Mathematical Structures in Computer Science (2019), 1–00
doi:10.1017/xxxxx

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

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

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

Department of Computer Science, Aarhus University, Denmark gratzer@cs.au.dk
Department of Informatics, University of Bergen, Norway hakon.gylterud@uib.no
Department of Mathematics, Stockholm University, Sweden anders.mortberg@math.su.se
Department of Informatics, University of Bergen, Norway elisabeth.stenholm@uib.no

(Received xx xxx xxx; revised xx xxx xxx; accepted xx xxx xxx)

Abstract

When working in Homotopy Type Theory and Univalent Foundations, the traditional role of the category of sets, $\mathcal{Set}$, is replaced by the category $h\mathcal{Set}$ of homotopy sets (h-sets); types with h-propositional identity types. Many of the properties of $\mathcal{Set}$ hold for $h\mathcal{Set}$ ((co)completeness, exactness, local cartesian closure, etc.). Notably, however, the univalence axiom implies that $\mathsf{Ob}\, h\mathcal{Set}$ 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.

Keywords: Set universes, Internal models, Univalent Foundations, Type Theory.

Competing interests: The authors declare no competing interests.

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*[a] (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 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

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

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, 2023)—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 et al., 2021)—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 by Gylterud and Stenholm (2023).

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, true, 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 (Escardó, 2019) and the basic system used in `UniMath` (Voevodsky 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 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 18 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, 2020).

Notation 2. *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, 2020, 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 3 ((Gylterud, 2020, Theorem 1),). *For two elements $x, y : \mathsf{V}^\infty$ the canonical map*

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

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 4 ((Gylterud, 2020, 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:

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

The first equivalence is the one constructed in Theorem 3. The second equivalence is proven by Gylterud (2020, 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 (2020). This elementhood relation, and its well-foundedness, will be used in later constructions.

Definition 5 (Elementhood,). *We define $\in : \mathsf{V}^\infty \to \mathsf{V}^\infty \to \mathcal{U}$ 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 4 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:

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

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:

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

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 $x, y : \mathsf{V}^\infty$:

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

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 the forthcoming paper (Gylterud and Stenholm, 2023) 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 6 ((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 7 ((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 8 (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 7, 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.[b] 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 12). 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 $\varphi : \mathsf{bool} \to \mathsf{V}^0$ is an embedding if it is injective:

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

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

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

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 10 (). *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 11 (). *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 12 (). V^0 *is an h-set.*

[b]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).

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 4. 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 (Gylterud and Stenholm, 2023).

Remark 13. 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 14 ( ). *We define the following function:*

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

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

Definition 15 ( ). *We define the following function:*

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

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 $\mathsf{V}^0 \simeq (\sum_{A:\mathcal{U}} A \hookrightarrow \mathsf{V}^0)$, realized by the maps sup^0 and desup^0 above. These results are due to Gylterud and Stenholm (2023), who extend 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 16 (Decoding,). *We define the decoding function* $\mathsf{El}^0 : \mathsf{V}^0 \to \mathcal{U}$ *by*

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

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

Theorem 17 ( ). *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 12 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 18 ( ). *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

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

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

We compute as follows:

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

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 19 (). *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 20 (). *V^0 is closed under the natural numbers.*

Proof. By Proposition 18 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 $\in$ (Gylterud and Stenholm, 2023).

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 21 ( ). *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 (Gylterud and Stenholm, 2023, Theorem 7). □

Proposition 22 ( ). V^0 *is closed under* Π*-types.*

Proof. Let $x : \mathsf{V}^0$ and $y : \mathsf{El}^0\, x \to \mathsf{V}^0$. By (Gylterud and Stenholm, 2023, Lemma 12) 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 23 ( ). 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 24 ( ). 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:

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

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 25 ( ). 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 26 ( ). *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 7 and for each $((x, y), p) : \mathsf{fib}\, f\, z$, the type $x = x_0$ is an h-proposition. □

Lemma 27 ( ). *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:

$$\begin{tikzcd} x = x' \arrow[rr, "\mathsf{ap}\,\mathsf{inl}"] \arrow[dr, "\mathsf{ap}\, f"'] & & \mathsf{inl}\, x = \mathsf{inl}\, x' \arrow[dl, "\mathsf{ap}\, h"] \\ & f\,x = f\,x' & \end{tikzcd}$$

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

Proposition 28 ( ). 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 26 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 27 we conclude that f is an embedding.

We now define the coproduct:

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

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

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

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

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:

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

The decoding holds up to definitional equality:

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

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 30. *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 31 ( ). *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 18 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:

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

Note that we have used the characterization of equivalence relations given by Lemma 30 in the next to last step. □

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 32 ( ). *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:

$$\begin{tikzcd} \sum_{X:\mathcal{U}_\ell} X \to \mathsf{V}^\infty_\ell \arrow[r, "\lambda(X,h).(X,\varphi\circ h)"] \arrow[d, "\mathsf{sup}"'] & \sum_{X:\mathcal{U}_{\ell^+}} X \to \mathsf{V}^\infty_{\ell^+} \arrow[d, "\mathsf{sup}"] \\ \mathsf{V}^\infty_\ell \arrow[r, "\varphi"'] & \mathsf{V}^\infty_{\ell^+} \end{tikzcd}$$

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.

$$\begin{tikzcd} (A, f) = (B, g) \arrow[r, "\mathsf{ap}\ (\lambda(X,h).(X,\varphi\circ h))"] \arrow[d, "\simeq"'] & (A, \varphi \circ f) = (B, \varphi \circ g) \arrow[d, "\simeq"] \\ \sum_{e:A\simeq B} f \sim g \circ e \arrow[r, "\lambda(e,H).(e,\mathsf{ap}\ \varphi\circ H)"'] & \sum_{e:A\simeq B} \varphi \circ f \sim \varphi \circ g \circ e \end{tikzcd}$$

The vertical maps are provided by Theorem 3. 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 33. 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 34 ( ). *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 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 35 ( ). *The map* El^0 *induces a full and faithful functor from $\mathcal{V}$ to $\mathcal{hSet}_{\mathcal{U}}$.*

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 36 ( ). *$\mathcal{V}$ 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 19, 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 37. *$\mathcal{V}$ 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 38 ( ). *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 39 ( ). *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 40. *$\mathcal{V}$ 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 et al., 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 41 ( ). *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 42. 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 43 (Category with families,). *A category with families (CwF) consists of:*

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

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

Here we have written $A \cdot \gamma$ for $\mathsf{Ty}\,(\gamma A)$, and $\int \mathsf{Ty}$ for the *category of elements* of Ty, i.e., the total space of the right fibration induced by Ty. 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 $\mathsf{Ty}\,\Gamma$ while the set of terms of type $A : \mathsf{Ty}\,\Gamma$ is given by $\mathsf{Tm}\,(\Gamma, A)$. The functoriality of Ty and Tm 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 $\int \mathsf{Ty}$ to $\mathcal{C}$ interprets context extension. A context $\Gamma : \mathsf{Ob}\,\mathcal{C}$ can be extended by a type $A : \mathsf{Ty}\,\Gamma$ in that context, to produce a new context $\Gamma.\,A : \mathsf{Ob}\,\mathcal{C}$.

Finally, the natural equivalence ties together substitutions and elements of Tm. 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 $(\Gamma, a) : \sum_{\gamma : \mathsf{Hom}\,(\Delta,\Gamma)} \mathsf{Tm}\,(\Delta, A \cdot\ \gamma)$.

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

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

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

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 $\mathsf{Ty}\,X$ 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:

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

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

Proposition 44 (). *$\mathcal{V}$ can be equipped with a CwF structure.*

Proof. We have given the putative definitions of Ty and Tm. 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 $A : \mathsf{Ty}\, X$. We define $X.\, A : \mathsf{V}^0$ as $\Sigma^0\, X\, A$. The natural equivalence then follows from the η principle of dependent sums. $\square$

By virtue of Proposition 41, 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 Ty and Tm 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 45. 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, 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 $(\mathcal{V}, \mathsf{Ty}, \mathsf{Tm})$ 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 $(\mathcal{V}, \mathsf{Ty}, \mathsf{Tm})$ supports this additional structure.

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

Definition 46 (Π-structure,). *A Π-structure on a CwF $(\mathcal{C}, \mathsf{Ty}, \mathsf{Tm})$ is defined by the following:*

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

Lemma 47 ( ). *The CwF $(\mathcal{V}, \mathsf{Ty}, \mathsf{Tm})$ supports a Π-structure.*

Proof. We begin by defining pi as follows:

$$\mathsf{pi}\, X\, A\, B := \lambda (x : \mathsf{El}^0\, (X)).\, \Pi^0\, (A\, x)\, (\lambda a.B\, (x, a))$$

Naturality in X is a straightforward computation. The definition of α_{pi}, after unfolding, reduces to the manifestly natural equivalence induced by currying:

$$\prod_{x:X} \prod_{y:Y(x)} Z(x,y) \simeq \prod_{p:\sum_{x:X} Y(x)} Z(p)$$

$\square$

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

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

Definition 48 (Σ-structure)**.** *A Σ-structure on a CwF $(\mathcal{C}, \mathsf{Ty}, \mathsf{Tm})$ consists of the following two pieces of data:*

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

Lemma 49. *The CwF $(\mathcal{V}, \mathsf{Ty}, \mathsf{Tm})$ supports a Σ-structure.*

Proof. We define sig as follows:

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

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. $\square$

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

Theorem 50. *$\mathcal{V}$ supports a model of extensional type theory with the standard connectives.*

We note that this result, combined with Proposition 41 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, 2012) models the core system of `Coq` system in intuitionistic ZF, and formalizes the model in `Coq` (The `Coq` Development Team, 2021). 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, 2020) who observed that this gives a universe of multisets, but that one can restrict it, as in Definition 9, 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 (Escardó and de Jong, 2023). Among many other things, they have two more proofs of Theorem 12 formalized. Various HITs for representing *finite* multisets have also been considered in HoTT/UF (Basold et al., 2017;

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* (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 (MEWOs)

In their recent paper, 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 $hSet_{\mathcal{U}}$

One reason to consider the category of iterative sets is to regard it as a replacement for $hSet_{\mathcal{U}}$. 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 $hSet_{\mathcal{U}}$ as the *Rezk completion* (Ahrens et al., 2015) of $\mathcal{V}$. Informally, this shows, modulo classical axioms, that $\mathcal{V}$ is a more rigid presentation of $hSet_{\mathcal{U}}$. 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 (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 $hSet_{\mathcal{U}}$. This was experimented with in `UniMath` (Voevodsky 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 50.

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 42 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 (Cohen et al., 2018) 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 et al., 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 $hSet_{\mathcal{U}}$ 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 $\mathsf{El}^0 : \mathsf{V}^0_{\mathcal{U}} \to hSet_{\mathcal{U}}$ 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 $\mathit{hSet}_{\mathcal{U}}$ 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, P. 1978. The Type Theoretic Interpretation of Constructive Set Theory. In **Macintyre, A.**, **Pacholski, L.**, **and Paris, J.**, editors, *Logic Colloquium '77*, volume 96 of *Studies in Logic and the Foundations of Mathematics*, pp. 55–66. Elsevier.

Aczel, P. 1999. On Relating Type Theories and Set Theories. In **Altenkirch, T.**, **Reus, B.**, **and Naraschewski, W.**, editors, *Types for Proofs and Programs*, pp. 1–18, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ahrens, B., **Kapulkin, K.**, **and Shulman, M.** 2015. Univalent categories and the Rezk completion. *Mathematical Structures in Computer Science*, 25(5):1010–1039.

Altenkirch, T. and Kaposi, A. 2016. Type theory in type theory using quotient inductive types. *SIGPLAN Not.*, 51(1):18–29.

Angiuli, C., **Cavallo, E.**, **Mörtberg, A.**, **and Zeuner, M.** 2021. Internalizing representation independence with univalence. *Proc. ACM Program. Lang.*, 5(POPL).

Awodey, S. 2013. Structuralism, invariance, and univalence. *Philosophia Mathematica*, 22(1):1–11.

Awodey, S. 2018. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286.

Barras, B. 2010. Sets in Coq, Coq in Sets. *Journal of Formalized Reasoning*, 3:29–48.

Barras, B. 2012. Semantical investigations in intuitionistic set theory and type theories with inductive families. Available at `http://www.lsv.fr/~barras/habilitation/`.

Basold, H., **Geuvers, H.**, **and van der Weide, N.** 2017. Higher Inductive Types in Programming. *Journal of Universal Computer Science*, 23(1):63–88.

Cartmell, J. 1986. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243.

Choudhury, V. and Fiore, M. 2019. The finite-multiset construction in HoTT. Talk at the International Conference on Homotopy Type Theory (HoTT 2019).

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

Curien, P.-L., **Garner, R.**, **and Hofmann, M.** 2014. Revisiting the categorical interpretation of dependent type theory. *Theoretical Computer Science*, 546:99–119. Models of Interaction: Essays in Honour of Glynn Winskel.

de Jong, T., **Kraus, N.**, **Forsberg, F.**, **and Xu, C.** 2023. Set-Theoretic and Type-Theoretic Ordinals Coincide. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–13, Los Alamitos, CA, USA. IEEE Computer Society.

Dybjer, P. 1996. Internal type theory. In **Berardi, S. and Coppo, M.**, editors, *Types for Proofs and Programs (TYPES 1995)*, volume 1158 of *Lecture Notes in Computer Science*, pp. 120–134, Berlin, Heidelberg. Springer Berlin Heidelberg.

Escardó, M. and de Jong, T. 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, M. H. 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, M. 2012. Discrete generalised polynomial functors. Slides from talk given at ICALP 2012.

Frumin, D., **Geuvers, H.**, **Gondelman, L.**, **and van der Weide, N.** 2018. Finite Sets in Homotopy Type Theory. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2018, pp. 201–214, New York, NY, USA. Association for Computing Machinery.

Gylterud, H. R. 2018. From multisets to sets in Homotopy Type Theory. *The Journal of Symbolic Logic*, 83(3):1132–1146.

Gylterud, H. R. 2020. Multisets in type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 169(1):1–18.

Gylterud, H. R. and Stenholm, E. 2023. Univalent Material Set Theory. arXiv:2312.13024 [math].

Hofmann, M. 1997. *Syntax and Semantics of Dependent Types*, pp. 79–130. Publications of the Newton Institute. Cambridge University Press.

Jacobs, B. 1993. Comprehension categories and the semantics of type dependency. *Theoretical Computer Science*, 107(2):169–207.

Jacobs, B. 1999. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland.

Joram, P. and Veltri, N. 2023. Constructive Final Semantics of Finite Bags. In **Naumowicz, A. and Thiemann, R.**, editors, *14th International Conference on Interactive Theorem Proving (ITP 2023)*, volume 268 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 20:1–20:19, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Kapulkin, K. and Lumsdaine, P. L. 2021. The Simplicial Model of Univalent Foundations (after Voevodsky). *Journal of the European Mathematical Society*, 23:2071–2126.

Leversha, G. 1976. *Formal Systems for Constructive Mathematics*. PhD thesis, University of Manchester.

Martin-Löf, P. 1975. An Intuitionistic Theory of Types: Predicative Part. In **Rose, H. E. and Shepherdson, J. C.**, editors, *Logic Colloquium '73*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pp. 73–118. North-Holland.

Martin-Löf, P. 1982. Constructive mathematics and computer programming. In **Cohen, L. J.**, **Łoś, J.**, **Pfeiffer, H.**, **and Podewski, K.-P.**, editors, *Logic, Methodology and Philosophy of Science VI, Proceedings of the Sixth International Congress of Logic, Methodology and Philosophy of Science, Hannover 1979*, volume 104 of *Studies in Logic and the Foundations of Mathematics*, pp. 153–175. North-Holland.

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

Mörtberg, A. 2018. Category structures on well-ordered sets in UniMath. Available at https://github.com/UniMath/UniMath/blob/master/UniMath/CategoryTheory/categories/wosets.v.

Palmgren, E. 2019. From type theory to setoids and back. arXiv:1909.01414 [math].

Rijke, E. and Spitters, B. 2015. Sets in homotopy type theory. *Mathematical Structures in Computer Science*, 25(5):1172–1202.

Rijke, E., **Stenholm, E.**, **Prieto-Cubides, J.**, **Bakke, F.**, **and others** 2021. The agda-unimath library. https://github.com/UniMath/agda-unimath/.

Salvesen, A. B. 1984. *Typeteori - en studie*. Master's thesis, University of Oslo.

Seely, R. A. G. 1984. Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(1):33–48.

Shulman, M. 2010. Stack semantics and the comparison of material and structural set theories. arXiv:1004.3802 [math.CT].

Streicher, T. 1991. *Semantics of Type Theory*. Birkhäuser Boston.

The `Agda` **Development Team** 2023. The `Agda` Programming Language.

The `Coq` **Development Team** 2021. The `Coq` Proof Assistant.

The Univalent Foundations Program 2013. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Self-published, Institute for Advanced Study.

Veltri, N. 2021. Type-Theoretic Constructions of the Final Coalgebra of the Finite Powerset Functor. In **Kobayashi, N.**, editor, *6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)*, volume 195 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 22:1–22:18, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Vezzosi, A. 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, A., **Mörtberg, A.**, **and Abel, A.** 2021. Cubical Agda: A dependently typed programming language with univalence and higher inductive types. *Journal of Functional Programming*, 31:e8.

Voevodsky, V. 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.

Voevodsky, V. 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.

Voevodsky, V. 2015. An experimental library of formalized mathematics based on the univalent foundations. *Mathematical Structures in Computer Science*, 25(5):1278–1294.

Voevodsky, V., **Ahrens, B.**, **Grayson, D.**, **and others** 2020. UniMath — a computer-checked library of univalent mathematics. https://github.com/UniMath/UniMath.

Werner, B. 1997. Sets in Types, Types in Sets. In *Proceedings of the Third International Symposium on Theoretical Aspects of Computer Software*, TACS '97, pp. 530–346, Berlin, Heidelberg. Springer-Verlag.