THE PURELY ITERATIVE CONCEPTION OF SET

ANSTEN KLEV

Abstract. According to the iterative conception of set, sets are formed in stages. According to the *purely* iterative conception of set, sets are formed by iterated application of a set-of operation. The cumulative hierarchy is a mathematical realization of the iterative conception of set. A mathematical realization of the purely iterative conception can be found in Peter Aczel's type-theoretic model of constructive set theory. Although Aczel's work is of obvious relevance to the philosophy of set theory, it is only rarely discussed in the philosophical literature. Here Aczel's model construction will be explained in a way that presupposes no previous familiarity with the theories on which it is based.

1. Introduction

According to the purely iterative conception of set, a set is anything formed by iterated application of a set-of operation, starting from the empty set or Urelemente. The adverb "purely" serves to distinguish this conception of set from the conception according to which sets are formed in, or at, stages. This latter description is a common gloss on the iterative conception of set (without the adverb), though the conception it glosses might perhaps better be called the stage conception of set. These are different conceptions of set, and it is useful to keep them apart terminologically.

In the cumulative and similarly defined set-theoretical hierarchies we have a mathematical realization of the stage conception of set. The aim of this article is to draw the reader's attention to, and explain, a mathematical realization of the purely iterative conception, namely a model of set theory where sets are formed by iterated application of a set-of operation. The model was constructed by Peter Aczel as part of an interpretation of a set theory known as Constructive ZF, or CZF, inside Per Martin-Löf's constructive type theory (Aczel, 1978, 1982). To some readers, such an interpretation might seem like a perfect case of explaining the obscure by the more obscure. As I hope to demonstrate, the main idea behind Aczel's model construction can be explained without presupposing any prior familiarity with these, to the reader perhaps obscure, formal systems. It is enough to be conversant with the notion of function and to be open to a few novel bits of symbolism.

If the law of excluded middle is added to CZF, then classical ZF results. It follows that if the law of excluded middle is added to constructive type theory, then Aczel's model interprets classical ZF. The model can likewise be made to interpret the set-theoretical Axiom of Choice, namely if the so-called Extensional Axiom of Choice is added to constructive type theory (Martin-Löf, 2006). As we shall see, also certain large cardinal notions can be accounted for. Although the presentation below will be constructivist in spirit, it should therefore also be of relevance to the classically minded.

Preliminary remarks on the two theories involved in Aczel's interpretation are given in Section 2. The contrast between the purely iterative and the stage conception of set will be further developed in Section 3. The rest of the article is dedicated to Aczel's model construction, starting in Section 4 with a general discussion of the set-of operation.

In Aczel's model, every set is formed by the application of a set-of operation to a function enumerating sets "already formed" (this temporal language is metaphorical and—as the formal definition of the model shows—disposable). The domains of these functions are, in Aczel's original definition, culled from constructive type theory. It is, however, possible to carry out essentially the same construction using only finite types as domains. The result, explained in Sections 5-6, is the universe of hereditarily finite sets. From this construction it is a short conceptual step to Aczel's full model, which will be briefly described in Section 7. Expanding the domains of the enumerating functions even further yields more encompassing models of set theory. In Section 8, I shall attempt to give the reader a sense of how such an expansion can be carried out while adhering to constructivist tenets.

Our presentation of Aczel's model construction will be elementary, presupposing no previous exposure to constructive type theory. Among the many philosophical implications of the construction, our primary interest will be with those that concern the iterative conception of set. Another philosophical implication, on which a brief remark may be inserted here, concerns the relation between type theory and set theory (for further discussion, see Klev, 2019a). In the light of Aczel's result, universes of sets appear as just some among the indefinitely many domains, or types, of mathematical object that there are. Domains of sets might be especially useful for certain metamathematical purposes, but they are not the only domains of mathematical objects. A pluralist ontology underlies type theory. The natural numbers, the integers, the rational numbers, the real numbers, the complex numbers, the Cartesian product of any of these with another, the space of functions from one to another, etc., all constitute their own autonomous domain, not reduced to elements in a universe of sets. By the same account, Aczel's model construction does not reduce the notion of set to the notion of type, in the way, say, the notions of ordered pair and function are reduced to the notion of set in set theory. Rather, it embeds sets into a richer mathematical ontology that also accommodates other types of mathematical object.

2. Preliminaries

Both constructive type theory and constructive set theory grew out of attempts to provide a foundation for constructive mathematics in the style of Bishop (1967). They follow quite different methodologies, however. Constructive set theory is meant to remain as close as possible to the formalism of classical axiomatic set theory. The theory CZF, in particular, is written in the language of single-sorted first-order predicate logic with identity and a constant, $\in$, for the binary relation of membership. Constructive type theory, by contrast, starts from scratch with its own conception of mathematical ontology and mathematical language. It accepts the Fregean function/argument analysis of logical grammar only to a limited extent, namely in its account of propositions. The basic unit of expression is not the proposition, but a larger structure called a judgement. The form of judgement

which we shall mainly be employing in this article is

$$a : \tau$$

to be read as "a is an object of type τ". A type is a sort, kind, or category of object, hence a judgement of this form places a within its sort, kind or category. If we assume that there is a type V of sets, then $v : V$ says that v is a set. The set-theoretical proposition $u \in v$, by contrast, expresses that the relation of membership holds between the two sets u and v. (Since membership is defined only between sets, the proposition $u \in v$ presupposes the two judgements $u : V$ and $v : V$.)

In the axioms of CZF, the most conspicuous difference with classical ZF is a weakening of Separation and Power Set. A set-theoretical formula is said to be bounded if all of its quantifiers occur as $\forall x \in y$ or $\exists x \in y$, given the usual contextual definition of these formula fragments. The classical Separation axiom of ZF is replaced in CZF by Bounded Separation, meaning that only bounded formulae may be used in separating a subset out of a given set. This restriction, as well as that on Power Set (which we shall not explain), is motivated by the requirement of predicativity. For instance, a definition of a subset of the natural numbers that quantifies over all sets is impredicative, since the set to be defined itself belongs to that range. A bounded formula, by contrast, quantifies only over the elements of sets "already defined". Aczel (1978) proved that, in spite of this weakening of the axioms, it is enough to add the law of excluded middle to CZF to obtain ZF. For more background on CZF and other constructive set theories, the reader might consult Crosilla (2020).

Although the notion of axiom makes sense within constructive type theory, this theory is more naturally regarded as entirely rules-based. Of special importance are the introduction and elimination rules associated with type-forming operators. (The terminology stems from Gentzen.) Suppose that A is a type formed by means of the type-forming operator Φ. The introduction rule for Φ prescribes how an object of type A is formed out of its parts. For instance, if A is a type of the form $B \times C$, then the introduction rule for $\times$ prescribes that an object of type A has the form $\langle b, c \rangle$, where b is of type B and c is of type C. The first step in Aczel's model construction is the provision of an introduction rule for a type $\mathbb{V}$. Under the finely grained notion of identity native to the type $\mathbb{V}$, it is not a type of sets, but rather a type of well-founded trees. As we shall see, we obtain a universe of sets by equipping $\mathbb{V}$ with relations of set equality and membership, yielding a structure $(\mathbb{V}, \doteq, \in)$. The definitions of these relations rely on the possibility of defining functions and predicates by induction on $\mathbb{V}$. Principles of proof and definition by induction are incorporated as the elimination rules associated with the type-forming operators. For instance, the elimination rule for the type $\mathbb{N}$ of natural numbers is simultaneously a principle of proof by induction and a principle of definition by induction (or recursion) on $\mathbb{N}$, and ditto for the elimination rule for the type $\mathbb{V}$.

The reader might balk at our apparently cavalier way in this article of postulating types and objects. There is method in this postulation, grounded in Martin-Löf's meaning explanations for his type theory (Martin-Löf, 1982, 1984, 1993). One is free to postulate a type or type former provided one gives the introduction rule, or rules, for it. An introduction rule, such as the rules ($\mathbb{HF}$-intro) and ($\mathbb{V}$-intro) below,

is a stipulation, but it is subject to the requirement that it serves to define the type or types in question. The introduction rules for a type or type former make up the clauses in an inductive definition, and one can make precise the requirements that must be met for such a definition to succeed (Dybjer, 1994, 2000), just as one can make precise the requirements that must be met for an explicit definition to succeed (in particular, any vicious form of circularity must be avoided). Taking this method to lie at the basis of mathematics brings with it a number of consequences for the philosophy of mathematics. It lies far outside the scope of this article to discuss these consequences. The reader is merely asked to grant, if only for the sake of argument, that the method is a rational one, and in particular, that it is not an instance of anything-goes postulationism.

3. The stage conception of set

The phrase "iterative conception of set" suggests a conception of set according to which sets are formed by iteration of some kind. The iteration in question is naturally spelled out as iterated set formation, which in turn may be spelled out as iterated application of a set-of operation. Gödel described such a conception of set, contrasting it to a class conception of set, already in 1947:

> This concept of set, however, according to which a set is something obtainable from the integers (or some other well-defined objects) by iterated application of the operation "set of", and not something obtained by dividing the totality of all existing things into two categories, has never led to any antinomy whatsoever. (Gödel, 1947, p. 519)

A set relative to a domain $\mathcal{D}$ of "well-defined objects" is anything that can be obtained from $\mathcal{D}$ by iterated application of the set-of operation. Unlike the naive class-conception of set, this conception has never led to any contradiction.

As usually presented in the literature, the iterative conception of set does not involve a set-of operation. A set-of operation features prominently in the stimulating presentation of Forster (2008), but it remains there an informal notion, likened to a wand that turns a collection of objects into the set of them. Standard presentations of the iterative conception of set rather follow Shoenfield (1967, 1977) and Boolos (1971, 1989) and emphasize the stage-wise formation of sets:[1] sets are formed in, or at, stages. At the initial stage, there are no sets. Passing from any stage, α, to the next, $\alpha + 1$, every collection of sets available at stage α that includes at least one set formed at stage α is made into a set. Thus, at stage one, following the initial stage, the empty set, and only the empty set, is formed. At stage two, the singleton of the empty set, $\{\emptyset\}$, is formed. At stage three, the sets $\{\{\emptyset\}\}$ and $\{\emptyset, \{\emptyset\}\}$ are formed, and so on. Also limit stages—stages, different from the initial stage, with no immediately preceding stage—are postulated. The sets available at a limit stage are all the sets formed before that stage.

A natural set-theoretic realization of this *stage conception of set*, as we might call it, is the cumulative hierarchy of sets. Stages are identified with ordinals, and the sets available at a stage themselves form a set, generated by means of the power-set and the union operations. Writing V_α for the set of sets available at stage α, where

[1] Recent examples include Barton (2024, ch. 4), Incurvati (2020, ch. 2), Linnebo (2017, ch. 10).

α is an ordinal, the cumulative hierarchy is defined by induction on the ordinals as follows:

$$\begin{array}{rcll} V_0 & = & \emptyset & \text{the empty set} \\ V_{\alpha+1} & = & \wp(V_\alpha) & \text{the power set of } V_\alpha \\ V_\lambda & = & \bigcup_{\alpha<\lambda} V_\alpha & \text{the union over all the } V_\alpha\text{'s for } \alpha < \lambda, \\ & & & \text{if } \lambda \text{ is a limit ordinal} \end{array}$$

The set of sets formed at stage $\alpha + 1$ is $V_{\alpha+1} - V_\alpha$, that is, the set of sets in $V_{\alpha+1}$ that are not in V_α.

From the definition of the cumulative hierarchy it may appear that sets are formed by iterated application of the power-set operation, taking unions at limit stages. Each set V_α is, however, just a record of all the sets available at stage α, that is, of all the sets formed at the stages up to and including stage α. In the definition of the cumulative hierarchy, the power-set operation is not used for generating new sets: it is an accumulator, not a generator. How the sets in $V_{\alpha+1} - V_\alpha$ are formed in the first place, how they are generated from those in V_α, is not accounted for by the definition of the cumulative hierarchy. This is so, mutatis mutandis, also in similarly defined hierarchies. Gödel's constructible hierarchy, for instance, is silent on how the sets in $L_{\alpha+1} - L_\alpha$ are formed. Although we can describe precisely what these sets are, we cannot say, on the basis of the definition of the constructible hierarchy, how the sets in $L_{\alpha+1} - L_\alpha$ are formed from those in L_α.

The sets in any V_α should thus not be thought as having been formed by the iteration of the power-set and union operations. These operations generate the sets V_α, but not, in general, every element of a V_α. Iteration as such, even, is not essential for characterizing the sets V_α, as shown by Button (2021). He characterizes the V_α's in abstract set-theoretical terms that involve no reference to iteration. More precisely, he offers a simple second-order theory in the language of set theory whose models, up to isomorphism, are precisely the V_α's.

Against this background, it seems to me useful to distinguish the stage conception of set described by Boolos, Shoenfield, and others from the purely iterative conception of set described by Gödel in the quotation above. There might be a common conception that unites them both. Under the purely iterative conception, the transitive closure of a set is a well-founded tree. If stages are required merely to be well-founded, hence if they may fail to be linearly ordered, then the tree that is the transitive closure of a set, v, may be identified with the stage at which v is formed. Each set, v, is formed at a unique stage, identified with the well-founded tree that is the transitive closure of v. A suitably general stage conception, such as Barton's weak iterative conception (Barton, 2024, p. 26), may therefore be taken to comprehend the purely iterative conception. To conceive of sets primarily as something formed in stages is, however, quite different from conceiving of sets as something formed by iteration of a set-of operation. A philosopher reflecting on the foundations of set theory ought therefore to keep the two conceptions apart. A good aid to that end are different mathematical realizations of the two conceptions, such as the cumulative hierarchy and Aczel's model, respectively. I will now proceed to explain the latter, starting from some general reflections on the set-of operation.

4. The set-of operation

The set-of operation takes certain objects and forms the set of them. Such an operation is often appealed to in informal descriptions of the notion of set, as for instance in the classical definitions (or "definitions") of Cantor (1895) or Dedekind (1888). The ubiquitous curly-braces notation is naturally understood as signifying a set-of operation: $\{a, b, c, \ldots\}$ is the set of $a, b, c, \ldots$, and $\{x \mid P(x)\}$ is the set of x's such that $P(x)$ is true. Yet a set-of operation is not a primitive notion of axiomatic set theory, which in the codification of Zermelo (1908) takes for granted, besides the notion of set itself, only the membership relation as a non-logical primitive.

Attempts to base a rigorous treatment of set theory on a set-of operation does indeed face serious challenges. Being an operation, it must be determined what the set-of operation operates on, what its admissible operands are. According to naive set theory, the set-of operation applies to a propositional function $P(x)$ over some underlying domain $\mathcal{D}$ and forms the set $\{x \mid P(x)\}$. Since no restrictions are placed on the function $P(x)$, and sets are themselves regarded as objects belonging to the domain $\mathcal{D}$, this principle leads, in by now familiar ways, to contradiction (though see Aczel, 1980). According to an alternative, combinatorial view, the set-of operation applies to a list of objects $a, b, c, \ldots$ to form the set $\{a, b, c, \ldots\}$, an object different from each of its members $a, b, c, \ldots$ This approach may avoid inconsistency, but does not obviously deliver infinite sets, since the notion of a list would appear to be finitary.[2]

Aczel, in his model construction, was able to overcome these difficulties by letting the set-of operation apply to functions with codomain the universe of sets. Since his aim was to give a type-theoretic interpretation of set theory, he could take type theory for granted, meaning that he had access, independently of set theory, to the notions of function and the domain and codomain of a function. (Recall that the domain of a function is the type of its arguments, whereas the codomain is the type of its values.) Let f be a function with domain A and codomain V, the universe of sets. The function may be thought of as enumerating sets $f(a)$ already formed. Aczel's set-of operation lets us form $\{f(a) \mid a : A\}$, the set of $f(a)$'s as a ranges over A.

What sets can be formed in this way depends of course on what domains are available for the enumerating functions and what means are available for defining functions on such domains. In constructive type theory, the domains available are required to be inductively generated in some way, meaning, intuitively, that each object of such a domain is systematically built up from its parts. Functions may be defined either explicitly or by induction on the build-up of the objects of the domain (think of definition by recursion on the natural numbers as a paradigmatic example). All of this may seem restrictive, but it suffices for constructing a model of CZF, and indeed also of proof-theoretically stronger set theories. The Axiom of Infinity, in particular, will be validated in such a model, as we shall see in Section 7.

[2]Such was the conclusion of Vopěnka (1979, p. 17), who went on to develop a theory of sets based on this finitist conception. Linnebo (2010, 2013) applies a combinatorial set-of operation to pluralities rather than lists and is able on that basis to validate classical ZF set theory.

5. The type $\mathbb{HF}$

5.1. **Finite type theory.** In explaining Aczel's model construction, I shall, for pedagogical reasons, begin by restricting myself to finite sets. Only finite types will be employed as domains of the enumerating functions. The resulting simpler model construction requires hardly any type theory at all, yet it incorporates all the main ideas of Aczel's original construction.

The first step is the definition of a type $\mathbb{HF}$, on which, in a second step, relations of extensional equality and membership will be defined, giving rise to a structure $(\mathbb{HF}, \doteq, \in)$ that is naturally regarded as the universe of hereditarily finite sets.

For the definition of $\mathbb{HF}$ we shall need one ground type of each finite cardinality. To achieve this in a systematic way, we postulate an empty type, $\bot$, and the following rule of type formation: whenever A is a type, then so is $S(A)$, the successor type of A.[3] The types we have available are therefore

$$\bot, S(\bot), S^2(\bot), \ldots, S^n(\bot), \ldots$$

The type $S(A)$ is stipulated to contain a copy of each object of A together with one additional object, whence its cardinality is one greater than that of A. Since there are zero objects of type $\bot$, and the cardinality increases by one with each application of S, there are precisely n objects of type $S^n(\bot)$. We shall use the suggestive notation $\mathbf{n}$ for $S^n(\bot)$, and we shall write

$$0_{\mathbf{n}}, \ldots n-1_{\mathbf{n}}$$

for the n objects of type $\mathbf{n}$.

These are the ground types. Besides the ground types, we shall also need function types. More precisely, we shall need unary first-order functions, i.e., functions whose domain and codomain are ground types. We stipulate that whenever A and B are ground types, then $(A)B$ is a type, namely the type of functions from A to B. A judgement of the form

$$f : (A)B$$

therefore means that f is a function from A to B. The reader is referred to (Martin-Löf, 1993, Lecture 4) for a detailed explanation of functions and function types. Here it suffices to say that a function f from A to B can be *applied* to every object of type A, and when applied to a of type A, it yields an object $f(a)$ of type B.

Since the type $\mathbf{n}$ is given by a list of n distinct objects, we may define a function f on $\mathbf{n}$ by specifying the value of $f(a)$ for each a of type $\mathbf{n}$. We call this definition by cases. It is what definition by induction on the build-up of objects becomes over a finite domain. A function f may also be defined in terms of already introduced functions by means of a single equation, $f(x) \equiv t[x]$. This is called explicit definition. A degenerate case of definition by cases is the definition, for any ground type B, of a function $R_B : (\bot)B$. This is an "empty function" with domain $\bot$ and codomain B. The existence, for every type B, of the function R_B follows from the existence of an empty type, $\bot$, and the permission to define functions by cases.

Let us take stock. We have the ground types $\mathbf{n}$, and we have the function types $(A)B$, where A and B are ground types. For any type B, we are allowed to define

[3] This method of generating the finite types is due to Martin-Löf; see (Nordström et al., 1990, p. 93).

a function f of type $(\mathbf{n})B$ either explicitly or by cases. In particular, we have, for every type B, an "empty function" $R_B : (\bot)B$.

5.2. **The type $\mathbb{HIF}$.** These are all the ingredients required for the definition of a new ground type, $\mathbb{HIF}$. For every finite type, $\mathbf{n}$, and function, f, from $\mathbf{n}$ into $\mathbb{HIF}$, we may form a new object, $\mathbf{set}(f)$, which under a set-theoretical reading is to be understood as the set of objects enumerated by f. This is to be the principle generating the objects of type $\mathbb{HIF}$. It may be compared to the principle saying that if b is of type B and c is of type C, then we may form a new object $\langle b, c\rangle$ of type $B \times C$. Both principles are, in effect, inductive definitions, and indeed, fundamental inductive definitions, in the terminology of Kleene (1952, § 53).

The task now is to formulate the principle generating the objects of type $\mathbb{HIF}$ in one or more rules. Following established type-theoretical terminology, any such rule is called an introduction rule for $\mathbb{HIF}$. A first attempt is the following formulation, where $\mathbf{n}$ is a parameter:

$$\frac{f : (\mathbf{n})\mathbb{HIF}}{\mathbf{set}(f) : \mathbb{HIF}}$$

This is, however, not a single rule, but a rule scheme. There are infinitely many rules here, one for each natural number, n. We can unify all of these rules into a single rule by introducing the type $\mathcal{F}$ of finite types. A judgement of the form

$$A : \mathcal{F}$$

is thus to mean that A is some $\mathbf{n}$. (The type $\mathcal{F}$ is also an inductively generated type of types—it is a universe in the terminology of Section 8.) Employing this form of judgement, we reach the official formulation of the introduction rule defining the type $\mathbb{HIF}$:

($\mathbb{HIF}$-intro) $$\frac{A : \mathcal{F} \qquad f : (A)\mathbb{HIF}}{\mathbf{set}(A, f) : \mathbb{HIF}}$$

Provided A is some $\mathbf{n}$ and f is a function from A into $\mathbb{HIF}$, then $\mathbf{set}(A, f)$ is an object of type $\mathbb{HIF}$. The domain A of the function f is included as an argument to the **set**-function mainly for the purposes of bookkeeping. We could leave it out and write simply $\mathbf{set}(f)$, as we already did above and shall do at times below.

The rule ($\mathbb{HIF}$-intro) makes precise the metaphor of a lasso collecting sets together and a wand forming the set of those sets (Forster, 2008, p. 98). Any function $f : (A)\mathbb{HIF}$ is a lasso comprehending a collection of sets, namely all the $f(a)$'s as a ranges over A. Letting the wand work its magic, we obtain $\mathbf{set}(A, f)$, a novel set. It is not enough merely to lasso the sets $f(a)$ together: an additional step is needed that turns this "preset" (in Forster's happy terminology) into a set.[4]

5.3. **Populating $\mathbb{HIF}$.** It may not be immediately clear how the generation of objects of type $\mathbb{HIF}$ gets off the ground, given that "$\mathbb{HIF}$" occurs in the premiss of ($\mathbb{HIF}$-intro). In order to apply this rule, we need a function from some $\mathbf{n}$ into $\mathbb{HIF}$, but, it would seem, we cannot define a function by cases on $\mathbf{n}$ unless we already have some objects of type $\mathbb{HIF}$ to use as values. If n is greater than zero, this is certainly so, but if n is zero, that is, $\mathbf{n}$ is the type $\bot$, then no objects of $\mathbb{HIF}$ need to be

[4]Forster imagines having also a second wand that turns a preset into the complement of the corresponding set: in addition to $\mathbf{set}(A, f)$, we have $\mathbf{coset}(A, f)$ with precisely the same introduction rule. The intuition that these sets are complements of each other needs to be captured by the definitions of extensional equality and elementhood.

known, since there is the "empty function" $R_{\mathbb{HF}} : (\bot)\mathbb{HF}$. The following inference therefore accords with the rule ($\mathbb{HF}$-intro):

$$\frac{\bot : \mathcal{F} \qquad R_{\mathbb{HF}} : (\bot)\mathbb{HF}}{\mathbf{set}(\bot, R_{\mathbb{HF}}) : \mathbb{HF}}$$

Since $R_{\mathbb{HF}}$ enumerates no objects of $\mathbb{HF}$, we may regard $\mathbf{set}(\bot, R_{\mathbb{HF}})$ as the empty set and make the following definition:

$$\emptyset \equiv \mathbf{set}(\bot, R_{\mathbb{HF}})$$

This way of introducing the empty set is quite similar to the way of the stage conception of set. We have obtained the empty set from the empty function, whereas in the stage conception, it is obtained, at stage one, from the empty collection. No sets are available at the initial stage, so only the empty collection can be used in set formation. (We are assuming here that all sets are pure.)

If this account of the empty set seems circular in its reliance on the existence of an empty domain, $\bot$, constructive type theory allows for an alternative approach: stipulate a second introduction rule,

($\mathbb{HF}$-intro$_2$) $$\emptyset : \mathbb{HF}$$

The empty set is thereby introduced explicitly as a base element of the inductively generated type $\mathbb{HF}$, just as 0 is the base element of the inductively generated type $\mathbb{N}$ of natural numbers. Proceeding in this way, the empty type, $\bot$, could be left out of $\mathcal{F}$. The empty type is, however, a necessary component of Aczel's model (it is used in the validation of Bounded Separation), hence we have included it also in $\mathcal{F}$. Once the type $\bot$ is available, there is, as we have just seen, no need for an additional introduction rule stipulating the existence of the empty set.

Having introduced $\emptyset$ into $\mathbb{HF}$, we can go on to introduce further objects. For instance, if $f : (\mathbf{1})\mathbb{HF}$ is defined by $f(0_{\mathbf{1}}) \equiv \emptyset$, then $\mathbf{set}(\mathbf{1}, f)$ is the singleton $\{\emptyset\}$, and if $g : (\mathbf{2})\mathbb{HF}$ is defined by the two equations

$$g(0_{\mathbf{2}}) \equiv \emptyset$$
$$g(1_{\mathbf{2}}) \equiv \{\emptyset\}$$

then $\mathbf{set}(\mathbf{2}, g)$ is the set $\{\emptyset, \{\emptyset\}\}$. More generally, given already constructed objects, $a_1, \ldots, a_n$, of type $\mathbb{HF}$, we can define a function $h : (\mathbf{n})\mathbb{HF}$ enumerating these and thence form $\mathbf{set}(\mathbf{n}, h)$, which is the set $\{a_1, \ldots, a_n\}$.

Urelemente may be included as well. They will have to come from some type, $\mathcal{A}$, which we include into $\mathbb{HF}$ by copying: an object a of type $\mathcal{A}$ will be copied as the Urelement $\mathrm{ur}(a)$. In addition to ($\mathbb{HF}$-intro) we thus have the following introduction rule for $\mathbb{HF}$:

($\mathbb{HF}$-intro$_{\mathrm{ur}}$) $$\frac{a : \mathcal{A}}{\mathrm{ur}(a) : \mathbb{HF}}$$

Urelemente defined in this way may be in the range of an enumerating function f, hence, intuitively, be members of $\mathbf{set}(f)$—"intuitively", since we have yet to define membership.

6. Turning $\mathbb{HF}$ into a universe of sets

6.1. **Strict identity.** The type $\mathbb{HF}$, just as any other type, being a type of *objects*, must come equipped with a notion of identity. We cannot speak of objects without

assuming some notion of identity, as Quine (1958) famously remarked. Type theory adds the thesis that what it means for objects of one type to be identical may not be the same as what it means for objects of another type to be identical: different types may be associated with different criteria of identity. For a type τ, and objects a, a' of type τ, let us write

$$a = a' : \tau$$

to mean that a and a' are identical according to the criterion of identity for τ. Following Sundholm (1999), let us call criterial identity the notion of identity captured by the criterion of identity for a type.

In constructive type theory, criterial identity is strict. It is sometimes called intensional, or definitional, identity, though what that means more precisely need not be discussed here (see Klev, 2019b, 2022). Let us merely note that the following equivalence holds:

$$\mathbf{set}(A, f) = \mathbf{set}(B, g) : \mathbb{HF} \quad \text{iff} \quad A = B : \mathbb{HF} \text{ and } f = g : (A)\mathbb{HF}$$

From this it follows that criterial identity on $\mathbb{HF}$ is not the notion of extensional equality that we expect to hold in a universe of sets. To see this, consider, for instance, the following two constant functions:

$$\begin{array}{ll} k_{\mathbf{1}} : (\mathbf{1})\mathbb{HF} & k_{\mathbf{2}} : (\mathbf{2})\mathbb{HF} \\ k_{\mathbf{1}}(x) \equiv \emptyset & k_{\mathbf{2}}(x) \equiv \emptyset \end{array}$$

Since $k_{\mathbf{1}}$ and $k_{\mathbf{2}}$ have different domains, they are different functions. The set of objects enumerated by $k_{\mathbf{1}}$ is, however, the same as the set of objects enumerated by $k_{\mathbf{2}}$, namely $\{\emptyset\}$. For another example, consider the two functions i, j of type $(\mathbf{2})\mathbb{HF}$ defined as follows:

$$\begin{array}{ll} i(0_{\mathbf{2}}) \equiv \emptyset & j(0_{\mathbf{2}}) \equiv \mathbf{set}(k_{\mathbf{1}}) \\ i(1_{\mathbf{2}}) \equiv \mathbf{set}(k_{\mathbf{1}}) & j(1_{\mathbf{2}}) \equiv \emptyset \end{array}$$

These are different functions, but they have the same range, indeed the set of objects enumerated by both is $\{\emptyset, \{\emptyset\}\}$.

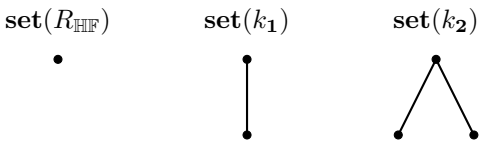
Under the set-theoretical interpretation that we are aiming at, each function f into $\mathbb{HF}$ may be regarded as a way of collecting objects in $\mathbb{HF}$. Different ways of collecting may yield what is, intuitively, the same set. Among the $f(a)$'s there might be repetitions, and the $f(a)$'s come in a certain order, but neither repetition nor order matters to set identity. A further complication is illustrated by the functions F and G of type $(\mathbf{2})\mathbb{HF}$ defined as follows:

$$\begin{array}{ll} F(0_{\mathbf{2}}) \equiv \emptyset & G(0_{\mathbf{2}}) \equiv \emptyset \\ F(1_{\mathbf{2}}) \equiv \mathbf{set}(k_{\mathbf{1}}) & G(1_{\mathbf{2}}) \equiv \mathbf{set}(k_{\mathbf{2}}) \end{array}$$

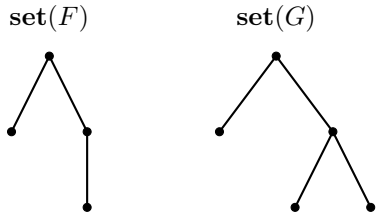
The values of $F(1_{\mathbf{2}})$ and $G(1_{\mathbf{2}})$ are not identical as objects of $\mathbb{HF}$, but they are extensionally equal—they are both the set $\{\emptyset\}$. Extensional equality must be defined such that repetition and order are neutralized and such that, when comparing $\mathbf{set}(A, f)$ and $\mathbf{set}(B, g)$ for extensional equality, one must be allowed to compare the $f(a)$'s with the $g(b)$'s for extensional equality.

Before moving on to the definition of extensional equality, let us consider the type $\mathbb{HF}$ as it is, equipped with strict identity. The objects of $\mathbb{HF}$ are not sets, but they may usefully be regarded as trees. Seen as growing downwards, $\mathbf{set}(A, f)$

is the tree whose immediate subtrees are all the $f(a)$'s as a ranges over A. For instance, the functions $R_{\mathbb{HF}}$, $k_{\mathbf{1}}$, and $k_{\mathbf{2}}$ give rise to the following trees:



The tree $\mathbf{set}(k_{\mathbf{2}})$ has two immediate subtrees, $k_{\mathbf{2}}(0_{\mathbf{2}})$ and $k_{\mathbf{2}}(1_{\mathbf{2}})$, both of which are equal to the tree $\mathbf{set}(R_{\mathbb{HF}})$, consisting of a single node. The functions F and G give rise to the following trees:



The tree $\mathbf{set}(G)$ has two immediate subtrees: $G(0_{\mathbf{2}})$, which is a single node, and $G(1_{\mathbf{2}})$, which is the tree $\mathbf{set}(k_{\mathbf{2}})$.

6.2. **Extensional equality.** The Axiom of Extensionality may be glossed as saying that sets u and v are the same if, and only if, they have the same members. What does it mean for u and v to have the same members? A careful answer, taking into account that sets may be given to us in different ways, is the following: each member x of u is the same as some member y of v, and each member y of v is the same as some member x of u. If x and y are themselves sets, then what it means for them to be identical is again determined by the Axiom of Extensionality. We continue checking for identity until we reach either the empty set or Urelemente.

Although we have not yet defined the membership relation, a variant of the algorithm just described is available that proceeds along the relation of immediate predecession.[5] The immediate predecessors of $\mathbf{set}(A, f)$ are all the $f(a)$'s as a ranges over A. In the intuitive curly-braces notation, the immediate predecessors of $\{f(a) \mid a : A\}$ are all the $f(a)$'s as a ranges over A. Any such $f(a)$, being an object of type $\mathbb{HF}$, may be assumed to be of the form $\mathbf{set}(A', f')$, where A' is of type $\mathcal{F}$ and f' is a function of type $(A')\mathbb{HF}$. Any of its immediate predecessors may, in turn, be assumed to have the form $\mathbf{set}(A'', f'')$, where A'' is of type $\mathcal{F}$ and f'' is a function of type $(A'')\mathbb{HF}$. Consider the recursive procedure, starting with $\mathbf{set}(A, f)$, of continuously choosing an immediate predecessor. However the procedure is executed, we obtain a sequence of objects of the form $\mathbf{set}(B, g)$ that form a branch in the tree $\mathbf{set}(A, f)$. It is clear from how the type $\mathbb{HF}$ has been explained, namely as inductively generated by the rule ($\mathbb{HF}$-intro), that this branch

[5]In the type $\mathbb{V}$, to be defined below, comparing objects for extensional equality is no longer an algorithm in the usual sense of a finite procedure. Firstly, objects in $\mathbb{V}$ may be infinitary. Secondly, it is in general not decidable whether a given object of $\mathbb{V}$ is extensionally equal to the empty set, defined as $\mathbf{set}(R_{\mathbb{V}})$: owing to how $\mathbb{V}$ is defined, such a decision procedure would yield a decision procedure for classical first-order logic.

will be finite, since the procedure of choosing immediate predecessors will eventually reach $\emptyset$.[6]

Let us write $u \prec v$ to mean that u is an immediate predecessor of v. As we have just seen, this is a well-founded relation, hence we may define predicates and relations by induction over it. Extensional equality, written $\doteq$, is defined by induction on the relation of immediate predecession as follows:

$$\begin{aligned} u \doteq v \quad \text{iff} \quad & \text{for each } x \prec u \text{ there is } y \prec v \text{ such that } x \doteq y, \text{ and} \\ & \text{for each } y \prec v \text{ there is } x \prec u \text{ such that } x \doteq y. \end{aligned}$$

In prose: u and v are extensionally equal if, and only if, each immediate predecessor of u can be matched with an extensionally equal immediate predecessor of v, and vice versa. Assuming that $x \doteq y$ has been defined for all immediate predecessors x of u and y of v, the proposition $u \doteq v$ is thus also defined. (Technically, this is a double induction, since we are appealing to the immediate predecessors of both u and v.)

This definition turns out to provide precisely what we want. Repetition and order are neutralized. Moreover, owing to the inductive nature of the definition, we are allowed, when comparing u with v for extensional equality, to assume that $x \doteq y$ has been defined for all immediate predecessors x of u and y of v. Recalling the various functions defined above, the reader can verify that the following propositions are all true:

$$\begin{aligned} \mathbf{set}(k_{\mathbf{1}}) &\doteq \mathbf{set}(k_{\mathbf{2}}) \\ \mathbf{set}(i) &\doteq \mathbf{set}(j) \\ \mathbf{set}(F) &\doteq \mathbf{set}(G) \end{aligned}$$

It can also be seen that extensional equality, thus defined, is an equivalence relation on $\mathbb{HF}$. In particular, $\emptyset \doteq \emptyset$ is true because $\emptyset$ has no immediate predecessors.

The definition of extensional equality is perhaps the most subtle part of Aczel's model construction. Our formulation above captures the main idea of Aczel's definition, but it is not quite his formulation of it. Aczel had to phrase the definition in the language of constructive type theory, and he had to do so in a way that the axioms of CZF (Bounded Separation, in particular) can be validated. In Aczel's formulation, the induction proceeds, not along the defined relation of immediate predecession, but rather along the build-up of objects. As already noted above, a principle of definition by induction on the build-up of the objects of a type is part and parcel of constructive type theory, given by the elimination rule for the type in question.

Since Urelemente have no predecessors, applying the definition of extensional equality above yields $\mathrm{ur}(a) \doteq \mathrm{ur}(a')$ for every a and a' in $\mathcal{A}$. To avoid this outcome, we must add a base case to the definition that takes care of Urelemente. All that is needed for this is an equivalence relation, ρ, on the type $\mathcal{A}$ from which the Urelemente are taken. The base case then has the following form:

$$\mathrm{ur}(a) \doteq \mathrm{ur}(a') \quad \text{iff} \quad \rho(a, a')$$

[6]That any inductively defined domain D has a well-founded structure, and in particular, that functions may be defined by induction on D, is a fundamental assumption of constructive type theory. See, for instance, how the rules of $\mathbb{N}$-elimination and $\mathbb{W}$-elimination are justified by Martin-Löf (1984, pp. 71–72, 81–82).

The relation ρ may be, but need not be, the relation of identity on $\mathcal{A}$.

The type $\mathbb{HF}$ is turned into a universe of sets by equipping it with extensional equality, yielding the domain $(\mathbb{HF}, \doteq)$. Objects in this domain are not individuated according to the strict criterial identity on $\mathbb{HF}$, but rather according to the coarser relation of extensional equality. Extensional equality is, of course, just one of indefinitely many equivalence relations definable on $\mathbb{HF}$. Gylterud (2020) defined an equivalence relation, $\sim$, that behaves similarly to extensional equality except it does not neutralize repetitions of the $f(a)$'s. The domain $(\mathbb{HF}, \sim)$ is therefore the domain of hereditarily finite multisets, i.e., sets in which one and the same element (itself a multiset) may have more than one occurrence.

Mathematics is full of similar examples, where a domain is, as it were, transformed by equipping it with a novel equivalence relation that is to serve as identity. We are taught to operate with rational numbers given in the form of fractions, and we learn that $\frac{m}{n}$ and $\frac{k \cdot m}{k \cdot n}$, whenever k and n are non-zero, are the same rational number—the same rational number, but not the same pair of integers, although a fraction, on the face of it, is just a pair of integers, (m, n), where n is non-zero. The domain of rational numbers is obtained by equipping the domain of such pairs with a novel relation of equality to serve as identity. The domain of real numbers, likewise, may be obtained by equipping the domain of Cauchy sequences of fractions with the relation of co-convergence to serve as identity.

6.3. **Membership.** It remains to define the membership relation. That u is a member of v means that u is extensionally equal to an immediate predecessor of v:

$$u \in v \text{ iff } \quad \text{there is } x \prec v \text{ such that } u \doteq x$$

It is then straightforward to verify the following implications:

$$\begin{aligned} u \in v \,\&\, v \doteq w &\implies u \in w \\ u \in v \,\&\, u \doteq w &\implies w \in v \end{aligned}$$

Equality and membership thus interact as one would expect. This completes the definition of the structure $(\mathbb{HF}, \doteq, \in)$.

7. Expanding to $\mathbb{V}$

The type $\mathbb{V}$ and the structure $(\mathbb{V}, \doteq, \in)$ that Aczel showed to be a model of CZF are defined precisely as $\mathbb{HF}$ and $(\mathbb{HF}, \doteq, \in)$ have been defined with the sole difference that more types are allowed as domains for the enumerating functions. In particular, there is an infinite type, $\mathbb{N}$, of natural numbers, and the types available are closed under type-forming operators more powerful than the successor operator, S, used in the definition of the finite types $\mathcal{F}$. Just as $\mathcal{F}$, the type $\mathcal{U}$, used in the definition of $\mathbb{V}$, is inductively generated. Explaining in detail every step of this inductive definition is beyond the scope of this article—for that, the reader should consult an introduction to constructive type theory, such as Martin-Löf (1984) or Nordström et al. (1990). We shall make do with the following informal description.

The type $\mathcal{U}$ contains, firstly, the following types:

$\bot$	empty type
$\mathbf{1}$	unit type (type of cardinality one)
$\mathbb{N}$	type of natural numbers

Secondly, it is closed under the following type-forming operations:

$A + B$	disjoint union
$\Pi(A, C)$	generalized Cartesian product
$\Sigma(A, C)$	generalized disjoint union

In this list, A and B are types, and C is a function from A into $\mathcal{U}$. For example, by induction on the natural numbers, $\mathbb{N}$, we can for any type B in $\mathcal{U}$, define a function C such that

$$C(n) \equiv \overbrace{B + B + \cdots + B}^{n \text{ times}},$$

the n-fold disjoint union of the type B with itself.

Every type in $\mathcal{U}$ is a ground type. As before, we also postulate the function type $(A)B$ whenever A and B are ground types. A function of type $(A)B$ may be defined either explicitly or by induction. Each type in $\mathcal{U}$ is inductively generated, hence it makes sense to define functions on such a type, A, by induction on the build-up of the objects of type A. On the type $\mathbb{N}$ of natural numbers this is just the method of recursive definition.

The type $\mathbb{V}$ is defined by the following rule:

($\mathbb{V}$-intro)
$$\frac{A : \mathcal{U} \qquad f : (A)\mathbb{V}}{\mathbf{set}(A, f) : \mathbb{V}}$$

An object of type $\mathbb{V}$ may be regarded as a labelled tree. Labels on the nodes are needed in order to ensure that the trees are individuated according to the strict notion of identity on $\mathbb{V}$. (It can, for instance, be seen that, in $\mathbb{V}$, one must distinguish different single-noded trees, since there are different empty types in $\mathcal{U}$, such as $\bot$ and $\bot + \bot$.) Just as we can see from the definition of $\mathbb{HF}$ that an object of type $\mathbb{HF}$ is a well-founded tree, meaning that no branch is infinite, so we can see from the definition of $\mathbb{V}$ that the trees in $\mathbb{V}$ are also well-founded. Unlike the trees in $\mathbb{HF}$, however, a tree in $\mathbb{V}$ may be infinite, since a node may have infinitely many immediate subtrees (infinitely many children, in a widespread anthropomorphic terminology). An example will given below.

Extensional equality, $\doteq$, and membership, $\in$, on $\mathbb{V}$ are defined precisely as they were defined on $\mathbb{HF}$, and our discussion above of $\mathbb{HF}$ and $(\mathbb{HF}, \doteq, \in)$ applies, mutatis mutandis, to $\mathbb{V}$ and $(\mathbb{V}, \doteq, \in)$. The additional structure in $\mathcal{U}$, however, makes possible the validation of all of CZF. This validation makes use of all the various type-forming operations available in $\mathcal{U}$. It moreover makes use of the so-called propositions-as-types principle, according to which the notions of proposition and inductively generated type are identified.[7] Going into the details of that validation lies outside the scope of this article. We shall, however, see that the Axiom of Infinity is validated in $(\mathbb{V}, \doteq, \in)$.

On both $\mathbb{HF}$ and $\mathbb{V}$ one can define a function that in standard set-theoretical notation may be written as follows:

$$\mathbf{s}(x) \equiv \text{“}x \cup \{x\}\text{”}$$

[7]Gambino and Aczel (2006) construct a type-theoretic model of CZF that does not rely on the propositions-as-types principle. They instead add a system of predicate logic to an underlying dependent type theory.

This is a function, not only on $\mathbb{HF}$ and $\mathbb{V}$, but also on the relational structures $(\mathbb{HF}, \doteq)$ and $(\mathbb{V}, \doteq)$, since it respects extensional equality:

$$x \doteq y \implies \mathbf{s}(x) \doteq \mathbf{s}(y)$$

It is, moreover, an injective but non-surjective function on these structures:

$$\mathbf{s}(x) \doteq \mathbf{s}(y) \implies x \doteq y$$

$$\neg(\mathbf{s}(x) \doteq \emptyset)$$

It follows that $(\mathbb{HF}, \doteq)$ and $(\mathbb{V}, \doteq)$ are Dedekind infinite: on both domains an injective, but non-surjective, function may be defined.

In the structure $(\mathbb{V}, \doteq, \in)$, something more holds: it contains an infinite set. Define a function $N : (\mathbb{N})\mathbb{V}$ by induction as follows:

$$\begin{aligned} N(0) &\equiv \emptyset \\ N(n') &\equiv \mathbf{s}(N(n)) \\ &\equiv N(n) \cup \{N(n)\} \end{aligned}$$

The sets enumerated by N are $\emptyset$, $\{\emptyset\}$, $\{\emptyset, \{\emptyset\}\}$, etc., in other words, the Von Neumann ordinals. Since $\mathbb{N}$ is of type $\mathcal{U}$, we may use ($\mathbb{V}$-intro) to tie all of these ordinals together into a single set, $\mathbf{set}(N)$, which we shall call ω,

$$\omega : \mathbb{V}$$

This is an infinite set, witnessing that the Axiom of Infinity is true in the structure $(\mathbb{V}, \doteq, \in)$.

Although ω, being an infinite set of sets, in a sense encompasses infinitely many applications of the **set**-operation, the proof of the judgement $\omega : \mathbb{V}$ is finite. We do not need to prove all of the infinitely many judgements

$$\emptyset : \mathbb{V},\ \{\emptyset\} : \mathbb{V},\ \{\emptyset, \{\emptyset\}\} : \mathbb{V}, \ldots$$

before we can pass on to assert $\omega : \mathbb{V}$. In order to apply the rule ($\mathbb{V}$-intro), it is enough to have available the function N. The domain of this function is infinite, but its definition is finite, namely it is defined by recursion on $\mathbb{N}$.

8. Expanding further

The presence of the type $\mathbb{N}$ in the type of types $\mathcal{U}$ allows the formation of an infinite set in $\mathbb{V}$. A question that naturally arises is whether an even more expansive type $\mathbb{V}^+$ may be defined after the pattern of $\mathbb{HF}$ and $\mathbb{V}$ that accommodates higher infinities, in particular infinities that in classical set theory would go under the title of large cardinals. A positive answer to the question is forthcoming if we replace $\mathcal{U}$ in the definition of $\mathbb{V}$ by a more expansive type of types.

In type-theoretical terminology, a type of types is called a *universe*. The notion was introduced by Martin-Löf (1975) after an earlier version of his type theory that had an axiom to the effect that there is a type of all types was shown to be inconsistent. Although a universe is a type of types, it is not the type of all types. Rather, a universe is a type inductively generated from certain base types by means of certain type-forming operations. The type $\mathcal{U}$, for instance, has as base types $\bot$, $\mathbf{1}$, and $\mathbb{N}$, and it is generated by the operators of disjoint union, generalized Cartesian product and generalized disjoint union. A universe is meant to capture the definition of types up to a certain stage: the universe reflects the definition of

types so far. It is therefore stipulated that a universe U is never itself an object of type U. A universe U may, however, belong to another, more encompassing, universe U'. Already Martin-Löf considered an infinite sequence, $U_0, U_1, U_2, \ldots$, of universes, forming a cumulative hierarchy of types with each universe U_n contained in the next U_{n+1}.

Martin-Löf described this sequence of universes in the metalanguage of type theory. As a way of internalizing this sequence in the language of type theory itself, Palmgren (1998) introduced a novel type-former called a universe operator, $\mathbf{u}$, which when applied to a universe produces the next universe, in the sense that U_{n+1} is the next universe after U_n. A universe that is closed under the universe operator $\mathbf{u}$ Palmgren called a *superuniverse*. Being a type-theoretical operator, $\mathbf{u}$ is associated with certain rules that determine its behaviour. A suitable generalization of these rules yields rules for a superuniverse operator, $\mathbf{su}$, which when applied to a universe produces the next superuniverse. A supersuperuniverse is a universe that is closed under the superuniverse operator.

The method used to construct the definition of $\mathbf{su}$ from that of $\mathbf{u}$ may now be iterated to yield a sequence $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots$ of ever more general universe operators. It thus makes sense to speak of "the next universe-operator", just as the sequence of universes $U_0, U_1, U_2, \ldots$ gave rise to the idea of the next universe. And just as the universe operator internalizes the passage from U_n to U_{n+1}, one might ask whether also the passage from $\mathbf{u}_n$ to $\mathbf{u}_{n+1}$ may be internalized in the language of constructive type theory.

Rathjen et al. (1998) gave rules for a series of operators that together achieve this and introduced a universe $\mathcal{M}$ closed under the operators in question. The universe operators in $\mathcal{M}$ are introduced through a bootstrap process, where every new such operator introduced allows the definition of further ones. More specifically, suppose A is a type in $\mathcal{M}$ and f is a function from A into universe operators. For instance, A might be $\mathbb{N}$, and f might be a function enumerating the universe operators $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots$ mentioned above. A new universe operator, $\mathbf{univ}(f)$, is introduced and stipulated to have the property that a universe formed by means of it is closed under all the universe operators $f(a)$, as a ranges over A.[8] The universe $\mathcal{M}$ is closed under all universe operators $\mathbf{univ}(f)$ definable in this way.

Let $\mathbb{V}^+$ be the type obtained by replacing the $\mathcal{U}$ in ($\mathbb{V}$-intro) with $\mathcal{M}$, and let $(\mathbb{V}^+, \doteq, \in)$ be the set-theoretical structure obtained by equipping $\mathbb{V}^+$ with extensional equality and membership. Rathjen et al. (1998) showed that this structure is a model, not only of CZF, but also of an axiom that classically entails the existence of inaccessible cardinals of all transfinite orders (all the so-called π_ν-numbers defined by Mahlo 1911).

References

Aczel, P. (1978). The type theoretic interpretation of constructive set theory. In Macintyre, A., Pacholski, L., and Paris, J., editors, *Logic Colloquium 77*, pages 55–66. North-Holland, Amsterdam. DOI: https://doi.org/10.1016/S0049-237X(08)71989-X

Aczel, P. (1980). Frege structures and the notions of proposition, truth and set. In Barwise, J., Keisler, H. J., and Kunen, K., editors, *The Kleene Symposium*, pages 31–59. North-Holland, Amsterdam. DOI: https://doi.org/10.1016/S0049-237X(08)71252-7

[8] Gödel (1933, pp. 46-47) described a similar process for defining transfinite types.

Aczel, P. (1982). The type theoretic interpretation of constructive set theory: choice principles. In Troelstra, A. S. and van Dalen, D., editors, *The L.E.J. Brouwer Centenary Symposium*, pages 1–40. North-Holland, Amsterdam. DOI: https://doi.org/10.1016/S0049-237X(09)70120-X

Barton, N. (2024). *Iterative Conceptions of Set*. Cambridge University Press, Cambridge. DOI: https://doi.org/10.1017/9781009227223

Bishop, E. (1967). *Foundations of Constructive Analysis*. McGraw-Hill, New York.

Boolos, G. (1971). The iterative conception of set. *Journal of Philosophy*, 68:215–232. DOI: https://doi.org/10.2307/2025204

Boolos, G. (1989). Iteration again. *Philosophical Topics*, 42:5–21. DOI: https://doi.org/10.5840/philtopics19891721

Button, T. (2021). Level theory, part 1: Axiomatizing the bare idea of a cumulative hierarchy of sets. *Bulletin of Symbolic Logic*, 27:436–460. DOI: https://doi.org/10.1017/bsl.2021.13

Cantor, G. (1895). Beiträge zur Begründung der transfiniten Mengenlehre (Erster Artikel). *Mathematische Annalen*, 46:481–512. DOI: https://doi.org/10.1007/BF02124929

Crosilla, L. (2020). Set theory: constructive and intuitionistic ZF. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Stanford. https://plato.stanford.edu/archives/sum2020/entries/set-theory-constructive/.

Dedekind, R. (1888). *Was sind und was sollen die Zahlen?* Vieweg und Sohn, Braunschweig.

Dybjer, P. (1994). Inductive families. *Formal Aspects of Computing*, 6:440–465. DOI: https://doi.org/10.1007/BF01211308

Dybjer, P. (2000). A general formulation of simultaneous inductive-recursive definitions in type theory. *Journal of Symbolic Logic*, 65:525–549. DOI: https://doi.org/10.2307/2586554

Forster, T. (2008). The iterative conception of set. *Review of Symbolic Logic*, 1:97–110. DOI: https://doi.org/10.1017/S1755020308080064

Gambino, N. and Aczel, P. (2006). The generalised type-theoretic interpretation of constructive set theory. *Journal of Symbolic Logic*, 71:67–103. DOI: https://doi.org/10.2178/jsl/1140641163

Gödel, K. (1933). The present situation in the foundations of mathematics. Cited from (Gödel, 1995).

Gödel, K. (1947). What is Cantor's Continuum Problem? *American Mathematical Monthly*, 54:515–525. DOI: https://doi.org/10.2307/2304666

Gödel, K. (1995). *Collected Works. Volume III. Unpublished Essays and Lectures*. Oxford University Press, Oxford. DOI: https://doi.org/10.1093/oso/9780195072556.001.0001

Gylterud, H. R. (2020). Multisets in type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 169:1–18. DOI: https://doi.org/10.1017/S0305004119000045

Incurvati, L. (2020). *Conceptions of Set and the Foundations of Mathematics*. Cambridge University Press, Cambridge. DOI: https://doi.org/10.1017/9781108596961

Kleene, S. C. (1952). *Introduction to Metamathematics*. Van Nostrand, New York.

Klev, A. (2019a). A comparison of type theory with set theory. In Centrone, S., Kant, D., and Sarikaya, D., editors, *Reflections on the Foundations of Mathematics*, pages 271–292. Springer, Dordrecht. DOI https://doi.org/10.1007/978-3-030-15655-8_12

Klev, A. (2019b). Eta-rules in Martin-Löf type theory. *Bulletin of Symbolic Logic*, 25:333–359. DOI: https://doi.org/10.1017/bsl.2019.21

Klev, A. (2022). Identity in Martin-Löf type theory. *Philosophy Compass*, 17(e12805). DOI: https://doi/10.1111/phc3.12805

Linnebo, O. (2010). Pluralities and sets. *Journal of Philosophy*, 107:144–164. DOI: https://doi.org/10.5840/jphil2010107311

Linnebo, O. (2013). The potential hierarchy of sets. *Review of Symbolic Logic*, 6:205–228. DOI: https://doi.org/10.1017/s1755020313000014

Linnebo, Ø. (2017). *Philosophy of Mathematics*. Princeton University Press, Princeton.

Mahlo, P. (1911). Über lineare transfinite Mengen. *Berichte über die Verhandlungen der Königlich Sächsischen Gesellschaft von Wissenschaften zu Leipzig. Mathematisch-physische Klasse*, 63:187–200.

Martin-Löf, P. (1975). An intuitionistic theory of types: Predicative part. In Rose, H. E. and Shepherdson, J. C., editors, *Logic Colloquium '73*, pages 73–118. North-Holland, Amsterdam. DOI: https://doi.org/10.1016/S0049-237X(08)71945-1

Martin-Löf, P. (1982). Constructive mathematics and computer programming. In Cohen, J. L., Łoś, J., et al., editors, *Logic, Methodology and Philosophy of Science VI, 1979*, pages 153–175.

North-Holland, Amsterdam. DOI: https://doi.org/10.1016/S0049-237X(09)70189-2

Martin-Löf, P. (1984). *Intuitionistic Type Theory*. Bibliopolis, Naples.

Martin-Löf, P. (1993). Philosophical aspects of intuitionistic type theory. Transcript of a lecture course given at Leiden University in the autumn semester of 1993. https://pml.flu.cas.cz/.

Martin-Löf, P. (2006). 100 years of Zermelo's axiom of choice: what was the problem with it? *The Computer Journal*, 49:345–350. DOI: https://doi.org/10.1093/comjnl/bxh162

Nordström, B., Petersson, K., and Smith, J. (1990). *Programming in Martin-Löf's Type Theory*. Oxford University Press, Oxford.

Palmgren, E. (1998). On universes in type theory. In Sambin, G. and Smith, J., editors, *Twenty-five Years of Constructive Type Theory*, pages 191–204. Clarendon Press, Oxford. DOI: https://doi.org/10.1093/oso/9780198501275.003.0012

Quine, W. v. O. (1958). Speaking of objects. *Proceedings and Addresses of the American Philosophical Association*, 31:5–22. DOI: https://doi.org/10.2307/3129242

Rathjen, M., Griffor, E. R., and Palmgren, E. (1998). Inaccessibility in constructive set theory and type theory. *Annals of Pure and Applied Logic*, 94:181–200. DOI: https://doi.org/10.1016/S0168-0072(97)00072-9

Shoenfield, J. R. (1967). *Mathematical Logic*. Addison-Wesley, Reading, MA.

Shoenfield, J. R. (1977). The axioms of set theory. In Barwise, J., editor, *Handbook of Mathematical Logic*, pages 321–344. North-Holland, Amsterdam. DOI: https://doi.org/10.1016/S0049-237X(08)71106-6

Sundholm, B. G. (1999). Identity: Absolute, criterial, propositional. In Childers, T., editor, *LOGICA Yearbook 1998*, pages 20–26. Filosofia, Prague.

Vopěnka, P. (1979). *Mathematics in the Alternative Set Theory*. Teubner, Leipzig.

Zermelo, E. (1908). Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische Annalen*, 65:261–281. DOI: https://doi.org/10.1007/BF01449999