Adding an Abstraction Barrier to ZF Set Theory*

Ciarán Dunne, J. B. Wells, and Fairouz Kamareddine

Heriot-Watt University

May 2020

Abstract. Much mathematical writing exists that is, explicitly or implicitly, based on set theory, often Zermelo-Fraenkel set theory (ZF) or one of its variants. In ZF, the domain of discourse contains only sets, and hence every mathematical object must be a set. Consequently, in ZF with the usual encoding of an ordered pair $\langle a, b\rangle$, formulas like $\{a\} \in \langle a, b\rangle$ have truth values, and operations like $\mathcal{P}(\langle a, b\rangle)$ have results that are sets. Such 'accidental theorems' do not match how people think about the mathematics and also cause practical difficulties when using set theory in machine-assisted theorem proving. In contrast, in a number of proof assistants, mathematical objects and concepts can be built of type-theoretic stuff so that many mathematical objects can be, in essence, terms of an extended typed λ-calculus. However, dilemmas and frustration arise when formalizing mathematics in type theory.

Motivated by problems of formalizing mathematics with (1) purely set-theoretic and (2) type-theoretic approaches, we explore an option with much of the flexibility of set theory and some of the useful features of type theory. We present ZFP: a modification of ZF that has ordered pairs as primitive, non-set objects. ZFP has a more natural and abstract axiomatic definition of ordered pairs free of any notion of representation. This paper presents axioms for ZFP, and a proof in ZF (machine-checked in Isabelle/ZF) of the existence of a model for ZFP, which implies that ZFP is consistent if ZF is. We discuss the approach used to add this abstraction barrier to ZF.

Keywords: set theory · formalisation of mathematics · theorem proving

1 Introduction

1.1 Background: Set Theory and Type Theory as Foundations

A large portion of the mathematical literature is based on set theory, explicitly or implicitly, directly or indirectly. Set theory is pervasive in mathematical culture. University mathematics programmes have introductory courses on set theory and many other courses that rely heavily on set-theoretic concepts (sets, classes, etc.), notation (comprehensions a.k.a. set-builders, power set, etc.), and reasoning.

* The final authenticated version is available online at https://doi.org/10.1007/978-3-030-53518-6_6.

Formal foundations for mathematics have been developed since the early 20th century, with both set-theoretic and type-theoretic approaches being considered. Although there are a number of set-theoretic foundations, for this paper it is sufficient to consider Zermelo-Fraenkel set theory (ZF), which anyway seems to be broadly accepted and reasonably representative of the strengths and weaknesses of set theory in actual practice. The core concept of ZF is the *set membership relation* $\in$, which acts on a domain of objects called *sets*. The theory is a collection of formulas (known as *axioms*) of first-order logic which characterise the membership relation. Logical deduction from these axioms yields a rich theory of sets. Moreover, mathematical objects such as ordered pairs, functions, and numbers can be represented as sets in ZF.

At roughly the same time as Zermelo was formulating his axiomatic set theory, Russell introduced the first type theory. Both Zermelo and Russell had the goal of rigorous, formal, logical reasoning free from the paradoxes that plagued the earlier systems of Cantor and Frege. Most modern type theories are descendants of Church's typed λ-calculus [9]. Many of the methods of modern type theory have been developed by computer scientists to solve problems in programming languages and formal verification. Types add layers of reasoning that help with soundness and representation independence. Some type theories have been used to formulate foundations of mathematics in which mathematical objects (e.g., groups, rings, etc.) are represented by terms and types of what is essentially a very fancy typed λ-calculus.

Formalizing mathematics that has been developed in a set-theoretic culture using a type-theoretic foundation can lead to dilemmas and frustration [6]. Subtyping may not work smoothly when formalising chains of structures such as the number systems and those belonging to universal algebra. There are also design choices in how to model predicates which can make proving some things easier but other things much harder. The rules of powerful type systems are also very complicated, so users require machine assistance to follow the typing rules, and even with machine support it can be quite challenging. In contrast, ZF-like set theories typically have very few 'types', e.g., there might be a type of sets and a type of logical formulas or perhaps a type of classes. When nearly every mathematical object you need is of 'type set' it is easy to obey the typing rules.

There are problems formalizing mathematics in pure ZF set theory also. When everything is of 'type set', a computer proof system has no easy way to know that it would be wasting its time to try to prove a theorem about ordinal numbers using lemmas and tactics for groups or rings, so automated support is more challenging. When representing mathematical objects (e.g., numbers) as sets, the bookkeeping of the intended 'type' of these objects is not avoided, but must be managed by the user outside the realm of a type system. In many not-too-tricky cases, a type inference algorithm can automatically infer type information that represents necessary preconditions for successful use of theorems and lemmas, but in pure set theory such automated inference is not very useful when the only type is 'set'.

Furthermore, practical computerisation in ZF requires abbreviation and definition mechanisms which first-order logic does not provide. Two contrasting examples of how this can be done are Metamath and Isabelle/ZF. Metamath [10] is mostly string based, and has 'syntax definitions' to introduce new constants, or syntax patterns. These definitions are given meaning by 'defining axioms' (whose correctness is not checked by the verifier). Isabelle/ZF is built on top of Isabelle/Pure, which is a fragment of intuitionistic higher-order logic that is based on Church's typed λ-calculus [11]. This means that meta-level activities such as variable binding, definitions, and abbreviations are handled by Isabelle/ZF in a type theory, albeit a very simple type theory. Isabelle also handles proof tactics in SML, which can be seen as another typed λ-calculus.

1.2 The Issue of Representation and the Case of the Ordered Pair

As discussed above, set theory can represent a multitude of mathematical objects as sets, but in some cases the user might prefer that some of their mathematical objects are genuinely not sets. The alternative of using a sophisticated type-theoretic foundation might not be the right solution, for a variety of reasons, some of which are mentioned above. So the user might ask: "May I please have a set theory which has genuine non-sets that I can use for purpose XYZ?"

There are indeed set theories with non-set objects [7], which are generally known as *urelements*, so named because they are often considered to be primordial, existing independently of and before the sets. A popular use for urelements is as 'atoms' whose only properties are being distinct from everything else and existing in large enough multitudes. Adding genuine non-sets takes some work, because the assumption that 'everything is a set' is deeply embedded in ZF's axioms. One example is the axiom of Extensionality,

$$\forall x, y : (\forall a : a \in x \leftrightarrow a \in y) \rightarrow x = y$$

which asserts that any two objects are equal if they have exactly the same set members. Because non-set objects of course have no set members, this ZF axiom forces them all to equal the empty set, meaning there can not be any.

Existing set theories with urelements generally (except see GST below) do not consider urelements with 'internal' structure that might include sets. The *ordered pair* is a simple and important example of a mathematical object with 'internal' structure which is not usually intended to be viewed as a set. Ordered pairs have been of enormous value in building theories of relations, functions, and spaces. The most widely used set-theoretical definition, by Kuratowski, defines the ordered pair $\langle a, b\rangle$ to be the set $\{\{a\}, \{a, b\}\}$. Because a is in all sets in $\langle a, b\rangle$ and b is only in one, a first-order logic formula using only the membership relation can check if an object is the first (or second) projection of an ordered pair. Kuratowski pairs satisfy the characteristic property of ordered pairs:

$$\langle a, b\rangle = \langle c, d\rangle \leftrightarrow (a = c \wedge b = d)$$

Like for any ZF representation of mathematical objects not thought of as sets, Kuratowski pairs have 'accidental theorems' such as $\{b\} \in \langle b, c\rangle$, and $\{\langle b, b\rangle\} = \{\{\{b\}\}\} = \langle\{b\}, \{b\}\rangle$, and $\{\langle 0, 0\rangle\} = \langle 1, 1\rangle$ with Von Neumann numbers.

The set representation of conceptually non-set objects raises issues. There are places in the literature where some mathematical objects are thought of as (or even explicitly stated to be) non-sets with no set members. One can find definitions or proofs by cases on 'type' that assume the case of sets never overlaps with the cases of pairs, numbers, etc. To view such writing as being founded on pure set theory requires either proving that none of the sets used overlap with the set representations used for abstract objects or inserting many tagging and tag-checking operations (see, e.g., the translation we give in definition 4.7 as part of proving a model for our system ZFP can be built in the pure set theory ZF). When formalizing and machine-checking mathematics, additional difficulties arise, some of which are mentioned above.

1.3 ZFP: Extending ZF Set Theory with Primitive Ordered Pairs

We aim to go beyond previous set theories with urelements to develop methods for extending set theories with genuine non-set objects whose internal structure can contain other objects including the possibility of sets. As a first instance of this aim, we achieve the objective of ZFP, a set theory with primitive non-set ordered pairs such that there is no limit on the 'types' of objects that sets and ordered pairs may contain. We axiomatise ZFP and prove its consistency relative to ZF. We hope that our explanation of how we did this will be useful guidance for other work extending set theories.

ZFP extends $\in$ with two new binary predicate symbols, π_1 and π_2, whose intended meanings are 'is first projection of' and 'is second projection of'. We define abbreviations for formulas $\mathsf{Set}(x)$ and $\mathsf{Pair}(x)$ that distinguish sets and ordered pairs by the rule that an ordered pair has a first projection and a set does not. ZFP's axioms are in two groups, one for sets and one for ordered pairs. We were able to generate nearly all of ZFP's axioms for sets by modifying the axioms of ZF by restricting quantifiers using $\mathsf{Set}(x)$ in the right places. The axiom of Foundation needed to be modified to handle sets and ordered pairs simultaneously. ZFP's axioms for ordered pairs specify the expected abstract properties, including that ordered pairs have no set members.

To prove ZFP is consistent if ZF is, we construct in ZF a model and prove it satisfies ZFP's axioms [4]. Building a model for a set theory with non-set objects with 'internal' structure that can include sets differs from building a model for a set theory with no urelements or with only simple urelements, because there can be new non-set objects at each stage of the construction. $\mathbf{W}$, the domain of our model, is similar to the domain $\mathbf{V}$ of the Von Neumann hierarchy. Each tier of $\mathbf{V}$ is constructed by taking the power set of the previous tiers. In contrast, when building the tiers of $\mathbf{W}$, each successor tier W_{α^+} is formed by taking the disjoint sum of the power set $\mathcal{P}(W_\alpha)$ and the cartesian product $(W_\alpha)^2$. Hence every object in $\mathbf{W}$ has a tag that tells whether it is intended to model a set or an ordered pair. This supports defining relations that model ZFP's $\in$, π_1, and π_2

which may only return true when their second argument is of the correct 'type'. This proof has been machine-checked in Isabelle/ZF.[1]

Although our model for ZFP is built purely of sets and implements ordered pairs as sets, another model could use other methods (e.g., type-theoretic) and implement ordered pairs differently. Hence, we have put an 'abstraction barrier' between the user of ZFP and the implementation of ordered pairs.

1.4 Related Work

Harrison [6] details the challenges that face both type-theoretic and set-theoretic foundations for formalised mathematics. Harrison makes the case for using set theory as 'machine code', leaving theorem proving to layers of code. Harrison suggests using a set theory with urelements to avoid the issue of 'accidental theorems'. Weidijk [13] formulates axiomatic set theories and type theories in AutoMath in order to compare them and assess their relative complexity.

A significant work aiming to make computer formalisation of set-theoretical mathematics practical is Farmer's Chiron [5], a conservative extension of the set theory NBG (itself a conservative extension of ZF). Chiron has additional features such as support for undefinedness, definite descriptions, quotation and evaluation of expressions, and a kind of types.

Aczel and Lunnon worked on Generalised Set Theory (GST) [1] with the aim of better supporting work in situation theory. GST extends set theory with a mechanism for primitive functions, as well as a number of other features. It appears that GST assumes the Anti-Foundation axiom instead of Foundation which ZF uses. Unfortunately, we failed to find a specification of the axioms of GST. Part of GST seems similar to our work but a technical comparison is difficult without the axioms.

Although ordered pairs now seem obvious, Kanamori's excellent history [8] shows a sequence of conceptual breakthroughs were needed to reach the modern ordered pair. How we built a model for ZFP was heavily inspired by the way Barwise [2] interprets KPU (Kripke-Platek set theory with Urelements) in KP.

1.5 Outline

Section 2 presents and discusses the first-order logic we use and definitions and axioms of ZF. Section 3 presents and discusses ZFP in the form of definitions and two collections of axioms, one for sets, and one for ordered pairs. Section 4 proves the existence in ZF of a model for the axioms of ZFP (which implies that ZFP is consistent if ZF is). Section 5 discusses the significance of these results, and how they will be used in further investigation.

[1] The source and a prettier HTML rendering of the Isabelle/ZF proofs is contained in a ZIP archive embedded in this PDF file as an 'attachment'. If your viewer does not support extraction of attachments embedded in PDF files, you can also see `http://www.macs.hw.ac.uk/~cmd1/cicm2020/ZFP.thy` for the source, and `http://www.macs.hw.ac.uk/~cmd1/cicm2020/ZFPDoc/index.html` for the HTML.

2 Formal Machinery

Let $X := Y$ be meta-level notation meaning that X stands for Y.

2.1 First-Order Logic with Equality

We use a fragment of first-order logic (FOL) with equality sufficient for defining ZF and ZFP. We consider only four binary infix *predicate symbols* including equality. The MBNF [12] specification of the syntax is:

$$a, \ldots, z \mathrel{\dot\in} \mathsf{Var} ::= \mathsf{v0} \mid \mathsf{v1} \mid \cdots \qquad \sim \mathrel{\dot\in} \mathsf{Pred} ::= \in \mid \pi_1 \mid \pi_2 \mid =$$
$$A, \ldots, Z \mathrel{\dot\in} \mathsf{Term} ::= x \mid \imath x : \varphi \qquad \varphi, \psi \mathrel{\dot\in} \mathsf{Form} ::= X \sim Y \mid \varphi \to \psi \mid \neg\varphi \mid \forall x : \varphi$$

We work with terms and formulas modulo α-conversion where $\forall x$ and $\imath x$ bind x. Except where explicitly specified otherwise, we require metavariables ranging over the set Var to have the attribute of *distinctness*. Two different metavariables with the distinctness attribute can not be equal. For example, $x = \mathsf{v9}$ and $x_1 = \mathsf{v27}$ and $y = \mathsf{v53}$ could hold simultaneously, but neither $x = \mathsf{v9} = x_1$ nor $x = \mathsf{v53} = y$ are allowed. This restriction applies only to metavariables: the same object-level variable can be used in nested scopes, e.g., the formula $(\forall\, \mathsf{v7} : \forall\, \mathsf{v7} : \mathsf{v7} \in \mathsf{v7})$ is fine and equal to $(\forall\, \mathsf{v0} : \forall\, \mathsf{v1} : \mathsf{v1} \in \mathsf{v1})$. We assume the usual abbreviations for logical connectives ($\wedge$, $\vee$, $\leftrightarrow$), for quantifiers ($\exists$, $\exists!$, $\forall x_1, \ldots, x_n$, $\exists x_1, \ldots, x_n$), and for predicate symbols ($\neq$, $\notin$, $\ni$).

A term can be a *definite description* $(\imath\, x : \varphi)$ which, if there is exactly one member x of the domain of discourse such that the formula φ is true, evaluates to that member and otherwise evaluates to a special value $\bot$ outside the domain of discourse such that any predicate symbol (including equality) with $\bot$ as an argument evaluates to false.[2] A term is said to be *undefined* or to *have no value* iff it evaluates to $\bot$. An alternative specification of definite descriptions that gives formulas the same meanings is eliminating them by the following rule (only the left case is given; the right case is similar):

$$((\imath\, x : \varphi) \sim Y) := (\exists\, x : x \sim Y \wedge \varphi) \wedge \exists!\, x : \varphi \text{ where } x \text{ is not free in } Y$$

2.2 Zermelo-Fraenkel Set Theory

The only predicate symbols ZF uses are the *membership relation* $\in$ and equality. ZF makes no use of the FOL predicate symbols π_1 and π_2, but instead we define these symbols as parts of abbreviations in section 2.3. We use the following abbreviations where $n \geq 3$ and a, c, x, y, and z are not free in the other arguments

[2] When working with functions that might be applied outside their domain, one might prefer to have $\bot = \bot$, but this is a bit more complex and not needed for this paper.

and b is not free in X:

$$(\forall b \in X : \varphi) := (\forall b : b \in X \to \varphi) \qquad (\exists b \in X : \varphi) := (\exists b : b \in X \wedge \varphi)$$
$$\cup X := (\imath y : \forall a : a \in y \leftrightarrow \exists z \in X : a \in z) \qquad X \subseteq Y := (\forall\, c \in X : c \in Y)$$
$$\{A, B\} := (\imath\, x : \forall c : c \in x \leftrightarrow (c = A \vee c = B)) \qquad X \cup Y := \cup \{X, Y\}$$
$$\mathcal{P}(X) := (\imath\, y : \forall\, z : z \in y \leftrightarrow z \subseteq X) \qquad \{A\} := \{A, A\}$$
$$\{A_1, \ldots, A_n\} := \{A_1\} \cup \{A_2, \ldots, A_n\} \qquad \emptyset := (\imath x : \forall a : a \notin x)$$
$$\{\, b \in X \mid \varphi \,\} := (\imath\, y : \forall b : b \in y \leftrightarrow (b \in X \wedge \varphi)) \qquad X^+ := X \cup \{X\}$$

These abbreviations are defined if their arguments are defined due to the axioms.

Definition 2.1. *The axioms of ZF are all the instances of the following formulas for every formula φ with free variables at most a, b, c_1 and c_2.*

1. *Extensionality:* $\forall x, y : (\forall a : a \in x \leftrightarrow a \in y) \to x = y$
2. *Union:* $\forall x : \exists y : \forall a : a \in y \leftrightarrow (\exists z \in x : a \in z)$
3. *Power Set:* $\forall x : \exists y : \forall z : z \in y \leftrightarrow z \subseteq x$
4. *Infinity (ugly version; see pretty version below):* $\exists y : (\exists\, z \in y : \forall\, b : b \notin z) \wedge (\forall\, x \in y : \exists\, s \in y : \forall\, c : c \in s \leftrightarrow (c \in x \vee c = x))$
5. *Replacement:* $\forall c_1, c_2, x : (\forall a \in x : \exists! b : \varphi) \to (\exists y : \forall b : b \in y \leftrightarrow \exists a \in x : \varphi)$
6. *Foundation:* $\forall x : x = \emptyset \vee (\exists y \in x : \neg \exists b \in x : b \in y)$

The axioms are due to Zermelo, except for Replacement which is due to Fraenkel and Skolem [3] and Foundation which is due to Von Neumann. Extensionality asserts that sets are equal iff they contain the same members. Union and Power Set state that $\cup X$ and $\mathcal{P}(X)$ are defined if X is defined; this implies the domain of discourse is closed under $\cup$ and $\mathcal{P}$. Infinity states that there exists a set containing $\emptyset$ which is closed under the ordinal successor operation; from this we can extract the Von Neumann natural numbers $\mathbb{N}$. Here is a prettier presentation of Infinity that we do not use as the axiom to avoid bootstrap confusion[3]:

$$\exists y : \emptyset \in y \wedge (\forall x \in y : x^+ \in y)$$

The powerful infinite axiom schema Replacement asserts the existence of the range of a function determined by any formula φ where the values of the variables a and b that make φ true have a functional dependency of b on a and where the domain of the function exists as a set. Foundation enforces the policy that there are no infinite descending chains of the form $X_0 \ni X_1 \ni \cdots$.

Lemma 2.2. *The following theorems of ZF are often presented as axioms. For every formula φ such that any free variable must be a, the following hold in ZF:*

1. *Empty Set:* $\exists x : \forall b : b \notin x$
2. *Pairing:* $\forall a, b : \exists x : \forall c : (c \in x \leftrightarrow (c = a \vee c = b))$
3. *Specification:* $\forall x : \exists y : \forall a : (a \in y \leftrightarrow (a \in x \wedge \varphi))$

[3] Provided *some* object exists, Replacement can build $\emptyset$, and then further axiom use can build operations like $\{B, C\}$, $\{B\}$, $X \cup Y$, and X^+, thus ensuring the terms $\emptyset$ and x^+ are defined in the pretty version of Infinity. We prefer getting that initial object from an axiom over using the FOL assumption that the domain of discourse is non-empty. The only axiom giving an object for free is Infinity. We find it confusing to use Infinity in proving the definedness of subterms of itself, so we don't.

2.3 Ordered Pairs in ZF

We define the Kuratowski ordered pair $\langle A, B\rangle$ and related operations as follows where a, b, p, and x are not free in A, B, and Q:

$$\langle A, B\rangle := \{\{A\}, \{A, B\}\}$$
$$A\ \pi_1\ Q := (\forall x \in Q : A \in x) \qquad B\ \pi_2\ Q := (\exists! x \in Q : B \in x)$$
$$A \times B := (\imath x : \forall p : p \in x \leftrightarrow (\exists c \in A, d \in B : p = \langle c, d\rangle))$$

We call a and b the *first* and *second projections* of $\langle a, b\rangle$ respectively. The first projection of an ordered pair q is in all sets in q, whereas the second is only in one.[4] The projection relations π_1 and π_2 only give meaningful results when the set Q on the right side of the relation is an ordered pair, i.e., this holds:

$$(\exists c, d : Q = \langle c, d\rangle) \rightarrow (\forall a, b : (a\ \pi_1\ Q \wedge b\ \pi_2\ Q) \leftrightarrow Q = \langle a, b\rangle)$$

Kuratowski ordered pairs are sets and have set members that are distinct from their projections. In fact, no matter which representation we use, there will always exist some x such that $x \in \langle a, b\rangle$ (for all but at most one ordered pair which can be represented by $\emptyset$). If A and B are defined, we can show the cartesian product $A \times B$ is defined using Replacement nested inside Replacement[5]:

$$A \times B = \cup\{\, z \mid \exists\, c \in A : z = \{\, p \mid \exists\, d \in B : p = \langle c, d\rangle \,\}\,\}$$

3 Extending ZF to ZFP

This section introduces Zermelo-Fraenkel Set Theory with Ordered Pairs (ZFP), a set theory with primitive non-set ordered pairs. ZFP axiomatises the membership predicate symbol $\in$ similarly to ZF. The ordered pair projection predicate symbols π_1 and π_2 are axiomatised in ZFP instead of being abbreviations that use $\in$ as in ZF. Ordered pairs in ZFP qualify as urelements because they contain no members via the set membership relation $\in$, but they are unusual urelements because they can contain arbitrary sets via the π_1 and π_2 relations.

3.1 Definitions and Axioms of ZFP

We use the metavariables p, q, P, and Q where it might help the reader to think 'ordered pair', and the metavariables s, x, y, z, X, Y, and Z where it might help the reader to think 'set'; this convention has no formal status and all FOL variables continue to range over all objects in the domain of discourse. We call b a *member* of x iff $b \in x$. We call b a *projection* of q iff $b\ \pi_1\ q$ or $b\ \pi_2\ q$. An

[4] This holds even in the case of $\langle a, a\rangle = \{\{a\}, \{a, a\}\} = \{\{a\}\}$.

[5] The traditional construction of $A \times B$ as $\{\, p \in \mathcal{P}(\mathcal{P}(A \cup B)) \mid \exists c \in A, d \in B : p = \langle c, d\rangle \,\}$ is only needed if the weaker Specification is preferred over Replacement. We avoid the traditional construction because it depends on a set representation of ordered pairs and thus will not work for ZFP.

ordered pair is any object with a projection, and a *set* is any object that is not an ordered pair. We use the following abbreviations where b is not free in Q and X and q is not free in A and B:

$$\begin{array}{ll} \mathsf{Pair}(Q) & := \exists b : b\ \pi_1\ Q \\ \forall_{\mathsf{Pair}}\, p : \varphi & := \forall p : \mathsf{Pair}(p) \to \varphi \\ \exists_{\mathsf{Pair}}\, p : \varphi & := \exists p : \mathsf{Pair}(p) \wedge \varphi \\ \imath_{\mathsf{Pair}}\, p : \varphi & := \imath p : \mathsf{Pair}(p) \wedge \varphi \\ (A, B) & := (\imath q : A\ \pi_1\ q \wedge B\ \pi_2\ q) \end{array} \qquad \begin{array}{ll} \mathsf{Set}(X) & := \neg\mathsf{Pair}(X) \\ \forall_{\mathsf{Set}}\, x : \varphi & := \forall x : \mathsf{Set}(x) \to \varphi \\ \exists_{\mathsf{Set}}\, x : \varphi & := \exists x : \mathsf{Set}(x) \wedge \varphi \\ \imath_{\mathsf{Set}}\, x : \varphi & := \imath x : \mathsf{Set}(x) \wedge \varphi \end{array}$$

We reuse the text of the abbreviation definitions for ZF for $\{A, B\}$, $X \cup Y$, $\{A\}$, and $\{A_1, \ldots, A_n\}$ where $n \geq 3$. We redefine the following abbreviations a bit differently for ZFP, where a, b, c, p, x, y, and z are not free in A, B, X and Y:

$$\begin{array}{ll} X \subseteq Y & := \mathsf{Set}(X) \wedge \mathsf{Set}(Y) \wedge (\forall\, c \in X : c \in Y) \\ \cup X & := (\imath_{\mathsf{Set}}\, y : \forall a : a \in y \leftrightarrow \exists z \in X : a \in z) \\ \mathcal{P}(X) & := (\imath_{\mathsf{Set}}\ y : \forall\, z : z \in y \leftrightarrow z \subseteq X) \\ \emptyset & := (\imath_{\mathsf{Set}}\, x : \forall a : a \notin x) \\ \{\, b \in X \mid \varphi \,\} & := (\imath_{\mathsf{Set}}\ y : \forall b : b \in y \leftrightarrow (b \in X \wedge \varphi)) \\ A \times B & := (\imath x : \forall p : p \in x \leftrightarrow (\exists c \in A, d \in B : p = (c, d))) \end{array}$$

These abbreviations are defined if their arguments are defined due to the axioms.

Definition 3.1. *The axioms of ZFP are all the instances of the following formulas for every formula φ with free variables at most a, b, c_1, c_2.*

- ***Sets:***
  - ***S1.*** *Set Extensionality:* $\forall_{\mathsf{Set}}\, x, y : (\forall a : a \in x \leftrightarrow a \in y) \to x = y$
  - ***S2.*** *Union:* $\forall_{\mathsf{Set}}\, x : \exists y : \forall a : a \in y \leftrightarrow (\exists z \in x : a \in z)$
  - ***S3.*** *Power Set:* $\forall_{\mathsf{Set}}\, x : \exists y : \forall z : z \in y \leftrightarrow z \subseteq x$
  - ***S4.*** *Infinity (ugly version):* $\exists y : (\exists_{\mathsf{Set}}\ z \in y : \forall\, b : b \notin z) \wedge (\forall\, x \in y : \exists\, s \in y : \forall\, c : c \in s \leftrightarrow (c \in x \vee c = x))$.
  - ***S5.*** *Replacement:* $\forall c_1, c_2, x : (\forall a \in x : \exists! b : \varphi) \to (\exists_{\mathsf{Set}}\, y : \forall b : b \in y \leftrightarrow \exists a \in x : \varphi)$
  - ***S6.*** *Foundation:* $\forall_{\mathsf{Set}}\, x : x = \emptyset \vee (\exists a \in x : \neg\exists b \in x : b\ \pi_1\ a \vee b\ \pi_2\ a \vee b \in a)$
- ***Ordered Pairs:***
  - ***P1.*** *Ordered Pair Emptiness:* $\forall_{\mathsf{Pair}}\, p : \forall a : a \notin p$
  - ***P2.*** *Ordered Pair Formation:* $\forall a, b : \exists p : a\ \pi_1\ p \wedge b\ \pi_2\ p$
  - ***P3.*** *Projection Both-Or-Neither:* $\forall p : (\exists a : a\ \pi_1\ p) \leftrightarrow (\exists b : b\ \pi_2\ p)$
  - ***P4.*** *Projection Uniqueness:* $\forall_{\mathsf{Pair}}\, p : (\exists! a : a\ \pi_1\ p) \wedge (\exists! b : b\ \pi_2\ p)$
  - ***P5.*** *Ordered Pair Extensionality:* $\forall_{\mathsf{Pair}}\, p, q : (\forall a : (a\ \pi_1\ p \leftrightarrow a\ \pi_1\ q) \wedge (a\ \pi_2\ p \leftrightarrow a\ \pi_2\ q)) \to p = q$

Lemma 3.2. *For every formula φ such that any free variable must be a, the following hold in ZFP:*

1. *Unordered/Set Pairing:* $\forall a, b : \exists x : \forall c : c \in x \leftrightarrow (c = a \vee c = b)$
2. *Specification:* $\forall_{\mathsf{Set}}\, x : \exists_{\mathsf{Set}}\, y : \forall a : a \in y \leftrightarrow (a \in x \wedge \varphi))$

3. *Cartesian Product Existence:*
   $\forall_{\mathsf{Set}}\, x, y : \exists_{\mathsf{Set}}\, z : \forall p : p \in z \leftrightarrow (\exists a \in x, b \in y : a\ \pi_1\ p \wedge b\ \pi_2\ p)$

For lemma 3.2 (3), note that the cartesian product $A \times B$ can be built in ZFP using the same construction given for ZF in section 2.3, which does not depend on any set representation of ordered pairs.

3.2 Discussion

Axioms for Sets. Each ZF axiom was transformed to make a ZFP axiom. First, because we use abbreviations for more readable axioms, those used in axioms needed to be modified for ZFP. The definition of $\subseteq$ (used in Power Set) was changed to ensure an ordered pair is neither a subset nor has a subset. The definition of $\emptyset$ (used in Foundation) was changed to ensure a defined result.

Second, some occurrences of $(\forall\, b : \psi)$ and $(\exists\, b : \psi)$ needed to enforce that ψ can be true only when b stands for a set. Where needed, such occurrences were changed to $(\forall_{\mathsf{Set}}\ b : \psi)$ respectively $(\exists_{\mathsf{Set}}\ b : \psi)$. Each quantifier needed individual consideration. If the sethood of b was already enforced by ψ only being true when b has at least 1 set member, there was no need for a change but a change might also clarify the axiom. If the truth of ψ was unaffected by any set members of b, there was no need for a change and this generally indicated that a change would go against the axiom's intention. We needed to understand the axiom's *intention* and *expected usage* because it was not written to specify where it is expected that 'X is a set' (because this always holds in ZF).

Finally, Foundation was extended to enforce a policy of no infinite descending chains through not just $\in$ but also π_1 and π_2, so that ZF proofs using Kuratowski ordered pairs (having no such chains) would continue to work in ZFP.

Consider the example of Power Set which states that for any set X there exists a set Y containing all of the subsets of X and nothing else, i.e., $\mathcal{P}(X)$:

$$\forall_{\mathsf{Set}}\, x : \exists y : \forall z : (z \in y \leftrightarrow z \subseteq x)$$

We could have left $\forall_{\mathsf{Set}}\, x$ as $\forall x$, because when x is an ordered pair it would act like $\emptyset$ and this would only add another reason that $\mathcal{P}(\emptyset)$ exists. However, we thought this would be obscure. It would not hurt to change $\exists y$ to $\exists_{\mathsf{Set}}\, y$ but there is no need to do so because the body forces y to contain a set member and hence rejects y being an ordered pair. We did not change $\forall z$ to $\forall_{\mathsf{Set}}\, z$ because this would allow y to contain extra junk ordered pairs that proofs expecting to get $\mathcal{P}(x)$ would have to do extra work using Replacement to filter out.

Axioms for Ordered Pairs. The ZFP axioms for ordered pairs specify the abstract properties of ordered pairs via the relations π_1 and π_2. These ordered pairs have no 'type' restrictions, i.e., each pair projection can be either a set or an ordered pair. Ordered Pair Emptiness (P1) ensures that no object has both a projection (ordered pairs only) and a set member (sets only). Ordered Pair Formation (P2) ensures that for every two objects b and c there exists

an ordered pair with b as first projection and c as second. Projection Both-Or-Neither (P3) ensures that every object either has no projections (sets) or both projections (ordered pairs). Projection Uniqueness (P4) ensures each ordered pair has exactly one first projection and one second projection. Ordered Pair Extensionality (P5) ensures that for every choice of first and second projections, there is exactly one ordered pair.

Comparing the Objects and Theorems of ZF and ZFP. A set is *pure* iff all its members are pure sets. Each ZF object is a pure set and is also a pure set of ZFP, but ZFP has additional impure sets which have members that are primitive ordered pairs or impure sets, and ZFP also has primitive ordered pairs. The set membership relation $\in$ of ZF is the restriction of the relation $\in$ of ZFP to pure sets. Let $\mathsf{Pure}(x)$ be a formula (implemented with transfinite recursion) that holds in ZFP when x is a pure set. For every ZF formula φ, let $\mathsf{PRestrict}(\varphi)$ be the ZFP formula obtained from φ by changing each subformula $(\forall\, x : \psi)$ to $(\forall\, x : \mathsf{Pure}(x) \to \psi)$. Then φ is a ZF theorem iff $\mathsf{PRestrict}(\varphi)$ is a ZFP theorem. If one wants to go the other direction and take a ZFP formula ψ and find a ZF formula ψ' that 'does the same thing', one must represent as ZF sets both (1) the primitive ordered pairs *and* (2) the sets of ZFP, and then one must either prevent or somehow manage the possible confusion between the representations of (1) and (2). Section 4.2 is an example of doing this rigorously.

Design Alternatives. We considered having the projections π_1 and π_2 be unary FOL function symbols, but this would require the term $\pi_1(x)$ to denote an object within the domain of discourse for every set x, so we avoided this. We considered having the pairing operator $(\cdot, \cdot)$ be a binary FOL function symbol. Using a binary function symbol would mean the graph model would have hyper-edges (i.e., connecting 3 or more nodes) which is more difficult to think about. Because we used two separate binary predicate symbols, one for each projection, we get a fairly standard-looking directed-graph model with ordinary edges. If we used a binary FOL function symbol $(\cdot, \cdot)$ for pairing, we could replace our axioms P2, P3, P4, and P5 by the characteristic property of ordered pairs:

$$\forall\, a, b, c, d : (a, b) = (c, d) \to (a = b \wedge c = d)$$

Our axioms can be seen as the result of applying a function-symbol-elimination transformation to this alternative.

Very early on, we considered simply using ZF's axioms as they are, adding a binary pairing function symbol, and adding the characteristic property of ordered pairs as an axiom. In this theory, formulas such as $\{b\} \in \langle b, c\rangle$ would be independent, because the representation of ordered pairs would be unknown (and need not even be definable in ZF), so some 'junk theorems' would no longer hold. We avoided this alternative for many reasons. First, Extensionality would force all but one ordered pair (which could be $\emptyset$) to have set members, so there would be 'junk theorems' such as $(a, b) \neq (c, d) \to \exists\, e : e \in (a, b) \leftrightarrow e \notin (c, d)$.

Second, we could not see how to do transfinite induction and recursion. Third, genuine non-sets make it easier to talk about the distinction between sets and conceptually non-set objects, e.g., to students. Fourth, we hope our approach might help a weak form of 'type checking', where a prover might more quickly solve or disprove subgoals, and if a user mistakenly requires a non-set to have a set member, this might be detected earlier and result in a more understandable failure message. Some further reasons are discussed in section 1.

4 A Model of ZFP

We define within ZF a model for ZFP, i.e., an interpretation of the domain and predicate symbols of ZFP. A translation from a ZFP formula ψ to a ZF formula ψ^* is defined to interpret ZFP formulas in the model. Terms and formulas in this section belong to ZF except for the arguments of $(\,\cdot\,)^*$. All axioms of ZFP hold under this translation, which implies that if ZF is consistent, so is ZFP [4]. That each axiom's translation holds has been checked in Isabelle/ZF.

4.1 The Cumulative Hierarchy W

Like the Von Neumann universe **V** used as the domain of a model of ZF, our domain **W** is a set hierarchy indexed by ordinal numbers.

An *ordinal* is a *transitive* set that is totally ordered by $\in$, which we specify formally by $\mathsf{Ord}(x) := (\forall\, y \in x : y \subseteq x) \wedge (\forall\, y, z \in x : y = z \vee y \in z \vee z \in y)$. Let α and β range over ordinals. Let $0 := \emptyset$, $1 := 0^+$, $2 := 1^+$, and so on. Ordinal β is a *successor* ordinal iff $\beta = \alpha^+$ for some α. Ordinal β is a *limit* ordinal iff β is neither 0 nor a successor ordinal. Let λ range over limit ordinals. Let $(x < y) := (x \in y \wedge \mathsf{Ord}(y))$ and define related symbols (e.g., $\leq$) as usual.

Any model of ZFP must have some way of distinguishing between the objects in its domain representing ZFP sets, and those that represent ZFP pairs, i.e., ZFP needs a domain split into two disjoint subdomains. We model this in ZF using Kuratowski ordered pairs and cartesian products to tag all domain objects with 0 ('set') or 1 ('ordered pair').

Definition 4.1. *For ordinal α, define the set W_α via transfinite recursion thus:*

$$W_0 = \emptyset, \qquad W_{\beta^+} = (\{0\} \times \mathcal{P}(W_\beta)) \cup (\{1\} \times (W_\beta)^2), \qquad W_\lambda = \bigcup_{\beta \in \lambda} W_\beta$$

Starting from $\emptyset$, each successor tier W_{β^+} is built by taking the disjoint union of the power set and cartesian square of the previous tier. Each limit tier W_λ is the union of all preceding tiers. The use of disjoint union to build each successor tier W_{β^+} gives a set-theoretic universe split into two. Although our disjoint union uses Kuratowski pairs with 0 and 1 tags, we could use instead any two definable injective operators from a large enough class (e.g., the universe) to disjoint classes that raise rank by at most a constant.

Let $\mathbf{W}$ be the proper class such that $x \in \mathbf{W}$ iff $x \in W_\alpha$ for some α. We use a bold upright serif font to emphasize that $\mathbf{W}$ is not a ZF set.[6] By the transfinite recursion theorem, given x there is a definite description $\mathsf{W}(x)$ that evaluates to W_α when x evaluates to α.[7] We express X belonging to $\mathbf{W}$ as follows:

Definition 4.2. $\mathcal{H}(X) := (\exists\, y : \mathsf{Ord}(y) \wedge X \in \mathsf{W}(y))$.

Let an *m-object* be any member of $\mathbf{W}$ (i.e., a ZF set x such that $\mathcal{H}(x)$ holds), an *m-set* be any m-object of the form $\langle 0, x\rangle$, and an *m-pair* be any m-object of the form $\langle 1, x\rangle$. The following result says every m-object x is either an m-set or an m-pair, and tells where in the hierarchy the contents of x are.

Lemma 4.3. *Suppose $\mathcal{H}(x)$, so that $x \in W_\alpha$. Then for some $\beta < \alpha$ either:*

$$x = \langle 0, x'\rangle \text{ where } x' \subseteq W_\beta, \quad \text{or} \quad x = \langle 1, \langle a, b\rangle\rangle \text{ where } a, b \in W_\beta.$$

It holds that $\mathbf{W}$ is a cumulative hierarchy:

Lemma 4.4. *If $\alpha \leq \beta$, then $W_\alpha \subseteq W_\beta$.*

4.2 Interpreting ZFP in ZF

As explained above, we interpret the sets and ordered pairs of ZFP as the members of $\mathbf{W}$. Lemma 4.3 says any m-object is an ordered pair whose left projection is an integer which decides its 'type' and whose right projection is either a set or an ordered pair. We define our interpretations of ZFP's predicate symbols:

Definition 4.5. *Let $\mathrel{\widehat{\in}}$, $\widehat{\pi}_1$, and $\widehat{\pi}_2$ be defined by these abbreviations:*

$$\begin{aligned} a \mathrel{\widehat{\in}} x &:= (\exists\, y : x = \langle 0, y\rangle \wedge a \in y) \\ a\ \widehat{\pi}_1\ p &:= (\exists\, u, v : p = \langle 1, \langle u, v\rangle\rangle \wedge a = u) \\ a\ \widehat{\pi}_2\ p &:= (\exists\, u, v : p = \langle 1, \langle u, v\rangle\rangle \wedge a = v) \end{aligned}$$

$\mathbf{W}$ is downward closed under these three relations. That is:

Lemma 4.6. *Suppose $\mathcal{H}(x)$, i.e., $x \in W_\alpha$ for some α. Suppose $a \mathrel{\widehat{\in}} x$, $a\ \widehat{\pi}_1\ x$, or $a\ \widehat{\pi}_2\ x$ for some a. Then $a \in W_\beta$ for some $\beta < \alpha$, and thus $\mathcal{H}(a)$.*

To interpret a ZFP formula φ in ZF, we must show the formula holds when quantification is restricted to the domain $\mathbf{W}$, and the predicate symbols are replaced by the interpretations defined above.

Definition 4.7. *Let φ be a ZFP formula. Define φ^* recursively as follows:*

$$\begin{aligned} (X \in Y)^* &:= (X^*) \mathrel{\widehat{\in}} (Y^*) & (\varphi \to \psi)^* &:= (\varphi^*) \to (\psi^*) \\ (X\ \pi_1\ Y)^* &:= (X^*)\ \widehat{\pi}_1\ (Y^*) & (\neg\varphi)^* &:= \neg(\varphi^*) \\ (X\ \pi_2\ Y)^* &:= (X^*)\ \widehat{\pi}_2\ (Y^*) & (\forall x : \varphi)^* &:= (\forall\, x : \mathcal{H}(x) \to (\varphi^*)) \\ x^* &:= x & (\imath\, x : \varphi)^* &:= (\imath\, x : \mathcal{H}(x) \wedge (\varphi^*)) \end{aligned}$$

[6] $\mathbf{W}$ is a mathematical object in some other set theories.

[7] A nested definite description is used that specifies the function f such that $f(\beta) = W_\beta$ for $\beta \leq \alpha$, i.e., f is an initial prefix of the hierarchy. Then $f(\alpha)$ is returned.

Lemma 4.8. $(\exists\, x : \varphi)^* \leftrightarrow (\exists\, x : \mathcal{H}(x) \wedge (\varphi^*))$.

Because the translation $(\cdot)^*$ inserts quite a lot of extra structure, a ZFP user wanting to understand "the ZF formula corresponding to the ZFP formula ψ" might be tempted to instead translate ZFP's $\in$ directly to ZF's $\in$ and ZFP's π_1 and π_2 to the ZF abbreviations for π_1 and π_2 defined in section 2.3. However, as discussed in section 1.2, the user then would need to carefully prove that no problems arise from the coincidences where a ZFP set x and a ZFP primitive ordered pair p would be represented by the same ZF set y.

Observe that the ZFP abbreviations Set and Pair from section 3.1 that act like unary predicates are interpreted in ZF as follows:

$$\mathsf{Pair}(x)^* := (\exists\, a : \mathcal{H}(a) \wedge a\ \widehat{\pi}_1\ x) \qquad\qquad \mathsf{Set}(x)^* := \neg(\mathsf{Pair}(x)^*)$$

These predicates are clearly meaningful within the model because:

Lemma 4.9. *Suppose that $\mathcal{H}(x)$, then we have that:*

$$\mathsf{Pair}(x)^* \leftrightarrow (\exists\, a, b : x = \langle 1, \langle a, b\rangle\rangle) \qquad\qquad \mathsf{Set}(x)^* \leftrightarrow (\exists\, y : x = \langle 0, y\rangle)$$

Now we reach our main result, which implies ZFP is consistent if ZF is [4]:

Theorem 4.10. *For each ZFP axiom φ, the translation φ^* holds in ZF.*

The proof of this theorem simply observes the conjunction of a number of lemmas, each of which shows for a ZFP axiom ϕ that ϕ^* holds in ZF. Most of these lemmas are straightforward. Here we show a representative example:

Lemma 4.11. *The translation of ZFP's Power Set axiom holds in ZF.*

Proof. First, we find the translation using definition 4.7 and lemma 4.8:

$$\forall x : \mathcal{H}(x) \rightarrow (\mathsf{Set}(x)^* \rightarrow (\exists y : \mathcal{H}(y) \wedge \forall z : \mathcal{H}(z) \rightarrow (z\ \widehat{\in}\ y \leftrightarrow ((z \subseteq x)^*))))$$

Let x be such that $\mathcal{H}(x)$, and suppose $\mathsf{Set}(x)^*$. By lemma 4.9, $x = \langle 0, x'\rangle$ for some set x'. Let $y = \langle 0, y'\rangle$ where $y' = \{0\} \times \mathcal{P}(x')$ be our candidate for the power set. We must show that y has the property $\forall z : \mathcal{H}(z) \rightarrow (z\ \widehat{\in}\ y \leftrightarrow (z \subseteq x)^*)$, and also that y is indeed a member of $\mathbf{W}$. Fix z and assume $\mathcal{H}(z)$, then:

$$\begin{aligned}
z\ \widehat{\in}\ y &\leftrightarrow z \in y' && \text{by def of } y \text{ and } \widehat{\in}\\
&\leftrightarrow z \in \{0\} \times \mathcal{P}(x') && \text{by def of } y'\\
&\leftrightarrow \exists z' : z = \langle 0, z'\rangle \wedge z' \subseteq x' && \text{by def of } \times \text{ and } \mathcal{P}\\
&\leftrightarrow \mathsf{Set}(z)^* \wedge (\forall a : a\ \widehat{\in}\ z \rightarrow a\ \widehat{\in}\ x) && \text{since } z = \langle 0, z'\rangle, z' \subseteq x'\\
&\leftrightarrow \mathsf{Set}(z)^* \wedge \mathsf{Set}(x)^* \wedge (\forall a : a\ \widehat{\in}\ z \rightarrow a\ \widehat{\in}\ x) && \text{since } \mathcal{H}(x),\ x = \langle 0, x'\rangle\\
&\leftrightarrow (z \subseteq x)^* && \text{because } \mathcal{H}(z)
\end{aligned}$$

It now remains to show that $\mathcal{H}(y)$. From $\mathcal{H}(x)$, we have that $x \in W_\alpha$ for some ordinal α. By lemma 4.4, $x \in W_{\alpha^+}$, and by lemma 4.3, $x' \subseteq W_\alpha$. Then:

$$\begin{aligned}
x' \subseteq W_\alpha &\to \mathcal{P}(x') \subseteq \mathcal{P}(W_\alpha) \\
&\to \{0\} \times \mathcal{P}(x') \subseteq \{0\} \times \mathcal{P}(W_\alpha) \\
&\to y' \subseteq \{0\} \times \mathcal{P}(W_\alpha) && \text{by def of } y' \\
&\to y' \subseteq W_{\alpha^+} && \text{because } \{0\} \times \mathcal{P}(W_\alpha) \subseteq W_{\alpha^+} \\
&\to y \in W_{\alpha^{++}} && \text{by def of } y = \langle 0, y' \rangle \\
&\to \mathcal{H}(y) && \text{by def of } \mathcal{H}
\end{aligned}$$

□

5 Conclusion

5.1 Summary of Contributions

Presenting ZF Set Theory using Definite Descriptions. In section 2, we give a formal presentation of ZF that accounts for the technical details, whilst also defining notation for widely used operations. Although correct formal definitions of this notation can be found in computer implementations of set theory, we have not seen definite descriptions used for this in published articles. Definite descriptions allow defining terms in a compact and readable way without needing to add FOL function symbols, extend the model, or otherwise appeal to the meta-level. We show precisely how Kuratowski pairs and their operations are defined and highlight issues arising from their set representations.

Axiomatizing ZFP. Motivated by issues with the set representation in pure ZF set theory of conceptually non-set objects, in section 3 we introduce Zermelo-Fraenkel Set Theory with Ordered Pairs, which extends ZF with predicate symbols π_1 and π_2 and axioms to implement primitive non-set ordered pairs. ZFP is akin to some alternative set theories that use urelements as genuine non-set objects in the domain, with the difference that ZFP's urelements have meaningful internal structure endowed by the axiomatisation of π_1 and π_2. The design of ZFP is deliberately similar to that of ZF, so that we can better understand the relationship between the two theories. We axiomatize ZFP, and discuss how the axioms of ZF were modified to yield the axioms of ZFP. As a result, we gain a set theory with two types of individuals, both of which have a notion of 'container', which is unusual as urelements are usually structureless. The primitive ordered pairs of ZFP are unlike those typical of set theory, as they are free from any notion of representation.

Showing ZFP Consistent. In section 4, we construct a transfinite hierarchy to be the domain of a model for ZFP and we define relations on this domain to be interpretations for $\in$, π_1, and π_2. We show that the resulting structure satisfies the axioms of ZFP, i.e., it is a model for ZFP. As a result, we show ZFP is consistent if ZF is.

5.2 Future Work

Model Theoretic Status of ZF and ZFP. Axiomatisations of both ZF and ZFP are given within this paper, and we are aware that the sets of ZFP behave in a similar fashion to those in ZF. We suggest employing model-theoretic techniques to give a more detailed formal account of the relationship between the formulas of both theories, as well as the models.

Implementing ZFP. Preliminary experiments have taken place in implementing ZFP as an object logic for Isabelle. Further work on this will allow comparing mathematics formalised in ZF and in ZFP, and thus allow comparing the expressivity, and automatability of both theories. Moreover, there is already a large library of mathematics formalised in Isabelle/ZF. Once the formal relationship between ZF and ZFP has been established, we will attempt to translate mathematics between both bases.

Towards Abstract Data Types in Set Theory. In this paper we identified a role performed by some sets in ZF, namely the role of being an ordered pair for some representation (e.g., Kuratowski), together with the FOL abbreviations for their relations. We axiomatised a new set theory in which this role can be performed by non-set objects, yet maintain the same existence conditions and abstract behaviour of this role. We will attempt to abstract and adapt this method, to yield set theories in which the members of mathematical structures can be genuine non-sets dedicated to their role. We believe such a framework could be helpful when using set theory to formalise mathematics.

References

[1] Peter Aczel. "Generalised set theory". In: *Logic, Language and Computation*. Vol. 1. CSLI Lecture Notes 58. 1996, pp. 1–17.

[2] Jon Barwise. *Admissible Sets and Structures*. Originally published by Springer in 1975. Cambridge University Press, 2017.

[3] Heinz-Dieter Ebbinghaus and Volker Peckhaus. *Ernst Zermelo*. Springer, 2007.

[4] Herbert B Enderton. *A mathematical introduction to logic*. 2nd ed. Harcourt/Academic Press, 2001.

[5] William M. Farmer. "Chiron: A Multi-paradigm Logic". In: *Studies in Logic, Grammar and Rhetoric* 10.23 (2007).

[6] John Harrison. *Let's make set theory great again!* `http://aitp-conference.org/2018/slides/JH.pdf`. Accessed 2020-05-27. 2018.

[7] M. Randall Holmes. "Alternative Axiomatic Set Theories". In: *The Stanford Encyclopedia of Philosophy*. Winter 2017. Stanford University, 2017.

[8] Akihiro Kanamori. "The Empty Set, the Singleton, and the Ordered Pair". In: *Bulletin of Symbolic Logic* 9.3 (2003), pp. 273–298.

[9] Ken Kubota. *Foundations of Mathematics. Genealogy and Overview.* `https://owlofminerva.net/files/fom_2018.pdf`. Accessed 2020-05-27. 2018.

[10] N. Megill and D. A. Wheeler. *Metamath: A Computer Language for Mathematical Proofs*. LULU Press, 2019. ISBN: 9780359702237.

[11] Lawrence C. Paulson. "Set theory for verification: I. From foundations to functions". In: *Journal of Automated Reasoning* 11.3 (1993), pp. 353–389.

[12] Dee Quinlan, J. B. Wells, and Fairouz Kamareddine. "BNF-Style Notation as It Is Actually Used". In: *Intelligent Computer Mathematics, CICM 2019, Proc.* Vol. 11617. LNCS. Springer, 2019, pp. 187–204.

[13] Freek Wiedijk. "Is ZF a hack?: Comparing the complexity of some (formalist interpretations of) foundational systems for mathematics". In: *Journal of Applied Logic* 4.4 (2006), pp. 622–645.