has **foundation** if $\prod_{x:V} \mathrm{Acc}\, x$.

Since accessibility is a proposition, we can define for a given ∈-structure, the subtype of accessible elements.

Definition 15 ( ). Given an ∈-structure, $(V, \in)$, define the type $V_{\mathrm{Acc}} := \sum_{x:V} \mathrm{Acc}\, x$ and define the binary relation $\in_{\mathrm{Acc}} := \lambda\,(x : V_{\mathrm{Acc}})\,(y : V_{\mathrm{Acc}}).\ \pi_0\, x \in \pi_0\, y$.

The subtype of accessible elements inherits the ∈-structure from the base type.

Proposition 9 ( ). *Given an $\in$-structure, $(V, \in)$, the pair $(V_{\mathrm{Acc}}, \in_{\mathrm{Acc}})$ forms an $\in$-structure.*

Proof. We need to prove extensionality for $(V_{\mathrm{Acc}}, \in_{\mathrm{Acc}})$. For $x, y : V_{\mathrm{Acc}}$ we have the following chain of equivalences:

$$
\begin{aligned}
(x = y) &\simeq (\pi_0\, x = \pi_0\, y) && (75)\\
&\simeq \prod_{z:V} z \in \pi_0\, x \simeq z \in \pi_0\, y && (76)\\
&\simeq \sum_{e : \sum_{z:V} z \in \pi_0\, x \,\simeq\, \sum_{z:V} z \in \pi_0\, y} \pi_0 \circ e \sim \pi_0 && (77)\\
&\simeq \sum_{e : \sum_{z:V} (z \in \pi_0\, x) \times (\mathrm{Acc}\, z) \,\simeq\, \sum_{z:V} (z \in \pi_0\, y) \times (\mathrm{Acc}\, z)} \pi_0 \circ e \sim \pi_0 && (78)\\
&\simeq \sum_{e : \sum_{z:V_{\mathrm{Acc}}} \pi_0\, z \in \pi_0\, x \,\simeq\, \sum_{z:V_{\mathrm{Acc}}} \pi_0\, z \in \pi_0\, y} \pi_0 \circ e \sim \pi_0 && (79)\\
&\simeq \prod_{z:V_{\mathrm{Acc}}} z \in_{\mathrm{Acc}} x \simeq z \in_{\mathrm{Acc}} y && (80)
\end{aligned}
$$

In step (75) we use Lemma 6. Step (76) is extensionality for $(V, \in)$ and step (77) is Corollary 2. In step (78) we use the fact that elements of accessible sets are accessible and thus $\mathrm{Acc}\, z$ is contractible. Step (79) is some rearranging of conjuncts in the base type together with the characterisation of equality in subtypes for the fibration. The last step is again an application of Corollary 2.

Chasing refl along this chain of equivalences we see that it is sent to $\lambda\, z.\, \text{id-equiv}$. □

The subtype of accessible elements satisfies foundation.

Theorem 2 ( ). *Given an $\in$-structure, $(V, \in)$, the $\in$-structure $(V_{\mathrm{Acc}}, \in_{\mathrm{Acc}})$ has foundation.*

Proof. There are two different accessibility predicates at play here. Let Acc and acc be accessibility with respect to $(V, \in)$ and let Acc' and acc' be accessibility with respect to $(V_{\mathrm{Acc}}, \in_{\mathrm{Acc}})$. We need to show $\prod_{x:V_{\mathrm{Acc}}} \mathrm{Acc}'\, x$, which we do by using the induction principle of Acc (modulo a transport):

$$
\begin{aligned}
&\alpha : \prod_{x:V_{\mathrm{Acc}}} \mathrm{Acc}'\, x\\
&\alpha(x, \mathrm{acc}\ f) := \mathrm{acc}'\ (\lambda\, (y : V_{\mathrm{Acc}})(p : \pi_0\, y \in x).\, \alpha(\pi_0\, y, f\ (\pi_0\, y)\, p))
\end{aligned}
$$
□

3 Internalisations of types in ∈-structures

Any element $a : V$ gives rise to a type $\mathrm{El}\, a$ (Definition 2), and in some sense a represents $\mathrm{El}\, a$ inside the bigger structure of $(V, \in)$. For instance, an operation from a to b is precisely a function from $\mathrm{El}\, a \to \mathrm{El}\, b$ (Proposition 7). A natural question to ask is: Which types can be represented as elements in this way in a given ∈-structure? In this section we introduce some basic vocabulary for talking about this kind of representation. We apply this by giving a very flexible formulation of the axiom of infinity – which constructively is often formulated as the existence of a set collecting the natural numbers for some chosen encoding of these. The flexibility of our formulation is that it takes the encoding as a parameter, making very few assumptions about it. We prove that for ∈-structures that satisfy replacement, the existence of a set of natural numbers is independent of encoding.

3.1 Internalisations and representations

For the rest of this section, fix an ∈-structure $(V, \in)$ and $A : \mathrm{Type}$.

Definition 16 ( ). **A $(V, \in)$-internalisation of** A is an element $a : V$ such that $\mathrm{El}\, a \simeq A$.

There can be several different internalisations of A. However, if we fix an encoding of the elements of A in V, there is at most one element in V which is an internalisation of A with respect to this encoding.

An encoding of the elements of A in V is a function $A \to V$. We will call this a representation because of the superficial similarity with classical representation theory, which can be seen as the study of functors $G \to \mathrm{Vec}_k$ for some group G and field k. In our case representations are functions $A \to V$ where A can be any type (not necessarily a group), and the codomain is some ∈-structure.

Definition 17 ( ). **A $(V, \in)$-representation of** A is a function $A \to V$.

Our main concern is going to be if a given representation gives rise to an internalisation of the domain in the ∈-structure itself. In other words, if the elements pointed out by the representation can be collected to a set such that the type of elements is exactly the type being represented.

Definition 18 ( ). Given a $(V, \in)$-representation of A, say $f : A \to V$, let **an internalisation of** f be an $a : V$ such that for every $z : V$ we have $z \in a \simeq \mathrm{fiber}\, f\, z$.

We say that a representation is internalisable if there is an internalisation of it. An internalisation of a representation is an internalisation of the domain.

Proposition 10 ( ). *Let $f : A \to V$ be a $(V, \in)$-representation of A and let $a : V$ be an internalisation of f. Then a is an internalisation of A.*

Proof. By Lemma 4.8.2 in the HoTT Book (The Univalent Foundations Program, 2013), we have:

$$\mathrm{El}\, a \equiv \left(\sum_{z:V} z \in a \right) \simeq \left(\sum_{z:V} \mathrm{fiber}\, f\, z \right) \simeq A,$$

showing that a is an internalisation of A. □

Once a representation of a type is fixed, the internalisation is uniquely defined. Thus, we will speak of *the* internalisation of a given representation.

Proposition 11 ( ). *Given a $(V, \in)$-representation of A, say $f : A \to V$, the internalisation of f is uniquely defined. That is, the type $\sum_{a:V} \prod_{z:V} (z \in a \simeq \mathrm{fiber}\, f\, z)$ is a proposition.*

Proof. Corollary of Proposition 4. □

The notion of representation gives us a way to separate the internalisation of a type into a structure part (how the elements are encoded) and the property that this structure can be internalised. In fact, an internalisation of a type is exactly the same as an internalisation of a representation of that type.

Proposition 12 ( ). *The type of all internalisable $(V, \in)$-representations of A is equivalent to the type of all internalisations of A, i.e. there is an equivalence*

$$\left(\sum_{f:A \to V} \sum_{a:V} \prod_{z:V} z \in a \simeq \mathrm{fiber}\, f\, z \right) \simeq \sum_{a:V} (\mathrm{El}\ a \simeq A)\,.$$

Proof. For $a : A$, we have the following chain of equivalences:

$$\begin{aligned} \left(\sum_{f:A \to V} \prod_{z:V} z \in a \simeq \mathrm{fiber}\, f\, z \right) &\simeq \sum_{f:A \to V} \sum_{e:\mathrm{El}\, a \simeq A} f \circ e = \pi_0 && (81) \\ &\simeq \sum_{e:\mathrm{El}\, a \simeq A} \sum_{f:A \to V} f = \pi_0 \circ e^{-1} && (82) \\ &\simeq (\mathrm{El}\, a \simeq A) && (83) \end{aligned}$$

Step (81) is an application of Corollary 2. We rearrange in step (82), and finally, step (83) is the fact that the total space of paths from a given point is contractible.

The desired equivalence follows from the equivalence above, after some rearranging. □

If a representation $f : A \to V$ is an embedding, we say that the representation is *faithful*. The truncation level of a representation determines the level of the internalisation. In the special case of a faithful representation, the internalisation is a mere set.

Proposition 13 ( ). *An internalisation of $f : A \to V$ is a $(k+1)$-type if and only if f is a k-truncated representation, for $k : \mathbb{N}_{-1}$.*

Proof. Let $a : V$ be an internalisation of f. Then a is a $(k+1)$-type if and only if $\mathrm{fiber}\, f\, z$ is a k-type, for all $z : V$. In other words, a is a $(k+1)$-type if and only if f is a k-truncated map. □

Note here that if f is a faithful representation, it is not necessarily the case that the type A itself is a set. A mere set in V can be the internalisation of a faithful representation of a type of any level, as long as it can be embedded into V. An exception to this is if V is of level 0, meaning that it *only* contains mere sets, then V itself is a set, all internalisable representations are faithful, and hence any type with an internalisable representation will be a set.

Representations and internalisations give an new perspective on the properties of Section 2. All of them, except foundation, can be seen as stating that a certain representation has an internalisation. For instance, exponentiation can be seen as stating that the representation sending a map $\mathrm{El}\ a \to \mathrm{El}\ b$ to its graph, can be internalised.

Replacement, in particular, can be seen in an interesting light. In classical set theory, replacement is the axiom schema which says that if you have a set you can replace its elements and still get a set, as long as the replacements are uniquely defined (by a formula in the first-order language of set theory). In terms of representations we can view this as follows: Given an internalisation of a type as a set, any other representation can be internalised by replacing the elements of the internalisation by their alternate representation. The various levels of replacement properties correspond to restrictions on what kind of representations can be replaced. For instance, 0-replacement gives the replacement property for faithful representations.

Proposition 14 ( ). *If $(V, \in)$ satisfies $(k+1)$-replacement and A has an internalisation, then any k-truncated representation $f : A \to V$ of A has an internalisation.*

Proof. Let $a : V$ be an internalisation of A, with a given $\alpha : \mathrm{El}\, a \simeq A$. Then apply $(k+1)$-replacement to $f \circ \alpha : \mathrm{El}\, a \to V$ to obtain $b : V$ such that for all $z : V$ there is an equivalence

$$z \in b \simeq \Big\| \sum_{x:\mathrm{El}\ a} f(\alpha\, x) = z \Big\|_k \tag{84}$$

Since f is k-truncated and α is an equivalence, we can swap the base type El a with A and drop the truncation, giving us an equivalence

$$z \in b \simeq \sum_{x:A} f\,x = z \tag{85}$$

Thus b internalises f. □

3.2 Natural Numbers / Infinity

There are many ways of formulating the axiom of infinity. The simplest formulation is perhaps $\exists\, u\, (\emptyset \in u \wedge \forall\, x \in u\ \exists\, y \in u\ x \in y)$, which can be found in some texts, such as *Set Theory* by Bell (Bell, 2011). This formulation depends on foundation for the result to actually be an infinite set, as this axiom is also satisfied by the co-hereditarily finite sets (the final coalgebra of the the finite powerset functor), by the element defined by the equation $x = \{\emptyset, x\}$. Another approach is to define the notion of a successor set, for instance $s\,x = x \cup \{x\}$ or $s\,x = \{x\}$ and then postulate the existence of a set containing $\emptyset$ and which is closed under s. This becomes an infinite set even without foundation, since the successor function preserves well-foundedness and increases rank.

In Aczel's CZF (Aczel, 1978), the axiom of infinity is formulated as the existence of a set of natural numbers, $\exists\, z\ \mathrm{Nat}(z)$. Aczel's formulation of this axiom determines the set of natural numbers uniquely if foundation is assumed. However, if foundation is not assumed, one may have fixed points of the successor function which can be thrown in while still satisfying Aczel's Nat predicate. For instance, given a quine atom, $q = \{q\}$, then for any n we have $\mathrm{Nat}(n) \to \mathrm{Nat}(n \cup q)$. This can be remedied by further assuming that n is accessible, in which case n is uniquely determined.

When choosing a property corresponding to the axiom infinity for $\in$-structures in general, we will leverage the fact that we are not bound by first-order logic, and try to give a direct and intuitive formulation, which does not depend on foundation or assumptions about accessibility. We will also keep to the principle that properties postulating existence of sets should be uniquely determined, or be explicitly given as extra structure. Therefore, we will *not* say that an $\in$-structure has natural numbers to mean that $\mathbb{N}$ has an internalisation, as the type $\sum_{a:V} \mathrm{El}\, a \simeq \mathbb{N}$ is not a proposition. Having an internalisation of a type is a structure, rather than a property. However, having an internalisation *with respect to a fixed representation* is a property (Proposition 11), and it is the one we will use.

We will follow Aczel in choosing the natural numbers as the canonical infinite set, but leave the exact encoding of the natural numbers as extra structure. This leaves room for some exotic representations of the natural numbers, as the internalisation might not even be a mere set. Once an encoding is given, the set of naturals is uniquely determined, and will from a certain point of view behave like the usual natural numbers.

Definition 19 ( ). Given an $\in$-structure $(V, \in)$ with a representation of $\mathbb{N}$, $f : \mathbb{N} \to V$, we say that $(V, \in)$ has **natural numbers represented by** f if f has an internalisation.

Suppose n is a set of natural numbers represented by f and let $e : \mathrm{El}\, n \simeq \mathbb{N}$ be the equivalence given by Proposition 10. Then we can define zero as

$$\mathrm{zero} : \mathrm{El}\, n \tag{86}$$
$$\mathrm{zero} := e^{-1}\, 0 \tag{87}$$

and we can define the successor function as

$$\mathrm{suc} : \mathrm{El}\, n \to \mathrm{El}\, n$$
$$\mathrm{suc}\, x = e^{-1} \circ \mathrm{s} \circ e$$

where s is the successor function on $\mathbb{N}$.

The usual induction principle holds, with respect to zero and suc.

Proposition 15 ( ). *Given an $\in$-structure $(V, \in)$ with natural numbers $n : V$ represented by $f : \mathbb{N} \to V$, let $P : \mathrm{El}\, n \to \mathrm{Type}$ be a type family on $\mathrm{El}\, n$. Given $P\, \mathrm{zero}$ and $\prod_{x:\mathrm{El}\, n} P\, x \to P\, (\mathrm{suc}\ x)$, there is an element of the type $\prod_{x:\mathrm{El}\, n} P\, x$.*

Proof. Let $e : \mathrm{El}\, n \simeq \mathbb{N}$ be the equivalence given by Proposition 10. The result follows from the induction principle on $\mathbb{N}$, transported along e. □

Example 1 ( ). Let $(V, \in)$ be an $\in$-structure with the empty set, 0-singletons and binary 0-union. Then we can define the usual von Neumann representation of the natural numbers $f : \mathbb{N} \to V$ recursively by $f\, 0 := \emptyset$ and $f\, (k + 1) := f\, k \cup_0 \{f\, k\}_0$. This will be a faithful representation, and an $\in$-structure having natural numbers represented by f will be equivalent to the usual characterisations of the von Neumann natural numbers in set theory, such as the axiom of infinity in CZF (in the well-founded sets).

Example 2. Let $(V, \in)$ be an $\in$-structure with the empty set, binary 1-union and 0-singletons, then let $f : \mathbb{N} \to V$ be defined by $f\, 0 := \emptyset$ and $f\, (k + 1) := f\, k \cup_1 \{\emptyset\}_0$. This representation of $\mathbb{N}$ is not faithful, since $f\, k$ has non-trivial automorphisms when $\mathrm{k} > 1$. For instance $f\, 3 = \{\emptyset, \emptyset, \emptyset\}_1$ has $3! = 6$ automorphisms. An internalisation, $n : V$, of this representation would be a multiset with the interesting property that $(f\, k \in n) \simeq \mathrm{Fin}(k!)$.

Interestingly, as long as there is enough replacement in the $\in$-structure, the exact choice of representation of $\mathbb{N}$ does not matter:

Proposition 16 ( ). *If* $(V, \in)$ *satisfies* $(k+1)$*-replacement and has natural numbers for some representation, then* $(V, \in)$ *has natural numbers represented by* f *for any* k*-truncated representation* f*.*

Proof. Suppose $(V, \in)$ has natural numbers for some representation. By Proposition 10 it follows that $\mathbb{N}$ has an internalisation. Thus the result follows by Proposition 14. □

4 Equivalence of extensional coalgebras and U-like $\in$-structures

There is a well-established coalgebraic reading of set theory in which models of set theory can be understood as coalgebras for the powerset functor on the category of classes. This can be traced back to Rieger (Rieger, 1957), but perhaps found more explicitly in work by Osius (Osius, 1974).[§] In this section we establish a similar correspondence between $\in$-structures and coalgebras. We define a hierarchy of functors P_U^n, relative to a universe, and stratified by type levels, whose extensional coalgebras correspond to $\in$-structures at the given type level.

Definition 20 ( ). For $n : \mathbb{N}_{-1}^{\infty}$, define $\mathrm{P}_U^{n+1} : \mathrm{Type} \to \mathrm{Type}$ by $\mathrm{P}_U^{n+1}\ X = \sum_{A:U} A \hookrightarrow_n X$.

Remark: P_U^{∞} is a polynomial functor, but for finite n, P_U^n is *not* polynomial. Note that P_U^0 is a U-restricted 'powerset functor' in that $\mathrm{P}_U^0\ X$ is the type of U-small subtypes of X.

The functorial action of P_U^n on a function $f : X \to Y$ is to postcompose with f and then take the $(n-1)$-image and the $(n-1)$-image inclusion. There is a size issue with this though. With the usual construction of the n-image as $\sum_{b:B} \| \sum_{a:A} g\, a = b \|_n$ (The Univalent Foundations Program, 2013), this type is in U if U is closed under truncation and if both the domain and the codomain are in U. In our case the domain is in U, but the codomain need not be. Is it possible to weaken the assumption that the codomain is in U (given that the domain is) and still conclude that the n-image is in U?

4.1 Images of small types

As a higher groupoid, the image of a function $f : X \to Y$ looks like it has the points from X, but the paths are the paths from Y resulting from

[§]More recent work in this direction is the work of Paul Taylor (Taylor, 2023) studying coalgebraic notions of well-foundedness and recursion.

applying f. For example, consider the function $f : \mathbf{1} \to \text{Type}$ given by $f\,x :\equiv \mathbf{2}$. The image of f is the type $\sum_{x:\text{Type}} \|\mathbf{2} = x\|_{-1}$. There is one point, namely $(\mathbf{2}, |\operatorname{refl}|_{-1})$. However, by univalence, there are two distinct paths $(\mathbf{2}, |\operatorname{refl}|_{-1}) = (\mathbf{2}, |\operatorname{refl}|_{-1})$: the identity equivalence and the equivalence that flips the elements in $\mathbf{2}$ (we only need to consider equality on the first coordinate).

More generally, the n-image looks like it has the k-cells, for $k \leq n+1$, from X, but the $(n+2)$-cells are the $(n+2)$-cells from Y between the $(n+1)$-cells resulting from applying f. *From this intuition it seems reasonable that if all the $(n+2)$-iterated identity types of Y lie in U, along with X, then the resulting higher groupoid n-image also lies in U.*

It turns out that it is enough to assume that this holds for (-1)-images. As pointed out below, this is not true for all univalent universes, but the results presented here will rely upon the assumption that this holds for our particular universe.

Rijke's modified join construction (Rijke, 2017) proves that a sufficient criterion for this smallness of images assumption to hold in a univalent universe is that U is closed under homotopy colimits (or has graph quotients), and that global function extensionality holds. Here, we have chosen instead to directly assume that images of U-small types into locally U-small types are U-small.

The formulation of this assumption relies on two natural notions of smallness — which are also found in Rijke's work (Rijke, 2017) along with the fact that being small a proposition. We generalise to a definition of local smallness at any level.

Definition 21 ( ). A type X is **essentially U-small** if there is $A : U$ such that $A \simeq X$. That is, $\sum_{A:U} A \simeq X$.

We proceed by induction to define local smallness at higher levels.

Definition 22 ( ). Let X be a type. We define the notion of X being n-locally U-small, for $n : \mathbb{N}^\infty$, as follows:

- X is 0-locally U-small if it is essentially U-small.
- X is $(n+1)$-locally U-small if for all $x, y : X$, the identity type $x = y$ is n-locally U-small.
- X is ∞-locally U-small.

We say that a 1-locally U-small type is **locally U-small**.

Lemma 7 ( ). *Being essentially U-small is a mere proposition.*

Proof. We must show that the type $\sum_{A:U} A \simeq X$ is a mere proposition. This type looks close to being of the form $\sum_{b:T} b = a$, which is known to be contractible, but since X is not a type in the universe we cannot directly apply univalence. Instead we give a direct proof based on the definition of being a mere proposition.

Let (A, α) and (B, β) be two elements of $\sum_{A:U} A \simeq X$. Applying univalence, we observe that $\mathrm{ua}\,(\beta^{-1} \cdot \alpha) : A = B$.

It remains to show that we get β by transporting α along the path $\mathrm{ua}\,(\beta^{-1} \cdot \alpha)$, in the family $\lambda(Y : U).Y \simeq X$. But this is easily computed with path algebra:

$$\begin{aligned} \mathrm{tr}^{\lambda Y.Y \simeq X}_{\mathrm{ua}\,(\beta^{-1} \cdot \alpha)}\, \alpha &= \alpha \cdot (\beta^{-1} \cdot \alpha)^{-1} \\ &= \alpha \cdot \alpha^{-1} \cdot \beta \\ &= \beta \end{aligned}$$ □

Now, the following assumption, which will be assumed throughout the rest of the paper, can be formulated:

Assumption 1 (Images of small types). For every small type $A : U$ and every locally U-small $X : \mathit{Type}$, and given a function $f : A \to X$, there is a U-small type $\mathrm{image}\ f : U$, and functions $\mathrm{surj}\, f : A \twoheadrightarrow \mathrm{image}\ f$ and $\mathrm{incl}\, f : \mathrm{image}\ f \hookrightarrow X$, such that for every $a : A$ we have $\mathrm{incl}\, f\ (\mathrm{surj}\, f\, x) \equiv f\, x$.

Remark: Not all univalent universes satisfy the assumption above. For instance, given any univalent universe, U, we can define the subuniverse of mere sets, Set_U. This subuniverse is univalent, and locally Set_U-small, but the image of the map $(\lambda\, x.\mathbf{2}) : \mathbf{1} \to \mathrm{Set}_U$ is not a set, as we saw above. Thus, the image of this map is not essentially Set_U-small.

While the work in this article is presented informally in type theory, there might be some benefit here to spell out the assumption as formulated in Agda. This formulation is uniform in universe levels (the parameters `i` and `j`) so that it can be applied at any level.

```
module _ {i j}
         {Domain : Type i} {Codomain : Type j}
         (_ : is-locally-small i Codomain)
         (f : Domain → Codomain) where

  postulate Image : Type i

  postulate image-inclusion : Image ↪ Codomain

  postulate image-quotient : Domain ↠ Image
```

```
postulate image-β
   : ∀ x → (image-inclusion ⟨ image-quotient ⟨ x ⟩ ⟩) ↦ f x

{-# REWRITE image-β #-}
```

This code can be found in the project source repository, where the file containing the above snippet is called `image-factorisation.agda` ( ).

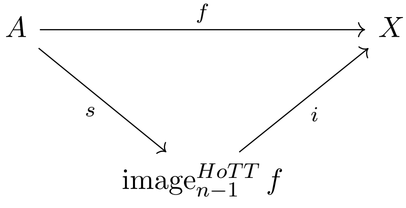
From the assumption of small (-1)-images, the smallness of n-images follows.

Proposition 17 ( )**.** *Let $n : \mathbb{N}^{\infty}_{-2}$. For every small type $A : U$ and every $(n+2)$-locally U-small X : Type, and given a function $f : A \to X$, there is a small type* $\text{image}_n\ f : U$*, together with an n-connected map* $\text{surj}_n\, f : A \twoheadrightarrow_n \text{image}_n\ f$ *and an n-truncated map* $\text{incl}_n\, f : \text{image}_n\ f \hookrightarrow_n X$*, such that for every $a : A$ we have* $\text{incl}\, f\ (\text{surj}\, f\, x) = f\, x$*.*

Proof. This is Proposition 2.2 in Christensen (2021), but we will also give our own proof here, formulated in a slightly different way. Let $A : U$ and $X : Type$ be n-locally U-small, and let $f : A \to X$. We proceed by induction on n.

For the base case we need to construct a type $\text{image}_{-2}\ f : U$ together with a (-2)-connected map $\text{surj}_{-2}\, f : A \twoheadrightarrow_{-2} \text{image}\ f$ and a (-2)-truncated map $\text{incl}_{-2}\, f : \text{image}\ f \hookrightarrow_{-2} X$. Note that any map is (-2)-connected and that (-2)-truncated maps are precisely equivalences. By assumption X is essentially U-small, i.e. there is a type $X' : U$ together with an equivalence $e : X \simeq X'$. So we take $\text{image}_{-2}\ f := X'$, $\text{surj}_{-2}\, f := e \circ f$ and $\text{incl}_{-2}\, f := e^{-1}$.

Suppose the proposition holds for n, and suppose X is $(n+1)$-locally U-small. Let $\text{image}^{HoTT}_n\ g$ denote the HoTT Book definition of the n-image of a function g (The Univalent Foundations Program, 2013, Def. 7.6.3). We will show that $\text{image}^{HoTT}_{n-1}\ f$ is essentially U-small. By Theorem 7.6.6 in (The Univalent Foundations Program, 2013) we have a factorisation



where s is $(n-1)$-connected and i is $(n-1)$-truncated. The map s is in particular surjective, so by the uniqueness of the image factorisation, $\text{image}^{HoTT}_{n-1}\ f \simeq \text{image}\ s$. Thus we are done if we can show that $\text{image}^{HoTT}_{n-1}\ f$

is locally U-small. Since being U-small is a proposition and s is surjective, it is enough to show that $s\,x = s\,y$ is essentially U-small for all $x, y : A$. But by the characterisation of equality in n-images we have the equivalence $(s\,x = s\,y) \simeq \mathrm{image}^{HoTT}_{n-2}\ \big(\mathrm{ap}_f\ x\,y\big)$. The domain $x = y$ of $\mathrm{ap}_f\ x\,y$ is in U and the codomain $f\,x = f\,y$ is n-locally U-small by assumption. Thus, by the induction hypothesis and the uniqueness of the $(n-2)$-image factorisation, $\mathrm{image}^{HoTT}_{n-2}\ \big(\mathrm{ap}_f\ x\,y\big) \simeq \mathrm{image}_{n-2}\ \big(\mathrm{ap}_f\ x\,y\big)$ and thus $s\,x = s\,y$ is essentially U-small.

For the case $n = \infty$, note that an ∞-connected map is an equivalence and that any map is ∞-truncated. Thus we take $\mathrm{image}_\infty\ f := A$, $\mathrm{surj}_\infty\ f :=$ id-equiv and $\mathrm{incl}_\infty\ f := f$. □

4.2 The functor P^n_U

With the construction of U-small n-images, we can define the functorial action of P^n_U. For $n : \mathbb{N}^\infty$, and two types X, Y such that Y is $(n+1)$-locally U-small, the $(n-1)$-image of any function $\phi : X \to Y$, $\mathrm{image}_{n-1}\ \phi$, lies in U by Proposition 17. Moreover, the image inclusion $\mathrm{incl}_{n-1}\ \phi : \mathrm{image}_{n-1}\ \phi \to Y$ is an $(n-1)$-truncated map. Thus we make the following definition:

Definition 23 ( ). Let $n : \mathbb{N}^\infty$. For types X, Y such that Y is $(n+1)$-locally U-small, and a function $\phi : X \to Y$ let $\mathrm{P}^n_U\ \phi : \mathrm{P}^n_U\ X \to \mathrm{P}^n_U\ Y$ be the function

$$\mathrm{P}^n_U\,\phi\ (A, f) := \big(\mathrm{image}_{n-1}\,(\phi \circ f), \mathrm{incl}_{n-1}\,(\phi \circ f)\big)$$

Note: For $n = \infty$ we get $\mathrm{P}^\infty_U\,\phi\ (A, f) \equiv (A, \phi \circ f)$.

Thinking of the elements in $\mathrm{P}^n_U\ X$ as slices over the type X, equality should be fiberwise equivalence. This is indeed the case, as the next proposition shows.

Proposition 18 ( ). *Given $n : \mathbb{N}^\infty$ and a type X, for $(A, f), (B, g) : \mathrm{P}^n_U\ X$ there is an equivalence*

$$((A, f) = (B, g)) \simeq \prod_{x:X} \mathrm{fiber}\ f\,x \simeq \mathrm{fiber}\ g\,x$$

Proof. For $(A, f), (B, g) : \mathrm{P}^{\infty}_U\ X$ we have the following chain of equivalences:

$$((A, f) = (B, g)) \simeq \sum_{\alpha : A = B} g \circ (\mathrm{coe}\ \alpha) = f \tag{88}$$

$$\simeq \sum_{\alpha : A \simeq B} g \circ \alpha = f \tag{89}$$

$$\simeq \sum_{\alpha : \sum_{x:X} \mathrm{fiber}\ f\,x \simeq \sum_{x:X} \mathrm{fiber}\ g\,x} \pi_0 \circ \alpha = \pi_0 \tag{90}$$

$$\simeq \prod_{x:X} \mathrm{fiber}\ f\,x \simeq \mathrm{fiber}\ g\,x \tag{91}$$

Step (88) is the usual characterisation of equality in ϕ-types, step (89) is the univalence axiom, step (90) is Lemma 4.8.2 in the HoTT Book (The Univalent Foundations Program, 2013, p. 186) and step (91) is Corollary 2. (Note: the equivalence $(\sum_{\alpha:A\simeq B} g \circ \alpha = f) \simeq \prod_{x:X}(\mathrm{fiber}\ f\,x \simeq \mathrm{fiber}\ g\,x)$ was proved as Lemma 5 in *Multisets in type theory* (Gylterud, 2019), by explicitly constructing an equivalence.)

For finite n, the type $\mathrm{P}^n_U\ X$ is a subtype of $\mathrm{P}^{\infty}_U\ X$, and thus has the same identity types. □

The characterisation of equality on P^n_U allows us to compute the type level.

Corollary 3 ( ). *Given $n : \mathbb{N}^{\infty}$ and a type X, the type $\mathrm{P}^n_U\ X$ is an n-type.*

Proof. For $(A, f), (B, g) : \mathrm{P}^n_U\ X$ and $x : X$, the types fiber $f\,x$ and fiber $g\,x$ are $(n-1)$-truncated. Thus the identity type $(A, f) = (B, g)$ is $(n-1)$-truncated. □

The functor P^n_U preserves homotopies.

Proposition 19 ( ). *Let $n : \mathbb{N}^{\infty}$. For types X, Y such that Y is $(n+1)$-locally U-small, and functions $\phi, \psi : X \to Y$, any homotopy $\epsilon : \phi \sim \psi$ induces a homotopy $\mathrm{P}^n_U\ \epsilon : \mathrm{P}^n_U\,\phi \sim \mathrm{P}^n_U\,\psi$. Moreover,* refl-htpy *is sent to* refl-htpy.

Proof. Given $\epsilon : \phi \sim \psi$ we construct the homotopy

$$\begin{aligned} &\mathrm{P}^n_U\,\epsilon : \mathrm{P}^n_U\,\phi \sim \mathrm{P}^n_U\,\psi \\ &\mathrm{P}^n_U\,\epsilon\,(A, f) :\equiv \mathrm{ap}_{\lambda\,\sigma.\,\mathrm{P}^n_U\,\sigma\,(A,f)}(\mathrm{funext}\ \epsilon) \end{aligned}$$

For P^n_U refl-htpy we have the following chain of equalities:

$$\begin{aligned} \mathrm{P}^n_U\,\text{refl-htpy}\,(A, f) &\equiv \mathrm{ap}_{\lambda\,\sigma.\,\mathrm{P}^n_U\,\sigma\,(A,f)}(\mathrm{funext}\ \text{refl-htpy}) \\ &= \mathrm{ap}_{\lambda\,\sigma.\,\mathrm{P}^n_U\,\sigma\,(A,f)}\,\mathrm{refl} \\ &\equiv \mathrm{refl} \end{aligned}$$

□

We recap the definition of an algebra for the functor P_U^n and a homomorphism between two algebras.

Definition 24 ( ). A P_U^n-algebra, (X, m) consists of an $(n+1)$-locally U-small type X and a map $m : \mathrm{P}_U^n\, X \to X$.

The restriction to $(n+1)$-locally U-small carrier types for P_U^n algebras, is because P_U^n is only functorial on such types.

Definition 25 ( ). Given two P_U^n algebras, (X, m) and (X', m'), a P_U^n-algebra homomorphism from (X, m) to (X', m') is a pair (ϕ, α) where $\phi : X \to X'$ and $\alpha : \phi \circ m \sim m' \circ \mathrm{P}_U^n\, \phi$.

For two P_U^n-algebra homomorphisms (ϕ, α) and (ψ, β) to be equal, there needs to be a homotopy between the underlying maps ϕ and ψ. But there is also a coherence condition on α and β, which can be seen as saying that the two natural ways of proving that $m'\,(\mathrm{P}_U^n\ \phi) =_X' \psi\,(m\,a)$ are equal identifications in X'. This holds trivially if X' is a mere set, and in that case only the homotopy between the maps is relevant.

Lemma 8 ( ). *For any two P_U^n-algebra homomorphisms (ϕ, α) and (ψ, β), from (X, m) to (X', m'), their identity type is characterised by:*

$$((\phi, \alpha) = (\psi, \beta)) \simeq \sum_{\epsilon : \phi \sim \psi} \prod_{a : \mathrm{P}_U^n\, X} \alpha\, a \cdot \mathrm{ap}_{m'}(\mathrm{P}_U^n\ \epsilon\, a) = \epsilon\,(m\,a) \cdot \beta\, a \tag{92}$$

Proof. We use the structure identity principle. Given an algebra homomorphism (ϕ, α), we need to define families $P : (X \to X') \to \mathrm{Type}$ and $Q : \prod_{\psi : X \to X'} (\psi \circ m \sim m' \circ \mathrm{P}_U^n\, \psi) \to P\,\psi \to \mathrm{Type}$, which should characterise the identity type of the base type and of the fibration respectively. Then we need to construct elements $p : P\,\phi$ and $q : Q\,\phi\,\alpha\,p$. Finally, we need to show that for all $\psi : X \to X'$, $(\phi = \psi) \simeq P\,\psi$ and that for all $\alpha' : \phi \circ m \sim m' \circ \mathrm{P}_U^n\, \phi$, $(\alpha = \alpha') \simeq Q\,\phi\,\alpha'\,p$.

The family P is defined as $P\,\psi :\equiv (\phi \sim \psi)$, and Q is defined as $Q\,\psi\,\beta\,p' :\equiv \prod_{a : \mathrm{P}_U^n\, X} \alpha\, a \cdot \mathrm{ap}_{m'}(\mathrm{P}_U^n\ p'\, a) = p'\,(m\,a) \cdot \beta\, a$. For the element in $P\,\phi$ we take $p :\equiv \text{refl-htpy}$, and for $Q\,\phi\,\alpha\,p$ and $(A, f) : \mathrm{P}_U^n\, X$ we have the following chain of equalities:

$$\alpha\,(A, f) \cdot \mathrm{ap}_{m'}(\mathrm{P}_U^n\ p\,(A, f)) = \alpha\,(A, f) \cdot \mathrm{ap}_{m'}\, \mathrm{refl} \tag{93}$$

$$= \alpha\,(A, f) \tag{94}$$

$$= p\,(m\,(A, f)) \cdot \alpha\,(A, f) \tag{95}$$

In step (93) we use Proposition 19.

It remains to construct the equivalences. By function extensionality, $(\phi = \psi) \simeq (\phi \sim \psi)$. Moreover, for $\alpha' : \phi \circ m \sim m' \circ \mathrm{P}^n_U\, \phi$ we have the following chain of equivalences:

$$\begin{aligned}
(\alpha = \alpha') &\simeq (\alpha \sim \alpha') && (96)\\
&\simeq \prod_{a : \mathrm{P}^n_U\, X} \alpha\, a \cdot \mathrm{ap}_{m'}\, \mathrm{refl} = \mathrm{refl} \cdot \alpha'\, a && (97)\\
&\simeq \prod_{a : \mathrm{P}^n_U\, X} \alpha\, a \cdot \mathrm{ap}_{m'} (\mathrm{P}^n_U\; p\, a) = p\, (m\, a) \cdot \alpha'\, a && (98)\\
&\equiv Q\, \phi\, \alpha'\, p && (99)
\end{aligned}$$

Step (96) is function extensionality, and in step (98) we again use Proposition 19. □

4.3 U-likeness

In classical set theory, the Mostowsky collapse relates any well-founded *set-like* relation to the well-founded hierarchy. A set-like relation is a class relation, say R, where $\{y \mid (x, y) \in R\}$ is a set for all x. The notion of "being small", i.e. being a set, is also one of the basic distinctions in algebraic set theory (Joyal and Moerdijk, 1995).

In type theory, types take the role of classes while the universe U provides a measure of smallness, akin to being a set in set theory. Hence, the notion of a U-like ∈-structure will mirror the classical notion of an extensional, set-like relation.

Definition 26 ( ). An ∈-structure, $(V, \in)$ is **U-like** if $\mathrm{El}\, a$ is essentially U-small for every $a : V$.

Remark: To simplify notation, we will coerce $\mathrm{El}\, a : U$ in U-like ∈-structures.

Proposition 20 ( ). *Being U-like is a mere proposition for any ∈-structure,* $(V, \in)$.

The interaction between internalisable types and U-likeness is straightforward:

Proposition 21 ( ). *An ∈-structure,* $(V, \in)$*, is U-like if any only if for each internalisable* $A : \mathrm{Type}$*, the type A is essentially U-small.*

Proof. We have the following chain of equivalences:

$$\begin{aligned}
&\prod_{A:\mathrm{Type}} \left(\sum_{a:V} \mathrm{El}\, a \simeq A\right) \to \sum_{X:U} A \simeq X \\
&\simeq \prod_{a:V} \left(\sum_{A:\mathrm{Type}} \mathrm{El}\, a \simeq A\right) \to \sum_{X:U} \mathrm{El}\, a \simeq X \qquad (100)\\
&\simeq \prod_{a:V} \left(\sum_{X:U} \mathrm{El}\, a \simeq X\right) \qquad (101)
\end{aligned}$$

In step (101) we use the fact that $\sum_{A:\mathrm{Type}} \mathrm{El}\, a \simeq A$ is contractible. □

We will now investigate in detail the relationship between U-like $\in$-structures and P^n_U-coalgebras.

Lemma 9 ($\in$-structures are coalgebras). *For a fixed V, having a U-like $\in$-structure on V is equivalent to having a coalgebra structure $V \hookrightarrow \mathrm{P}^\infty_U\ V$, which is an embedding. This equivalence sends the relation $\in : V \to V \to$* Type *to the coalgebra $\lambda\, a.(\mathrm{El}\ a, \pi_0)$ and, in the other direction, it sends the coalgebra $m : V \to \mathrm{P}^\infty_U\ V$ to the relation $\lambda\, b\, a.\,\mathrm{fiber}\ (\pi_1\,(m\, a))\ b$.*

Proof. We have the following chain of equivalences:

$$\begin{aligned}
V \to \sum_{A:U} (A \to V) &\simeq \left(V \to \sum_{F:V\to\mathrm{Type}} \sum_{E:U} \left(\left(\sum_{b:V} F\, b\right) \simeq E\right)\right) \qquad (102)\\
&\simeq \sum_{\in:V\to V\to\mathrm{Type}} \prod_{a:V} \sum_{E:U} \left(\left(\sum_{b:V} b \in a\right) \simeq E\right) \qquad (103)
\end{aligned}$$

The equivalence (102) follows by substituting $\sum_{A:U}(A \to V)$ with the equivalent type
$\sum_{F:V\to\mathrm{Type}} \sum_{E:U} (\sum_{b:V} F\, b) \simeq E$, since a fibration $A \to V$, with a total type $A : U$ is equivalent to a family over V with U-small total type, as being U-small is a mere proposition. The equivalence (103) is essentially currying.

Chasing a coalgebra, respectively an $\in$-relation, along the chain of equivalences we see that it computes as stated.

The type $\sum_{\in:V\to V\to\mathrm{Type}} \prod_{a:V} \sum_{E:U} ((\sum_{b:V} b \in a) \simeq E)$ is exactly the type of $\in$-structures on V which are U-like, except for the extensionality requirement. It remains to show that the coalgebra is an embedding if and only if the corresponding $\in$-relation, given by the equivalence above, is extensional.

Given a coalgebra $m : V \to \mathrm{P}_U^{\infty}\, V$, let $\in$ be the corresponding relation given by the equivalence. By Proposition 18 we have, for any $x, y : V$, an equivalence

$$e : (m\,x = m\,y) \simeq \left(\prod_{z:V} z \in x \simeq z \in y\right)$$

This equivalence sends refl to $\lambda\, z.\, \text{id-equiv}$. Let $e' : x = y \to \prod_{z:V} z \in x \simeq z \in y$ be the map defined by path induction as $e'\ \text{refl} :\equiv \lambda\, z.\, \text{id-equiv}$. The following diagram commutes:

$$\begin{array}{ccc} x = y & \xrightarrow{\quad \mathrm{ap}_m \quad} & m\,x = m\,y \\ & {\scriptstyle e'} \searrow \qquad \swarrow {\scriptstyle e} & \\ & \prod_{z:V} z \in x \simeq z \in y & \end{array}$$

Since e is an equivalence it follows that ap_m is an equivalence if and only if e' is an equivalence. □

Lemma 10. *Given a U-like $\in$-structure, $(V, \in)$, and any $n : \mathbb{N}_{-1}$, the following are equivalent:*

1. *$b \in a$ is an n-type for all $a, b : V$,*
2. *$\lambda\, a.(\mathrm{El}\ a, \pi_0)$ restricts to a map $V \hookrightarrow \mathrm{P}_U^{n+1}\, V$.*

Proof. The fibres of $\pi_0 : (\sum_{b:V} b \in a) \to V$ are exactly the types $b \in a$. □

The following theorem combines the two previous lemmas in order to characterise U-like $\in$-structures of various levels in terms of coalgebras.

Theorem 3 ( ). *For a fixed V and for $n : \mathbb{N}_{-1}^{\infty}$, having a U-like $\in$-structure on V such that $b \in a$ is an n-type for all $a, b : V$ is equivalent to having a coalgebra structure $V \hookrightarrow \mathrm{P}_U^{n+1}\, V$.*

Proof. Simple corollary of the previous two lemmas. □

5 Fixed-point models

Rieger's theorem is a result in set theory which states that any class with a set-like binary relation, which is a fixed-point of the related powerset functor, is a model of ZFC$^-$ (ZF without foundation) (Rieger, 1957). In this section we prove an analogous result, considering the family of functors P_U^n as higher level generalisations of the powerset functor. Any fixed-point for P_U^n is in particular a P_U^n-coalgebra for which the coalgebra map is an embedding.

Thus, by Theorem 3, such a fixed-point gives rise to a U-like $\in$-structure. This induced $\in$-structure will satisfy almost all the properties defined in Section 2, at some level, the only exception being foundation.

So we assume in this section that we are given a type V and an equivalence $\sup : \mathrm{P}_U^n\, V \simeq V$. Let $\mathrm{desup} : V \simeq \mathrm{P}_U^n\, V$ be the inverse of sup. We will specifically show that the induced $\in$-structure on V has the following properties:

- Empty set.
- U-restricted n-separation.
- If V is $(n+1)$-locally U-small, it has ∞-unordered I-tupling for all $(n-1)$-truncated types $I : U$.
- If V is $(k+1)$-locally U-small, for some $k \leq n$ then it has:
  - k-unordered I-tupling for all $I : U$,
  - k-replacement,
  - k-union.
- V has exponentiation for all ordered pairing structures.
- V has natural numbers represented by f for any $(n-1)$-truncated representation $f : \mathbb{N} \to V$.

Notation: Given $x : V$, we will use the notation $\overline{x} : U$ and $\widetilde{x} : \overline{x} \hookrightarrow_{n-1} V$ for the type and $(n-1)$-truncated map such that $x = \sup(\overline{x}, \widetilde{x})$, i.e. $\overline{x} = \pi_0\,(\mathrm{desup}\ x)$ and $\widetilde{x} = \pi_1\,(\mathrm{desup}\ x)$.

Definition 27 ( ). Let $(V, \in)$ be the $\in$-structure on V given by Theorem 3. Note that for $x, y : V$,

$$y \in x \equiv \mathrm{fiber}\ \widetilde{x}\, y$$

Proposition 22 ( ). *For any $x : V$ we have that $\overline{x} \simeq \mathrm{El}\ x$.*

Proof. We have the following:

$$\mathrm{El}\ x \equiv \left(\sum_{y:V} \mathrm{fiber}\ \widetilde{x}\, y \right) \simeq \overline{x} \tag{104}$$

where the equivalence in the last step is Lemma 4.8.2 in the HoTT Book (The Univalent Foundations Program, 2013, p. 142). ∎

Proposition 23 ( ). *The type V is an n-type.*

Proof. $V \simeq \mathrm{P}^n_U\, V$ and $\mathrm{P}^n_U\, V$ is an n-type by Corollary 3. □

Many of the constructions of the properties defined in Section 2 require us to use sup on the k-image of some function, and its k-truncated inclusion map. For this, the k-image needs to be U-small. We therefore need to assume that V is appropriately locally U-small in several of the following theorems.

5.1 Empty set

The empty set is a special case of unordered tupling but it does not require any assumption about local smallness. Therefore we note this special case separately.

Theorem 4 ( ). *The $\in$-structure $(V, \in)$ has **empty set**.*

Proof. The empty type $\mathbf{0}$ embeds into any type, in particular, it embeds into V. Let $f : \mathbf{0} \to V$ be the unique map from $\mathbf{0}$ to V. Since the map is an embedding, it is also an $(n-1)$-truncated map. The empty set is defined as

$$\emptyset := \sup(\mathbf{0}, f)$$

For $x : V$ we have the following chain of equivalences:

$$x \in \emptyset \simeq \mathrm{fiber}\ f\ x \simeq \left(\sum_{s:\mathbf{0}} f\ s = x\right) \simeq \mathbf{0}$$ □

5.2 Restricted separation

Theorem 5 ( ). *The $\in$-structure $(V, \in)$ has **U-restricted n-separation**.*

Proof. For $x : V$ and $\Phi : \mathrm{El}\, x \to (n-1)\text{-}\mathrm{Type}_U$ we construct the term $\{x \mid \Phi\}$ as follows:

$$\{x \mid \Phi\} := \sup\left(\sum_{a:\overline{x}} \Phi\,(\widetilde{x}\, a, (a, \mathrm{refl})), \lambda(a, \_).\widetilde{x}\, a\right).$$

The map $\lambda(a, \_).\widetilde{x}\, a$ is $(n-1)$ truncated as it is the composition of the $(n-1)$-truncated map $\widetilde{x}$ and the projection $\pi_0 : \sum_{a:\overline{x}} \Phi\,(\widetilde{x}\, a, (a, \mathrm{refl})) \to \overline{x}$, which is $(n-1)$-truncated since Φ is a family of $(n-1)$-truncated types.

For $z : V$ we then get the following chain of equivalences:

$$
\begin{aligned}
z \in \{x \mid \Phi\} &= \sum_{(a,\_):\sum_{a:\overline{x}} \Phi\,(\widetilde{x}\,a,(a,\mathrm{refl}))} \widetilde{x}\,a = z \\
&\simeq \sum_{(a,\_):\sum_{a:\overline{x}} \widetilde{x}\,a = z} \Phi\,(\widetilde{x}\,a,(a,\mathrm{refl})) \\
&\simeq \sum_{e:(z\in x)} \Phi\,(z,e)
\end{aligned}
$$

□

5.3 Unordered tupling

For unordered tupling we can construct the k-truncated version if V is $(k+1)$-locally U-small, and we can construct the ∞-truncated version if the indexing type has lower type level than V.

Theorem 6 ( ). *Let $k : \mathbb{N}^\infty$ be such that $k \leq n$. If V is $(k+1)$-locally U-small then the $\in$-structure $(V, \in)$ has **k-unordered I-tupling** for all $I : U$.*

Proof. Let $I : U$ and $v : I \to V$, and construct the element $\{v\}_k : V$ by

$$\{v\}_k := \sup\left(\mathrm{image}_{k-1}\ v, \mathrm{incl}_{k-1}\ v\right)$$

Since $k \leq n$, $\mathrm{incl}_{k-1}\ v$ is an $(n-1)$-truncated map. Then, for any $z : V$, we have the following chain of equivalences:

$$
\begin{aligned}
z \in \{v\}_k &= \mathrm{fiber}\left(\mathrm{incl}_{k-1}\ v\right)\ z \\
&\simeq \left\| \mathrm{fiber}\ v\ z \right\|_{k-1} \\
&\equiv \left\| \sum_{i:I} v\,i = z \right\|_{k-1}
\end{aligned}
$$

□

Corollary 4 ( ). *Let $k : \mathbb{N}^\infty$ be such that $k \leq n$. If V is $(k+1)$-locally U-small then V contains the k-unordered 1-tupling $\{x\}_k$, and the k-unordered 2-tupling $\{x, y\}_k$, for any $x, y : V$.*

Theorem 7 ( ). *If V is $(n+1)$-locally U-small then it has an ordered pairing structure.*

Proof. Follows from the previous corollary together with Theorem 4, Theorem 1 and Proposition 23. □

Theorem 8 ( ). *Let $k : \mathbb{N}$ be such that $k < n$. If V is $(n+1)$-locally U-small, then the $\in$-structure $(V, \in)$ has ∞-**unordered I-tupling** for all $I : k\text{-}\mathrm{Type}_U$.*

Proof. Let $I : U$ and $v : I \to V$. We construct the element $\{v\}_\infty : V$ as in Theorem 6:

$$\{v\}_\infty := \sup\left(\mathrm{image}_{n-1}\ v, \mathrm{incl}_{n-1}\ v\right)$$

Then, for any $z : V$, we have the following chain of equivalences:

$$
\begin{aligned}
z \in \{v\}_\infty &= \mathrm{fiber}\left(\mathrm{incl}_{k-1}\ v\right)\ z && (105)\\
&\simeq \left\| \mathrm{fiber}\ v\ z \right\|_{n-1} && (106)\\
&\equiv \left\| \sum_{i:I} v\, i = z \right\|_{n-1} && (107)\\
&\simeq \sum_{i:I} v\, i = z && (108)
\end{aligned}
$$

where step (108) is the fact that $\sum_{i:I} v\, i = z$ is $(n-1)$-truncated since V is an n-type and I is a k-type for some $k < n$. □

5.4 Replacement

Theorem 9 ( ). *Let $k : \mathbb{N}^\infty$ be such that $k \leq n$. If V is $(k+1)$-locally U-small, then the $\in$-structure $(V, \in)$ has k-**replacement**.*

Proof. For $x : V$ and $f : \mathrm{El}\ x \to V$, let $\phi : \overline{x} \to V$ be the function that sends $a : \overline{x}$ to $f\ (a, \mathrm{refl})$. We define the following element:

$$\{f(y) \mid y \in x\} := \sup\left(\mathrm{image}_{k-1}\ \phi, \mathrm{incl}_{k-1}\ \phi\right)$$

Since $k \leq n$, $\mathrm{incl}_{k-1}\ \phi$ is an $(n-1)$-truncated map. For $z : V$ we have the following chain of equivalences:

$$
\begin{aligned}
z \in \{f(y) \mid y \in x\} &= \mathrm{fiber}\left(\mathrm{incl}_{k-1}\ \phi\right) z && (109)\\
&\simeq \left\| \mathrm{fiber}\ \phi\ z \right\|_{k-1} && (110)\\
&\simeq \left\| \sum_{s:\mathrm{El}\ x} f\, s = z \right\|_{k-1} && (111)
\end{aligned}
$$

where (111) uses Proposition 22. □

5.5 Union

Theorem 10 ( ). *Let $k : \mathbb{N}^\infty$ be such that $k \leq n$. If V is $(k+1)$-locally U-small, then the $\in$-structure $(V, \in)$ has k-**union**.*

Proof. For $x : V$, let $\phi_x : \sum_{a:\overline{x}} \overline{(\widetilde{x}\, a)} \to V$ be the function that sends (a, b) to $\widetilde{(\widetilde{x}\, a)}\, b$. We then define the union by

$$\bigcup_k x := \sup\left(\text{image}_{k-1}\left(\phi_x\right), \text{incl}_{k-1}\left(\phi_x\right)\right)$$

Since $k \leq n$, $\text{incl}_{k-1}\left(\phi_x\right)$ is an $(n-1)$-truncated map. For $z : V$ we thus have the following chain of equivalences:

$$z \in \bigcup_k x = \text{fiber}\left(\text{incl}_{k-1}\left(\phi_x\right)\right) z \tag{112}$$

$$\simeq \left\| \text{fiber}\ \phi_x\, z \right\|_{k-1} \tag{113}$$

$$\simeq \left\| \sum_{a:\overline{x}} z \in \widetilde{x}\, a \right\|_{k-1} \tag{114}$$

$$\simeq \left\| \sum_{y:V} (z \in y) \times (y \in x) \right\|_{k-1}, \tag{115}$$

where (115) uses Proposition 22. □

5.6 Exponentiation

The property of having exponentiation is relative to an ordered pairing structure. It turns out that the construction of ordered pairs does not matter. $(V, \in)$ has exponentiation for any ordered pairing structure. In order to prove this though, we first need some lemmas.

Lemma 11 ( ). *Let A be a type and B a type family over A. For any two functions $\phi, \psi : \prod_{a:A} B\, a$ there is an equivalence*

$$(\phi = \psi) \simeq \sum_{e:A\simeq A} \prod_{a:A} (a, \phi\, a) = (e\, a, \psi\, (e\, a))$$

Moreover, this equivalence sends refl *to* $(\text{id-equiv}, \text{refl-htpy})$*.*

Proof. For $\phi, \psi : \prod_{a:A} B\, a$ we have the following chain of equivalences:

$$(\phi = \psi) \simeq (\phi \sim \psi) \tag{116}$$

$$\simeq \sum_{e:A\simeq A} \sum_{r:\text{id}_A \sim e} \prod_{a:A} \text{tr}^B_{r\, a}(\phi\, a) = \psi\, (e\, a) \tag{117}$$

$$\simeq \sum_{e:A\simeq A} \prod_{a:A} \sum_{p:a=e\, a} \text{tr}^B_p(\phi\, a) = \psi\, (e\, a) \tag{118}$$

$$\simeq \sum_{e:A\simeq A} \prod_{a:A} (a, \phi\, a) = (e\, a, \psi\, (e\, a)) \tag{119}$$

Step (116) is function extensionality. In step (117) we use the fact that the type $\sum_{e:A\simeq A} \mathrm{id}_A \sim e$ is contractible, with (id-equiv, refl-htpy) as the center of contraction. Finally, step (119) is the characterisation of equality in Σ-types.

Chasing refl along the chain of equivalences, we get:

$$\begin{aligned} \mathrm{refl} &\mapsto \text{refl-htpy} \\ &\mapsto (\text{id-equiv}, \text{refl-htpy}, \text{refl-htpy}) \\ &\mapsto (\text{id-equiv}, \lambda\, a.(\mathrm{refl}, \mathrm{refl})) \\ &\mapsto (\text{id-equiv}, \text{refl-htpy}) \end{aligned}$$

□

Lemma 12 ( ). *Let $\langle -, - \rangle : V \to V \to V$ be an ordered pairing structure on V. Given a small type $A : U$ and a type family $B : A \to \mathrm{Type}$ together with $(n-1)$-truncated maps $f : A \hookrightarrow_{n-1} V$ and $g : \prod_{a:A} B\, a \hookrightarrow_{n-1} V$, there is an $(n-1)$-truncated map*

$$\mathrm{graph}_{f,g} : \left(\prod_{a:A} B\, a\right) \hookrightarrow_{n-1} V$$

Proof. Given $\phi : \prod_{a:A} B\, a$, we first need to construct an element in V. To this end, we construct the $(n-1)$-truncated map

$$\begin{aligned} &F_\phi : A \hookrightarrow_{n-1} V \\ &F_\phi\, a = \langle f\, a, g\, a\, (\phi\, a) \rangle \end{aligned}$$

This map is $(n-1)$-truncated as it is the composition of the maps $\langle -, - \rangle$, $\lambda\, (a, b).(f\, a, g\, a\, b)$ and $\lambda\, a.(a, \phi\, a)$. The first is (-1)-truncated by assumption, and thus $(n-1)$-truncated. The second is $(n-1)$-truncated since f is $(n-1)$-truncated and $g\, a$ is $(n-1)$-truncated for all $a : A$. To see that the last map is $(n-1)$-truncated, for any pair $(a, b) : \sum_{a:A} B\, a$ we have the following chain of equivalences:

$$\left(\sum_{a':A} (a', \phi\, a') = (a, b)\right) \simeq \left(\sum_{a':A} \sum_{p:a'=a} \mathrm{tr}^B_p (\phi\, a') = b\right) \simeq (\phi\, a = b)$$

where we have used the fact that the type $\sum_{a':A} a' = a$ is contractible, with (a, refl) as the center of contraction. Since V is n-truncated, it follows that $\sum_{a':A} (a', \phi\, a') = (a, b)$ is $(n-1)$-truncated, and thus $\lambda\, a.(a, \phi\, a)$ is an $(n-1)$-truncated map.

This gives us the underlying map

$$\begin{aligned} &\mathrm{graph}_{f,g} : \left(\prod_{a:A} B\, a\right) \to V \\ &\mathrm{graph}_{f,g}\ \phi := \sup\, (A, F_\phi) \end{aligned}$$

What is left is to show that this map is $(n-1)$-truncated. For this we show that $\mathrm{ap}_{\mathrm{graph}_{f,g}} : \phi = \psi \to \mathrm{graph}_{f,g}\ \phi = \mathrm{graph}_{f,g}\ \psi$ is $(n-2)$-truncated for all $\phi, \psi : \prod_{a:A} B\, a$.

Let $\phi, \psi : \prod_{a:A} B\, a$, let $\alpha : (\phi = \psi) \simeq \sum_{e:A \simeq A} \prod_{a:A} (a, \phi\, a) = (e\, a, \psi\, (e\, a))$ be the equivalence given by Lemma 11 and let $\beta : (\sum_{e:A\simeq A} F_\phi \sim F_\psi \circ e) \simeq ((A, F_\phi) = (A, F_\psi))$ be the equivalence given by the usual characterisation of identity in Σ-types together with univalence and function extensionality. We start by observing that

$$\mathrm{ap}_{\mathrm{graph}_{f,g}} \sim \mathrm{ap}_{\mathrm{sup}} \circ \beta \ \circ \ (\lambda\, (e, H).(e, \mathrm{ap}_{\lambda\, (a,b).\langle f\, a, g\, a\, b\rangle} \circ H)) \circ \alpha$$

All three of α, β and $\mathrm{ap}_{\mathrm{sup}}$ are equivalences, and hence $(n-2)$-truncated maps. For the last map, it is enough to show that for all $e : A \simeq A$ and all $a : A$

$$\mathrm{ap}_{\lambda\, (a,b).\langle f\, a, g\, a\, b\rangle} : (a, \phi\, a) = (e\, a, \psi\, (e\, a)) \to F_\phi\, a = F_\psi\, (e\, a)$$

is $(n-2)$-truncated. But this follows from the fact that the composition $\lambda\, (a,b).\langle f\, a, g\, a\, b\rangle$ of two $(n-1)$-truncated maps is $(n-1)$-truncated. □

Theorem 11 (Exponentiation). *The $\in$-structure $(V, \in)$ has* ***exponentiation****, for any ordered pairing structure.*

Proof. Let $\langle -, - \rangle : V \to V \to V$ be an ordered pairing structure on V. Given $x, y : V$, we define the element

$$\begin{aligned} &y^x : V \\ &y^x := \sup\, (\overline{x} \to \overline{y}, \mathrm{graph}_{\widetilde{x}, \lambda\, \_.\widetilde{y}}) \end{aligned}$$

For $f : V$ we then have the following chain of equivalences:

$$\begin{aligned} f \in y^x &\simeq \sum_{\phi:\overline{x}\to\overline{y}} f = \mathrm{graph}_{\widetilde{x}, \lambda\, \_.\widetilde{y}}\ \phi && (120) \\ &\simeq \sum_{\phi:\overline{x}\to\overline{y}} \prod_{z:V} z \in f \simeq z \in \mathrm{graph}_{\widetilde{x}, \lambda\, \_.\widetilde{y}}\ \phi && (121) \\ &\simeq \sum_{\phi:\overline{x}\to\overline{y}} \prod_{z:V} z \in f \simeq \sum_{a:\overline{x}} \langle \widetilde{x}\, a, \widetilde{y}\, (\phi\, a)\rangle = z && (122) \\ &\simeq \sum_{\phi:\mathrm{El}\ x \to \mathrm{El}\ y} \prod_{z:V} z \in f \simeq \sum_{a:\mathrm{El}\ x} \langle \pi_0\, a, \pi_0\, (\phi\, a)\rangle = z && (123) \\ &\simeq \text{operation}\ x\, y\, f && (124) \end{aligned}$$

In step (121) we use extensionality and in step (123) we use Proposition 22. Finally, in step (124) we use Proposition 6. □

5.7 Natural numbers

Theorem 12 ( ). *The $\in$-structure $(V, \in)$ has natural numbers represented by f, for any $(n-1)$-truncated representation f.*

Proof. Let $f : \mathbb{N} \to V$ be an $(n-1)$-truncated representation of $\mathbb{N}$. We construct the following element:

$$n := \sup(\mathbb{N}, f)$$

Then for any $z : V$ we have

$$z \in n \equiv \text{fiber } \widetilde{n}\, z = \text{fiber } f\, z \qquad \square$$

6 The initial P^n_U-algebra

The most straightforward way to construct a fixed-point of a functor is by constructing its initial algebra. For polynomial functors, the initial algebras are well understood. They are called W-types and are used to create a variety of inductive data structures, such as models of set theory in type theory: The initial algebra for P^∞_U formed the basis for Aczel's setoid model of CZF in Martin-Löf type theory (Aczel, 1978). The initial algebras for P^n_U, for finite n, are subtypes of this type.

However, the functor P^n_U is not a polynomial functor, for finite n. So, its initial algebra will not be a simple W-type. Looking carefully at the type $\sum_{A:U} A \hookrightarrow_{n-1} X$ one can see that it is not strictly positive, since the type $A \hookrightarrow_{n-1} X := \sum_{f:A\to X} \prod_{x:X} \text{is-}(n-1)\text{-type}\,(\text{fiber } f\, x)$ contains a negative occurrence of X. Being strictly positive is a usual requirement for inductive definitions, so it is a bit surprising that P^n_U still has an initial algebra.

In this section, we will construct the initial algebra for P^n_U, which then becomes a fixed-point $\in$-structure of level n. One interesting aspect of doing this proof in HoTT is that the carrier for a P^n_U-algebra can lie on any type level.

We start by recalling Aczel's W-type.

Definition 28 ( ). Let $V^\infty := W_{A:U} A$, and denote its (uncurried) constructor $\sup^\infty : \mathrm{P}^\infty_U\, V^\infty \to V^\infty$.

Remark: The superscript of the name, V^∞, indicates that there is no bound on the level of the type. In fact, V^∞ has the same type level as U.

The pair $(V^\infty, \sup^\infty)$ is the initial algebra for the functor $\mathrm{P}^\infty_U : \text{Type} \to \text{Type}$, and hence a fixed-point of P^∞_U. Any fixed-point has a canonical coalgebra structure, and by Lemma 9 this gives rise to a U-like $\in$-structure. Thus, we will denote by $\in^\infty : V^\infty \to V^\infty \to \text{Type}$, this elementhood relation on V^∞. **Note:** by construction, $x \in^\infty \sup^\infty(A, f) \equiv \text{fiber}\, f\, x$.

This is the $\in$-structure explored in *Multisets in type theory* (Gylterud, 2019), which also introduced some of the axioms of the previous sections. In *From multisets to sets in Homotopy Type Theory* (Gylterud, 2018) a 0-level $\in$-structure was carved out as a subtype of V^∞. And we will now prove that this 0-level structure is, as previously suspected, the initial P^0_U algebra. In fact, we will generalise the construction to form a type V^n for all $n : \mathbb{N}$ and show that V^n is the initial algebra for P^n_U.

6.1 Iterative n-types

The types V^n form a stratified hierarchy of $\in$-structures, where V^∞ is at the top. This is analogous to how the n-types in U form a hierarchy with U itself at the top.

Definition 29 ( ). Given $n : \mathbb{N}_{-1}$, define by induction on V^∞, a predicate $\text{is-it-}n\text{-type} : V^\infty \to \text{Type}$ by:

$$\text{is-it-}n\text{-type}\,(\sup^\infty(A, f)) := (\text{is-}n\text{-trunc-map}\ f) \times \left(\prod_{a:A} \text{is-it-}n\text{-type}(f\,a)\right)$$

This predicate is propositional, and an element of V^∞ for which the predicate is true is called *an iterative n-type.*

Remark: We could define iterative (-2)-types analogously, but there are no elements in V^∞ which satisfy this predicate, as the existence of such an element would imply that V^∞ is essentially U-small.

Definition 30 ( ). For $n : \mathbb{N}_{-1}$, let $V^{n+1} := \sum_{x:V^\infty} \text{is-it-}n\text{-type}\, x$ denote the type of iterative n-types, which is a subtype of V^∞.

Even though the type V^n is not itself U-small, it has U-small identity types

Proposition 24 ( ). *The type V^n is locally U-small.*

Proof. First, we observe that, since being an iterative $(n-1)$-type is a proposition, V^n is a subtype of V^∞. It was shown in *Multisets in type theory* (Gylterud, 2019, Lemma 3) that the latter is locally U-small. Since the equality of a subtype is the underlying equality of the base type, it follows that V^n is locally U-small. □

In the following proofs we will leave out elements of types which are propositions, unless they are necessary, in order to increase readability. For the full details of the proofs, please see the Agda formalisation.

Proposition 25 ( ). *The map* $\sup^{\infty} : \mathrm{P}_U^{\infty}\, V^{\infty} \to V^{\infty}$ *restricts to a map* $\sup^{n} : \mathrm{P}_U^{n}\, V^{n} \to V^{n}$.

Proof. Let $A : U$ and $f : A \hookrightarrow_{n-1} V^n$. The element $\sup^{\infty}(A, \pi_0 \circ f)$ is an iterative $(n-1)$-type:

- $\pi_0 \circ f$ is $(n-1)$-truncated since f is $(n-1)$-truncated, and π_0 is the inclusion of a subtype, which is an embedding, and thus $(n-1)$-truncated.
- For every $a : A$ we have $\pi_1\,(f\,a) : \text{is-it-}(n-1)\text{-type}(\pi_0\,(f\,a))$.

Thus we construct the map

$$\begin{aligned}
&\sup^{n} : \mathrm{P}_U^{n}\, V^{n} \to V^{n}\\
&\sup^{n}(A, f) := (\sup^{\infty}(A, \pi_0 \circ f), _)
\end{aligned}$$

□

Proposition 26 ( ). *There is a map* $\mathrm{desup}^{n} : V^{n} \to \mathrm{P}_U^{n}\, V^{n}$.

Proof. Let $x : V^n$, without loss of generality we may assume that $x \equiv (\sup^{\infty}(A, f), (p, q))$ for some $A : U$, $f : A \to V^{\infty}$, $p : \text{is-}(n-1)\text{-trunc-map}\ f$ and $q : \prod_{a:A} \text{is-it-}(n-1)\text{-type}(f\,a)$. The following function:

$$\lambda\, a.(f\,a, q\,a) : A \to V^n$$

is $(n-1)$-truncated. This is because the composition $\pi_0 \circ (\lambda\, a.(f\,a, q\,a))$ is $(n-1)$-truncated, by p, and π_0 is $(n-1)$-truncated as it is the inclusion of a subtype. This implies that the right factor, $\lambda\, a.(f\,a, q\,a)$, is $(n-1)$-truncated. Thus we construct the map

$$\begin{aligned}
&\mathrm{desup}^{n} : V^{n} \to \mathrm{P}_U^{n}\, V^{n}\\
&\mathrm{desup}^{n}(\sup^{\infty}(A, f), (p, q)) := (A, \lambda\, a.(f\,a, q\,a))
\end{aligned}$$

□

Proposition 27 ( ). *The maps* $\sup^{n}$ *and* desup^{n} *form an equivalence.*

Proof. For $(\sup^{\infty}(A, f), (p, q)) : V^n$ we have

$$\begin{aligned}
\pi_0\,(\sup^{n}\,(\mathrm{desup}^{n}\,(\sup^{\infty}(A, f), (p, q))))&\\
&\equiv \pi_0\,(\sup^{n}(A, \lambda\, a.(f\,a, q\,a)))\\
&\equiv (\sup^{\infty}(A, f))\\
&\equiv \pi_0\,(\sup^{\infty}(A, f), (p, q))
\end{aligned}$$

Thus, by the characterisation of equality in subtypes, $\sup^{n} \circ \mathrm{desup}^{n} \sim \mathrm{id}$.

For $A : U$ and $f : A \hookrightarrow_{n-1} V^n$ we have

$$\pi_0\,(\mathrm{desup}^{n}\,(\sup^{n}(A, f))) \equiv A$$

and

$$\begin{aligned}&\pi_0\,(\pi_1(\mathrm{desup}^n\,(\sup^n\,(A,f))))\\&\qquad\equiv \pi_0\,(\pi_1(\mathrm{desup}^n\,(\sup^\infty\,(A,\pi_0\circ f),(\_,\pi_1\circ f))))\\&\qquad\equiv \lambda\,a.(\pi_0\,(f\,a),\pi_1\,(f\,a))\\&\qquad\equiv \pi_0\,f\end{aligned}$$

Thus by the characterisation of equality in Σ-types, and equality in subtypes, $\mathrm{desup}^n \circ \sup^n \sim \mathrm{id}$. □

Theorem 13 ( ). *V^n is a fixed-point to the functor P^n_U.*

Proof. Corollary of the previous proposition. □

Theorem 14. *There is an ∈-structure $(V^n, \in^n)$, on V^n, of level n.*

Proof. This follows from Theorem 13 together with Theorem 3. □

Remark: By construction, for $x, y : V^n$ we have $y \in^n x \equiv \pi_0\, y \in^\infty \pi_0\, x$.

Remark: The ∈-structure $(V^0, \in^0)$ is exactly the ∈-structure studied in *From multisets to sets in Homotopy Type Theory* (Gylterud, 2018), which there was established to be equivalent to the model of set theory given in the HoTT Book (The Univalent Foundations Program, 2013).

6.2 Induction principle and recursors for V^n

V^n is a composite type: a Σ-type over a W-type and a predicate involving induction on that W-type. Since it is not a primitive inductive type it does not come with a ready-to-use induction principle. But, to some extent, $\sup^n : \mathrm{P}^n_U\ V^n \to V^n$ acts as a constructor, for which we have a corresponding induction principle. The expected computation rules do not hold definitionally, but are instead proven to hold up to identity.

Proposition 28 (Elimination for V^n). *Given any family $P : V^n \to \mathrm{Type}$ and a function*

$$\phi : \textstyle\prod_{A:U} \prod_{f:A\hookrightarrow_{n-1} V^n} \left(\prod_{a:A} P\,(f\,a)\right) \to P\,(\sup^n\,(A,f))$$

there is a function $\mathrm{elim}_{V^n}\ P\ \phi : \prod_{x:V^n} P\,x$.

Furthermore, there is a path $\mathrm{elim}_{V^n}\ P\,\phi\,(\sup^n(A,f)) = \phi\,A\,f\,(\mathrm{elim}_{V^n}\ P\,\phi\circ f)$, for any $A : U$ and $f : A \hookrightarrow_{n-1} V^n$.

Proof. Given $x : V^n$, we may assume that $x \equiv (\sup^\infty\,(A,f),(p,q))$ where $A : U$, $f : A \to V^\infty$, p : is-$(n-1)$-trunc-map f and $q : \prod_{a:A}$ is-it-$(n-1)$-type$(f\,a)$. The function $\mathrm{elim}_{V^n}\ P\ \phi$ is defined as follows

$$\mathrm{elim}_{V^n}\ P\ \phi\ x := \mathrm{tr}^P_\alpha\,(\phi\,A\,(\pi_1\,(\mathrm{desup}^n\ x))\,(\lambda\,a.\,\mathrm{elim}_{V^n}\ P\ \phi\ (f\,a,q\,a)))$$

where $\alpha : \sup^n (\mathrm{desup}^n\ x) = x$ by Proposition 27. To construct the path, we use univalence to do equivalence induction. Let Q be the type family

$$Q : \prod_{B:\mathrm{Type}} V^n \simeq B \to \mathrm{Type}$$
$$Q\,B\,e := \prod_{p:\prod_{b:B} P\,(e^{-1}\,b)} \prod_{b:B} \mathrm{tr}^P_\alpha \left(p\ \left(e\ \left(e^{-1}\,b\right)\right)\right) = p\,b$$

where $\alpha : e^{-1}\ \left(e\ \left(e^{-1}\,b\right)\right) = e^{-1}\,b$ is the proof that e is a retraction applied to the element $e^{-1}\,b$. Then, for the case $B := V^n$ and $e := \text{id-equiv}$ we have

$$Q\,V^n\ \text{id-equiv} \equiv \prod_{p:\prod_{x:V^n} P\,(a)} \prod_{x:V^n} (p\,x = p\,x)$$

which is inhabited by $\lambda\,p.\,\text{refl-htpy}$. By univalence we thus have an element of the type $Q\,B\,e$ for any type B and equivalence $e : V^n \simeq B$. For $B := V^n$ and $e := \mathrm{desup}^n$ we construct the term

$$p : \prod_{(A',f'):\mathrm{P}^n_U\,V^n} P\,(\sup^n (A', f'))$$
$$p\,(A', f') := \phi\,A'\,f'\,(\mathrm{elim}_{V^n}\ P\ \phi\ \circ f')$$

Then we have an element of the type $Q\ (\mathrm{P}^n_U\,V^n)\ \mathrm{desup}^n\ p\ (A, f)$, which unfolds to

$$\mathrm{tr}^P_\alpha\,(\phi\,A\,(\pi_1\,(\mathrm{desup}^n\ x))\,(\lambda\,a.\,\mathrm{elim}_{V^n}\ P\ \phi\ (f\,a, q\,a)))$$
$$= \phi\,A\,f\,(\mathrm{elim}_{V^n}\ P\ \phi\ \circ f)$$

where $\alpha : \sup^n\ (\mathrm{desup}^n\ (\sup^n (A', f'))) = \sup^n (A', f')$ is the same element as was used in the construction of $\mathrm{elim}_{V^n}\ P\ \phi$. ☐

As usual, we can consider the specialisation from eliminators to recursors, and in the recursors the computation rules end up holding definitionally. Since $(n-1)$-truncated maps are also functions, V^n actually has recursors for both P^∞_U-algebras and P^n_U-algebras.

Proposition 29 (Untruncated recursion for V^n). *Given any type X and map $m : \mathrm{P}^\infty_U\,X \to X$ there is a map $\mathrm{P}^\infty\text{-rec}_{V^n}\ X\ m : V^n \to X$ such that for $(A, f) : \mathrm{P}^n_U\,V^n$ the following definitional equality holds:*

$$\mathrm{P}^\infty\text{-rec}_{V^n}\ X\ m\ (\sup^n (A, f)) \equiv m\,(\mathrm{P}^\infty_U(\mathrm{P}^\infty\text{-rec}_{V^n}\ X\ m)\ (A, f))$$

Proof. Given $x : V^n$, we may assume that $x \equiv (\sup^\infty(A, f), (p, q))$ where $A : U$, $f : A \to V^\infty$, p : is-$(n-1)$-trunc-map f and $q : \prod_{a:A}$ is-it-$(n-1)$-type$(f\,a)$. The map P^∞-rec_{V^n} is defined as:

$$\mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m\; x := m\,(A, \lambda\, a.\, \mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m\; (f\,a, q\,a))$$

For $(A, f) : \mathrm{P}^n_U\, V^n$ we thus have the following chain of definitional equalities:

$$\begin{aligned}
\mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m\; (\sup^n(A, f)) &\equiv \mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m\; (\sup^\infty(A, \pi_0 \circ f), (_, \pi_1 \circ f))\\
&\equiv m\,(A, \mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m \circ f)\\
&\equiv m\,(\mathrm{P}^\infty_U(\mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m)\; (A, f))
\end{aligned}$$

□

Corollary 5 (Truncated recursion for V^n). *Given a* P^n_U*-algebra* (X, m) *there is a map* P^n-$\mathrm{rec}_{V^n}(X, m) : V^n \to X$ *such that for* $(A, f) : \mathrm{P}^n_U\, V^n$ *the following definitional equality holds:*

$$\mathrm{P}^n\text{-}\mathrm{rec}_{V^n}(X, m)\,(\sup^n(A, f)) \equiv m\,(\mathrm{P}^n_U(\mathrm{P}^n\text{-}\mathrm{rec}_{V^n}(X, m))\; (A, f))$$

Proof. The map P^n-$\mathrm{rec}_{V^n}(X, m)$ is defined as:

$$\mathrm{P}^n\text{-}\mathrm{rec}_{V^n}(X, m) := \mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; (\lambda\,(A, f).m\,(\mathrm{image}_{n-1}\, f, \mathrm{incl}_{n-1}\, f))$$

For $(A, f) : \mathrm{P}^n_U\, V^n$ we have the following chain of definitional equalities:

$$\begin{aligned}
&\mathrm{P}^n\text{-}\mathrm{rec}_{V^n}(X, m)\,(\sup^n(A, f))\\
&\quad\equiv \mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m'\,(\sup^n(A, f))\\
&\quad\equiv m'\,(A, \mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m' \circ f)\\
&\quad\equiv m\,(\mathrm{image}_{n-1}(\mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m' \circ f), \mathrm{incl}_{n-1}(\mathrm{P}^\infty\text{-}\mathrm{rec}_{V^n}\; X\; m' \circ f))\\
&\quad\equiv m\,(\mathrm{image}_{n-1}(\mathrm{P}^n\text{-}\mathrm{rec}_{V^n}(X, m) \circ f), \mathrm{incl}_{n-1}(\mathrm{P}^n\text{-}\mathrm{rec}_{V^n}(X, m) \circ f))\\
&\quad\equiv m\,(\mathrm{P}^n_U(\mathrm{P}^n\text{-}\mathrm{rec}_{V^n}(X, m))\; (A, f))
\end{aligned}$$

where $m' := \lambda\,(A, f).m\,(\mathrm{image}_{n-1}\, f, \mathrm{incl}_{n-1}\, f)$. □

6.3 Initiality of $\mathbf{V}^n$

In the proof of initiality we are to show that the type of homomorphisms from the initial algebra, into any other algebra, is contractible. If we were working with mere sets, it would be sufficient to show that the underlying maps of the homomorphisms are equal to a specified canonical map. But being a homomorphism is actually a structure when the types involved are of higher levels. So, the proof of contractibility has to coherently transfer this structure when proving that every homomorphism is equal to the center of contraction. This is achieved by using the induction principle for V^n and the characterisation of the identity type on P^n_U-algebra homomorphisms.

Theorem 15 ( ). *The algebra* (V^n, sup^n) *is initial: given any other* P_U^n*-algebra* (X, m) *the type of algebra homomorphisms from* (V^n, sup^n) *to* (X, m) *is contractible.*

Proof. The center of contraction is given by $(\mathrm{P}^n\text{-rec}_{V^n}\,(X, m), \text{refl-htpy})$. We will use the characterisation of equality between P_U^n-algebra homomorphisms given by Lemma 8 to show that any other homomorphism is equal to $\mathrm{P}^n\text{-rec}_{V^n}\,(X, m)$.

Let (ϕ, α) be another P_U^n-algebra homomorphism from (V^n, sup^n) to (X, m). We need to do two things:

- Construct a homotopy $\epsilon : \phi \sim \mathrm{P}^n\text{-rec}_{V^n}\,(X, m)$.
- For each $(A, f) : \mathrm{P}_U^n\ V^n$, construct a path

$$\alpha\,(A, f) \cdot \mathrm{ap}_m\,(\mathrm{P}_U^n\,\epsilon\,(A, f)) = \epsilon\,(\mathrm{sup}^n\,(A, f)) \cdot \text{refl-htpy}\,(A, f).$$

To construct a homotopy from ϕ to $\mathrm{P}^n\text{-rec}_{V^n}\,(X, m)$ we use the elimination principle on V^n, by Proposition 28, with the type family $P\,x :\equiv (\phi\,x = \mathrm{P}^n\text{-rec}_{V^n}\,(X, m)\,x)$. This means that given $A : U$, $f : A \hookrightarrow_{n-1} V^n$ and $H : \prod_{a:A} \phi\,(f\,a) = \mathrm{P}^n\text{-rec}_{V^n}\,(X, m)\,(f\,a)$, we need to construct a path

$$\phi\,(\mathrm{sup}^n\,(A, f)) = \mathrm{P}^n\text{-rec}_{V^n}\,(X, m)\,(\mathrm{sup}^n\,(A, f))$$

We have the following chain of equalities:

$$\begin{aligned}
&\phi\,(\mathrm{sup}^n\,(A, f)) \\
&= m\,(\mathrm{P}_U^n\,\phi\,(A, f)) && (125)\\
&\equiv m\,(\mathrm{image}_{n-1}(\phi \circ f), \mathrm{incl}_{n-1}(\phi \circ f)) && (126)\\
&= m\,(\mathrm{image}_{n-1}(\mathrm{P}^n\text{-rec}_{V^n}\,(X, m) \circ f), \mathrm{incl}_{n-1}(\mathrm{P}^n\text{-rec}_{V^n}\,(X, m) \circ f)) && (127)\\
&\equiv m\,(\mathrm{P}_U^n(\mathrm{P}^n\text{-rec}_{V^n}\,(X, m))\,(A, f)) && (128)\\
&\equiv \mathrm{P}^n\text{-rec}_{V^n}\,(X, m)\,(\mathrm{sup}^n\,(A, f)) && (129)
\end{aligned}$$

Step (125) is the path $\alpha\,(A, f)$ and step (127) is the path

$$\mathrm{ap}_{\lambda\,h.m\,(\mathrm{image}_{n-1}\,h, \mathrm{incl}_{n-1}\,h)}(\text{funext } H).$$

So let

$$\sigma := \lambda\,A\,f\,H.\alpha\,(A, f) \cdot \mathrm{ap}_{\lambda\,h.m\,(\mathrm{image}_{n-1}\,h, \mathrm{incl}_{n-1}\,h)}(\text{funext } H)$$

then by Proposition 28 we have a homotopy

$$\mathrm{elim}_{V^n}\ P\ \sigma : \phi \sim \mathrm{P}^n\text{-rec}_{V^n}\,(X, m).$$

It remains to construct the second component of the Σ-type in Lemma 8. We have the following chain of equalities:

$$\begin{aligned}
&\alpha\,(A,f)\cdot \mathrm{ap}_m\,(\mathrm{P}^n_U\,(\mathrm{elim}_{V^n}\ P\ \sigma)\ (A,f)) && (130)\\
&\equiv \alpha\,(A,f)\cdot \mathrm{ap}_m\,\Big(\mathrm{ap}_{\lambda\,\psi.\,\mathrm{P}^n_U\,\psi(A,f)}\,(\mathrm{funext}\ (\mathrm{elim}_{V^n}\ P\ \sigma))\Big) && (131)\\
&= \alpha\,(A,f) && (132)\\
&\quad\cdot \mathrm{ap}_m\,\Big(\mathrm{ap}_{\lambda\,h.(\mathrm{image}_{n-1}\,h,\mathrm{incl}_{n-1}\,h)}\,\big(\mathrm{ap}_{-\circ f}\,(\mathrm{funext}\ (\mathrm{elim}_{V^n}\ P\ \sigma))\big)\Big) &&\\
&= \alpha\,(A,f)\cdot \mathrm{ap}_{\lambda\,h.m\,(\mathrm{image}_{n-1}\,h,\mathrm{incl}_{n-1}\,h)}\,\big(\mathrm{ap}_{-\circ f}\,(\mathrm{funext}\ (\mathrm{elim}_{V^n}\ P\ \sigma))\big) && (133)\\
&= \alpha\,(A,f)\cdot \mathrm{ap}_{\lambda\,h.m\,(\mathrm{image}_{n-1}\,h,\mathrm{incl}_{n-1}\,h)}\,(\mathrm{funext}\ ((\mathrm{elim}_{V^n}\ P\ \sigma)\circ f)) && (134)\\
&\equiv \sigma\,A\,f\ ((\mathrm{elim}_{V^n}\ P\ \sigma)\circ f) && (135)\\
&= \mathrm{elim}_{V^n}\ P\ \sigma\,(\mathrm{sup}^n\,(A,f)) && (136)\\
&= \mathrm{elim}_{V^n}\ P\ \sigma\,(\mathrm{sup}^n\,(A,f))\cdot \text{refl-htpy}\,(A,f) && (137)
\end{aligned}$$

In steps (132) and (133) we use the fact that ap and function composition commute. Step (134) uses the fact that function extensionality respects precomposition. Finally, step (136) is the computation rule for elim_{V^n}. □

6.4 Properties

Since V^n is a fixed-point for P^n_U, the induced ∈-structure satisfies the properties as shown in Section 5. But as V^n is the initial algebra for P^n_U, it also satisfies foundation, which is not true of all fixed-points for P^n_U. More explicitly, we list the properties which $(V^n, \in^n)$ satisfies.

Theorem 16 ( ). *For $n : \mathbb{N}^\infty$, the ∈-structure $(V^n, \in^n)$ satisfies the following properties:*

- *empty set,*
- *U-restricted n-separation,*
- *∞-unordered I-tupling, for all $k : \mathbb{N}_{-1}$ such that $k < n$ and k-truncated types $I : U$,*
- *k-unordered I-tupling, for all $k : \mathbb{N}_{-1}$ such that $k \le n$ and $I : U$,*
- *k-replacement, for all $k : \mathbb{N}_{-1}$ such that $k \le n$,*
- *k-union, for all $k : \mathbb{N}_{-1}$ such that $k \le n$,*
- *exponentiation, for any ordered pairing structure,*
- *natural numbers for any $(n-1)$-truncated representation,*

- *foundation.*

Proof. The first two properties follow directly from Theorem 4 and Theorem 5 respectively since V^n is a fixed-point for P_U^n. The next four properties follow from Theorem 8, Theorem 6, Theorem 9 and Theorem 10 respectively, together with the fact that V^n is locally small (Proposition 24), and thus $(k+1)$-small for all $k : \mathbb{N}$.

By Theorem 11, $(V^n, \in^n)$ has exponentiation for any ordered pairing structure. (Note that $(V^n, \in^n)$ has at least one ordered pairing structure by Theorem 7.)

For natural numbers, the result follows from Theorem 12, since V^n is a fixed-point for P_U^n.

Lastly, we use the induction principle for V^n to show that $(V^n, \in^n)$ has foundation. Given $A : U$ and $f : A \hookrightarrow_{n-1} V^n$ we need to construct an element of type

$$\left(\prod_{a:A} \mathrm{Acc}\,(f\,a)\right) \to \mathrm{Acc}\,(\sup^n(A, f)).$$

Therefore suppose we have $p : \prod_{a:A} \mathrm{Acc}\,(f\,a)$. We construct the following element:

$$\mathrm{acc}\ \big(\lambda\, y\, (a, q).\, \mathrm{tr}_q^{Acc}(p\, a)\big) : \mathrm{Acc}\,(\sup^n(A, f)).$$

It then follows by Proposition 28 that we have

$$\prod_{x:V^n} \mathrm{Acc}\ x.$$

□

7 V^n as an n-type universe of n-types

We have seen that the type V^n can be equipped with a binary relation $\in^n$, making it a model of our higher level generalisation of material set theory. There is a second perspective on V^n, namely as a type theoretic universe à la Tarski. This has already been explored in detail for the type V^0 in Paper I, showing that it is a mere set universe of mere sets which is closed under all the usual type formers and which has definitional decoding. Here we will show that the corresponding universe construction can be done for every V^n.

Let $(\mathrm{V}^n, \in^n)$ be the n-level $\in$-structure given by Theorem 14. Note that the type $x \in^n y$ is an $(n-1)$-type, for any elements $x, y : \mathrm{V}^n$.

Proposition 30 ( ). *The type* V^n *is an* n*-type.*

Proof. This follows both from Proposition 1, and from Proposition 23 together with Theorem 13. □

Definition 31 ( ). For $n : \mathbb{N}^\infty$, define the decoding function on V^n as the family

$$\begin{aligned} \mathrm{El}^n &: \mathrm{V}^n \to U \\ \mathrm{El}^n\ x &:= \overline{x} \end{aligned}$$

We take this as the decoding function rather than El from Definition 2. This is so that the decoding holds up to definitional equality, since we have

$$\mathrm{El}^n\,(\sup^n(A, f)) \equiv A.$$

But the two families are equivalent by Proposition 22.

Proposition 31 ( ). *For $n : \mathbb{N}^\infty$, the decoding $\mathrm{El}^n\ a$ of any element $a : \mathrm{V}^n$, is an n-type.*

Proof. The map $\widetilde{a}$ is an $(n-1)$-truncated map from $\mathrm{El}^n\ a$ into V^n. The domain $\mathrm{El}^n\ a$ is equivalent to the total space of fibers of $\widetilde{a}$, which is an n-type as the base, V^n, is an n-type and each fiber is an $(n-1)$-type. □

Propositions 30 and 31 thus show that V^n is an n-type universe of n-types. Section 3 explored internalisations of types in $\in$-structures. The type of internalisations of a type was shown to be equivalent to the type of internalisable representations of that type (Proposition 11). In the case of V^n, the internalisable representations are the $(n-1)$-truncated ones.

Proposition 32 ( ). *For $n : \mathbb{N}^\infty$ and a type $A : U$ together with a representation $f : A \to \mathrm{V}^n$, the representation is internalisable if and only if it is $(n-1)$-truncated.*

Proof. Suppose that f is internalisable, i.e. we have an element of the type

$$\sum_{a:\mathrm{V}^n} \prod_{z:\mathrm{V}^n} z \in^n a \simeq \mathrm{fiber}\, f\ z.$$

Since the type $z \in^n a$ is an $(n-1)$-type for all $z : \mathrm{V}^n$ it follows that $\mathrm{fiber}\, f\ z$ is an $(n-1)$-type for all $z : \mathrm{V}^n$. Conversely, suppose that f is $(n-1)$-truncated. Then we take for a the element $\sup^n(A, f)$ and for the family of equivalences, the family of identity equivalences. □

Proposition 33 ( ). *For $n : \mathbb{N}^\infty$ and $A : U$, the type of $(n-1)$-truncated representations of A, $A \hookrightarrow_{n-1} \mathrm{V}^n$, is equivalent to the type of internalisations of A, $\sum_{a:A} \mathrm{El}^n\ a \simeq A$.*

Proof. This follows by Proposition 12 and the fact that the families El and El^n are equivalent, together with the previous proposition. Alternatively, $\sup^n$ is an equivalence making the following diagram commute

$$\begin{array}{ccc} \mathrm{P}^n_U\,\mathrm{V}^n & \xrightarrow{\mathrm{sup}^n} & \mathrm{V}^n \\ & {\scriptstyle \pi_0} \searrow \quad \swarrow {\scriptstyle \mathrm{El}^n} & \\ & U & \end{array}$$

Therefore the type of fibers of π_0 over A, which is $A \hookrightarrow_{n-1} \mathrm{V}^n$, is equivalent to the type of fibers of El^n over A, which is equivalent to $\sum_{a:A} \mathrm{El}^n\ a \simeq A$ by univalence. □

The universes are cumulative both with regards to universe levels and with regards to type levels. So far we have assumed only two universes, U and Type, but assume for the next proposition a (cumulative) hierarchy of universes $U_0, U_1, \cdots, U_\ell, \cdots$ and let V^n_ℓ denote the initial algebra to the functor $\mathrm{P}^n_{U_\ell}$.

Proposition 34 ( ). *For $n : \mathbb{N}^\infty$, there is an internalisation of V^n_ℓ in $\mathrm{V}^n_{\ell+1}$.*

Proof. The map

$$\begin{aligned} &\phi : \mathrm{V}^\infty_\ell \to \mathrm{V}^\infty_{\ell+1} \\ &\phi\,(\mathrm{sup}^\infty\,(A, f)) := \mathrm{sup}^\infty\,(A, \phi \circ f) \end{aligned}$$

is an embedding, as shown in Paper I. To show that it restricts to iterative n-types, let $\mathrm{sup}^\infty\,((A, f), \_) : \mathrm{V}^n_\ell$, i.e. f is $(n-1)$-truncated and $f\,a$ is an iterative n-type for all $a : A$. Then $\phi \circ f$ is $(n-1)$-truncated as it is the composition of two $(n-1)$-truncated maps. Moreover, by the induction hypothesis $\phi\,(f\,a)$ is an iterative n-type since $f\,a$ is an iterative n-type, for all $a : A$.

The map ϕ is therefore an embedding $\mathrm{V}^n_\ell \hookrightarrow \mathrm{V}^n_{\ell+1}$. By Proposition 33 the map gives rise to an internalisation of V^n_ℓ, namely $\mathrm{sup}^{n+1}\,(\mathrm{V}^n_\ell, \phi)$. □

Proposition 35 ( ). *For $n : \mathbb{N}^\infty$, there is an embedding $\mathrm{V}^n \hookrightarrow \mathrm{V}^{n+1}$.*

Proof. This is simply the fact that an iterative n-type is also an iterative $(n+1)$-type. □

Note that there is a size issue with internalising V^n in V^{n+1}. If there was an element $v : \mathrm{V}^{n+1}$ such that $\mathrm{El}^{n+1}\ v \simeq \mathrm{V}^n$, then the type V^n would be essentially U-small, which would induce a paradox. However, assuming again a hierarchy of universes, the type V^n_ℓ can be internalised in $\mathrm{V}^{n+1}_{\ell+1}$.

The universe V^n also contains all the usual types and type formers, assuming they exist in the underlying universe U.

Proposition 36 ( ). *For $n : \mathbb{N}^\infty$, the universe V^n contains the following types and type formers:*

- *the empty type, unit type and booleans,*
- *the natural numbers,*
- Π-*types,*
- Σ-*types,*
- *coproducts and*
- *identity types.*

Proof. For the empty type, unit type and booleans, we internalise them as the elements $\emptyset$, $\{\emptyset\}_0$ and $\{\emptyset, \{\emptyset\}_0\}_0$ respectively, using Theorems 4 and 6. The natural numbers have an internalisation as they have an $(n-1)$-truncated representation by Example 1.

For Π-types and Σ-types, suppose we have an element $a : \mathrm{V}^n$ and a map $b : \mathrm{El}^n\ a \to \mathrm{V}^n$. In the first case, note that $\mathrm{graph}_{\widetilde{a}, \lambda\, i. \widetilde{(b\, i)}}$ in Lemma 12 gives an $(n-1)$-truncated representation of $\prod_{i : \mathrm{El}^n\ a} \mathrm{El}^n\,(b\, i)$. For Σ-types, we use the ordered pairing structure given by Theorem 7. The map $\lambda\,(i, j). \langle \widetilde{a}\, i, \widetilde{(b\, i)}\, j \rangle$ is an $(n-1)$-truncated representation of $\sum_{i : \mathrm{El}^n\ a} \mathrm{El}^n\,(b\, i)$.

Given $a, b : \mathrm{V}^n$, the map

$$
\begin{aligned}
&f : \mathrm{El}^n\ a + \mathrm{El}^n\ b \to \mathrm{V}^n \\
&f\,(\mathrm{inl}\ i) := \langle \emptyset, \widetilde{a}\, i \rangle \\
&f\,(\mathrm{inr}\ j) := \langle \{\emptyset\}_0, \widetilde{b}\, j \rangle
\end{aligned}
$$

is $(n-1)$-truncated. To see this, note that for $z : \mathrm{V}^n$ we have the following chain of equivalences:

$$
\begin{aligned}
\mathrm{fiber}\ f\ z &\equiv \left(\sum_{s : \mathrm{El}^n\ a + \mathrm{El}^n\ b} f\ s = z \right) && (138) \\
&\simeq \left(\sum_{i : \mathrm{El}^n\ a} \langle \emptyset, \widetilde{a}\, i \rangle = z \right) + \left(\sum_{j : \mathrm{El}^n\ b} \langle \{\emptyset\}_0, \widetilde{b}\, j \rangle = z \right) && (139) \\
&\simeq \left(\sum_{((s,t),p) : \mathrm{fiber}\ \langle -,- \rangle\ z} (\mathrm{fiber}\ \widetilde{a}\, t) \times (s = \emptyset) \right) && (140) \\
&\quad + \left(\sum_{((s,t),p) : \mathrm{fiber}\ \langle -,- \rangle\ z} \left(\mathrm{fiber}\ \widetilde{b}\, t\right) \times (s = \{\emptyset\}_0) \right)
\end{aligned}
$$

The last type is $(n-1)$-truncated as all the components are $(n-1)$-truncated and the two summands are mutually exclusive since $\emptyset \neq \{\emptyset\}_0$. The map f therefore gives an internalisation of the coproduct $\mathrm{El}^n\ a + \mathrm{El}^n\ b$. For identity types we note that the map $\lambda\, p. \emptyset : a = b \to \mathrm{V}^n$ is $(n-1)$-truncated as it is a function from an $(n-1)$-type into an n-type. So the type $a = b$ is internalisable. □

Note that all the codes in V^n are constructed using $\sup^n$. Therefore, the decoding of a code is definitionally equal to the type being encoded.

As V^n is an n-type universe of n-types, a natural question to ask is which n-types lie in the universe. For V^1 we have seen that the classifying space of any group can be internalised. In particular, we have an internalisation of the higher inductive type of the circle in V^1. Using the same idea, we can internalise any n-type in V^{n+1}.

Proposition 37 ( ). *For $n : \mathbb{N}^\infty$, any U-small n-type can be internalised in V^{n+1}.*

Proof. Let $A : U$ be an n-type. Let $f : A \to \mathrm{V}^{n+1}$ be the constant function sending any element to $\emptyset$. This is an n-truncated map as the domain is an n-type and the codomain is an $(n+1)$-type. The element $\sup^{n+1}(A, f)$ is thus an internalisation of A in V^{n+1}. □

For V^0, the statement that any mere set has an internalisation is essentially the axiom of wellfounded materialisation (Shulman, 2010). The statement that any n-type can be internalised in V^n can thus be thought of as a higher level generalisation of wellfounded materialisation. It amounts to being able to equip every n-type with some higher iterative structure.

Proposition 38. *For $n : \mathbb{N}^\infty$, the universe V^n is* ***not*** *a univalent universe.*

Proof. The quickest way to see this is to note that V^n contains (at least) two distinct internalisations of the unit type: $\{\emptyset\}_0$ and $\{\{\emptyset\}_0\}_0$ (but there are of course many more). As these both decode to the unit type but are not equal, univalence fails. □

From the reasoning above it follows that V^n does not even have partial univalence (Sattler and Vezzosi, 2020) (univalence restricted to k-types, for some k). The reason why univalence fails is essentially that the decoding of an element in V^n only returns the indexing type of the children to the root, seeing the element as a tree, regardless of the rest of the tree. Thus, several trees can internalise the same type, but be distinct as trees, i.e. as elements of the universe.

8 Conclusion and future work

In this paper we defined the concept of an $\in$-structure and gave generalisations of the axioms of constructive set theory to higher level structures. As instances of such higher level $\in$-structures, we generalised the construction of the type of iterative sets to obtain the initial algebras, V^n, to the functors P^n_U. These were shown to model the $\in$-structure properties at the same level or lower. Moreover, V^n was shown to be an n-type universe of n-types with definitional decoding.

8.1 Related work

Gallozzi Gallozzi, 2019 constructs a family of interpretations of set theory in Homotopy Type Theory indexed on two type levels, k and h. His interpretation is a generalisation of Aczel's model. In one direction, he generalises by taking as the type of sets Aczel's W-type, but over the small universe of k-types, rather than the whole universe U. In the other direction, he uses h-truncated Σ-types, while Aczel uses untruncated Σ-types. Gallozzi then shows that this models Aczel's CZF Aczel, 1978 and Myhill's CST Myhill, 1975.

These models are setoid models—equality is not interpreted as the identity type, as opposed to our models. They are not $\in$-structures in our sense either, in that our version of extensionality does not hold. Of course, he shows that his interpretation of extensionality holds. The k-level W-type used by Gallozzi is a $(k+1)$-type, and it is a subtype of our type V^{k+1}, so any set in that universe is also a set in our universe.

The HoTT Book model of set theory is equivalent to the iterative sets model. It is not clear, however, how to generalise that construction as we have done with the iterative sets, as this would require more complex higher inductive types.

8.2 Future work

As we have seen, the universe V^n embeds into the universe V^{n+1}. So the higher universes contain the types of the lower universes. However, it remains to see which new types appear at each level. For instance, we showed that the higher inductive type of the circle, and more generally the classifying space of any group, lies in V^1. For the universe V^n one would like to construct an internalisation of (at least some) proper n-types. This amounts to giving the type some kind of higher iterative structure, which would be an interesting direction for further study.

References

Aczel, Peter (1978). "The Type Theoretic Interpretation of Constructive Set Theory". In: *Logic Colloquium '77*. Ed. by A. MacIntyre, L. Pacholski, and J. Paris. North–Holland, Amsterdam-New York, pp. 55–66.

Bell, John L. (2011). *Set theory: Boolean-valued models and independence proofs*. 3rd. Vol. 47. Oxford Logic Guides. Oxford University Press. ISBN: 978-0198568520.

Bezem, Marc et al. (2022). *Symmetry*. An undergraduate textbook written in univalent style, emphasizing symmetry in logic. Licensed under CC-BY-SA-4.0. Online: GitHub. URL: `https://github.com/UniMath/SymmetryBook`.

Christensen, J. Daniel (Sept. 2021). *Non-accessible localizations*. DOI: `10.48550/arXiv.2109.06670`. URL: `http://arxiv.org/abs/2109.06670`.

Gallozzi, Cesare (2019). "Homotopy type-theoretic interpretations of constructive set theories". In: *Mathematical Structures in Computer Science*, pp. 1–32. DOI: `10.1017/S0960129519000148`.

Gylterud, Håkon Robbestad (2018). "From multisets to sets in homotopy type theory". In: *The Journal of Symbolic Logic* 83.3, pp. 1132–1146. DOI: `10.1017/jsl.2017.84`.

— (2019). "Multisets in type theory". In: *Mathematical Proceedings of the Cambridge Philosophical Society*, pp. 1–18. DOI: `10.1017/S0305004119000045`.

Gylterud, Håkon Robbestad and Elisabeth Bonnevier (2020). "Non-wellfounded sets in HoTT". In: *arXiv preprint arXiv:2001.06696v2*.

Joyal, André and Ieke Moerdijk (Sept. 14, 1995). *Algebraic Set Theory*. Google-Books-ID: E0Xhn32xzZAC. Cambridge University Press. 136 pp. ISBN: 978-0-521-55830-3.

Kraus, Nicolai (July 2015). "Truncation levels in homotopy type theory". PhD thesis.

Myhill, John (1975). "Constructive Set Theory". In: *The Journal of Symbolic Logic* 40.3. Publisher: Association for Symbolic Logic, pp. 347–382. ISSN: 0022-4812. DOI: `10.2307/2272159`. URL: `https://www.jstor.org/stable/2272159`.

Osius, Gerhard (1974). "Categorical set theory: a characterization of the category of sets". In: *Journal of Pure and Applied Algebra* 4.1, pp. 79–119.

Rieger, Ladislav (1957). "A contribution to Gödel's axiomatic set theory, I". eng. In: *Czechoslovak Mathematical Journal* 07.3. Publisher: Institute of Mathematics, Academy of Sciences of the Czech Republic, pp. 323–357. ISSN: 0011-4642. URL: `https://eudml.org/doc/11888`.

Rijke, Egbert (2017). "The join construction". In: *ArXiv e-prints*. arXiv: `1701.07538 [math.CT]`.

Rijke, Egbert et al. (2024). *The agda-unimath library.* URL: https://github.com/UniMath/agda-unimath/.

Sattler, Christian and Andrea Vezzosi (July 8, 2020). "Partial Univalence in n-truncated Type Theory". In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '20. New York, NY, USA: Association for Computing Machinery, pp. 807–819. ISBN: 978-1-4503-7104-9. DOI: 10.1145/3373718.3394759. URL: https://doi.org/10.1145/3373718.3394759.

Shulman, Michael (Apr. 2010). "Stack semantics and the comparison of material and structural set theories". In: DOI: 10.48550/arXiv.1004.3802. URL: https://arxiv.org/abs/1004.3802.

Taylor, Paul (2023). *Well founded coalgebras and recursion.* URL: http://www.paultaylor.eu/ordinals/welfcr.pdf.

The Agda development team (2024). *Agda.* URL: https://github.com/agda/agda.

The Univalent Foundations Program (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: homotopytypetheory.org. URL: http://homotopytypetheory.org/book.

Paper III

Terminal Coalgebras and Non-wellfounded Sets in Homotopy Type Theory

Håkon Robbestad Gylterud, *Elisabeth Stenholm* and Niccolò Veltri

Abstract

Non-wellfounded material sets have previously been modeled in Martin-Löf type theory by Lindström using setoids (Lindström, 1989). In this paper we construct models of non-wellfounded material sets in Homotopy Type Theory (HoTT) where equality is interpreted as the identity type. The first model satisfies Scott's Anti-Foundation Axiom (SAFA) and dualises the construction of iterative sets. The second model satisfies Aczel's Anti-Foundation Axiom (AFA), and is constructed by adaptation of Aczel–Mendler's terminal coalgebra theorem to type theory, which requires propositional resizing.

In a bid to extend coalgebraic theory and anti-foundation axioms to higher type levels, we formulate generalisations of AFA and SAFA, and construct a hierarchy of models which satisfies the SAFA generalisations. These generalisations build on the framework of Univalent Material Set Theory, previously developed by two of the authors (Paper II).

Since the model constructions are based on M-types, the paper also includes a characterisation of the identity type of M-types as indexed M-types.

Our results are formalised in the proof-assistant Agda.

1 Introduction

In non-wellfounded set theory, the concept of a material set is expanded beyond the cumulative hierarchy. The allowance for non-wellfounded sets, such as the quine atom $q := \{q\}$, makes it easier to study circular phenomena and structures such as transitions systems and streams. In what follows, we seek to integrate non-wellfounded set theory into Homotopy Type Theory (HoTT)—a relatively new framework for mathematics, which supports higher dimensional structures as first-class citizens with the powerful *Univalence Axiom* and higher inductive types. Our aim is to take classical notions from universal coalgebra and non-wellfounded set theory and extend them to higher-dimensional structures.

Wellfounded material set theory has been studied in Martin-Löf type theory since 1978 with the introduction of Aczel's setoid model of Constructive Zermelo–Fraenkel set theory (CZF) (Aczel, 1978). Non-wellfounded set theory in Martin-Löf type theory was studied already in 1989 by Lindström, when she constructed a setoid based model of constructive ZF^- (ZF without the axiom of foundation) + Aczel's anti-foundation axiom (AFA) (Lindström, 1989).

These two models of material set theory were, as mentioned, setoid based, meaning that equality was interpreted as a binary relation distinct from Martin-Löf's identity type. This was rectified in the model presented in the HoTT Book (The Univalent Foundations Program, 2013), which constructed a model of wellfounded set theory using a higher inductive type, in which equality was interpreted as the identity type.

Gylterud (2018) then constructed a model, $(V^0, \in)$, equivalent to the HoTT Book model, but which did not require higher inductive types for the construction. This construction and its properties have been further explored in Paper I. One important aspect of V^0 is its role as the initial algebra of the U-restricted powerset functor $\mathrm{P}^0_U : \mathrm{Type} \to \mathrm{Type}$, which maps $X \mapsto \sum_{A:U} A \hookrightarrow X$. One of the ideas we explore here is to construct the terminal coalgebra for P^0_U to use as a model of non-wellfounded sets, filling out the question mark in the table below.

	Setoid	Identity type
Foundation	Aczel 1978	Gylterud 2018
Anti-foundation	Lindström 1989	?

We show that the terminal coalgebra for P^0_U would indeed yield a model of Aczel's anti-foundation axiom (AFA):

> **AFA**: Any (directed) graph can be uniquely decorated with sets such that elementhood between the sets coincides with edges in the graph.

As we shall see, the classical Aczel–Mendler construction (Aczel and Mendler, 1989) can be adapted to the HoTT setting and constructs a terminal coalgebra for P^0_U, but it requires propositional resizing—an impredicative axiom.

In addition to the Aczel–Mendler construction, we provide a new construction, V^0_∞, of non-wellfounded sets in HoTT which dualises the construction of V^0, but which surprisingly *does not* yield a terminal coalgebra for P^0_U. It is a third fixed point—neither initial nor terminal. This type is a model of Scott's anti-foundation axiom (SAFA), an alternative anti-foundation axiom to AFA. SAFA is based on the concept of *Scott extensionality*. A graph is Scott extensional if equality of nodes in the graph coincides with isomorphism of unfolding trees (more on that later).

> **SAFA**: Every Scott extensional graph can be injectively decorated with sets, and the graph of all sets with edges symbolising elementhood is Scott extensional.

We also explore possible extensions of anti-foundation axioms to higher types. In HoTT, there is a fundamental notion of n-type arising from the iterative application of identity types. The 0-types are the sets, where much of classical mathematics takes place. But even for down-to-earth mathematics such as combinatorics, higher types can play a role. Groupoids, that is 1-types, show up for instance in Joyal's theory of combinatorial species. We therefore propose generalisations of both AFA and SAFA to n-types, and our model construction V_∞^0 is presented as a general construction, V_∞^n, which then satisfies k-SAFA for each $k \leq n$.

The construction of V_∞^n is based on M-types. These types were constructed in HoTT in "Non-wellfounded trees in Homotopy Type Theory" (Ahrens, Capriotti, and Spadotti, 2015). We provide some further general results about M-types. In particular, we fully characterise the identity types of M-types as indexed M-types.

1.1 Contributions

The main contributions of this paper are the following:

- Construction of a fixed point for each of the non-polynomial functors $X \mapsto (\sum_{A:U} A \hookrightarrow_n X)$, which is distinct from both the initial algebra and the terminal coalgebra.
- Adapting Aczel–Mendler's construction to type theory, assuming propositional resizing.
- Applying the HoTT version of Aczel–Mendler to construct a terminal coalgebra for the U-restricted powerset functor.
- Demonstration that this terminal coalgebra yields a model of set theory incorporating Aczel's anti-foundation axiom, with the identity type serving as equality.
- Show that Scott's anti-foundation axiom has a constructive model in HoTT, with the identity type as equality.
- A characterisation of the identity types of M-types as indexed M-types.

1.2 Formalisation

The results in this paper has been formalised in the Agda proof assistant (The Agda development team, 2024a). Our formalisation builds on the agda-unimath library (Rijke et al., 2024), which is an extensive library of formalised mathematics from the univalent point of view. The results in Section 7 are formalised using Cubical Agda—an extension of Agda with features from cubical type theory (The Agda development team, 2024b). But as the proofs in this article demonstrate, they can be carried out in the same framework as the rest of the article.

The formalisation of Sections 1–5 in this paper has been included in a larger library on material set theory in HoTT, which can be found here: `https://git.app.uib.no/hott/hott-set-theory`. As the formalisation is structured slightly differently than the outline of this paper, there are a few results which do not have an exact counterpart in the code base. All these results are simple corollaries or variations of results which have been formalised. Importantly, all the main results are fully formalised.

The formalisation of Section 7 can be found at: `https://github.com/niccoloveltri/aczel-mendler`. Throughout the paper there will also be clickable links to specific lines of Agda code corresponding to a given result. These will be shown as the Agda logo .

1.3 Notations and conventions

The notation throughout the paper will follow common practice in HoTT. We use some categorical notations, including coercion from categories to their types of objects: We take $x : C$ to mean $x : \mathrm{Ob}_C$.

The ambient type theory is assumed to contain M-types. This is not a very restrictive assumption as it has been shown by Ahrens, Capriotti, and Spadotti (2015) that M-types can be constructed from inductive types in HoTT.

Convention Throughout the paper we will take the type of truncation levels to be the type $\mathbb{N}_{-2}^{\infty}$, i.e. the usual truncation levels, but with a supremum, ∞, such that $\|P\|_\infty \equiv P$. Moreover, for computations we have $\infty - 1 = \infty = \infty + 1$. We will also use $\mathbb{N}_{-1}^{\infty}$ for the subset of truncation levels excluding -2, and $\mathbb{N}_{-2}$ and $\mathbb{N}_{-1}$ for the ones further excluding ∞.

We will also take liberties with coercions of subtypes into their ambient type to enhance the readability of theorems and proofs. Since the results are all formalised in Agda, we allow ourselves this simplification without worry of any loss of rigour. The same goes for using some essentially small types in some places instead of their small replacements.

2 Coalgebras on Type

The notion of an F-coalgebra is usually formulated for functors on categories. In HoTT, there is a whole spectrum of notions of categories depending on how much saturation (or univalence) one wants to require and whether one wants to restrict the type level of homomorphisms or objects or both. At one end of this spectrum we find the *wild categories*, where objects and homomorphisms can be of any type level and no saturation is required.

In this setting we will be interested in wild functors $\mathrm{F} : \mathrm{Type} \to \mathrm{Type}$, which is an operation on types with an action $(X \to Y) \to (\mathrm{F}\ X \to \mathrm{F}\ X)$, which we denote by juxtaposition $\mathrm{F}\ f$, which preserves composition and the identity function.

An F-coalgebra is a pair (A, α), where $A : \mathrm{Type}$ and $\alpha : A \to \mathrm{F}\ A$. As is usual in universal coalgebra, we require no comonadicity of F nor coassociativity of α (i.e. α being algebra for F as a comonad). We will also here settle on some notation for standard notions of universal coalgebra, adapted to the HoTT setting.

Definition 1 (The wild category of F-coalgebras)**.** Let $\mathrm{F} : \mathrm{Type} \to \mathrm{Type}$ be a wild endofunctor on the wild category of types and functions. The **wild category of F-coalgebras**, denoted $\mathrm{F}\text{-Coalg}$, is the wild category for which

- The type of objects is the type of F-coalgebras:

$$\sum_{A:\mathrm{Type}} A \to \mathrm{F}\ A.$$

- Given two coalgebras (A, α) and (B, β), the type of F-coalgebra homomorphisms from (A, α) to (B, β) is the type

$$\sum_{f:A \to B} \beta \circ f \sim \mathrm{F}\ f \circ \alpha.$$

- The underlying map of the identity homomorphism on (A, α) is id_A and the homotopy is constructed as usual by the functoriality of F on the identity homomorphism.
- Given a homomorphism (f, H) from (A, α) to (B, β) and (g, K) from (B, β) to (C, γ), the underlying map of their composition is given by $g \circ f$, and the homotopy is the usual composition of squares together with the functoriality of F on composition.

It is important to note that since the carrier of the codomain, B, can be of any type level, the second component of $\mathrm{Hom}_{\mathrm{F}\text{-Coalg}}\,(A, \alpha)\,(B, \beta)$, namely $\beta \circ f \sim \mathrm{F}\ f \circ \alpha$, is a structure, not just a property.

We will also use coalgebras for (wild) functors on indexed types. These are functorial operations $\mathrm{F} : (I \to \mathrm{Type}) \to (I \to \mathrm{Type})$ for some $I : \mathrm{Type}$. We will call these *indexed functors* and *indexed coalgebras.*

Definition 2 (The wild category of indexed F-coalgebras)**.** Given an index $I : \mathrm{Type}$, let $\mathrm{F} : (I \to \mathrm{Type}) \to (I \to \mathrm{Type})$ be a wild endofunctor on the wild category of I-indexed type families and fiberwise maps. The **wild category of I-indexed F-coalgebras**, $\mathrm{F}\text{-Coalg}$, is the wild category for which

- The type of objects is the type of I-indexed F-coalgebras:

$$\sum_{A:I\to\text{Type}} \prod_{i:I} A\,i \to \text{F}\ A\,i.$$

- Given two coalgebras $(A,\alpha),(B,\beta)$, the type of F-coalgebra homomorphisms from (A,α) to (B,β) is the type

$$\sum_{f:\prod_{i:I} Ai\to Bi} \prod_{i:I} \beta\,i \circ f\,i \sim \text{F}\ f\,i \circ \alpha\,i.$$

- The underlying map of the identity homomorphism on (A,α) is $\lambda i.\ \text{id}_{A\,i}$ and the homotopy is constructed as usual by the functoriality of F on the identity homomorphism.
- Given a homomorphism (f,H) from (A,α) to (B,β) and (g,K) from (B,β) to (C,γ), the underlying map of their composition is given by $\lambda i.\, g\,i \circ f\,i$, and the homotopy is the usual composition of squares together with the functoriality of F on composition.

Definition 3. An F-coalgebra (A,α) is **extensional** if $\alpha : A \to F\,A$ is an embedding.

Through the lens of type levels, we can also see a close connection between two important properties of coalgebras, being *terminal* and being *simple*:

Definition 4. An F-coalgebra (A,α) is **terminal** if for every other F-coalgebra, (B,β), the type of homomorphisms into (A,α), namely

$$\text{Hom}_{F\text{-Coalg}}\,(A,\alpha)\,(B,\beta)$$

is contractible.

Definition 5. An F-coalgebra (A,α) is **simple** if for every other F-coalgebra, (B,β), the type of homomorphism into (A,α), namely

$$\text{Hom}_{F\text{-Coalg}}\,(A,\alpha)\,(B,\beta)$$

is a proposition.

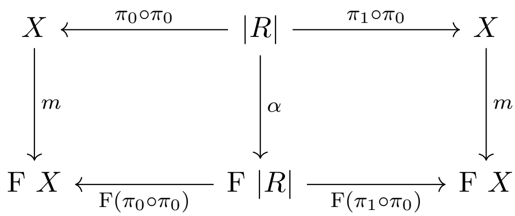
The following is immediate from the definitions:

Lemma 1. *A terminal F-coalgebra is simple.*

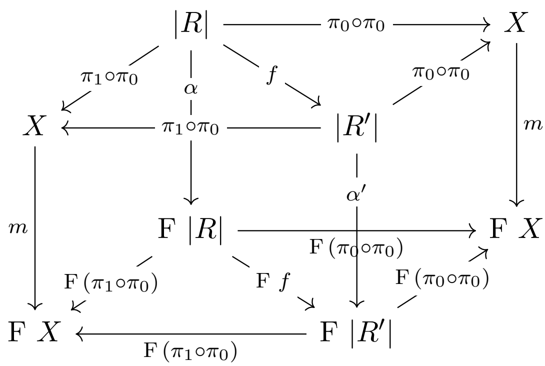
2.1 Bisimulation

Bisimulation is another central notion of coalgebra theory. In short, a bisimulation is just a span in the category of F-coalgebras, or a relation on the coalgebra that relates elements in a way compatible with the coalgebra structure. We can arrange the bisimulations on a particular F-coalgebra into a (wild) category.

Although bisimulations are essentially spans, when working with dependent types, it is also useful to think of the bisimulation as stemming from a relation $R : X \to X \to \text{Type}$. Thus, the carrier of the bisimulation is (without loss of generality) the Σ-type: $|R| := \sum_{(x,x'):X\times X} R\,x\,x'$. From this carrier, we have projections $\pi_0 \circ \pi_0 : |R| \to X$ and $\pi_1 \circ \pi_0 : |R| \to X$, which should be F-coalgebra homomorphisms.



A morphism of bisimulations can be thought of as an F-coalgebra homomorphism between the bisimulations as F-coalgebras, along with a filling of the left and right triangular prisms of the following diagram:



Definition 6 (The wild category of F-bisimulations on an F-coalgebra)**.** Let $\text{F} : \text{Type} \to \text{Type}$ be a wild endofunctor on the wild category of types and functions, and let (X, m) be an F-coalgebra. The **wild category of F-bisimulations on** (X, m), denoted $\text{F-Bisim}_{(X,m)}$, is the wild category for which

- The type of objects is the type of spans:

$$\sum_{R:X\times X\to \mathrm{Type}}\ \sum_{\alpha:|R|\to \mathrm{F}\ |R|} (m\circ\pi_0\circ\pi_0 \sim \mathrm{F}\,(\pi_0\circ\pi_0)\circ\alpha) \\ \times (m\circ\pi_1\circ\pi_0 \sim \mathrm{F}\,(\pi_1\circ\pi_0)\circ\alpha)$$

- Given two F-bisimulations $(R,(\alpha,(H_0,H_1)))$ and $(R',(\alpha',(H_0',H_1')))$, the type of F-bisimulation homomorphisms from the first to the second is

$$\sum_{(f,K):\mathrm{Hom}_{\mathrm{F\text{-}Coalg}}\,(|R|,\alpha)\,(|R'|,\alpha')} \big((\pi_0\circ\pi_0,H_0')\circ(f,K)=(\pi_0\circ\pi_0,H_0)\big) \\ \times \big((\pi_1\circ\pi_0,H_1')\circ(f,K)=(\pi_1\circ\pi_0,H_1)\big)$$

- The first component of the identity homomorphism on $(R,(\alpha,(H_0,H_1)))$ is the identity

$$\mathrm{id}:\mathrm{Hom}_{\mathrm{F\text{-}Coalg}}\,(|R|,\alpha)\,(|R|,\alpha).$$

  The higher homotopies follow from the functoriality of F.
- Given a homomorphism $((f,K),(p_0,p_1))$ from $(R,(\alpha,(H_0,H_1)))$ to $(R',(\alpha',(H_0',H_1')))$ and a homomorphism $((g,J),(q_0,q_1))$ from there to $(R'',(\alpha'',(H_0'',H_1'')))$, the underlying homomorphism of their composition is given by

$$(g,J)\circ(f,K):\mathrm{Hom}_{\mathrm{F\text{-}Coalg}}\,(|R|,\alpha)\,(|R''|,\alpha'').$$

  The higher homotopies follow from the functoriality of F.

When doing set level mathematics, a bisimulation homomorphism from (R,α) to (R',α') (the homotopies being propositions) would simply be an F-coalgebra homomorphism from the total space of the first relation to the total space of the second. But since we have no restrictions on the type levels of the carrier types, we also need coherence on the homotopies involved in the bisimulations.

In universal coalgebra, there are many equivalent formulations of being a simple F-coalgebra (Rutten, 2000). One of the equivalent formulations is that the identity bisimulation is the terminal bisimulation. The definition below is a strengthening of the classical definitions, allowing proof relevant bisimulations and coalgebras with higher homotopies.

Definition 7. Let (X,m) be an F-coalgebra. We define **the identity bisimulation**, $(=,(\alpha,(H_0,H_1)))$, on (X,m), by noting that $\pi_0\circ\pi_0:|=|\to X$ is an equivalence, and letting $\alpha\,((x,x),\mathrm{refl})=\mathrm{F}\,(\pi_0\circ\pi_0)^{-1}(m\,x)$. Likewise, H_0 and H_1 are defined by path induction.

Definition 8. Let (X, m) be an F-coalgebra. We say that (X, m) **is bisimulation simple** if $(=, (\alpha, (H_0, H_1)))$ is terminal. That is: for every other bisimulation $(R, (\alpha', (H_0', H_1')))$ on (X, m) the type of F-bisimulation homomorphisms from $(R, (\alpha', (H_0', H_1')))$ to $(=, (\alpha, (H_0, H_1)))$ is contractible.

A type family is a pair (A, P) where $A : \mathrm{Type}$ and $P : A \to \mathrm{Type}$. An equivalence of families between two families (A, P) and (B, Q) is a pair (α, σ) where $\alpha : A \simeq B$ and $\sigma : \prod_{a:A} Q(\alpha\, a) \simeq P\, a$. By univalence, we can transfer results about one family along such an equivalence to a result about the other family. We will now use this to relate equality of homomorphisms with bisimulation homomorphisms into the identity bisimulation.

Lemma 2 ( ). *Let* (X, m) *be an* F*-coalgebra. There is an equivalence of type families between bisimulations with homomorphisms into* $(=, (\alpha, (H_0, H_1)))$ *and equality of pairs of homomorphisms into* (X, m)*. That is, there is an equivalence of families between*

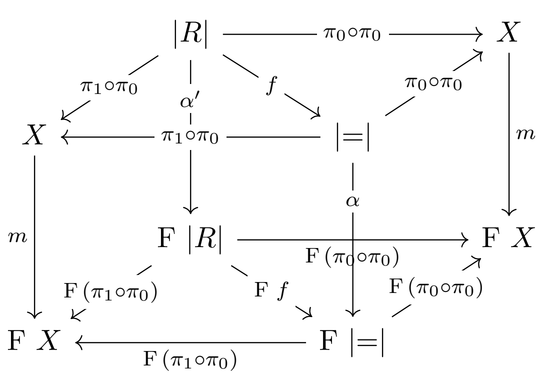
$$(\mathrm{F\text{-}Bisim}_{(X,m)}, \mathrm{Hom}_{\mathrm{F\text{-}Bisim}_{(X,m)}}\ \_\ (=, (\alpha, (H_0, H_1))))$$

and the pair consisting of

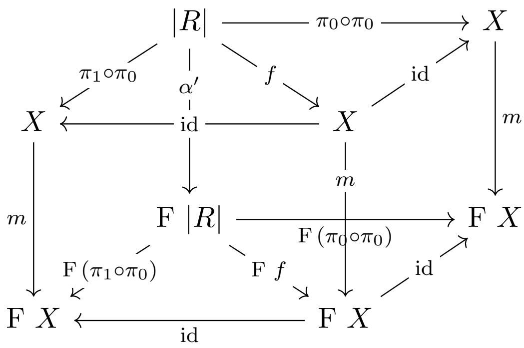
$$\sum_{(Y,n):\mathrm{F\text{-}Coalg}} (\mathrm{Hom}_{\mathrm{F\text{-}Coalg}}\,(Y, n)\,(X, m)) \times (\mathrm{Hom}_{\mathrm{F\text{-}Coalg}}\,(Y, n)\,(X, m))$$

and $\lambda(\_, (f, g)).f = g$.

Proof. Let $(R, (\alpha', (H_0', H_1')))$ be an F-coalgebra bisimulation on (X, m). By definition $(|R|, \alpha')$ is an F-coalgebra, and $(\pi_0 \circ \pi_0, H_0')$ and $(\pi_1 \circ \pi_0, H_1')$ are homomorphisms. The type of bisimulation homomorphisms from $(R, (\alpha', (H_0', H_1')))$ to $(=, (\alpha, (H_0, H_1)))$ is the type of fillings of the following diagram:



Note that the projection $\pi_0 \circ \pi_0 : |=| \to X$ is an equivalence. Applying this equivalence and using the fact that $\mathrm{F}\,(\pi_0 \circ \pi_0)\,(\alpha\,((x, x), \mathrm{refl})) = m\, x$, we see that this is equivalent to having a filling of the following diagram:



Such a filling is equivalent to $(\pi_0 \circ \pi_0, H'_0) = (\pi_1 \circ \pi_0, H'_1)$.

Likewise, any pair of parallel coalgebra homomorphisms (f, H) and (g, I) from (Y, n) to (X, m) gives rise to a bisimulation by letting $R\,x\,x' = \sum_{y:Y} (f\,y = x) \times (g\,y = x')$. One can check that going back and forth yields equivalent results. Thus, by univalence we have an equivalence of type families. □

Lemma 3 ( ). *An* F*-coalgebra is bisimulation simple if and only if it is simple.*

Proof. By Lemma 2, equality between homomorphisms into (X, m) is equivalent to bisimulation homomorphisms into $(=, (\alpha, (H_0, H_1)))$. Thus, if equality between homomorphisms is contractible (since $\mathrm{Hom}_{\mathrm{F\text{-}Coalg}}\,(Y, n)\,(X, m)$ is a proposition), then $(=, (\alpha, (H_0, H_1)))$ terminal, and vice versa. □

Corollary 1. *Let* (X, m) *be a terminal* F*-coalgebra. Then* (X, m) *is bisimulation simple, i.e.* $(=, (\alpha, (H_0, H_1)))$ *is the terminal bisimulation.*

2.2 Coalgebraic view of set theory

There is a coalgebraic viewpoint of material set theory, where one replaces the usual $\in$-relation on V with a coalgebra structure $V \to P(V)$ in the category of classes and class functors. The functor P is the powerset functor on classes which assigns to each class the class of subsets of the class. The axiom of foundation says that V is the initial P-algebra, while Aczel's anti-foundation axiom says that V is the terminal coalgebra. Other P-coalgebras are what is known in set theory as *set-like* models of set theory, and the Mostowski collapsing theorem can be framed in these terms. See for instance Paul Taylor's work on these topics (Taylor, 2023).

In Paper II, two of the authors of the current paper developed this coalgebraic viewpoint of material set theory inside HoTT, generalising it from sets to types of arbitrary type levels. Since the models developed later use this framework, we will quickly revisit the central definitions here.

The powerset functor on classes has a close correspondent in HoTT, namely the U-restricted powerset functor:

$$\begin{aligned} \mathrm{P}^0_U &: \mathrm{Type} \to \mathrm{Type} \\ \mathrm{P}^0_U\ X &:= \sum_{A:U} A \hookrightarrow X. \end{aligned}$$

The functorial action of P^0_U is taking the forward image along the function:

$$\mathrm{P}^0_U\, f\,(A, v) = (\mathrm{image}(f \circ v), \mathrm{incl}(f \circ v)).$$

By applying the type theoretic replacement principle (Rijke, 2017), the image lands in U (and thus the functorial action is well-defined) if the codomain of f is locally U-small. We will therefore restrict the application of this function to locally small types.

This notion of powerset is different from the one attained by regarding subtypes as maps into the type of U-small propositions. The two notions coincide on types in U, but differ on large types. In particular, $X \mapsto (X \to \mathrm{hProp}_U)$ cannot have a fixed point, due to Cantor's paradox. There is however no such obstacle for P^0_U, which is already known to have an initial algebra. As we shall see later in this article, it also has a terminal coalgebra, assuming propositional resizing, and a third fixed point (without assuming any resizing). All fixed points are extensional coalgebras, which means that they model the set theoretic extensionality axiom.

In univalent material set theory, one lifts the requirement of having to deal only with subtypes, and generalises to coalgebras for the polynomial functor P^∞_U:

$$\begin{aligned} \mathrm{P}^\infty_U &: \mathrm{Type} \to \mathrm{Type} \\ \mathrm{P}^\infty_U\ X &:= \sum_{A:U} A \to X. \end{aligned}$$

The functorial action for P^∞_U is simply postcomposition:

$$\mathrm{P}^\infty_U\, f(A, v) = (A, f \circ v).$$

Extensional coalgebras for this functor correspond to what are called $\in$-structures in univalent material set theory. There is also a hierarchy of functors between P^0_U and P^∞_U, where we restrict to n-truncated maps:

$$\begin{aligned} \mathrm{P}^{n+1}_U &: \mathrm{Type} \to \mathrm{Type} \\ \mathrm{P}^{n+1}_U\ X &:= \sum_{A:U} A \hookrightarrow_n X. \end{aligned}$$

The subscripted hooked arrow, $A \hookrightarrow_n X$, denotes an n-truncated function $A \to X$. The n here ranges from -1 to ∞, so that P^n_U is defined for all n from 0 to ∞. The type $\mathrm{P}^1_U\, X$, for instance, is the type of coverings of X.

The functorial action on P^n_U is taking n-images of the composition:

$$\mathrm{P}^n_U\, f(A, v) = (\mathrm{image}_n(f \circ v), \mathrm{incl}_n(f \circ v)).$$

Just as for P^0_U, unless $n = \infty$, this is only well-defined on locally small types.

We will almost exclusively focus on the anti-foundation axioms in this paper, but at times we will see some examples where we will use things like the empty set, $\emptyset$, and paring/finite unordered tupling. In univalent material set theory unordered tuples must be subscripted with their type level. We will only use type level 0 and type level 1 in the examples, so it is sufficient here to note that $\{a_0, \cdots, a_{n-1}\}_0$ is the usual set theoretic tupling where repetition is ignored, while $\{a_0, \cdots, a_{n-1}\}_1$ is multiset tupling where for instance $\emptyset \in \{\emptyset, \emptyset\}$ becomes a type with two elements. There is also the notion of ordered pairing, but it is uniform in type level and consists of a choice of embedding $\langle -, - \rangle : V \times V \hookrightarrow V$. See Paper II for details.

Notation: As we do not work with several universes in this article, we will often suppress mention of U in P^n_U and simply write P^n.

Since, we will use it already in the definition of the anti-foundation axioms, we will now take the opportunity to introduce the terminal coalgebra of P^∞ which we will call V^∞_∞:

$$\mathrm{V}^\infty_\infty := \mathbb{M}_{A:U} A.$$

This M-type comes equipped with a coalgebra structure $\mathrm{desup}^\infty : \mathrm{V}^\infty_\infty \to \mathrm{P}^\infty\, \mathrm{V}^\infty_\infty$, which is an equivalence. Let $\sup^\infty : \mathrm{P}^\infty\, \mathrm{V}^\infty_\infty \to \mathrm{V}^\infty_\infty$ denote the inverse of desup^∞. For any other P^∞-coalgebra, (X, m) there is a unique coalgebra homomorphism $\mathrm{corec}^\infty\,(X, m) : (X, m) \to (\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)$. We will sometimes suppress the coalgebra (X, m) and only write corec^∞, when the coalgebra is clear from the context.

3 The identity type of an M-type

The M-types are a class of coinductive types, dual to the inductive W-types. Intuitively, while the elements of W-types are well-founded trees with specified branching types, the M-types are the types of all trees with that branching type. Formally, each M-type is the terminal coalgebra of a polynomial functor which specifies the branching type. A polynomial functor is one which is induced by a container (Abbot, Altenkirch, and Ghani, 2005; Altenkirch et al., 2015). Put simply, a polynomial functor Type $\to$ Type is one of the form $X \mapsto \sum_{a:A} B\, a \to X$, for some A : Type and $B : A \to$ Type. The data A,B is called a container and denoted $A \triangleleft B$. The functor $X \mapsto \sum_{a:A} B\, a \to X$, as induced by the container $A \triangleleft B$ is denoted by $[\![A \triangleleft B]\!] : \mathrm{Type} \to \mathrm{Type}$. The M-type $\mathrm{M}_{a:A} B\, a : \mathrm{Type}$ is the underlying type of the terminal coalgebra of $[\![A \triangleleft B]\!]$ and its coalgebra map is denoted:

$$\mathrm{desup}_{A,B} : \mathrm{M}_{a:A} B\, a \to [\![A \triangleleft B]\!](\mathrm{M}_{a:A} B\, a).$$

There is also an indexed version of polynomial functors, containers and M-types. The indexed versions generalise from functors $\mathrm{Type} \to \mathrm{Type}$ to functors $(I \to \mathrm{Type}) \to (J \to \mathrm{Type})$. An indexed polynomial functor maps $X \mapsto \lambda j. \sum_{a:A_j} \prod_{b:B_j a} \to X(w\, j\, b)$, for some $A : J \to \mathrm{Type}$ and $B : \prod_{j:J} A\, j \to \mathrm{Type}$ and $w : \prod_{j:J} \prod_{a:Aj} B_j a \to I$. The data A,B,w is called an indexed container* and is denoted $A \triangleleft (B, w)$. The induced polynomial functor is denoted $[\![A \triangleleft (B, w)]\!] : (I \to \mathrm{Type}) \to (J \to \mathrm{Type})$. The indexed M-types are the terminal coalgebras for indexed polynomial endofunctors, i.e. when $I = J$.

Throughout the rest of this section, let $A \triangleleft B$ be a container. For convenience, we introduce some notation for $[\![A \triangleleft B]\!]$-coalgebras. This notation goes back to Aczel (1978), where it was applied to its prototypical W-type, but we will use it for coalgebras in general.

Notation: Given $m : X \to [\![A \triangleleft B]\!]\, X$, and $x : X$ we will denote by $\overline{x} : A$ and $\widetilde{x} : B\, \overline{x} \to X$ the unique elements defined by $m\, x = (\overline{x}, \widetilde{x})$, that is $\overline{x} := \pi_0\, (m\, x)$ and $\widetilde{x} := \pi_1\, (m\, x)$. This notation suppresses the map m, but it should be clear from the context which map the notation refers to, whenever it is used. This notation will also be used for large Type coalgebras $m : X \to \sum_{I:\mathrm{Type}} I \to X$.

The identity type of a W-type can be characterised inductively (Gylterud, 2019). For elements $x, y : \mathrm{W}_{a:A}\, B\, a$ there is an equivalence:

$$(x = y) \simeq \sum_{p:\overline{x}=\overline{y}} \prod_{b:B\,\overline{x}} \widetilde{x}\, b = \widetilde{y}\, (\mathrm{tr}_p^B\, b).$$

The goal of this section is to give a similar characterisation of the identity type of M-types: The identity type between two elements of an M-type is an indexed M-type (Theorem 1). This characterisation is slightly more involved than the one for W-types, which was straightforward induction, and goes through some results of bisimulation.

3.1 Characterisation of bisimulations of polynomial functors and the identity type of M-types

We will now characterise the identity type of an M-type as an indexed M-type. This result is not surprising, but is very useful for working with M-types

*Note that what we here call *indexed container* is what Altenkirch et al. (2015) call a *doubly indexed container*, which is *not* the same as what they call indexed containers.

in HoTT. When we later construct a model of Scott's non-wellfounded sets, this characterisation is critical to prove local smallness of the model. Furthermore, the characterisation of the identity type follows from a characterisation of bisimulations of polynomial functors as coalgebras for a related indexed polynomial functor.

Definition 9 ( ). Given an $[\![A \lhd B]\!]$-coalgebra (X, m), we define the $(X \times X)$-indexed polynomial functor

$$\begin{aligned}
&\mathrm{E}_{(X,m)} : (X \times X \to \mathrm{Type}) \to (X \times X \to \mathrm{Type}) \\
&\mathrm{E}_{(X,m)}\ R\,(x, y) := \sum_{p : \overline{x} = \overline{y}} \prod_{b : B\,\overline{x}} R\,(\widetilde{x}\,b, \widetilde{y}\,(\mathrm{tr}^B_p\,b)).
\end{aligned}$$

The functorial action is postcomposition on the second component.

Note that the identity type is an $\mathrm{E}_{(X,m)}$-coalgebra, for any pair (X, m). Specifically, define the following map by path induction:

$$\begin{aligned}
&\gamma : \prod_{(x,y) : X \times X} x = y \to \mathrm{E}_{(X,m)}\,(=)\,(x, y) \\
&\gamma\,(x, x)\ \mathrm{refl} := (\mathrm{refl}, \text{refl-htpy}).
\end{aligned}$$

As an intermediate step towards showing equivalence of E-coalgebras and $[\![A \lhd B]\!]$-bisimulations, it will also be helpful to define the following $(X \times X)$-indexed polynomial functor on (X, m):

Definition 10. Given an $[\![A \lhd B]\!]$-coalgebra (X, m), we define the $(X \times X)$-indexed polynomial functor

$$\begin{aligned}
&\mathrm{D}_{(X,m)} : (X \times X \to \mathrm{Type}) \to (X \times X \to \mathrm{Type}) \\
&\mathrm{D}_{(X,m)}\ R\,(x, y) := \\
&\quad \sum_{(a,\phi) : [\![A \lhd B]\!]\,|R|} (m\,x = [\![A \lhd B]\!]\,(\pi_0 \circ \pi_0)\,(a, \phi)) \\
&\qquad\qquad \times (m\,y = [\![A \lhd B]\!]\,(\pi_1 \circ \pi_0)\,(a, \phi)).
\end{aligned}$$

The functorial action sends a fiberwise map

$$g : \prod_{(x,y) : X \times X} R\,(x, y) \to R'\,(x, y)$$

to the map which acts on the first component by $[\![A \lhd B]\!]\,(\mathrm{tot}\ g)$.

Intuitively, the operations $\mathrm{E}_{(X,m)}$ and $\mathrm{D}_{(X,m)}$ both unfold a relation one step as though it was a bisimulation. The difference is that D uses the realisation of the polynomial, while E uses the polynomial directly. But they are in fact equivalent:

Lemma 4 ( ). *Given an* $[\![A \lhd B]\!]$*-coalgebra* (X, m) *and a relation* $R : X \times X \to \mathrm{Type}$*, for any pair* $(x, y) : X \times X$ *we have a natural equivalence*

$$\mathrm{E}_{(X,m)}\ R\,(x,y) \simeq \mathrm{D}_{(X,m)}\ R\,(x,y),$$

which maps $(p, \sigma) : \mathrm{E}_{(X,m)}\ R\,(x,y)$ *to the element*

$$(\overline{x}, \lambda\, b.((\widetilde{x}\, b, \widetilde{y}\,(\mathrm{tr}^B_p\, b)), \sigma\, b)) : [\![A \lhd B]\!]\, |R|.$$

Proof. We have the following chain of equivalences:

$$\begin{aligned}
\mathrm{E}_{(X,m)}\ R\,(x,y) &\simeq \sum_{p:\overline{x}=\overline{y}}\ \sum_{\phi_1 : B\,\overline{x}\to X} (\widetilde{y} \circ \mathrm{tr}^B_p = \phi_1) \times \left(\prod_{b:B\,\overline{x}} R\,(\widetilde{x}\, b, \phi_1\, b)\right) && (1)\\
&\simeq \sum_{\phi_1 : B\,\overline{x}\to X} (m\, y = (\overline{x}, \phi_1)) \times \left(\prod_{b:B\,\overline{x}} R\,(\widetilde{x}\, b, \phi_1\, b)\right) && (2)\\
&\simeq \sum_{a:A}\ \sum_{\phi_0, \phi_1 : B\, a\to X} (m\, x = (a, \phi_0)) \times (m\, y = (a, \phi_1)) && (3)\\
&\qquad\qquad \times \left(\prod_{b:B\, a} R\,(\phi_0\, b, \phi_1\, b)\right)\\
&\simeq \mathrm{D}_{(X,m)}\ R\,(x,y). && (4)
\end{aligned}$$

By chasing $(p, \sigma) : \mathrm{E}_{(X,m)}\ R\,(x,y)$ along the equivalences one sees that it is mapped as stated. □

Proposition 1 ( ). *For any* $[\![A \lhd B]\!]$*-coalgebra* (X, m) *there is an equivalence of types*

$$\mathrm{E}_{(X,m)}\text{-Coalg} \simeq [\![A \lhd B]\!]\text{-Bisim}_{(X,m)},$$

which sends a map $f : \prod_{(x,y):X\times X} R\,(x,y) \to \mathrm{E}_{(X,m)}\ R\,(x,y)$ *to the map*

$$\lambda\,((x,y), r).(\overline{x}, \lambda\, b.((\widetilde{x}\, b, \widetilde{y}\,(\mathrm{tr}^B_{\pi_0\,(f\, r)}\, b)), \pi_1\,(f\, r)\, b)) : |R| \to [\![A \lhd B]\!]\, |R|.$$

Proof. Given a relation $R : X \times X \to \mathrm{Type}$, by Lemma 4 there is an equivalence

$$\begin{aligned}
&\prod_{(x,y):X\times X} R\,(x,y) \to \mathrm{E}_{(X,m)}\ R\,(x,y)\\
&\quad\simeq \prod_{(x,y):X\times X} R\,(x,y) \to \mathrm{D}_{(X,m)}\ R\,(x,y) && (5)\\
&\quad\simeq \sum_{\alpha:|R|\to[\![A\lhd B]\!]\,|R|} (m \circ \pi_0 \circ \pi_0 \sim [\![A \lhd B]\!]\,(\pi_0 \circ \pi_0) \circ \alpha) && (6)\\
&\qquad\qquad \times (m \circ \pi_1 \circ \pi_0 \sim [\![A \lhd B]\!]\,(\pi_1 \circ \pi_0) \circ \alpha)\,.
\end{aligned}$$

The desired equivalence then follows by applying the equivalence above to the second component of the type $\mathrm{E}_{(X,m)}$-Coalg. By chasing $f : \prod_{(x,y):X\times X} R\,(x,y) \to \mathrm{E}_{(X,m)}\; R\,(x,y)$ along the equivalence we see that it is mapped as stated. □

Proposition 2 ( ). *Given a* $[\![A \lhd B]\!]$*-coalgebra* (X,m)*, let* e *be the equivalence given by Proposition 1. Then for any* $\mathrm{E}_{(X,m)}$*-coalgebras* (R,f) *and* (R',f') *there is an equivalence of types*

$$\mathrm{Hom}_{\mathrm{E}_{(X,m)}\text{-Coalg}}\,(R,f)\,(R',f') \simeq \mathrm{Hom}_{[\![A\lhd B]\!]\text{-Bisim}_{(X,m)}}\,(e\,(R,f))\,(e\,(R',f')).$$

Proof. Applying the equivalence e given by Proposition 1 on (R,f) and (R',f'), denote the components of the result as:

- $\alpha : |R| \to [\![A \lhd B]\!]\,|R|$,
- $\alpha' : |R'| \to [\![A \lhd B]\!]\,|R'|$,
- $H_0 : m \circ \pi_0 \circ \pi_0 \sim [\![A \lhd B]\!]\,(\pi_0 \circ \pi_0) \circ \alpha$,
- $H_0' : m \circ \pi_0 \circ \pi_0 \sim [\![A \lhd B]\!]\,(\pi_0 \circ \pi_0) \circ \alpha'$,
- $H_1 : m \circ \pi_1 \circ \pi_0 \sim [\![A \lhd B]\!]\,(\pi_1 \circ \pi_0) \circ \alpha$
- $H_1' : m \circ \pi_1 \circ \pi_0 \sim [\![A \lhd B]\!]\,(\pi_1 \circ \pi_0) \circ \alpha'$.

Let e' denote the equivalence given by Lemma 4. We have a chain of equivalences

$$\begin{aligned}
&\mathrm{Hom}_{\mathrm{E}_{(X,m)}\text{-Coalg}}\,(R,f)\,(R',f') \\
&\quad\simeq \mathrm{Hom}_{\mathrm{D}\text{-Coalg}}\,(R, e'\circ f)\,(R', e'\circ f') && (7)\\
&\quad\simeq \sum_{g:\prod_{(x,y):X\times X} R\,(x,y)\to R'\,(x,y)}\ \sum_{K:\alpha'\circ\mathrm{tot}\ g\sim[\![A\lhd B]\!]\,(\mathrm{tot}\ g)\circ\alpha} && (8)\\
&\qquad\qquad \left(H_0'\cdot K = H_0\right)\times\left(H_1'\cdot K = H_1\right)\\
&\quad\simeq \sum_{g:\prod_{(x,y):X\times X} R\,(x,y)\to R'\,(x,y)}\ \sum_{K:\alpha'\circ\mathrm{tot}\ g\sim[\![A\lhd B]\!]\,(\mathrm{tot}\ g)\circ\alpha} && (9)\\
&\qquad\qquad \left(H_0'\cdot K = \mathrm{tr}_{\mathrm{refl}}^{\lambda\,h.m\circ h\sim[\![A\lhd B]\!]\,h\circ\alpha}\,H_0\right)\\
&\qquad\qquad \times\left(H_1'\cdot K = \mathrm{tr}_{\mathrm{refl}}^{\lambda\,h.m\circ h\sim[\![A\lhd B]\!]\,h\circ\alpha}\,H_1\right)\\
&\quad\simeq \sum_{g:|R|\to|R'|}\ \sum_{p:\pi_0\circ\pi_0\circ g=\pi_0\circ\pi_0}\ \sum_{q:\pi_1\circ\pi_0\circ g=\pi_1\circ\pi_0}\ \sum_{K:\alpha'\circ\mathrm{tot}\ g\sim[\![A\lhd B]\!]\,(\mathrm{tot}\ g)\circ\alpha} && (10)\\
&\qquad\qquad \left(H_0'\cdot K = \mathrm{tr}_{p}^{\lambda\,h.m\circ h\sim[\![A\lhd B]\!]\,h\circ\alpha}\,H_0\right)\\
&\qquad\qquad \times\left(H_1'\cdot K = \mathrm{tr}_{q}^{\lambda\,h.m\circ h\sim[\![A\lhd B]\!]\,h\circ\alpha}\,H_1\right)\\
&\quad\simeq \mathrm{Hom}_{[\![A\lhd B]\!]\text{-Bisim}_{(X,m)}}\,(e\,(R,f))\,(e\,(R',f')), && (11)
\end{aligned}$$

as desired. □

Now we are ready to characterise the identity type on $\mathbb{M}_{a:A} B\, a$ as an indexed M-type.

Theorem 1 ( ). *The pair* $(=, \gamma)$ *is the terminal* $\mathrm{E}_{(\mathbb{M}_{a:A} B\, a, \mathrm{desup}_{A,B})}$-*coalgebra.*

Proof. Let e be the equivalence given by Proposition 1. Then $e\,(=, f)$ is the terminal $[\![A \triangleleft B]\!]$-coalgebra bisimulation on $(\mathbb{M}_{a:A} B\, a, \mathrm{desup}_{A,B})$, by Corollary 1. The terminality of $(=, f)$ then follows by Proposition 2. □

4 AFA and SAFA in ∈-structures

Most axioms of set theory, such as paring, union, separation and even infinity, replacement or powerset, are *set existence axioms* — they inform a student which sets they can construct within the theory. All the sets the student can construct from these axioms alone are *wellfounded*. Classically, wellfounded sets are those without an infinite membership chain:

$$a_0 \ni a_1 \ni a_2 \ni \cdots$$

Constructively, well-foundedness is instead formulated as an induction principle for ∈ or using an accessibility predicate. In both constructive and classical traditions, the most prominent theories include an axiom which states that, in fact, all sets are wellfounded. This axiom is called regularity or *foundation*. It's a standard, classical result that the axiom of foundation is independent of the rest. What is more, under certain assumptions[†] any structure defined by sets can be defined by well-founded sets.

When one removes the requirement that every material set must be well-founded, two questions arise:

1. Which non-wellfounded sets exist?
2. When are two non-wellfounded sets equal?

Anti-foundation axioms are properties of ∈-structures which give answers to these two questions. In this text we consider two such axioms. The first is Aczel's Anti-Foundation Axiom (AFA), and the second is Scott's Anti-Foundation Axiom (SAFA). These answer the question slightly differently, and in this section we will try to capture the formulation of these in a way which generalises to ∈-structures to higher type levels.

The second question arises because extensionality does not fully determine the equality between non-wellfounded sets. For instance, if two sets satisfy

[†] AC is more than sufficient, but the much milder axiom of well-founded materialisation is enough (cf. discussion in Shulman, 2010, after Lemma 6.46).

the equations $x = \{x, y\}_0$ and $y = \{x\}_0$, both $x = y$ and $x \neq y$ are possible – of course not in the same $\in$-structure. The 0-subscript on the pairing is crucial, because if we used multiset pairing, and let $x = \{x, y\}_1$, it follows that $x \neq y$, since a pair is never a singleton. This foreshadows the main thesis of this section, that the difference between Aczel's and Scott's conceptions of non-wellfounded sets is a matter of truncation level, from the perspective of HoTT.

In elementary terms, AFA states that given any graph there is a unique assignment of sets to the nodes of the graph, such that the elementhood relations between the assigned sets coincides with the edges of the graph. This both gives a way of constructing non-wellfounded sets (by giving a graph) and a way of proving equalities between non-wellfounded sets (showing that they can decorate the same node in a graph).

SAFA states that every graph where nodes have unique unfolding trees can be decorated with sets (in the same sense as in AFA) and that for sets isomorphism of unfolding trees determines equality. Additionally, the decoration is injective (since equality of nodes is determined by their unfolding trees) and is unique among such decorations. This may at the moment sound baroque and even ad hoc, but we will attempt to shed light on this.

Why all these graphs? An answer to this question comes from universal coalgebra. An $\in$-structure being, in general a coalgebra for the functor P^∞, and specifically a P^n-coalgebra in the case of n-level structures (Theorem 3 in Paper II), the non-well founded sets come from coalgebra maps into the structures. In ordinary mathematics, a graph is exactly a coalgebra $X \to \mathrm{P}^0\ X$. This emphasises looking at the out-edges from a node, and a coalgebra map into an $\in$-structure translates out-edges to elements. So, what we will call a decoration of a graph is precisely a coalgebra homomorphism from the induced coalgebra of the graph into the $\in$-structure the graph lives in.

4.1 Graphs and decorations

Usually in mathematics, we think of graphs as structures consisting of nodes and edges. However, in the formulation of the anti-foundation axioms we will work with a slightly different notion of graph, as simply a set of pairs. This leaves the domain of nodes implicit, which simplifies the definition of a decoration. Another way of thinking of it is that the domain of nodes in g is always the entirety of V.

Definition 11 ( ). In an $\in$-structure $(V, \in)$ with ordered pairing structure $\langle -, - \rangle$, an element $g : V$ is a **graph** if all its elements are pairs. That is, there is a map

$$\prod_{e:V} e \in g \to \sum_{(x,y):V\times V} e = \langle x, y\rangle,$$

or equivalently, for every $e : V$ such that $e \in g$ there are source $e : V$ and target $e : V$ such that $e = \langle \text{source } e, \text{target } e\rangle$.

Remark: The notation "source e" and "target e" suppresses mention of the specific proof element of $e \in V$ which is used to construct source e and target e. However, this is justified since $\sum_{(x,y):V\times V} e = \langle x, y\rangle$ is a proposition, and thus any choice of such proof object yields equal results.

Definition 12 ( ). Given a graph $g : V$ in a $\in$-structure $(V, \in)$ with ordered pairing structure $\langle -, -\rangle$, define the type Target g, the subtype of V consisting of targets of edges in g, by Target $g := \sum_{y:V} \exists_{x:V} \langle x, y\rangle \in g$.

Since the domain of nodes in the graph is left implicit, a decoration will be a universally defined function $d : V \to V$, where the convention is that $d\,x$ is empty if there are no edges $\langle x, y\rangle \in g$. When there is an edge $\langle x, y\rangle$ this edge should give rise to an elementhood relation $d\,y \in d\,x$. In fact, there should for every $z : V$ be an equivalence between $z \in d\,x$ and the edges in $\langle x, y\rangle \in g$ for which $z = d\,y$:

Definition 13 ( ). For $n : \mathbb{N}^{\infty}_{-1}$, an $(n+1)$**-decoration** of a graph $g : V$ in an $\in$-structure $(V, \in)$, with an ordered pairing structure $\langle -, -\rangle$, is a map $d : V \to V$ together with an element of the type

$$\prod_{x,z:V} z \in d\,x \simeq \left\| \sum_{y:V} \langle x, y\rangle \in g \times d\,y = z \right\|_n .$$

The truncation level restricts the level of $d\,x$, so that, for instance, in 0-level $\in$-structures $d\,x$ will be a set. The notion of 0-decoration is equivalent to the classical notion of decoration as a function satisfying the equation $d(x) = \{\, d(y) \mid \langle x, y\rangle \in g, \}_0$ (cf. Aczel, 1988, Chapter 1). And, in terms of univalent material set theory[‡], an n-decoration is a function satisfying the equation $d(x) = \{\, d(y) \mid \langle x, y\rangle \in g, \}_n$.

The notion of ∞-decoration is one where there is no truncation yielding simply:

$$z \in d\,x \simeq \left(\sum_{y:V} \langle x, y\rangle \in g \times d\,y = z \right) .$$

[‡]For discussion on n-truncated set comprehension and replacement, see Definitions 7 and 8 in Paper II

Intuitively it says that $d\,y$ occurs in $d\,x$ precisely as many times as $\langle x, y\rangle$ occurs in g (and that all elements of $d\,x$ are of the form $d\,y$).

There are two simple observations we can make if we know the level of the $\in$-structure.

- In an n-level $\in$-structure, an $(n+1)$-decoration is also an ∞-decoration since the type $\sum_{y:V}\langle x, y\rangle \in g \times d\,y = z$ has type level n.
- In an n-level $\in$-structure, an ∞-decoration is also n-decoration, but the opposite is not always the case. For instance, in level 0, if $d : V \to V$ is an ∞-decoration, we know that $\sum_{y:V}\langle x, y\rangle \in g \times d\,y = z$ is a proposition since it is equivalent to $z \in d\,x$ which is a proposition. Hence, the propositional truncation in the requirement for a 0-truncation is superfluous and d is also a 0-decoration. However, the graph $g = \{\langle a, b\rangle, \langle a, c\rangle\}_0$ cannot have an ∞-decoration in any 0-level structure, if a, b and c are distinct, since $d\,b = d\,c = \emptyset$ and thus $\emptyset \in d\,a \simeq \left(\sum_{y:V}\langle a, y\rangle \in g \times d\,y = \emptyset\right) \simeq 2$ which is not a proposition. But, being wellfounded, g has a 0-decoration, namely the one which assigns $d\,x = \{\,\emptyset \mid x = a\,\}_0$.

At level 0, the ∞-decorations are the injective 0-decorations. This does not mean that d is injective on all of V—that would yield a contradiction—but rather that it becomes injective when restricted to the sets which are nodes in the graph (i.e. occurs in an edge). Classically, Scott's axiom is formulated in terms of injective decorations, but we will instead use ∞-decorations as this generalises to higher type levels.

4.2 Coalgebraic characterisation of n-decorations

The $\in$-structures are the same as P^∞-coalgebras, and the usual characterisation of decorations as coalgebra maps into V extends in our settings to coalgebra maps into P^∞. This is essentially what is proved in Proposition 4 below. However, to make characterisation convenient, either the functorial action must be adjusted for each n, or the underlying structure must be of level n. We opt to adjust the functorial action.

Definition 14. Let $n : \mathbb{N}^\infty_{-1}$, and define a wild functor $\mathrm{P}^\infty_n : \mathrm{Type} \to \mathrm{Type}$ by $\mathrm{P}^\infty_n\,X := \sum_{A:U} A \to X$ on types and on functions by $\mathrm{P}^\infty_n\,f\,(A, v) := (\mathrm{image}_n(f \circ v), \mathrm{incl}_n(f \circ v))$.

Remark: Notice that P^∞_n is like a hybrid between P^∞ and P^n: Since P^∞_n and P^∞ have the same action on types, a coalgebra for one is automatically a coalgebra for the other. On the other hand, if two P^∞_n-coalgebras factor into P^n-coalgebras, the type of P^∞_n-coalgebra homomorphisms is equivalent to the type of P^n-coalgebra homomorphisms. The following commutative diagram summarises the relationship between P^n and P^∞_n. The unnamed arrows are the n-image map and the inclusion of n-truncated functions into functions.

$$\begin{array}{ccccccc}
\mathrm{P}^\infty_n\, X & & & \xrightarrow{\mathrm{P}^\infty_n\, f} & & & \mathrm{P}^\infty_n\, Y \\
\| & & & & & & \| \\
\mathrm{P}^\infty\, X & \twoheadrightarrow & \mathrm{P}^n\, X & \xrightarrow{\mathrm{P}^n\, f} & \mathrm{P}^n\, Y & \hookrightarrow & \mathrm{P}^\infty\, Y
\end{array}$$

Let us for the rest of the subsection fix $n : \mathbb{N}^\infty_{-1}$ and a U-like $\in$-structure $(V, \in)$ and its associated P^∞-coalgebra structure $m_\in : V \to \mathrm{P}^\infty\, V$. Assume also that V is locally small and let $x \approx y$ denote the small type equivalent to the identity type for $x, y : V$.

If we have a graph in V, there are several ways of constructing a coalgebra from it. Below, we define two closely related P^∞-coalgebra structures: $m_g : V \to \mathrm{P}^\infty\, V$ and $n_g : \mathrm{Target}\ g \to \mathrm{P}^\infty\, (\mathrm{Target}\ g)$, which will help characterise decorations and define Scott's anti-foundation axiom.

Proposition 3 ( ). *For each graph $g : V$, there is a P^∞-coalgebra structure on V which we will call $m_g : V \to \mathrm{P}^\infty\, V$ such that $\pi_0(m_g\, x) \simeq \sum_{y:V} \langle x, y\rangle \in g$ and $\pi_1(m_g\, x) : \pi_0(m_g\, x) \to V$ becomes $\pi_0 : \left(\sum_{y:V} \langle x, y\rangle \in g\right) \to V$ when transported along this equivalence.*

Proof. Given $x : V$ let $m_g\, x := (\sum_{e:\overline{g}} \mathrm{source}\,(\widetilde{g}\, e) \approx x, \mathrm{target} \circ \widetilde{g} \circ \pi_0)$, and observe that:

$$\begin{aligned}
\sum_{e:\overline{g}} \mathrm{source}\,(\widetilde{g}\, e) \approx x &\simeq \sum_{y:V} \sum_{e:\overline{g}} (\mathrm{source}\,(\widetilde{g}\, e) = x) \times (\mathrm{target}\,(\widetilde{g}\, e) = y) && (12)\\
&\simeq \sum_{y:V} \sum_{e:\overline{g}} \langle \mathrm{source}\,(\widetilde{g}\, e), \mathrm{target}\,(\widetilde{g}\, e)\rangle = \langle x, y\rangle && (13)\\
&\simeq \sum_{y:V} \mathrm{fiber}\ \widetilde{g}\, \langle x, y\rangle && (14)\\
&\equiv \sum_{y:V} \langle x, y\rangle \in g, && (15)
\end{aligned}$$

Note that the following diagram commutes

$$\begin{array}{ccc}
\sum_{e:\overline{g}} \mathrm{source}\,(\widetilde{g}\, e) \approx x & \xrightarrow{\ \simeq\ } & \sum_{y:V} \langle x, y\rangle \in g \\
& {\scriptstyle \mathrm{target} \circ \widetilde{g} \circ \pi_0} \searrow \quad \swarrow {\scriptstyle \pi_0} & \\
& V &
\end{array}$$

up to definitional equality. □

Remark: Ignoring size issues, justified by Proposition 3, we will simply write:

$$m_g\, x = \left(\sum_{y:V} \langle x, y\rangle \in g, \pi_0\right).$$

This is clearer to read than coercing along an equivalence. A more careful treatment, without notational abuse, is found in the formalisation.

Lemma 5 ( ). *If a graph $g : V$ is an n-type in $(V, \in)$ (i.e. $e \in g$ is an $n-1$ type) then $\pi_1\,(m_g\, x) : \pi_0\,(m_g\, x) \to V$ is $(n-1)$-truncated, and hence m_g factors into a P^n-coalgebra $m_{n,g} : V \to \mathrm{P}^n\, V$.*

Proof. The map $\mathrm{target} \circ \widetilde{g} \circ \pi_0$ is $(n-1)$-truncated since, for any $y : V$, we have an equivalence

$$\begin{aligned} \mathrm{fiber}\,(\mathrm{target} \circ \widetilde{g} \circ \pi_0)\, y &\simeq \mathrm{fiber}\ \pi_0\, y && (16)\\ &\simeq \langle x, y\rangle \in g, && (17) \end{aligned}$$

and the last type is $(n-1)$-truncated. □

Proposition 4. *For each graph $g : V$ and map $d : V \to V$ there is an equivalence of types between d being an n-decoration of g and being a P^∞_n-coalgebra homomorphism from $m_g : V \to \mathrm{P}^\infty_n\, V$ to $m_\in : V \to \mathrm{P}^\infty_n\, V$. Hence, there is an equivalence of types between n-decorations of g and P^∞_n-coalgebra homomorphisms from m_g to $m_\in$.*

Proof. Given a graph $g : V$ and a map $d : V \to V$ we have the following chain of equivalences:

$$\begin{aligned} &(m \circ d \sim \mathrm{P}^\infty_n\ d \circ m_g)\\ &\quad\simeq \prod_{x:V}\prod_{z:V} \mathrm{fiber}\ \widetilde{(d\, x)}\, z \simeq \mathrm{fiber}\,(\mathrm{incl}_{n-1}\,(d \circ \mathrm{target} \circ \widetilde{g} \circ \pi_0))\, z && (18)\\ &\quad\simeq \prod_{x:V}\prod_{z:V} z \in d\, x \simeq \left\| \sum_{(y,p):\mathrm{fiber}\ d\, z} \langle x, y\rangle \in g \right\|_{n-1} && (19)\\ &\quad\simeq \prod_{x:V}\prod_{z:V} z \in d\, x \simeq \left\| \sum_{y:V} \langle x, y\rangle \in g \times d\, y = z \right\|_{n-1} && (20) \end{aligned}$$

□

Proposition 5 ( ). *For each graph $g : V$, the coalgebra m_g restricts to* $\mathrm{Target}\ g$. *We will call this coalgebra structure $n_g : \mathrm{Target}\, g \to \mathrm{P}^\infty\,(\mathrm{Target}\, g)$ and the subtype inclusion $\pi_0 : \mathrm{Target}\, g \to V$ is a P^∞-coalgebra homomorphism.*

Proof. First, note that for any $e : \overline{g}$, $\mathrm{target}\,(\widetilde{g}\,e)$ lies in $\mathrm{Target}\; g$ as it is the child of $\mathrm{source}\,(\widetilde{g}\,e)$. Thus let $n_g(x, \_) = (\sum_{e:\overline{g}} \mathrm{source}\,(\widetilde{g}\,e) \approx x, (\lambda(e, \_).(\mathrm{target}\,(\widetilde{g}\,e), \_)))$, for which we can check that π_0 is a P^∞-coalgebra homomorphism:

$$
\begin{aligned}
&\mathrm{P}^\infty\,\pi_0\,(n_g\,(x, \_)) \\
&\quad = (\sum_{e:\overline{g}} \mathrm{source}\,(\widetilde{g}\,e) \approx x, \pi_0 \circ (\lambda(e, \_).(\mathrm{target}\,(\widetilde{g}\,e), \_))) && (21)\\
&\quad = (\sum_{e:\overline{g}} \mathrm{source}\,(\widetilde{g}\,e) \approx x, (\lambda(e, \_).\,\mathrm{target}\,(\widetilde{g}\,e))) && (22)\\
&\quad = (\sum_{y:V} \langle x, y\rangle \in g, \pi_0) && (23)\\
&\quad = m_g\,x && (24)\\
&\quad = m_g\,(\pi_0\,(x, \_)) && (25)
\end{aligned}
$$

□

Remark: For n_g, just as for m_g, we will slightly abuse notation, justified by Proposition 5, and write:

$$n_g\,(x, \_) = \left(\sum_{y:V} \langle x, y\rangle \in g, \lambda(y, e).(y, |(x, e)|)\right).$$

Again, a more careful treatment is found in the formalisation.

4.3 Aczel's anti-foundation axiom

Aczel's anti-foundation axiom can now be formulated as generalised properties for any truncation level. We will demonstrate that if one could construct terminal coalgebras for the P^n functors, the resulting $\in$-structures would satisfy the generalised properties.

Definition 15 ( ). An $\in$-structure $(V, \in)$, with an ordered pairing structure, has **Aczel n-anti-foundation** (n-AFA), for $n : \mathbb{N}_0^\infty$, if for every graph $g : V$ the type of n-decorations of g is contractible. Equivalently, this can be split into two parts:

- n-AFA_1: For every graph $g : V$ the type of n-decorations of g is inhabited
- n-AFA_2: For every graph $g : V$ the type of n-decorations of g is a proposition.

The classical AFA axiom is equivalent to Aczel 0-anti-foundation, since 0-decorations are the usual decorations, and contractible is the HoTT way of saying “exists unique”.

As decorations are P^n-coalgebras, one type that would model AFA is the terminal P^n-coalgebra.

Theorem 2 ( ). *Suppose (V, m) is the terminal P^n-coalgebra and that V is locally U-small. Then the induced $\in$-structure has Aczel n-anti-foundation.*

Proof. It was shown in Paper II that (V, m) has an ordered pairing structure. Let $g : V$ be a graph. By Proposition 4 we need to show that the type of P^n-coalgebra homomorphisms from the corresponding graph coalgebra, given by Proposition 3, into (V, m) is contractible. But this follows from terminality of (V, m). □

4.4 Scott's anti-foundation axiom

Recall that, classically, SAFA is the statement that every Scott extensional graph has a unique injective decoration and V itself is Scott extensional. A graph is defined as being Scott extensional if equality on the nodes is tree isomorphism of the corresponding unfolding trees. Note that two trees are isomorphic if there is an isomorphism between the children of the roots, such that the subtrees of each related pair of children are tree isomorphic. We can see this as the unfolding step in a P^∞-bismulation.

The terminal P^∞-coalgebra, V_∞^∞, can be thought of as the type of trees, and the map induced by its terminality, $\mathrm{corec}^\infty\,(A, m) : A \to V_\infty^\infty$, is the unfolding of a coalgebra or graph into a tree (starting in a given node). Because of univalence, the identity type in V_∞^∞ is precisely tree isomorphism. This means that we can express Scott extensionality for a graph as saying that $\mathrm{corec}^\infty\,(\mathrm{Target}\ g, n_g)$ is an embedding. Every function in HoTT has an associated action on paths, which becomes an equivalence for an embedding. So, if $\mathrm{corec}^\infty\,(\mathrm{Target}\ g, n_g)$ is an embedding, its action on paths of the graph provides an equivalence between equality in the graph and isomorphism of its unfolding trees.

On higher type levels, it is a bit strong to require an embedding. For instance, in multisets (which are the material set theory equivalent of groupoids), we would like to consider a graph like $\{\langle\emptyset, \emptyset\rangle, \langle\emptyset, \emptyset\rangle\}_1$ as a Scott extensional representation of the complete binary tree. However, this tree has many non-trivial automorphisms in V_∞^∞, which our single node, $\emptyset$, does not have. An embedding would require nodes in the graph to come pre-filled with these automorphisms, but in our models this is not required. We therefore define the notion of a graph being Scott n-extensional.

Definition 16 ( ). Given a graph $g : V$ and $n : \mathbb{N}^{\infty}_{-1}$, we say that g is Scott $(n+1)$-extensional if the tree unfolding map $\mathrm{corec}^{\infty}\,(\mathrm{Target}\ g, n_g)$ is n-truncated.

Clearly, being Scott n-extensional implies being Scott $(n+1)$-extensional, and by the reasoning above, Scott 0-extensional is the usual notion of Scott extensional in level 0 $\in$-structures. Furthermore, if the graph is a set level graph (meaning that Target g is a set and n_g factors through P^1), then it is automatically Scott 1-extensional.

We can now define Scott's anti-foundation axiom for $\in$-structures of any type level.

Definition 17 ( ). A U-like $\in$-structure $(V, \in)$, with an ordered pairing structure, has **Scott *n*-anti-foundation** (n-SAFA), for $n : \mathbb{N}^{\infty}_{0}$, if the two properties n-SAFA$_1$ and SAFA$_2$ hold:

- n-SAFA$_1$: Any Scott n-extensional graph $g : V$ has an ∞-decoration.
- SAFA$_2$: For any graph g the type of ∞-decorations is a proposition.

The classical notion of SAFA then corresponds to what is defined above as Scott 0-anti-foundation. SAFA$_2$ is the same as ∞-AFA$_2$, and since being Scott ∞-extensional is a vacuous requirement, we get that ∞-SAFA is equivivalent to ∞-AFA.

5 The coiterative hierarchy

The coiterative hierarchy is a dualisation of a specific construction of the iterative hierarchy (Gylterud, 2018). That construction starts with the type of all *wellfounded* trees and picks out the subset of those which are hereditarily sets (i.e. in each node each immediate subtree is unique). The coiterative hierarchy is constructed dually, starting from the type of all (possibly non-wellfounded) trees, and picking out those which are co-hereditarily sets. That is, no matter how far we go into the tree, in each node the immediate subtrees are always distinct.

In Paper II, the construction of an iterative hierarchy of sets was extended to a hierarchy of n-types, V^n. When dualising to coiterative sets we will keep this level of generality and construct a coiterative hierarchy of n-types, V^n_{∞}. The first level, V^0_{∞} is then the coiterative sets.

The iterative hierarchy was carved out from the W-type $\mathrm{V}^{\infty} := \mathrm{W}_{A:U}\, A$, as a subtype, using an inductive predicate $\text{is-it-}n\text{-type} : \mathrm{V}^{\infty} \to \mathrm{Type}$. The coiterative hierarchy will, dually, be carved out as a subtype from the M-type, $\mathrm{V}^{\infty}_{\infty} := \mathrm{M}_{A:U}\, A$, and a coinductive predicate $\text{is-coit-}n\text{-type} : \mathrm{V}^{\infty}_{\infty} \to \mathrm{Type}$.

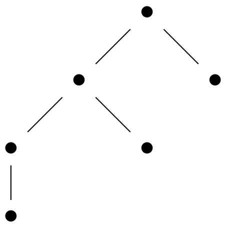


Figure 1: This tree represents an iterative set: $\{\{\{\emptyset\}_0, \emptyset\}_0, \emptyset\}_0$.

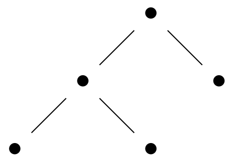


Figure 2: This tree does not represent an iterative set because the left child of the root has two equal children. It does however represent the iterative multiset: $\{\{\emptyset, \emptyset\}_1, \emptyset\}_1$.

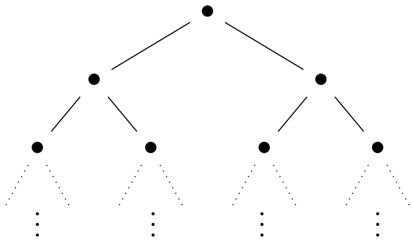


Figure 3: The full binary tree is not a coiterative set. But rather a multiset $b = \{b, b\}_1$.

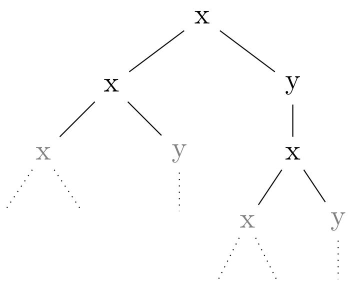


Figure 4: This infinite binary tree represents the set x which is part of the solution to the equations $x = \{x, y\}_0$ and $y = \{x\}_0$

Definition 18 ( ). For $n : \mathbb{N}_{-2}$, define the predicate:

$$\begin{aligned}
&\text{is-coit-}(n+1)\text{-type} : \mathbb{N} \to \mathrm{V}^{\infty}_{\infty} \to \mathrm{Type} \\
&\text{is-coit-}(n+1)\text{-type}_0 \; x :\equiv \text{is-}n\text{-trunc-map} \; \widetilde{x} \\
&\text{is-coit-}(n+1)\text{-type}_{(\mathrm{suc}\,k)} \; x :\equiv \prod_{a:\overline{x}} \text{is-coit-}(n+1)\text{-type}_k \, (\widetilde{x}\, a) \, .
\end{aligned}$$

Proposition 6 ( ). *The type* is-coit-n-type$_k$ x *is a proposition for any* n, k *and* $x : \mathrm{V}^{\infty}_{\infty}$.

Proof. This follows by induction on k and the fact that being an $(n-1)$-truncated map is a proposition. □

A coiterative n-type is then a tree which is a coiterative n-type at every level.

Definition 19 ( ). For $n : \mathbb{N}_{-1}$, define the predicate:

$$\begin{aligned} &\text{is-coit-}n\text{-type} : \mathrm{V}^{\infty}_{\infty} \to \mathrm{Type} \\ &\text{is-coit-}n\text{-type}\ x := \prod_{k:\mathbb{N}} \text{is-coit-}n\text{-type}_k\ x \end{aligned}$$

Proposition 7 ( ). *The type* is-coit-n-type x *is a proposition for any* $x : \mathrm{V}^{\infty}_{\infty}$.

Proof. A family of propositions is again a proposition. □

Now we can define the type of coiterative n-types.

Definition 20 (The coiterative hierarchy). For $n : \mathbb{N}_{-1}$, let V^{n}_{∞} denote the type of coiterative n-types:

$$\mathrm{V}^{n}_{\infty} := \sum_{x:\mathrm{V}^{\infty}_{\infty}} \text{is-coit-}n\text{-type}\, x.$$

Proposition 8 ( ). V^{n}_{∞} *is a subtype of* $\mathrm{V}^{\infty}_{\infty}$, *i.e. there is an embedding* $\mathrm{V}^{n}_{\infty} \hookrightarrow \mathrm{V}^{\infty}_{\infty}$.

In particular, the identity type on V^{n}_{∞} is the same as the identity type on $\mathrm{V}^{\infty}_{\infty}$.

5.1 V^{n}_{∞} is a fixed point for P^{n}

The elements in V^{n}_{∞} are non-wellfounded trees where all branchings are $(n-1)$-truncated maps. So when one removes the root from a tree, one gets a small type and an $(n-1)$-truncated map from that type into V^{n}_{∞}. Similarly, if one has a small type and an $(n-1)$-truncated map from that type into V^{n}_{∞} then one can construct a tree in V^{n}_{∞}. Hence, we will show that V^{n}_{∞} is a fixed point to P^{n}.

Lemma 6 ( ). *For any* $x : \mathrm{V}^{\infty}_{\infty}$, *there is an equivalence*

$$\text{is-coit-}n\text{-type}\ x \simeq \left(\text{is-}n\text{-trunc-map}\ \widetilde{x} \times \prod_{a:\overline{x}} \text{is-coit-}n\text{-type}\,(\widetilde{x}\, a) \right).$$

Proof. Follows by induction over $\mathbb{N}$. □

Theorem 3 ( ). V^{n}_{∞} *is a fixed point for* P^{n}.

Proof. Since $\mathrm{V}^{\infty}_{\infty}$ is the terminal P^{∞}-coalgebra, it is in particular a fixed point for P^{∞}.

Let $x : \mathrm{V}^{n}_{\infty}$, then by Lemma 6, the element $(\overline{x}, \widetilde{x})$ lies in $\mathrm{P}^{n}\ \mathrm{V}^{n}_{\infty}$. By the same token, given $A : U$ and $f : A \hookrightarrow_{n-1} \mathrm{V}^{n}_{\infty}$, the element $\sup^{\infty}(A, f)$ is a coiterative n-type. □

For the two maps given by Theorem 3 we introduce the following notation:

$$\mathrm{desup}^n : \mathrm{V}^n_\infty \to \mathrm{P}^n\ \mathrm{V}^n_\infty,$$
$$\mathrm{sup}_n : \mathrm{P}^n\ \mathrm{V}^n_\infty \to \mathrm{V}^n_\infty\,.$$

Proposition 9 ( ). *The inclusion* $\mathrm{V}^n_\infty \hookrightarrow \mathrm{V}^\infty_\infty$ *is a* P^∞*-coalgebra homomorphism from* $(\mathrm{V}^n_\infty, \mathrm{desup}^n)$ *to* $(\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)$.

Proof. Follows by structural induction on the elements of V^n_∞. □

5.2 Non-terminality of V^0_∞ as a P^0-coalgebra

Even though V^n_∞ is a fixed point for P^n and is a subtype of the terminal P^∞-coalgebra, it turns out to **not** be the terminal P^n-coalgebra. At least V^0_∞ is not the terminal P^0-coalgebra. But we conjecture this result to hold for all n. This is surprising since the dual construction gives the initial algebra of P^n (Theorem 15 of Paper II). Intuitively, the reason is that in the wellfounded setting tree isomorphism coincides with bisimulation, while in the non-wellfounded setting it does not.

For V^n_∞ to be terminal, any graph (considered as a P^n-coalgebra) should have a unique representative in V^n_∞. But V^n_∞ contains more than one representative of some graphs, i.e. we can construct a P^n-coalgebra for which there are two distinct P^n-coalgebra homomorphisms into V^n_∞. One of the maps sends each node to its unfolding tree. Because the functorial action of P^n takes the $(n-1)$-image of the composite map, i.e. it collapses some structure, there is also a P^n-coalgebra homomorphism which maps the nodes to another tree.

Theorem 4. V^0_∞ *is not the terminal* P^0*-coalgebra.*

Proof. Consider the following P^0-coalgebra (X, m), represented as a graph:

The unfolding trees of the two nodes as given by $\mathrm{corec}^\infty\,(X, m) : X \to \mathrm{V}^\infty_\infty$ are distinct, so $\mathrm{corec}^\infty\,(X, m)$ factors as a P^0-coalgebra homomorphism, $f : X \to \mathrm{V}^0_\infty$, from (X, m) to $(\mathrm{V}^0_\infty, \mathrm{desup}_0)$, such that $f\,x \neq f\,y$.

On the other hand, let g be the map that sends both nodes to the infinite unary tree, which we will denote $q : V^0_\infty$:

$$\begin{array}{c}\bullet\\ \downarrow\\ \bullet\\ \downarrow\\ \vdots\end{array}$$

Clearly, g is also a P^0-coalgebra homomorphism:

$$\mathrm{P}^0\, g\,(m\,x) = (\mathrm{image}\,(g \circ \widetilde{x}), \mathrm{incl}) = (1, \lambda\_.q) = (\overline{q}, \widetilde{q}) = \mathrm{desup}_0(g\,x)$$

and likewise for y:

$$\mathrm{P}^0\, g\,(m\,y) = (\mathrm{image}\,(g \circ \widetilde{y}), \mathrm{incl}) = (1, \lambda\_.q) = (\overline{q}, \widetilde{q}) = \mathrm{desup}_0(g\,y).$$

However, since $f\,x \neq f\,y$ and $g\,x = g\,y$, we get that f and g are two distinct P^0-coalgebra homomorphisms from (X, m) to $(\mathrm{V}^0_\infty, \mathrm{desup}_0)$. □

5.3 Local smallness of V^n_∞

The functorial action of P^n takes the n-image of a map. In order for this to be small, the domain must be small and the codomain appropriately locally small. In particular, when we are considering maps into V^n_∞, we use the fact that this type is locally small, as we will show in this section. This result uses univalence and follows from the characterisation of the identity on an M-type as an indexed M-type.

The idea is that, by univalence, the indexed functor $\mathrm{E}_{(V^\infty_\infty, \mathrm{desup})}$ is equivalent to the indexed functor $\mathrm{E}'_{(V^\infty_\infty, \mathrm{desup})}$, for which the corresponding indexed M-type is small.

Definition 21 ( ). Given X : Type and $m : X \to \left(\sum_{A:Type} A \to X\right)$, define the following functor

$$\begin{aligned}\mathrm{E}'_{(X,m)} &: (X \times X \to \mathrm{Type}) \to (X \times X \to \mathrm{Type})\\ \mathrm{E}'_{(X,m)}\ R\ (x, y) &:= \sum_{e:\overline{x} \simeq \overline{y}} \prod_{a:\overline{x}} R\,(\widetilde{x}\,a, \widetilde{y}\,(e\,a)).\end{aligned}$$

Proposition 10 ( ). *Given* X : Type *and* $m : X \to \left(\sum_{A:Type} A \to X\right)$, *there is a natural family of equivalences*

$$\mathrm{E}_{(X,m)}\ R\ (x, y) \simeq \mathrm{E}'_{(X,m)}\ R\ (x, y).$$

Proof. Follows by univalence. □

This gives us an alternative characterisation of the identity type on V^∞_∞.

Theorem 5 ( ). *The identity type on V^∞_∞ is the terminal* E'*-coalgebra.*

Proof. By Theorem 1, the identity type on V^∞_∞ is the terminal E-coalgebra. Since the functors E and E' are naturally equivalent by Proposition 10, the identity type is also the terminal coalgebra for E'. ☐

Note that by the theorem above, there is for any $x, y : \mathrm{V}^\infty_\infty$ an equivalence

$$(x = y) \simeq \sum_{e:\overline{x}\simeq\overline{y}} \prod_{a:\overline{x}} \widetilde{x}\, a = \widetilde{y}\,(e\, a).$$

Theorem 6 ( ). V^∞_∞ *is locally U-small.*

Proof. Since $\mathrm{E}'_{(\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)}$ is an indexed polynomial functor, it has a corresponding indexed M-type which is the terminal $\mathrm{E}'_{(\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)}$-coalgebra. In their paper on non-wellfounded trees in HoTT, Ahrens, Capriotti, and Spadotti (2015) constructed indexed M-types from inductive types. From this construction one can observe that the universe level of the constructed indexed M-type does not depend on the indexing type. In our case, the universe level of the indexed M-type corresponding to $\mathrm{E}'_{(\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)}$ is the least upper bound of the universe levels of $\overline{x} \simeq \overline{y}$ and $\overline{x}$, which is U.

Since (the carrier of) any two terminal $\mathrm{E}'_{(\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)}$-coalgebras are equivalent, it follows that V^∞_∞ is locally U-small. ☐

Corollary 2 ( ). V^n_∞ *is locally U-small.*

Proof. By Proposition 8, V^n_∞ is a subtype of V^∞_∞ and thus has the same identity type. The result then follows from the fact that V^∞_∞ is locally U-small, by Theorem 6. ☐

5.4 V^n_∞ is a simple P^∞-coalgebra

The first requirement to satisfy SAFA is that the type of ∞-decorations is a proposition. By the characterisation of ∞-decorations as P^∞-coalgebra homomorphisms it is sufficient for the model to be a simple P^∞-coalgebra. Thus, we show this for V^n_∞.

Note that we do not prove that V^n_∞ is simple as a P^n coalgebra. In fact, the proof of non-terminality of V^0_∞ demonstrates that it is not simple as a P^0 coalgebra.

Definition 22. Let X and Y be types, and let $f : X \to Y$. Given a binary relation $R : Y \times Y \to \mathrm{Type}$ we define a binary relation on X:

$$\begin{aligned} R\, f &: X \times X \to \mathrm{Type} \\ R\, f\,(x, x') &:= R\,(f\, x, f\, x'). \end{aligned}$$

Moreover, given a fiberwise map $g : \prod_{(y,y'):Y\times Y} R\,(y,y') \to R'\,(y,y')$ between relations R and R', define the fiberwise map:

$$\begin{aligned} g\,f &: \prod_{(x,x'):X\times X} R\,f\,(x,x') \to R'\,f\,(x,x') \\ g\,f\,(x,x') &:= g\,(f\,x, f\,x'). \end{aligned}$$

Proposition 11 ( ). *Let* (X,m) *and* (Y,n) *be* P^∞*-coalgebras and let* (f,α) *be a* P^∞*-coalgebra homomorphism from* (X,m) *to* (Y,n)*. For any binary relation* $R : Y\times Y \to \mathrm{Type}$ *and every pair of elements* $x,x' : X$*, there is an equivalence*

$$e_R : \mathrm{E}'_{(Y,n)}\ R\,(f\,x, f\,x') \simeq \mathrm{E}'_{(X,m)}\,(R\,f)\,(x,x').$$

This family of equivalences is natural, i.e. for every fiberwise map $g : \prod_{(y,y'):Y\times Y} R\,(y,y') \to R'\,(y,y')$ *the following diagram commutes:*

$$\begin{array}{ccc} \mathrm{E}'_{(Y,n)}\ R\,(f\,x, f\,x') & \xrightarrow{\ \mathrm{E}'_{(Y,n)}\ g\,(f\,x, f\,x')\ } & \mathrm{E}'_{(Y,n)}\ R'\,(f\,x, f\,x') \\ {\scriptstyle e_R}\big\downarrow & & \big\downarrow{\scriptstyle e_{R'}} \\ \mathrm{E}'_{(X,m)}\,(R\,f)\,(x,x') & \xrightarrow[\ \mathrm{E}'_{(X,m)}\,(g\,f)\,(x,x')\ ]{} & \mathrm{E}'_{(X,m)}\,(R'\,f)\,(x,x') \end{array}$$

Moreover, for equality we have

$$e_=\,(\text{id-equiv}, \text{refl-htpy}) = (\text{id-equiv}, \text{refl-htpy}).$$

Proof. For $x,x' : X$, the two types are

$$\mathrm{E}'_{(Y,n)}\ R\,(f\,x, f\,x') \equiv \sum_{e:\overline{f\,x}\simeq\overline{f\,x'}}\ \prod_{a:\overline{f\,x}} R\,(\widetilde{(f\,x)}\,a, \widetilde{(f\,x')}\,(e\,a)), \tag{26}$$

$$\mathrm{E}'_{(X,m)}\,(R\,f)\,(x,x') \equiv \sum_{e:\overline{x}\simeq\overline{x'}}\ \prod_{a:\overline{x}} R\,(f\,(\widetilde{x}\,a), f\,(\widetilde{x'}\,(e\,a))), \tag{27}$$

Note that we have paths

$$\alpha\,x : (\overline{f\,x}, \widetilde{f\,x}) = (\overline{x}, f\circ\widetilde{x}), \tag{28}$$

$$\alpha\,x' : (\overline{f\,x'}, \widetilde{f\,x'}) = (\overline{x'}, f\circ\widetilde{x'}). \tag{29}$$

The desired equivalence is thus given by transporting along these paths. Naturality follows from the fact that transport preserves families. The action of $e_=$ on $(\text{id-equiv}, \text{refl-htpy})$ follows by path induction. □

Proposition 12 ( ). *Let (X, m) and (Y, n) be P^∞-coalgebras and let (f, α) be a P^∞-coalgebra homomorphism from (X, m) to (Y, n). Given an $\mathrm{E}'_{(X,m)}$-coalgebra (R, σ), we can define an $\mathrm{E}'_{(Y,n)}$-coalgebra $(f\,R, f\,\sigma)$.*

Proof. Intuitively, $f\,R$ relates any two elements which are the images of two related elements in X. Formally, define the relation $f\,R : Y \times Y \to \mathrm{Type}$ by

$$f\,R\,(y, y') := \sum_{(x,-):\mathrm{fiber}\ f\,y}\ \sum_{(x',-):\mathrm{fiber}\ f\,y'} R\,(x, x').$$

For $f\,\sigma$, it is enough by path induction to construct an element

$$f\,\sigma\,(f\,x, f\,x')\,((x, \mathrm{refl}), (x', \mathrm{refl}), r) : \sum_{e:\overline{f\,x} \simeq \overline{f\,x'}}\ \prod_{a:\overline{f\,x}} f\,R\,(\widetilde{(f\,x)}\,a, \widetilde{(f\,x')}\,(e\,a))$$

Using the (inverse of the) equivalence $e_{f\,R}$ in Proposition 11, it is enough to construct an element of the type

$$\sum_{e:\overline{x} \simeq \overline{x'}}\ \prod_{a:\overline{x}} f\,R\,(f\,(\widetilde{x}\,a), f\,(\widetilde{x'}\,(e\,a))).$$

For this we take

$$(\pi_0\,(\sigma\,r), \lambda\,a.((\widetilde{x}\,a, \mathrm{refl}), (\widetilde{x'}\,(e\,a), \mathrm{refl}), \pi_1\,(\sigma\,r))). \qquad \square$$

Proposition 13 ( ). *Let (X, m) and (Y, n) be P^∞-coalgebras and let (f, α) be a P^∞-coalgebra homomorphism from (X, m) to (Y, n). Let (R, σ) be an $\mathrm{E}'_{(X,m)}$-coalgebra and let (S, ϕ) be an $\mathrm{E}'_{(Y,n)}$-coalgebra. There is an equivalence*

$$\hom\,((R, \sigma), (S\,f, e_S \circ \phi)) \simeq \hom\,((f\,R, f\,\sigma), (S, \phi)),$$

where e_S is the equivalence given in Proposition 11.

Proof. Define the map

$$h : \prod_{(x,x'):X \times X} R\,(x, x') \to f\,R\,(f\,x, f\,x')$$
$$h\,(x, x')\,r := ((x, \mathrm{refl}), (x', \mathrm{refl}), r).$$

We have the following chain of equivalences

$$\begin{aligned}
&\hom\,((R,\sigma),(S\,f,e_S\circ\phi)) \\
&\quad\equiv \sum_{g:\prod_{(x,x'):X\times X} R\,(x,x')\to S\,(f\,x,f\,x')} \tag{30}\\
&\qquad\prod_{(x,x'):X\times X} e_S\,(x,x')\circ\phi\,(f\,x,f\,x')\circ g\,(x,x') \\
&\qquad\qquad\sim \mathrm{E}'_{(X,m)}\;g\,(x,x')\circ\sigma\,(x,x') \\
&\quad\simeq \sum_{g:\prod_{(x,x'):X\times X} R\,(x,x')\to S\,(f\,x,f\,x')} \tag{31}\\
&\qquad\prod_{(x,x'):X\times X} \phi\,(f\,x,f\,x')\circ g\,(x,x') \\
&\qquad\qquad\sim e_S^{-1}\,(x,x')\circ\mathrm{E}'_{(X,m)}\;g\,(x,x')\circ\sigma\,(x,x') \\
&\quad\simeq \sum_{g:\prod_{(y,y'):Y\times Y} f\,R\,(y,y')\to S\,(y,y')} \tag{32}\\
&\qquad\prod_{(x,x'):X\times X} \phi\,(f\,x,f\,x')\circ g\,f\,(x,x')\circ h\,(x,x')\sim e_S^{-1}\,(x,x') \\
&\qquad\qquad\circ\,\mathrm{E}'_{(X,m)}\,(\lambda\,(x,x').g\,f\,(x,x')\circ h\,(x,x'))\,(x,x')\circ\sigma\,(x,x') \\
&\quad\equiv \sum_{g:\prod_{(y,y'):Y\times Y} f\,R\,(y,y')\to S\,(y,y')} \tag{33}\\
&\qquad\prod_{(x,x'):X\times X} \phi\,(f\,x,f\,x')\circ g\,f\,(x,x')\circ h\,(x,x')\sim e_S^{-1}\,(x,x') \\
&\qquad\qquad\circ\,\mathrm{E}'_{(X,m)}\,(g\,f)\,(x,x')\circ\mathrm{E}'_{(X,m)}\;h\,(x,x')\circ\sigma\,(x,x') \\
&\quad\simeq \sum_{g:\prod_{(y,y'):Y\times Y} f\,R\,(y,y')\to S\,(y,y')} \tag{34}\\
&\qquad\prod_{(x,x'):X\times X} \phi\,(f\,x,f\,x')\circ g\,f\,(x,x')\circ h\,(x,x')\sim \mathrm{E}'_{(Y,n)}\;g\,(f\,x,f\,x') \\
&\qquad\qquad\circ\,e_{f\,R}^{-1}\,(x,x')\circ\mathrm{E}'_{(X,m)}\;h\,(x,x')\circ\sigma\,(x,x') \\
&\quad\equiv \sum_{g:\prod_{(y,y'):Y\times Y} f\,R\,(y,y')\to S\,(y,y')} \tag{35}\\
&\qquad\prod_{(x,x'):X\times X} \phi\,(f\,x,f\,x')\circ g\,f\,(x,x')\circ h\,(x,x') \\
&\qquad\qquad\sim \mathrm{E}'_{(Y,n)}\;g\,(f\,x,f\,x')\circ f\,\sigma\,(f\,x,f\,x')\circ h\,(x,x')
\end{aligned}$$

$$\simeq \sum_{g:\prod_{(y,y'):Y\times Y} f\,R\,(y,y') \to S\,(y,y')} \prod_{(y,y'):Y\times Y} \phi\,(y,y') \circ g\,(y,y') \sim \mathrm{E}'_{(Y,n)}\; g\,(y,y') \circ f\,\sigma\,(y,y') \tag{36}$$

$$\equiv \hom\,((f\,R, f\,\sigma), (S, \phi)) \tag{37}$$

Step (34) uses the naturality of the equivalence in Proposition 11. □

Corollary 3 ( ). *Let (X, m) and (Y, n) be P^∞-coalgebras and let (f, α) be a P^∞-coalgebra homomorphism from (X, m) to (Y, n). Suppose that equality on Y is the terminal $\mathrm{E}'_{(Y,n)}$-coalgebra. Then equality on X is the terminal $\mathrm{E}'_{(X,m)}$-coalgebra if and only if f is an embedding.*

Proof. Let m be the $\mathrm{E}'_{(X,m)}$-coalgebra map for equality on X and let m' be the $\mathrm{E}'_{(Y,n)}$-coalgebra map for equality on Y. Let e be the equivalence given in Proposition 11, for equality on Y. Note that since $e\,(\text{id-equiv}, \text{refl-htpy}) = (\text{id-equiv}, \text{refl-htpy})$, ap_f is an $\mathrm{E}'_{(X,m)}$-coalgebra homomorphism from $(=_X, m)$ to $(\lambda\,(x, x').f\,x =_Y f\,x', e \circ m')$.

For any $\mathrm{E}'_{(X,m)}$-coalgebra (R, σ), we have an equivalence

$$\hom((R, \sigma), (\lambda\,(x, x').f\,x = f\,x', e \circ m')) \simeq \hom((f\,R, f\,\sigma), (=_Y, m'))$$

given by Proposition 13. By the terminality of $(=_Y, m')$ it therefore follows that $(\lambda\,(x, x').f\,x =_Y f\,x', e \circ m')$ is the terminal $\mathrm{E}'_{(X,m)}$-coalgebra.

Since ap_f is an $\mathrm{E}'_{(X,m)}$-coalgebra homomorphism from the identity $(=_X, m)$ to $(\lambda\,(x, x').f\,x =_Y f\,x', e \circ m')$ and the terminal $\mathrm{E}'_{(X,m)}$-coalgebra is unique up to unique isomorphism, it follows that ap_f is a family of equivalences if and only if $(=_X, m)$ is the terminal $\mathrm{E}'_{(X,m)}$-coalgebra. □

Corollary 4. $(\mathrm{V}^n_\infty, \mathrm{desup}^n)$ *is a simple* P^∞*-coalgebra.*

Proof. V^n_∞ embeds into V^∞_∞, hence it satisfies the conditions of Corollary 3. $\mathrm{E}'_{(\mathrm{V}^n_\infty, \mathrm{desup}^n)}$-coalgebras are equivalent to P^∞-bimsimulations on $(\mathrm{V}^n_\infty, \mathrm{desup}^n)$, so it follows that $(\mathrm{V}^n_\infty, \mathrm{desup}^n)$ is a bisimulation simple P^∞-coalgebra, hence simple. □

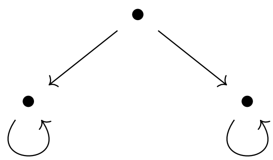
One can also use Corollary 3 to replace the condition in 0-SAFA_1 with g being simple:

Corollary 5. *Given an $\in$-structure $(V, \in)$, a graph $g : V$ being Scott 0-extensional is equivalent to (Target_g, n_g) being a simple P^∞-coalgebra.*

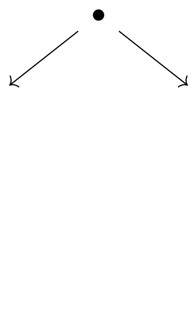
5.5 Coalgebra homomorphisms into V^n_∞

How do we construct a map from a P^n-coalgebra, say (X, m), into V^n_∞? An obvious approach is to view (X, m) as a P^∞-coalgebra and show that $\mathrm{corec}^\infty : X \to \mathrm{V}^\infty_\infty$ lands in V^n_∞, where corec^∞ is the underlying map of the unique P^∞-coalgebra homomorphism from (X, m) to $(\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)$. Unfortunately, this is not always the case.

Viewing (X, m) as a graph, corec^∞ maps each node to its unfolding tree. Consider now the P^0-coalgebra represented by the following graph:



The topmost node is mapped by corec^∞ to the tree



which is not an element of V^0_∞ as the branching at the root is not an embedding.

However, if corec^∞ is an $(n-1)$-truncated map, then it lands in V^n_∞.

Proposition 14 ( ). *Given a* P^n*-coalgebra* (X, m), *if* $\mathrm{corec}^\infty : X \to \mathrm{V}^\infty_\infty$ *is an* $(n-1)$*-truncated map, then for all* $x : X$, $\mathrm{corec}^\infty\ x$ *is a coiterative* n*-type.*

Proof. For $x : X$ we need to show that

$$\prod_{k:\mathbb{N}} \text{is-coit-}n\text{-type}_k\,(\mathrm{corec}^\infty\ x).$$

Proceed by induction on k.

For the base case, note that since corec^∞ is a P^∞-coalgebra homomorphism, we have

$$\widetilde{(\mathrm{corec}^\infty\ x)} = \mathrm{corec}^\infty \circ \widetilde{x}.$$

Both these maps are $(n-1)$-truncated, and therefore the composition is $(n-1)$-truncated.

Similarly, for the induction step, since corec^{∞} is a homomorphism, it is enough to show that

$$\prod_{a:\overline{x}} \text{is-coit-}\, n\, \text{-type}_k \left(\mathrm{corec}^{\infty} \left(\widetilde{x}\, a\right)\right).$$

But this follows from the induction hypothesis. □

Definition 23 ( ). Given a P^n-coalgebra (X, m) for which corec^{∞} is an $(n-1)$-truncated map, let

$$\mathrm{corec}^n : X \to \mathrm{V}^n_{\infty}$$

denote the restriction of corec^{∞} into V^n_{∞} by Proposition 14.

The map corec^n is a P^n-coalgebra homomorphism. This is an instance of a useful lemma about which maps into V^n_{∞} are P^n-coalgebra homomorphisms.

Lemma 7 ( ). *Let (X, m) be a P^n-coalgebra and let $f : X \hookrightarrow_{n-1} \mathrm{V}^n_{\infty}$. Then there is an equivalence of types between f being a P^n-coalgebra homomorphism and $\pi_0 \circ f$ being a P^{∞}-coalgebra homomorphism.*

Proof. Consider the following diagram:

$$\begin{array}{ccccc}
X & \xrightarrow{\quad f \quad} & \mathrm{V}^n_{\infty} & \xrightarrow{\quad \pi_0 \quad} & \mathrm{V}^{\infty}_{\infty} \\
{\scriptstyle m}\downarrow & & \downarrow{\scriptstyle \mathrm{desup}^n} & & \\
\mathrm{P}^n\ X & \xrightarrow{\mathrm{P}^n\ f} & \mathrm{P}^n\ \mathrm{V}^n_{\infty} & & \downarrow{\scriptstyle \mathrm{desup}^{\infty}} \\
\downarrow & & \downarrow & & \\
\mathrm{P}^{\infty}\ X & \xrightarrow[\mathrm{P}^{\infty}\ f]{} & \mathrm{P}^{\infty}\ \mathrm{V}^n_{\infty} & \xrightarrow[\mathrm{P}^{\infty}\ \pi_0]{} & \mathrm{P}^{\infty}\ \mathrm{V}^{\infty}_{\infty}
\end{array}$$

The map from $\mathrm{P}^n\ \mathrm{V}^n_{\infty}$ to $\mathrm{P}^{\infty}\ \mathrm{V}^n_{\infty}$ is an embedding as it simply forgets that the map in the second coordinate is $(n-1)$-truncated. Additionally, $\mathrm{P}^{\infty}\ \pi_0$ is an embedding since π_0 is an embedding. Thus, it is equivalent to show that the upper left square commutes, and showing that the two maps are equal when postcomposed with the forgetful map and $\mathrm{P}^{\infty}\ \pi_0$.

The square on the right commutes as the inclusion V^n_{∞} into $\mathrm{V}^{\infty}_{\infty}$ is a P^{∞}-coalgebra homomorphism (Proposition 9), and since f is an $(n-1)$-truncated map, the lower left square also commutes. It therefore follows that there is an equivalence between the upper left square commuting and the outer square commuting. □

Proposition 15 ( ). *Let* (X, m) *be a* P^n*-coalgebra for which* corec^∞ *is an* $(n-1)$*-truncated map, then* corec^n *is an* $(n-1)$*-truncated map, and it is a* P^n*-coalgebra homomorphism into* $(\mathrm{V}^n_\infty, \mathrm{desup}^n)$.

Proof. Since corec^∞ is an $(n-1)$-truncated map and $\pi_0 : \mathrm{V}^n_\infty \to \mathrm{V}^\infty_\infty$ is an embedding, it follows that corec^n is an $(n-1)$-truncated-map. By Lemma 7, since $\mathrm{corec}^\infty \equiv \pi_0 \circ \mathrm{corec}^n$ is a P^∞-coalgebra homomorphism, it follows that corec^n is a P^n-coalgebra homomorphism. □

Even though $(\mathrm{V}^n_\infty, \mathrm{desup}^n)$ is not the terminal P^n-coalgebra, it is *almost* terminal — it is terminal with respect to truncated maps.

Theorem 7 ( ). *Let* (X, m) *be a* P^n*-coalgebra for which* corec^∞ *is an* $(n-1)$*-truncated map. Then the following type is contractible:*

$$\sum_{(f,\alpha):\mathrm{Hom}_{\mathrm{P}^n\text{-Coalg}}\,(X,m)\,(\mathrm{V}^n_\infty,\mathrm{desup}^n)} \text{is-}(n-1)\text{-trunc-map } f$$

Proof. First we note that by Lemma 7, the type of P^n-coalgebra homomorphisms from (X, m) to $(\mathrm{V}^n_\infty, \mathrm{desup}^n)$ for which the underlying map is $(n-1)$-truncated, is a subtype of the type of P^∞-coalgebra homomorphisms from (X, m) to $(\mathrm{V}^\infty_\infty, \mathrm{desup}^\infty)$. Specifically, we have the following chain of equivalences and embeddings:

$$\begin{aligned}
&\sum_{f:X\hookrightarrow_{n-1}\mathrm{V}^n_\infty} \mathrm{desup}^n \circ f \sim \mathrm{P}^n\ f \circ m \\
&\quad\simeq \sum_{f:X\hookrightarrow_{n-1}\mathrm{V}^n_\infty} \mathrm{desup}^\infty \circ \pi_0 \circ f \sim \mathrm{P}^\infty\,(\pi_0 \circ f) \circ m && (38)\\
&\quad\hookrightarrow \sum_{f:X\to\mathrm{V}^n_\infty} \mathrm{desup}^\infty \circ \pi_0 \circ f \sim \mathrm{P}^\infty\,(\pi_0 \circ f) \circ m && (39)\\
&\quad\hookrightarrow \sum_{f:X\to\mathrm{V}^\infty_\infty} \mathrm{desup}^\infty \circ f \sim \mathrm{P}^\infty\ f \circ m && (40)
\end{aligned}$$

The last step is an instance of the fact that embeddings are monomorphisms.

By Proposition 15, the first type in the chain above is inhabited. Since any inhabited type which embeds into a proposition is contractible, it follows that the first type is contractible. □

Note that this does not contradict the counter example to terminality above since the second map in that case is not an embedding.

6 The coiterative hierarchy as a model of set theory

A result that dates back to the fifties is that any fixed point of the powerset functor is a model of ZFC^- (ZFC without foundation/regularity) (Rieger, 1957). In a previous paper by two of the authors (Paper II) a corresponding result was shown for models of set theory in HoTT — the powerset functor in this case being P^0. Specifically, a fixed point of P^0 in HoTT is a model of

- empty set,
- unordered pairing,
- restricted separation,
- replacement,
- union,
- exponentiation,
- infinity/natural numbers.

In fact, natural higher type level generalisations of these axioms were defined and it was shown that fixed points of P^n satisfy the axioms at level n or less[§] (Section 5 in Paper II). Moreover, the type V^n was shown to be the initial algebra of the functor P^n and as such was shown to model the axiom of foundation, in addition to the ones above.

Since V^n_∞ is a fixed point of P^n it is also a model of the axioms above. However, since it is not the initial algebra, it is *not* a model of foundation. Neither is it the terminal coalgebra, and thus *not* a model of Aczel's anti-foundation axiom. In this section we will show that it is instead a model of Scott's anti-foundation axiom.

The definition of the elementhood relation on V^n_∞ is the one which is induced by its coalgebra structure. The idea is that the elements of a tree are the children of the root.

Definition 24. For $x, y : \mathrm{V}^n_\infty$, define the elementhood relation between them as

$$x \in_n y := \mathrm{fiber}\ \widetilde{y}\ x.$$

The relation $\in_n$ is extensional: the canonical map

$$x = y \to \prod_{z:\mathrm{V}^n_\infty} z \in_n x \simeq z \in_n y$$

is an equivalence. A type with an extensional binary relation in this sense is what is called an $\in$-structure in Paper II. We will use the definitions of the properties corresponding to the set theoretic axioms defined there.

[§]There is also a requirement about the fixed point being appropriately locally small.

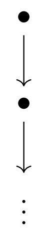
The following result is an instance of the results in Section 5 of Paper II, which shows that a (locally small) fixed point of P^n models all defined properties except foundation.

Theorem 8 ( ). *For $n : \mathbb{N}^\infty$, $(\mathrm{V}^n_\infty, \in_n)$ satisfies the following properties, as defined in Paper II:*

- *empty set,*
- *U-restricted n-separation,*
- *∞-unordered I-tupling, for all $k : \mathbb{N}_{-1}$ such that $k < n$ and k-types $I : U$,*
- *k-unordered I-tupling, for all $k : \mathbb{N}_{-1}$ such that $k \leq n$ and $I : U$,*
- *k-replacement, for all $k : \mathbb{N}_{-1}$ such that $k \leq n$,*
- *k-union, for all $k : \mathbb{N}_{-1}$ such that $k \leq n$,*
- *exponentiation, for any ordered pairing structure,*
- *natural numbers for any $(n-1)$-truncated representation.*

6.1 V^n_∞ models Scott's anti-foundation axiom

As V^n_∞ is not the initial P^n-algebra, $(\mathrm{V}^n_\infty, \in_n)$ is not a model of foundation. Indeed, V^n_∞ contains anti-wellfounded sets, the simplest one being the infinite unary tree:



So $(\mathrm{V}^n_\infty, \in_n)$ is a model of non-wellfounded sets. However, as discussed at the start of this paper, there are several anti-foundation axioms in material set theory, so we need to state specifically which anti-foundation axiom $(\mathrm{V}^n_\infty, \in_n)$ is a model of. In this section we will show that $(\mathrm{V}^n_\infty, \in_n)$ has Scott n-anti-foundation.

By Theorem 1 in Paper II and Theorem 8, $(\mathrm{V}^n_\infty, \in_n)$ has an ordered pairing structure. Let $\langle -, - \rangle : \mathrm{V}^n_\infty \times \mathrm{V}^n_\infty \hookrightarrow V^n_\infty$ denote this structure.

Theorem 9 ( ). *For each $n : \mathbb{N}^\infty_0$ the $\in$-structure $(\mathrm{V}^n_\infty, \in_n)$ has the Scott k-anti-foundation property (k-SAFA) for any $k \leq n$.*

Proof. SAFA_2 is immediate from V^n_∞ being a simple P^∞-coalgebra by Corollary 4 and Proposition 4.

For n-SAFA$_1$, let $g : \mathrm{V}^n_\infty$ be a Scott n-extensional graph. The P^∞-coalgebra homomorphism $\mathrm{corec}^\infty(\mathrm{Target}\ g, n_g) : \mathrm{Target}\ g \to \mathrm{V}^\infty_\infty$ factors through the embedding $\mathrm{V}^n_\infty \hookrightarrow \mathrm{V}^\infty_\infty$, since $g : \mathrm{V}^n_\infty$ is Scott n-extensional. Denote this map $d' : \mathrm{Target}\ g \to V^n_\infty$.

To obtain from this a P^∞-coalgebra homomorphism from (V^n, m_g), and thus an ∞-decoration by Proposition 4, let $d\,x = \sup^n\left(\sum_{y:\mathrm{V}^n_\infty} \langle x, y\rangle \in g, \lambda(y,e).d'\,(y, |(x,e)|)\right)$. This is a valid application of $\sup^n$ since $\sum_{y:\mathrm{V}^n_\infty} \langle x, y\rangle \in g$ is essentially small and d' is $(k-1)$-truncated and thus its composition with the map $\left(\sum_{y:\mathrm{V}^n_\infty} \langle x, y\rangle \in g\right) \to \mathrm{Target}\ g$ sending (y,e) to $(y, |(x,e)|)$ is $(n-1)$-truncated. It remains to check that the coalgebra homomorphism square commutes, i.e. $\mathrm{desup}\,(d\,x) = \mathrm{P}^\infty\,d\,(m_g\,x)$. Note that the first component of both $\mathrm{desup}\,(d\,x)$ and $\mathrm{P}^\infty\,d\,(m_g\,x)$ is $\sum_{y:\mathrm{V}^n_\infty} \langle x, y\rangle \in g$. For the second component we have the following chain of equalities:

$$\begin{aligned}
\pi_1\ (\mathrm{P}^\infty\,d\,(m_g\,x)) &= d \circ \pi_0 \\
&= \lambda(y,e).d\,y \\
&= \lambda(y,e).\sup^n\left(\sum_{z:\mathrm{V}^n_\infty} \langle y, z\rangle \in g, \lambda(z,e').d'\,(z, |(y,e')|)\right) \\
&= \lambda(y,e).\sup^n\left(\mathrm{P}^\infty\,d'\left(\sum_{z:\mathrm{V}^n_\infty} \langle y, z\rangle \in g, \lambda(z,e').(z, |(y,e')|)\right)\right) \\
&= \lambda(y,e).\sup^n\left(\mathrm{P}^\infty\,d'\,(n_g\,(y, |(x,e)|))\right) \\
&= \lambda(y,e).d'(y, |(x,e)|) \\
&= \pi_1\ (\mathrm{desup}\,(d\,x))
\end{aligned} \qquad \square$$

7 The terminal P^0-coalgebra

In this section we describe a general construction of terminal coalgebras for functors satisfying a certain accessibility condition. This is a formalization in type theory of a theorem due to Aczel and Mendler (1989), which dates back to the late 80s and states that every *set-based* endofunctor on the category of proper classes has a terminal coalgebra. We describe how to translate the original proof of Aczel and Mendler, written in the language of set theory with reasoning based on classical logic, into the constructive setting of HoTT. In the type theoretic statement of the theorem, proper classes are replaced by large types, and sets are replaced by small types. The notion of set-based functor is replaced by a certain accessibility condition with respect to small types. We were able to remove all invocations of choice principles from the original proof, but not all impredicativity. In fact, the existence of terminal coalgebras is guaranteed only under the assumption of *propositional resizing*, a form of impredicativity for propositions. Here we recall the principle in a formulation given by Jong and Escardó (2023).

Definition 25 ( ). The principle of **propositional resizing** states that every proposition P : Type is essentially small, i.e. it is equivalent to a small proposition $Q : U$.

We do not assume propositional resizing globally, but we precisely mark all theorems that require its assumption.

Remember that P^0 does not have a functorial action on *all* functions, only on ones with locally small codomain. In the presence of propositional resizing, these can also be functions with set-valued codomain. This means that the Aczel–Mendler theorem does not immediately apply to P^0. Nevertheless, in the last part of this section we will show how to appropriately adjust the statement and proof of the theorem in order to construct terminal coalgebras also for "functors" such as P^0.

7.1 U-based functors

Aczel and Mendler's theorem applies to set-based endofunctors on proper classes, where, intuitively, a functor is set-based when its value on a proper class X is the colimit of values on small subsets of X. Before reformulating this accessibility condition in our type theoretic setting, we recall some definitions and establish some notation.

In this section, we globally assume functors to be **set-valued**, i.e $F\,X$ is a set, independently of the type level of X.

Definition 26 ( ). Let $\alpha : A \to FA$ be a coalgebra. We say that α is

- U**-simple** if, for all $B : U$ and coalgebras $\beta : B \to FB$, the type of coalgebra homomorphisms from β to α is a proposition;
- U**-terminal** if, for all $B : U$ and coalgebras $\beta : B \to FB$, the type of coalgebra homomorphisms from β to α is contractible;

Aczel and Mendler write "strongly extensional" instead of "U-simple". Assuming propositional resizing, the Aczel–Mendler theorem guarantees the existence of a U-terminal coalgebra for every functor F. But the existence of a terminal coalgebra is guaranteed only in case F satisfies an accessibility condition. This condition is a type-theoretic reformulation (and slight generalization) of Aczel and Mendler's notion of set-based functor.

Definition 27 ( ). A functor F is U**-based** if, for any large type X : Type and $x : FX$, there is a small type $Y : U$, a function $\iota : Y \to X$ and element $y : FY$ such that $F\,\iota\,y = x$.

The existential quantification in the above statement is strong, i.e. it is a Σ-type without propositional truncation around it. Intuitively, F is U-based when FX is the colimit of FY, where Y ranges over small generalized elements of X. Notice that the definition is slightly different from the one of Aczel and Mendler, as the they require Y to be a subset of X, i.e. ι is an embedding in their definition. This restriction is not crucial in the construction of the terminal coalgebra, so we remove it from the definition.

Notice that Definition 27 admits a slight reformulation, that will become useful later on: a functor is U-based whenever for all X : Set the function

$$(\lambda(A, f, a).\, F\, f\, a) : \left(\sum_{(A,f):\mathrm{P}^\infty\, X} FA \right) \to FX$$

has a section $base_F : FX \to \sum_{(A,f):\mathrm{P}^\infty\, X} FA$.

7.2 Relation lifing and precongruences

There are many ways to lift a (possibly proof-relevant) relation on a type X to a relation on FX (Staton, 2011). Many of these liftings work well only when the functor F preserves weak pullbacks. This restriction can be avoided by employing Aczel and Mendler's notion of relation lifting.

Definition 28 ( ). Given X : Type, the **relation lifting** E_F takes a relation $R : X \times X \to \mathrm{Type}$ and produces a relation $E_F\, R : FX \times FX \to \mathrm{Type}$ as follows:

$$E_F\, R\,(x, y) := (F\,[-]_R\, x = F\,[-]_R\, y)$$

where $[-]_R$ is the point constructor of the set quotient X/R.

Notice that $E_F\, R$ is always propositionally-valued since $F(X/R)$ is always a set. If R is valued in U instead of Type, there is no guarantee that $E_F\, R$ is also valued in U, as $F(X/R)$ may not be locally U-small. But this is true under the assumption of propositional resizing.

Definition 29 ( ). Given a coalgebra $\alpha : X \to FX$, a relation $R : X \times X \to \mathrm{Type}$ is called a **precongruence** if the following type is inhabited:

$$\text{is-precong}_\alpha\, R := \prod_{x,y:X} R\,(x, y) \to E_F\, R\,(\alpha\, x, \alpha\, y)$$

The type of propositionally-valued precongruences on the coalgebra α is denoted $\mathrm{Precong}_\alpha$, and we write $\mathrm{Precong}^U_\alpha$ for the type of small precongruences. Every precongruence simple coalgebra is also U-precongruence simple, but the opposite implication is not necessarily true. It becomes true if we assume the principle of propositional resizing.

Definition 30 ( ). A coalgebra $\alpha : X \to FX$ is called **precongruence simple** if, for all $x, y : X$ such that $R\,(x, y)$ for some $R : \mathrm{Precong}_\alpha$, then also $x = y$. We call it **U-precongruence simple** if the latter holds for $R : \mathrm{Precong}^U_\alpha$.

Aczel and Mendler require the precongruence in the definition of simple coalgebra (which they call "s-extensional") to be a congruence, i.e. an equivalence relation on X. We do not require symmetry and transitivity, as reflexivity is sufficient for our purpouses. The terminology "simple" comes from Rutten (2000), denoting coalgebras for which bisimulation implies equality. We generalize the notion from bisimulation to reflexive precongruence.

The maximal precongruence on a coalgebra α is the propositional truncation of the disjoint union of all its small precongruences:

$$x \sim_\alpha y := \left\| \sum_{R : \mathrm{Precong}^U_\alpha} R\,(x, y) \right\|_{-1}$$

It is possible to show that $(\sim_\alpha) : \mathrm{Precong}_\alpha$ and, assuming propositional resizing, also $(\sim_\alpha) : \mathrm{Precong}^U_\alpha$. We can form the set quotient $X/{\sim_\alpha}$, which satisfies a number of important properties. First, $X/{\sim_\alpha}$ has an F-coalgebra structure $\alpha^q : X/{\sim_\alpha} \to F(X/{\sim_\alpha})$ defined by structural recursion. The case of the point constructor is given as follows: $\alpha^q\,[x]_{\sim_\alpha} := F\,[-]_{\sim_\alpha}(\alpha\, x)$. The constructor $[-]_{\sim_\alpha}$ is a coalgebra homomorphism between α and α^q. Moreover, the coalgebra α^q is U-precongruence simple.

Proposition 16 ( ). *The coalgebra $\alpha^q : X/{\sim_\alpha} \to F(X/{\sim_\alpha})$ is U-precongruence simple.*

Proof. Applying the elimination principle of set quotients, it is sufficient to show that given $x, y : X$, a propositionally-valued reflexive precongruence $R : X/{\sim_\alpha} \times X/{\sim_\alpha} \to U$ and a proof of $R\,([x]{\sim_\alpha}, [y]{\sim_\alpha})$, then $x \sim_\alpha y$. In other words, we need to find a small propositionally-valued reflexive precongruence $S : X \times X \to U$ such that $S\,(x, y)$. Take $S\,(a, b) := R\,([a]{\sim_\alpha}, [b]{\sim_\alpha})$. Notice that the types $(X/{\sim_\alpha})/R$ and X/S are isomorphic, and the underlying function $c : (X/{\sim_\alpha})/R \to X/S$ makes the following square commute:

$$\begin{array}{ccc} X & \xrightarrow{[-]_S} & X/S \\ {\scriptstyle [-]_{\sim_\alpha}}\big\downarrow & & \big\uparrow{\scriptstyle c} \\ X/{\sim_\alpha} & \xrightarrow[{[-]_R}]{} & (X/{\sim_\alpha})/R \end{array} \tag{41}$$

Let $a, b : X$ and suppose $S\,(a, b)$. The following sequence of equalities proves that S is a precongruence:

$$\begin{aligned} F\,[-]_S\,(\alpha\,a) &= F\,(c \circ [-]_R \circ [-]_{\sim_\alpha})\,(\alpha\,a) && (42)\\ &= F\,c\,(F\,[-]_R\,(\alpha^q\,[a]_{\sim_\alpha})) && (43)\\ &= F\,c\,(F\,[-]_R\,(\alpha^q\,[b]_{\sim_\alpha})) && (44)\\ &= F\,(c \circ [-]_R \circ [-]_{\sim_\alpha})\,(\alpha\,b) && (45)\\ &= F\,[-]_S\,(\alpha\,b) && (46) \end{aligned}$$

Step (42) follows by (41) and step (44) is the fact that R is a precongruence. Finally, in step (46) we use (41) again. □

Proposition 17 ( ). *Every U-precongruence simple coalgebra is U-simple.*

Proof. Let $\alpha : X \to FX$ be a precongruence simple coalgebra and let f, g be two coalgebra homomorphisms from another coalgebra $\beta : Y \to FY$ to α. For all $y : Y$ we need to show that $f\,y = g\,y$. From the precongruence simplicity of α, it is sufficient to find a reflexive precongruence relating $f\,y$ and $g\,y$. Consider the relation:

$$R'\,x\,x' :\equiv \sum_{y:Y} (x = f\,y) \times (x' = g\,y)$$

and its propositional reflexive closure $R\,x\,x' :\equiv \|R'\,x\,x' + (x = x')\|_{-1}$. It is not hard to show that R is a precongruence on α, which moreover relates $f\,y$ and $g\,y$ as $|\,\mathrm{inl}(y, \mathrm{refl}, \mathrm{refl})| : R\,(f\,y)\,(g\,y)$. □

Corollary 6. *Assuming propositional resizing, the coalgebra $\alpha^q : X/{\sim_\alpha} \to F(X/{\sim_\alpha})$ is U-simple.*

7.3 The U-terminal coalgebra

The U-terminal coalgebra of a functor F is built in two steps. First, define the *weakly* U-terminal coalgebra as the disjoint union of all small coalgebras:

$$w\nu F_U :\equiv \sum_{X:U} \sum_{\alpha : X \to FX} X. \tag{47}$$

Every small coalgebra $\alpha : X \to FX$ clearly injects in the union $\alpha^* : X \to w\nu F_U$, $\alpha^* \, x := (X, \alpha, x)$. The coalgebra structure $\zeta : w\nu F_U \to F(w\nu F_U)$ is given by $\zeta\,(X, \alpha, x) := F\,\alpha^*\,(\alpha\,x)$. It easy to prove that α^* is a coalgebra homomorphism between α and ζ.

In order to turn the weakly U-terminal coalgebra into a *strong* U-terminal coalgebra, we quotient its carrier $w\nu F_U$ by the maximal precongruence on ζ: $\nu F_U := w\nu F_U/{\sim_\zeta}$. We know this has a coalgebra structure ζ^q. Moreover, given a small coalgebra $\alpha : X \to FX$, there is a coalgebra homomorphism from it to ζ^q given by the composition of α^* and $[-]_{\sim_\zeta}$. Invoking Corollary 6, which assumes propositional resizing, we know that this is the only such coalgebra homomorphism.

Theorem 10 ( ). *Assuming propositional resizing, the coalgebra* $\zeta^q : \nu F_U \to F(\nu F_U)$ *is* U*-terminal.*

7.4 The Aczel–Mendler theorem

We finally show how the U-terminal coalgebra ζ^q is also *terminal* with respect to large coalgebras, provided the functor F is U-based.

First, notice that P^∞ is not only a polynomial functor, but a polynomial monad. Its unit $\eta : X \to \mathrm{P}^\infty\, X$ is $\eta\, x := (1, \lambda * .x)$. The Kleisli extension $\mathrm{bind}\, g : \mathrm{P}^\infty\, X \to \mathrm{P}^\infty\, Y$ of a function $g : X \to \mathrm{P}^\infty\, Y$ is obtained by forming the disjoint union of all indexing types:

$$\mathrm{bind}\, g\,(A, f) := \left(\sum_{a:A} \pi_0\,(g(f a)),\ \lambda(a, y).\, \pi_1\,(g(f a))y\right)$$

Given $g : X \to \mathrm{P}^\infty\, X$, its Kleisli extension can be iterated a finite number of times:

$$\begin{aligned} &\mathrm{bind} : \mathbb{N} \to (X \to \mathrm{P}^\infty\, X) \to \mathrm{P}^\infty\, X \to \mathrm{P}^\infty\, X \\ &\mathrm{bind}^0 \quad g\, z := z \\ &\mathrm{bind}^{n+1} g\, z := \mathrm{bind}\, g\,(\mathrm{bind}^n\, g\, z). \end{aligned} \tag{48}$$

It can also be iterated an infinite number of times, by collecting all the finite approximations:

$$\begin{aligned} &\mathrm{bind}^\infty\, g : \mathrm{P}^\infty\, X \to \mathrm{P}^\infty\, X \\ &\mathrm{bind}^\infty\, g\, z := \left(\sum_{n:\mathbb{N}} \pi_0\,(\mathrm{bind}^n\, g\, z),\ \lambda(n, x).\, \pi_1\,(\mathrm{bind}^n\, g\, z)x\right) \end{aligned}$$

Given a large coalgebra $\alpha : X \to FX$ for a U-based functor F, one can construct a P^∞-coalgebra structure on X as follows: $\widehat{\alpha} x := \pi_0\,(\mathit{base}_F\,(\alpha\, x))$.

Proposition 18 ( ). *Let F be a U-based functor and $\alpha : X \to F X$ a large coalgebra. For all $z : \mathrm{P}^{\infty}\, X$, there is a function $\alpha_z : \pi_0\, z \to F(\pi_0\, (\mathrm{bind}\, \widehat{\alpha}\, z))$ such that the following diagram commutes:*

$$\begin{array}{ccc} \pi_0\, z & \xrightarrow{\quad \pi_1\, z \quad} & X \\ {\scriptstyle \alpha_z}\big\downarrow & & \big\downarrow{\scriptstyle \alpha} \\ F(\pi_0\, (\mathrm{bind}\, \widehat{\alpha}\, z)) & \xrightarrow[F\, (\pi_1\, (\mathrm{bind}\, \widehat{\alpha}\, z))]{} & F\, X \end{array}$$

Proof. Let $a : \pi_0\, z$. Since F is U-based, there exist $A : U$, $\iota : A \to X$ and $y : F A$ such that $F\, \iota\, y = \alpha\, (\pi_1\, z\, a)$. In other words $y \equiv \pi_2\, (base_F\, (\alpha\, (\pi_1\, z\, a)))$. Take $\alpha_z\, a := F\, (\lambda x.\, (a, x))\, y$. □

The construction of Proposition 18 can be iterated, producing a family of functions $\alpha^n_z : \pi_0\, (\mathrm{bind}^n\, \widehat{\alpha}\, z) \to F(\pi_0\, (\mathrm{bind}^{n+1}\, \widehat{\alpha}\, z))$ indexed by a natural number n, which makes the following family of diagrams commute:

$$\begin{array}{ccc} \pi_0\, (\mathrm{bind}^n\, \widehat{\alpha}\, z) & \xrightarrow{\quad \pi_1\, (\mathrm{bind}^n\, \widehat{\alpha}\, z) \quad} & X \\ {\scriptstyle \alpha^n_z}\big\downarrow & & \big\downarrow{\scriptstyle \alpha} \\ F(\pi_0\, (\mathrm{bind}^{n+1}\, \widehat{\alpha}\, z)) & \xrightarrow[F\, (\pi_1\, (\mathrm{bind}^{n+1}\, \widehat{\alpha}\, z))]{} & F\, X \end{array} \tag{49}$$

Proposition 19 ( ). *Let F be a U-based functor and $\alpha : X \to F X$ a large coalgebra. Then each $z : \mathrm{P}^{\infty}\, X$ determines a small coalgebra $\alpha^{\infty}_z : X_z \to F(X_z)$ and a coalgebra homomorphism k_z from α^{∞}_z to α.*

Proof. Define the carrier X_z as $\pi_0\, (\mathrm{bind}^{\infty}\, \widehat{\alpha}\, z)$ and its coalgebra structure as

$$\alpha^{\infty}_z\, (n, x) := F\, (\lambda y.\, n + 1, y)\, (\alpha^n_z\, x).$$

There is a function $k_z\, (n, x) := \pi_1\, (\mathrm{bind}^n\, \widehat{\alpha}\, z)\, x$ between X_z and X. The fact that this is a coalgebra homomorphism between α^{∞}_z and α follows from the commutativity of the family of diagrams in (49). □

Notice also the existence of a function $u_z : \pi_0\, z \to X_z$ sending x to the pair $(0, x)$, which makes the triangle below commute. Since k_z is a coalgebra homomorphism, the square below also commutes:

$$\begin{array}{ccccc} & & \pi_0\, z & & \\ & {\scriptstyle u_z}\swarrow & & \searrow{\scriptstyle \pi_1\, z} & \\ X_z & & \xrightarrow{\quad k_z \quad} & & X \\ {\scriptstyle \alpha^{\infty}_z}\big\downarrow & & & & \big\downarrow{\scriptstyle \alpha} \\ F\, X_z & & \xrightarrow[F\, k_z]{} & & F\, X \end{array} \tag{50}$$

Given $z : \mathrm{P}^\infty X$, a multiset of elements in X, and $w : \mathrm{P}^\infty X_z$ a multiset of X_z, the latter determines also a multiset w' of X as follows: $w' := \mathrm{P}^\infty k_z\, w$. The small coalgebras associated to z and w' by Proposition 19 are in a strong relationship with each other. We refer the interested reader to the formalization for a proof of this technical lemma.

Lemma 8 ( ). *Let F be a U-based functor and $\alpha : X \to FX$ a large coalgebra. For all $z : \mathrm{P}^\infty X$ and $w : \mathrm{P}^\infty X_z$, there is a coalgebra homomorphism $l_{z,w}$ between $\alpha^\infty_{w'}$ and α^∞_z that makes the following diagram commute:*

$$\begin{array}{ccc} \pi_0\, w' & = \!= \!= \!= & \pi_0 w \\ {\scriptstyle u_{w'}}\downarrow & & \downarrow{\scriptstyle \pi_1\, w} \\ X_{w'} & \xrightarrow[l_{z,w}]{} & X_z \end{array} \tag{51}$$

We are now ready to prove the main result of Aczel and Mendler (1989).

Theorem 11 ( ). *Let F be a U-based functor. If a coalgebra is U-terminal then it is also terminal.*

Proof. Let $\beta : Y \to FY$ be a U-terminal coalgebra and let $\alpha : X \to FX$ be a large coalgebra. We construct a coalgebra homomorphism from α to β. Given $x : X$, we get $\eta\, x : \mathrm{P}^\infty X$ and therefore, by Proposition 19, a small coalgebra $\alpha^\infty_{\eta\, x} : X_{\eta\, x} \to F(X_{\eta\, x})$. From U-terminality, there exists a unique coalgebra homomorphism h_x between $\alpha^\infty_{\eta\, x}$ and β.

We now show how this homomorphism can be lifted to one initiating from the large coalgebra α. First, a function $h : X \to Y$ can be defined as $h\, x := h_x\,(u_{\eta\, x} *)$, which is a coalgebra homomorphism:

$$\begin{aligned} F\, h\,(\alpha\, x) &= F\, h\,(F\, k_{\eta\, x}\,(\alpha^\infty_{\eta\, x}\,(u_{\eta\, x} *))) && (52)\\ &= F\,(h \circ k_{\eta\, x})\,(\alpha^\infty_{\eta\, x}\,(u_{\eta\, x} *)) && (53)\\ &= F\, h_x\,(\alpha^\infty_{\eta\, x}\,(u_{\eta\, x} *)) && (54)\\ &= \beta\,(h_x\,(\alpha^\infty_{\eta\, x}\,(u_{\eta\, x} *))) && (55)\\ &\equiv \beta\,(h\, x) && (56) \end{aligned}$$

Step (52) follows from (50) and step (55) is the fact that h_x is a coalgebra homomorphism. The validity of step (54), i.e. the equation $h \circ k_{\eta\, x} = h_x$, can be justified as follows. Let $a : X_{\eta\, x}$ and define $a' : X$ as $a' := k_{\eta\, x}\, a$. We have the following sequence of equalities:

$$h\,(k_{\eta\, x}\, a) \equiv h_{a'}\,(u_{\eta\, a'} *) = h_x\,(l_{\eta\, x, \eta\, a}\,(u_{\eta\, a'} *)) = h_x\,(\pi_1\,(\eta\, a) *) \equiv h_x\, a$$

The second equality holds since $h_{a'}$ is the unique coalgebra homomorphism from $\alpha^\infty_{\eta\, a'}$ to β, and the fact that h_x and $l_{\eta\, x, \eta\, a}$ (which was introduced in

Lemma 8) are both coalgebra homomorphisms. The third equality is an instance of (51).

The coalgebra homomorphism h is unique. Given another one h' and an element $x : X$, we have the following sequence of equalities:

$$h\,x \equiv h_x\,(u_{\eta\,x}\,*) = h'\,(k_{\eta\,x}\,(u_{\eta\,x}\,*)) = h'\,(\pi_1\,(\eta\,x)\,*) \equiv h'\,x$$

The second equality holds since h_x is the unique coalgebra homomorphism from $\alpha^{\infty}_{\eta\,x}$ to β, and the fact that h' and $k_{\eta\,x}$ are both coalgebra homomorphisms. The third equality is an instance of the triangle in (50). □

Plugging together Theorems 10 and 11, we obtain the general terminal coalgebra theorem of Aczel and Mendler. Assuming propositional resizing, there is a U-terminal coalgebra $\zeta^q : \nu F_U \to F(\nu F_U)$ for any functor F. If the latter happens to be U-based, then this coalgebra is also terminal with respect to large coalgebras.

Theorem 12 ( ). *Let F be a U-based functor. Assuming propositional resizing, the coalgebra $\zeta^q : \nu F_U \to F(\nu F_U)$ is terminal.*

7.5 Adjusting the theorem for P^0

The powerset construction P^0 is not a functor, as it only acts on functions $f : X \to Y$ with locally small codomain. Y can also be restricted to be a set if one assumes propositional resizing. Crucially this means that the Aczel–Mendler theorem described so far does not apply to it. Luckily, this can be remedied with a few small modifications.

First, let us call F a **set-valued functor** if FX is a set and F acts exclusively on set-valued functions, i.e. its action on functions is typed $\prod_{X:\mathrm{Type},Y:\mathrm{Set}}(X \to Y) \to F\,X \to F\,Y$. Clearly P^0 is a set-valued functor in this sense, assuming propositional resizing.

The notion of U-basedness in Definition 27 also needs to be adjusted. Let Set_U be the type of sets in U. We now say that a set-valued functor is **Set_U-based** if, for any large *set* X : Set and $x : FX$, there is a small *set* $Y : \mathrm{Set}_U$, a function $\iota : Y \to X$ and element $y : FY$ such that $F\,\iota\,y = x$. In other words, both X and Y in the definition are required to be sets. This is important for the results of Section 7.4 to go though when functors only act on set-valued functions. For example, the bottom functions in (49) and (50) are well-defined only if X is a set. Similarly, the functions $l_{z,w}$ in Lemma 8 can only be a coalgebra morphism in case X_z is a set.

Proposition 20 ( ). *P^0 is Set_U-based.*

Proof. Let X : Set and $x : \mathrm{P}^0\,X$. Notice that $\pi_0\,x : U$ is a set, since $\pi_1\,x : \pi_0\,x \to X$ is an embedding and X is a set. Therefore we can return the triple consisting of the small set $\pi_0\, x$, the function $\pi_1\, x : \pi_0\, x \to X$ and the element $(\pi_0\, x, \mathrm{id}) : \mathrm{P}^0(\pi_0\, x)$. $\square$

The weakly U-terminal coalgebra in (47) also needs to be modified. This is because $w\nu F_U$ is not a set, so there cannot be any coalgebra homomorphism targeting it. The solution is to take its *set truncation* of $\|w\nu F_U\|_0$ instead. It is straightforward to define a coalgebra structure on it using the elimination principle of set truncation.

Finally, assuming that X is a set in the definition of Set_U-basedness restricts the notion of terminal coalgebra in Definition 26 to work only for coalgebras with a set carrier. We say that a coalgebra $\alpha : A \to FA$ is **terminal with respect to sets** if, for all B : Set and coalgebras $\beta : B \to FB$, the type of coalgebra homomorphisms from β to α is contractible.

With all these restrictions in place, the Aczel–Mendler Theorem 12 still works.

Theorem 13 ( ). *Let F be a Set_U-based set-valued functor. Assuming propositional resizing, the coalgebra $\zeta^q : \nu F_U \to F(\nu F_U)$ is terminal with respect to sets.*

As a corollary, we obtain a terminal coalgebra for the powerset functor P^0.

Corollary 7 ( ). *Assuming propositional resizing,* P^0 *admits a terminal coalgebra with respect to sets.*

8 Conclusion

In this paper we constructed a non-initial and non-terminal fixed point of the (restricted) powerset functor and showed that it is a model of material set theory with Scott's anti-foundation axiom. Moreover, we constructed the terminal coalgebra of the same functor, assuming propositional resizing. This is then a model of material set theory with Aczel's anti-foundation axiom.

8.1 Related work

The result that the subtype of coiterative sets, of the type of non-wellfounded trees, is a model of Scott's anti-foundation axiom can be found in the classical literature in the paper of D'Agostino and Visser (2002). They consider universes of multisets and define two functors, Δ and Γ, from sets to multisets for which the terminal coalgebras exist classically. They then show that the subclass of unisets (i.e. multisets with, coiteratively, only one occurrence of each element) for Δ and Γ are models of AFA and SAFA respectively. Their functor Γ corresponds to our functor P^{∞}, and their model of SAFA corresponds to our model, except, of course, that they work within the framework of classical set theory. They also prove in the same paper that the relation between nodes in graphs of having isomorphic unfolding trees is precisely the relation of Γ-bisimulation, which corresponds to our Theorem 1.

8.2 Future work

There are still some questions that remain unanswered, especially the initial motivation of this paper: to construct the terminal coalgebra of the powerset functor. The construction in the last section relies in a crucial way on propositional resizing. Is there a way to construct the terminal coalgebra, without any constructively questionable assumptions? Is it possible to show that assuming the existence of the terminal coalgebra implies some classical principle? Or is it independent altogether?

References

Abbot, Michael, Thorsten Altenkirch, and Neil Ghani (Sept. 2005). "Containers: Constructing strictly positive types". In: *Theoretical Computer Science* 342.1, pp. 3–27.

Aczel, Peter (1978). "The Type Theoretic Interpretation of Constructive Set Theory". In: *Logic Colloquium '77*. Ed. by A. MacIntyre, L. Pacholski, and J. Paris. North–Holland, Amsterdam-New York, pp. 55–66.

— (1988). *Non-well-founded sets*. Vol. 14. CSLI Lecture Notes. With a foreword by Jon Barwise [K. Jon Barwise]. Stanford University, Center for the Study of Language and Information, Stanford, CA, pp. xx+137. ISBN: 0-937073-22-9.

Aczel, Peter and Nax Mendler (1989). "A final coalgebra theorem". In: *Category Theory and Computer Science*. Ed. by David H. Pitt et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 357–365.

Ahrens, Benedikt, Paolo Capriotti, and Régis Spadotti (2015). "Non-wellfounded trees in homotopy type theory". In: *13th International Conference on Typed Lambda Calculi and Applications*. Vol. 38. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, pp. 17–30.

Altenkirch, Thorsten et al. (Jan. 2015). "Indexed containers". en. In: *Journal of Functional Programming* 25, e5. ISSN: 0956-7968, 1469-7653. DOI: 10.1017/S095679681500009X. URL: https://www.cambridge.org/core/journals/journal-of-functional-programming/article/indexed-containers/FB9C7DC88A65E7529D39554379D9765F.

D'Agostino, Giovanna and Albert Visser (2002). "Finality regained: a coalgebraic study of Scott-sets and multisets". In: *Archive for Mathematical Logic* 41.3, pp. 267–298. ISSN: 0933-5846. DOI: 10.1007/s001530100110. URL: https://doi.org/10.1007/s001530100110.

Gylterud, Håkon Robbestad (2018). "From multisets to sets in homotopy type theory". In: *The Journal of Symbolic Logic* 83.3, pp. 1132–1146. DOI: 10.1017/jsl.2017.84.

— (2019). "Multisets in type theory". In: *Mathematical Proceedings of the Cambridge Philosophical Society*, pp. 1–18. DOI: 10.1017/S0305004119000045.

Jong, Tom de and Martín Hötzel Escardó (2023). "On Small Types in Univalent Foundations". In: *Logical Methods in Computer Science* 19.2. DOI: 10.46298/LMCS-19(2:8)2023.

Lindström, Ingrid (1989). "A construction of non-well-founded sets within Martin-Löf's type theory". In: *The Journal of Symbolic Logic* 54.1, pp. 57–64. ISSN: 0022-4812. DOI: 10.2307/2275015. URL: https://doi.org/10.2307/2275015.

Rieger, Ladislav (1957). "A contribution to Gödel's axiomatic set theory, I". eng. In: *Czechoslovak Mathematical Journal* 07.3. Publisher: Institute of Mathematics, Academy of Sciences of the Czech Republic, pp. 323–357. ISSN: 0011-4642. URL: https://eudml.org/doc/11888.

Rijke, Egbert (2017). "The join construction". In: *ArXiv e-prints*. arXiv: 1701.07538 [math.CT].

Rijke, Egbert et al. (2024). *The agda-unimath library*. URL: https://github.com/UniMath/agda-unimath/.

Rutten, Jan J. M. M. (2000). "Universal coalgebra: a theory of systems". In: *Theoretical Computer Science* 249.1, pp. 3–80. DOI: 10.1016/S0304-3975(00)00056-6.

Shulman, Michael (Apr. 2010). "Stack semantics and the comparison of material and structural set theories". In: DOI: 10.48550/arXiv.1004.3802. URL: https://arxiv.org/abs/1004.3802.

Staton, Sam (2011). "Relating coalgebraic notions of bisimulation". In: *Log. Methods Comput. Sci.* 7.1. DOI: 10.2168/LMCS-7(1:13)2011.

Taylor, Paul (2023). *Well founded coalgebras and recursion*. URL: http://www.paultaylor.eu/ordinals/welfcr.pdf.

The Agda development team (2024a). *Agda*. URL: https://github.com/agda/agda.

— (2024b). *Cubical Agda*. URL: https://agda.readthedocs.io/en/v2.6.4.3/language/cubical.html.

The Univalent Foundations Program (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: homotopytypetheory.org. URL: http://homotopytypetheory.org/book.