Abstract. The main objective of this paper is to show that a certain formulae-as-classes interpretation based on generalized set recursive functions provides a self-validating semantics for Constructive Zermelo-Fraenkel Set theory, $\text{CZF}$. It is argued that this interpretation plays a similar role for $\text{CZF}$ as the constructible hierarchy for classical set theory, in that it can be employed to show that $\text{CZF}$ is expandable by several forms of the axiom of choice without adding more consistency strength.

1 Introduction

The general topic of Constructive Set Theory ($\text{CST}$) originated in John Myhill’s endeavour (see [16]) to discover a simple formalism that relates to Bishop’s constructive mathematics as classical Zermelo-Fraenkel Set Theory with the axiom of choice relates to classical Cantorian mathematics. $\text{CST}$ provides a standard set theoretical framework for the development of constructive mathematics in the style of Errett Bishop [8]. One of the hallmarks of constructive set theory is that it possesses (due to Aczel [1, 2, 3]) a canonical interpretation in Martin-Löf’s intuitionistic type theory (see [12, 13]) which is considered to be the most acceptable foundational framework of ideas that make precise the constructive approach to mathematics. The interpretation employs the Curry-Howard ‘propositions as types’ idea in that the axioms of constructive set theory get interpreted as provably inhabited types.

In constructive or intuitionistic set theories often the questions arises whether adding a particular set-theoretic statement to the given set theory leads to a theory which is equiconsistent with the original theory. The most famous methods for showing equiconsistency and independence of statements from axiom systems in the classical context are Gödel’s constructible hierarchy $L$ and Cohen’s method of forcing or the closely related technique of Boolean-valued models. In classical $\text{ZF}$ the constructible hierarchy has been utilized to show that augmenting $\text{ZF}$ by the axiom of choice and the generalized continuum hypothesis gives rise to a theory that is equiconsistent with $\text{ZF}$. This leads to the question whether $L$ can be fruitfully employed in intuitionistic set theories as well. Robert Lubarsky has established the following result for intuitionistic Zermelo-Fraenkel set theory, $\text{IZF}$, in [11]:

---

*Research partly supported by NSF Grant DMS-0301162 and United Kingdom Engineering and Physical Sciences Research Council Grant GR/R 15856/01.*
Theorem 1.1. Let $V = L$ be the statement that all sets are constructible. For a formula $\varphi$, let $\varphi^L$ be the result of relativizing all quantifiers in $\varphi$ to $L$. We then have

$$IZF \vdash (V = L)^L$$
$$IZF \vdash \varphi \Rightarrow IZF \vdash \varphi^L.$$ 

While 1.1 is an interesting result, it cannot be utilized to provide similar consequences as in the classical scenario. Classically, $L$ is a very well-behaved class that is “constructed” from the ordinals and can be well-ordered via a definable well-ordering, whence AC holds in $L$. In intuitionistic set theories, however, the ordinals are rather uncontrollable sets that cannot be shown to be linearly ordered, that is, given ordinals $\alpha, \beta$ it is not possible to conclude that $\alpha \in \beta$ or $\alpha = \beta$ or $\beta \in \alpha$. As a result, when working in an intuitionistic context, we have to seek a different approach if we want to show that certain forms of the axiom of choice can be added without gaining more consistency strength. In this paper it is argued that the formulae-as-classes interpretation can do for constructive set theories what the constructible hierarchy does for classical set theory. The formulae-as-classes interpretation is closely related to Aczel’s formulae-as-types interpretation of set theory in Martin-Löf type theory. In diverges from the latter, however, in that it is an interpretation of set theory into set theory and, crucially, in that it provides a semantics for Constructive Zermelo-Fraenkel Set Theory, CZF, which can be formalized in CZF itself (a self-validating semantics).

1.1 The system CZF

In this subsection we will summarize the language and axioms for CZF. The language of CZF is the same first order language as that of classical Zermelo-Fraenkel Set Theory, ZF whose only non-logical symbol is $\in$. The logic of CZF is intuitionistic first order logic with equality. Among its non-logical axioms are Extensionality, Pairing and Union in their usual forms. CZF has additionally axiom schemata which we will now proceed to summarize.

Infinity: $\exists x \forall u [u \in x \leftrightarrow (\emptyset = u \lor \exists v \in x \ u = v + 1)]$ where $v + 1 = v \cup \{v\}$.

Set Induction: $\forall x [\forall y \in x \phi(y) \rightarrow \phi(x)] \rightarrow \forall x \phi(x)$

Restricted Separation: $\forall a \exists b \forall x [x \in b \leftrightarrow x \in a \land \phi(x)]$

for all restricted formulae $\phi$. A set-theoretic formula is restricted if it is constructed from prime formulae using $\lnot, \land, \lor, \rightarrow, \forall x \in y$ and $\exists x \in y$ only.

Strong Collection: For all formulae $\phi$,

$$\forall a [\forall x \in a \exists y \phi(x, y) \rightarrow \exists b [\forall x \in a \exists y \in b \phi(x, y) \land \forall y \in b \exists x \in a \phi(x, y)]] .$$

Subset Collection: For all formulae $\psi$,

$$\forall a \forall b \exists c \forall u [\forall x \in a \exists y \in b \psi(x, y, u) \rightarrow \exists d \in c [\forall x \in a \exists y \in d \psi(x, y, u) \land \forall y \in d \exists x \in a \psi(x, y, u)]] .$$

The Subset Collection schema easily qualifies as the most intricate axiom of CZF. To explain this axiom in different terms, we introduce the notion of fullness (cf. [1]).
Definition 1.2. As per usual, we use \((x, y)\) to denote the ordered pair of \(x\) and \(y\). We use \(\text{Fun}(g), \text{dom}(R), \text{ran}(R)\) to convey that \(g\) is a function and to denote the domain and range of any relation \(R\), respectively.

For sets \(A, B\) let \(A \times B\) be the cartesian product of \(A\) and \(B\), that is the set of ordered pairs \((x, y)\) with \(x \in A\) and \(y \in B\). Let \(\mathcal{A}B\) be the class of all functions with domain \(A\) and with range contained in \(B\). Let \(\text{mv}(\mathcal{A}B)\) be the class of all sets \(R \subseteq A \times B\) satisfying \(\forall u \in A \exists v \in B \langle u, v \rangle \in R\). A set \(C\) is said to be \textit{full in} \(\text{mv}(\mathcal{A}B)\) if \(C \subseteq \text{mv}(\mathcal{A}B)\) and

\[
\forall R \in \text{mv}(\mathcal{A}B) \exists S \in C S \subseteq R.
\]

The expression \(\text{mv}(\mathcal{A}B)\) should be read as the collection of \textit{multi-valued functions} from the set \(A\) to the set \(B\).

Additional axioms we shall consider are:

\textbf{Exponentiation: } \(\forall x\forall y \exists z \ z = x^y\).

\textbf{Fullness: } \(\forall x\forall y \exists z \ z \text{ is full in } \text{mv}(x^y)\).

The next result is an equivalent rendering of [1], 2.2. We include a proof for the reader’s convenience.

Proposition 1.3. Let \(\text{CZF}^\sim\) be \(\text{CZF}\) without Subset Collection.

(i) \(\text{CZF}^\sim \vdash \text{Subset Collection } \iff \text{Fullness}\).

(ii) \(\text{CZF} \vdash \text{Exponentiation}\).

Proof. (i): For “\(\Rightarrow\)” let \(\phi(x, y, u)\) be the formula \(y \in u \land \exists z \in B \langle y = \langle x, z \rangle \rangle\). Using the relevant instance of Subset Collection and noticing that for all \(R \in \text{mv}(\mathcal{A}B)\) we have \(\forall x \in A \exists y \in A \times B \phi(x, y, R)\), there exists a set \(C\) such that \(\forall R \in \text{mv}(\mathcal{A}B) \exists S \in C S \subseteq R\).

“\(\Leftarrow\)” Let \(C\) be full in \(\text{mv}(\mathcal{A}B)\). Assume \(\forall x \in A \exists y \in B \phi(x, y, u)\). Define \(\psi(x, w, u) := \exists y \in B \langle w = \langle x, y \rangle \land \phi(x, y, u) \rangle\). Then \(\forall x \in A \exists w \psi(x, w, u)\). Thus, by Strong Collection, there exists \(v \subseteq A \times B\) such that

\[
\forall x \in A \exists y \in B \left[\langle x, y \rangle \in v \land \phi(x, y, u)\right] \land \forall x \in A \forall y \in B \left[\langle x, y \rangle \in v \to \phi(x, y, u)\right].
\]

As \(C\) is full, we find \(w \in C\) with \(w \subseteq v\). Consequently, \(\forall x \in A \exists y \in \text{ran}(w) \phi(x, y, u)\) and \(\forall y \in \text{ran}(w) \exists x \in A \phi(x, y, u)\), where \(\text{ran}(w) := \{v : \exists z \langle z, v \rangle \in \{w\}\}\).

Whence \(D := \{\text{ran}(w) : w \in C\}\) witnesses the truth of the instance of Subset Collection pertaining to \(\phi\).

(ii) Let \(C\) be full in \(\text{mv}(\mathcal{A}B)\). If now \(f \in \mathcal{A}B\), then \(\exists R \in C R \subseteq f\). But then \(R = f\). Therefore \(\mathcal{A}B = \{f \in C : f \text{ is a function}\}\).

Let \(\text{TND}\) be the principle of excluded third, i.e. the schema consisting of all formulae of the form \(A \lor \neg A\). The first central fact to be noted about \(\text{CZF}\) is:

Proposition 1.4. \(\text{CZF} + \text{TND} = \text{ZF}\).
Proof: Note that classically Collection implies Separation. Powerset follows classically from Exponentiation.

On the other hand, it was shown in [18], Theorem 4.14, that CZF has only the strength of Kripke-Platek Set Theory (with the Infinity Axiom), KP (see [3]), and, moreover, that CZF is of the same strength as its subtheory CZF⁻, i.e., CZF minus Subset Collection. To stay in the world of CZF one has to keep away from any principles that imply TND. Moreover, it is perhaps fair to say that CZF is such an interesting theory owing to the non-derivability of Powerset and Separation. Therefore one ought to avoid any principles which imply Powerset or Separation.

The first large set axiom proposed in the context of constructive set theory was the Regular Extension Axiom, REA, which Aczel introduced to accommodate inductive definitions in CZF (cf. [3]).

Definition 1.5. A is inhabited if \( \exists x \in A \). An inhabited set A is regular if A is transitive, and for every \( a \in A \) and set \( R \subseteq a \times A \) if \( \forall x \in a \exists y (\langle x, y \rangle \in R) \), then there is a set \( b \in A \) such that
\[
\forall x \in a \exists y \in b (\langle x, y \rangle \in R) \land \forall y \in b \exists x \in a (\langle x, y \rangle \in R).^{1}
\]

2 The axiom of choice in constructive set theories

Among the axioms of set theory, the axiom of choice is distinguished by the fact that is it the only one that one finds ever mentioned in workaday mathematics. In the mathematical world of the beginning of the 20th century, discussions about the status of the axiom of choice were important. In 1904 Zermelo proved that every set can be well-ordered by employing the axiom of choice. While Zermelo argued that it was self-evident, it was also criticized as an excessively non-constructive principle by some of the most distinguished analysts of the day, notably Borel, Baire, and Lebesgue. At first blush this reaction against the axiom of choice utilized in Cantor’s new theory of sets is surprising as the French analysts had used and continued to use choice principles routinely in their work. However, in the context of 19th century classical analysis only the Axiom of Dependent Choices, DC, is invoked and considered to be natural, while the full axiom of choice is unnecessary and even has some counterintuitive consequences.

Unsurprisingly, the axiom of choice does not have a unambiguous status in constructive mathematics either. On the one hand it is said to be an immediate consequence of the constructive interpretation of the quantifiers. Any proof of \( \forall x \in A \exists y \in B \phi(x, y) \) must yield a function \( f : A \rightarrow B \) such that \( \forall x \in A \phi(x, f(x)) \). This is certainly the case in Martin-Löf’s intuitionistic theory of types. On the other hand, it has been observed that the full axiom of choice cannot be added to systems of extensional constructive set theory without yielding constructively unacceptable cases of excluded middle (see [9]). In extensional intuitionistic set theories, a proof of a statement \( \forall x \in A \exists y \in B \phi(x, y) \), in general, provides only a function \( F \), which when fed a proof \( p \) witnessing \( x \in A \), yields \( F(p) \in B \) and \( \phi(x, F(p)) \). Therefore, in the main, such an \( F \) cannot be rendered a function of \( x \) alone. Choice will then hold over sets which have a canonical proof function, where a constructive function \( h \) is a canonical proof

---

1In particular, if \( R : a \rightarrow A \) is a function, then the image of \( R \) is an element of \( A \).
function for $A$ if for each $x \in A$, $h(x)$ is a constructive proof that $x \in A$. Such sets having natural canonical proof functions “built-in” have been called bases (cf. [21], p. 841).

The particular form of constructivism adhered to in this paper is Martin-Löf’s intuitionistic type theory (cf. [12, 13]). Set-theoretic choice principles will be considered as constructively justified if they can be shown to hold in the interpretation in type theory. Moreover, looking at set theory from a type-theoretic point of view has turned out to be valuable heuristic tool for finding new constructive choice principles. For more information on choice principles in the constructive context see [19].

2.1 Old acquaintances

In many a text on constructive mathematics, axioms of countable choice and dependent choices are accepted as constructive principles. This is, for instance, the case in Bishop’s constructive mathematics (cf. [8] as well as Brouwer’s intuitionistic analysis (cf. [21], Ch. 4, Sect. 2). Myhill also incorporated these axioms in his constructive set theory [16].

The weakest constructive choice principle we shall consider is the Axiom of Countable Choice, $\text{AC}_\omega$, i.e. whenever $F$ is a function with domain $\omega$ such that $\forall i \in \omega \exists y \in F(i)$, then there exists a function $f$ with domain $\omega$ such that $\forall i \in \omega f(i) \in F(i)$.

A mathematically very useful axiom to have in set theory is the Dependent Choices Axiom, $\text{DC}$, i.e., for all formulae $\psi$, whenever

$$\forall x \in a \left( \exists y \in a \psi(x, y) \right)$$

and $b_0 \in a$, then there exists a function $f : \omega \to a$ such that $f(0) = b_0$ and

$$\forall n \in \omega \psi(f(n), f(n + 1)).$$

Even more useful is the Relativized Dependent Choices Axiom, $\text{RDC}$. It asserts that for arbitrary formulae $\phi$ and $\psi$, whenever

$$\forall x \left[ \phi(x) \to \exists y (\phi(y) \land \psi(x, y)) \right]$$

and $\phi(b_0)$, then there exists a function $f$ with domain $\omega$ such that $f(0) = b_0$ and

$$\forall n \in \omega \left[ \phi(f(n)) \land \psi(f(n), f(n + 1)) \right].$$

2.2 Operations on classes

The interpretation of constructive set theory in type theory not only validates all the theorems of $\text{CZF}$ (resp. $\text{CZF} + \text{REA}$) but many other interesting set-theoretic statements, including several new choice principles which will be described next. To state these principles we need to introduce various operations on classes.

**Remark 2.1. Class notation:** In doing mathematics in $\text{CZF}$ we shall exploit the use of class notation and terminology, just as in classical set theory. Given a formula $\phi(x)$ there may not exist a set of the form $\{x : \phi(x)\}$. But there is nothing wrong with thinking about such collection. So, if $\phi(x)$ is a formula in the language of set theory we may form a class $\{x : \phi(x)\}$. We allow $\phi(x)$ to have free variables other than $x$, which are considered parameters.
upon which the class depends. Informally, we call any collection of the form \( \{ x : \phi(x) \} \) a *class*. However formally, classes do not exist, and expressions involving them must be thought of as abbreviations for expressions not involving them. Classes \( A, B \) are defined to be equal if \( \forall x [ x \in A \iff x \in B ] \).

We may also consider an augmentation of the language of set theory whereby we allow atomic formulas of the form \( y \in A \) and \( A = B \) with \( A, B \) being classes. There is no harm in taking such liberties as any such formula can be translated back into the official language of set theory by re-writing \( y \in A \) and \( \{ x : \phi(x) \} = \{ y : \psi(y) \} \) as \( \phi(y) \) and \( \forall z [ \phi(z) \iff \psi(z) ] \), respectively (with \( z \) not in \( \phi(x) \) and \( \psi(y) \)).

**Definition 2.2.** Let \( \text{CZF}_{\text{Exp}} \) denote the modification of \( \text{CZF} \) with Exponentiation in place of Subset Collection.

**Remark 2.3.** In all the results of this paper, \( \text{CZF} \) could be replaced by \( \text{CZF}_{\text{Exp}} \), that is to say, for the purposes of this paper it is enough to assume Exponentiation rather than Subset Collection.

**Definition 2.4.** (\( \text{CZF} \))

If \( A \) is a set and \( B_x \) are classes for all \( x \in A \), we define a class \( \prod_{x \in A} B_x \) by:

\[
\prod_{x \in A} B_x := \{ f : A \to \bigcup_{x \in A} B_x \mid \forall x \in A (f(x) \in B_x) \}. \tag{1}
\]

If \( A \) is a class and \( B_x \) are classes for all \( x \in A \), we define a class \( \sum_{x \in A} B_x \) by:

\[
\sum_{x \in A} B_x := \{ \langle x, y \rangle \mid x \in A \land y \in B_x \}. \tag{2}
\]

If \( A \) is a class and \( a, b \) are sets, we define a class \( I(A, a, b) \) by:

\[
I(A, a, b) := \{ z \in 1 \mid a = b \land a, b \in A \}. \tag{3}
\]

If \( A \) is a class and for each \( a \in A \), \( B_a \) is a set, then

\[
W_{a \in A} B_a
\]

is the smallest class \( Y \) such that whenever \( a \in A \) and \( f : B_a \to Y \), then \( \langle a, f \rangle \in Y \).

**Lemma 2.5.** (\( \text{CZF} \))

If \( A, B, a, b \) are sets and \( B_x \) are sets for all \( x \in A \), then \( \prod_{x \in A} B_x, \sum_{x \in A} B_x \) and \( I(A, a, b) \) are sets.

**Proof.** First of all, we need to prove that \( \bigcup_{x \in A} B_x \) is a set. Indeed, \( g = \{ \{ x, \{ x, B_x \} \} \mid x \in A \} \), and so \( \bigcup \bigcup g = \{ z, x, B_x \mid z \in x, x \in A \} \) is a set by Union. Now

\[
\text{ran}(g) = \{ y \in \bigcup \bigcup g \mid \exists x \in \bigcup \bigcup g (\langle x, y \rangle \in g) \}
\]

and \( \bigcup_{x \in A} B_x = \bigcup \text{ran}(g) \) are sets by Bounded Separation and Union.

1. The class of all functions from \( A \) to \( \bigcup_{x \in A} B_x \) is a set by Exponentiation and

\[
\prod_{x \in A} B_x := \{ f : A \to \bigcup_{x \in A} B_x \mid \forall x \in A (f(x) \in B_x) \} \]
is a set by Bounded Separation, since $\forall x \in A \exists y \in \text{ran}(f) \exists y' \in \text{ran}(g)(\langle x, y \rangle \in f \land \langle x, y' \rangle \in g \land y \in y')$.

2: Using from above that $\bigcup_{x \in A} B_x$ is a set, by Pairing, Union and Replacement we obtain a set

$$A \times \bigcup_{x \in A} B_x = \{ \langle x, y \rangle \mid x \in A \land y \in \bigcup_{x \in A} B_x \}.$$  

Now, the set

$$\sum_{x \in A} B_x := \{ \langle x, y \rangle \in A \times \bigcup_{x \in A} B_x \mid x \in A \land y \in B_x \}$$

exists by Bounded Separation, since $x \in A \land y \in B_x$ can be rewritten as

$$x \in A \land \exists y' \in \text{ran}(g)(\langle x, y' \rangle \in g \land y \in y').$$

3: $\text{I}(A, a, b)$ is a set by Bounded Separation. $\square$

**Lemma 2.6.** $\text{(CZF + REA)}$

If $A$ is a set and $B_x$ is a set for all $x \in A$, then $\mathbf{W}_{a \in A} B_a$ is a set.

**Proof.** This follows from [3], Corollary 5.3. $\square$

### 2.3 Inductively defined classes

In the following we shall introduce several inductively defined classes, and, moreover, we have to ensure that such classes can be formalized in CZF.

We define an *inductive definition* to be a class of ordered pairs. If $\Phi$ is an inductive definition and $\langle x, a \rangle \in \Phi$ then we write

$$\frac{x}{a} \Phi$$

and call $\frac{x}{a}$ an (inference) step of $\Phi$, with set $x$ of *premisses* and conclusion $a$. For any class $Y$, let

$$\Gamma_\Phi(Y) = \{ a \mid \exists x (x \subseteq Y \land \frac{x}{a} \Phi) \}.$$

The class $Y$ is $\Phi$-closed if $\Gamma_\Phi(Y) \subseteq Y$. Note that $\Gamma$ is monotone; i.e. for classes $Y_1, Y_2$, whenever $Y_1 \subseteq Y_2$, then $\Gamma(Y_1) \subseteq \Gamma(Y_2)$.

We define the class *inductively defined* by $\Phi$ to be the smallest $\Phi$-closed class. The main result about inductively defined classes states that this class, denoted $\text{I}(\Phi)$, always exists.

**Lemma 2.7.** $\text{(CZF) (Class Inductive Definition Theorem)}$ For any inductive definition $\Phi$ there is a smallest $\Phi$-closed class $\text{I}(\Phi)$.

Moreover, call a set $G$ of ordered pairs good if

$$(*) \quad \langle a, y \rangle \in G \Rightarrow y \in \Gamma_\Phi(G^{\in a}).$$

where

$$G^{\in a} = \{ y' \mid \exists x \in a \langle x, y' \rangle \in G \}.$$
Letting \( J = \bigcup \{ G \mid G \text{ is good} \} \) and \( J^a = \{ x \mid \langle a, x \rangle \in J \} \), it holds
\[
I(\Phi) = \bigcup_a J^a,
\]
and for each \( a \),
\[
J^a = \Gamma_{\Phi}(\bigcup_{x \in a} J^x).
\]
\( J \) is uniquely determined by the above, and its stages \( J^a \) will be denoted by \( \Gamma^a_{\Phi} \).

**Proof.** [2], section 4.2 or [4], Theorem 5.1. \( \square \)

### 2.4 Maximal choice principles

**Lemma 2.8.** (CZF)

There exists a smallest \( \Pi_1 \Sigma_1 \)-closed class, i.e. a smallest class \( Y \) such that the following holds:

(i) \( n \in Y \) for all \( n \in \mathbb{N} \);
(ii) \( \omega \in Y \);
(iii) \( \prod_{x \in A} B_x \in Y \) and \( \sum_{x \in A} B_x \in Y \) whenever \( A \in Y \) and \( B_x \in Y \) for all \( x \in A \).

Likewise, there exists a smallest \( \Pi_1 \Sigma_1 \text{-closed class, i.e. a smallest class } Y^* \), which, in addition to the closure conditions (i)–(iii) above, satisfies:

(iv) \( I(A, a, b) \in Y^* \) whenever \( A \in Y^* \) and \( a, b \in A \).

**Proof.** The classes \( Y \) and \( Y^* \) are inductively defined, and therefore exist by Lemma 2.7. To be precise, the respective inductive definitions of these classes are given by the classes \( \Phi_1, \ldots, \Phi_5 \) consisting of the following pairs:

(i) \( n \Phi_1 \), for all \( n \in \mathbb{N} \);
(ii) \( \omega \Phi_2 \);
(iii) \( \{ \text{dom}(g) \} \cup \text{ran}(g) \Phi_3 \), for all functions \( g \) with \( \text{dom}(g) = A \);
(iv) \( \{ \text{dom}(g) \} \cup \text{ran}(g) \Phi_4 \), for all functions \( g \) with \( \text{dom}(g) = A \);
(v) \( \{ A \} \Phi_5 \), if \( a, b \in A \).

(Clause (v) is only needed to define \( Y^* \).) \( \square \)

**Lemma 2.9.** (CZF + REA)

There exists a least \( \Pi_1 \Sigma_1 \text{-closed class, i.e. a smallest class } Y_w \) that in addition to the clauses (i),(ii),(iii) of Lemma 2.8 satisfies:

(vi) \( W_{a \in A} B_a \in Y_w \) whenever \( A \in Y_w \) and \( B_x \in Y_w \) for all \( x \in A \).

Likewise, there exists a smallest \( \Pi_1 \Sigma_1 \text{-closed class, i.e. a least class } Y^*_w \), which, in addition to the closure conditions above, satisfies clause (iv) of Lemma 2.8.
The Formulae-as-Classes Interpretation of Constructive Set Theory

Proof. Virtually the same as for Lemma 2.8. □

Definition 2.10. The \( \Pi \Sigma \)-generated sets are the sets in the smallest \( \Pi \Sigma \)-closed class, i.e. \( Y \). Similarly one defines the \( \Pi \Sigma I \), \( \Pi \Sigma W \) and \( \Pi \Sigma WI \)-generated sets.

A set \( P \) is a base if for any \( P \)-indexed family \( (X_a)_{a \in P} \) of inhabited sets \( X_a \), there exists a function \( f \) with domain \( P \) such that, for all \( a \in P \), \( f(a) \in X_a \).

\( \Pi \Sigma - AC \) is the statement that every \( \Pi \Sigma \)-generated set is a base. Similarly one defines the axioms \( \Pi \Sigma I - AC \), \( \Pi \Sigma W I - AC \), and \( \Pi \Sigma W - AC \).

Lemma 2.11. (i) \( \text{(CZF)} \) For every \( A \in Y^* \) there exists a \( B \in Y \) with a bijection \( h: B \rightarrow A \).

(ii) \( \text{(CZF + REA)} \) For every \( A \in Y_w^* \) there exists a \( B \in Y_w \) with a bijection \( h: B \rightarrow A \).

Proof. See the lemma following Theorem 3.7 in [3]. □

Corollary 2.12. (i) \( \text{(CZF)} \) \( \Pi \Sigma - AC \) and \( \Pi \Sigma I - AC \) are equivalent.

(ii) \( \text{(CZF + REA)} \) \( \Pi \Sigma W - AC \) and \( \Pi \Sigma WI - AC \) are equivalent.

Proof. \( \Pi \Sigma I - AC \) obviously implies \( \Pi \Sigma - AC \), since \( Y \subseteq Y^* \). To prove the converse, assume \( \Pi \Sigma - AC \), \( A \in Y^* \), and \( \forall x \in A \exists y \varphi(x, y) \), where \( \varphi \) is a formula of \( \text{CZF} \). Take a \( B \) and a bijection \( h: A \rightarrow B \) which exists by the previous Lemma; then \( \forall x \in B \exists y \varphi(h^{-1}(x), y) \).

By \( \Pi \Sigma - AC \),

\[ \exists f: B \rightarrow V \forall x \in B \varphi(h^{-1}(x), f(x)). \]

This yields

\[ \forall x \in A \varphi(h^{-1} \circ h(x), f \circ h(x)) \]

so that \( \forall x \in A \varphi(x, f \circ h(x)) \).

The proof of (ii) is similar. □

3 Interpreting bounded formulae as sets

Notation. For sets \( x \) and \( y \), we define \( \sup(x, y) \) as \( (x, y) \). If \( \alpha = \sup(A, f) \), where \( f \) is a function with domain \( A \), we define \( \bar{\alpha} := A \) and \( \tilde{\alpha} := f \).

Definition 3.1. \( \text{(CZF)} \)

Utilizing Lemma 2.7 we define a class \( V(Y^*) \) by the following rule:

\[
\frac{a \in Y^*}{\sup(a, f) \in V(Y^*)},
\]

(4)

The class \( V(Y) \) is defined in the same vein by replacing \( Y^* \) by \( Y \) in the foregoing clause.

Definition 3.2. \( \text{(CZF)} \)

The (class) functions \( \dot{=} : V(Y^*) \times V(Y^*) \rightarrow Y^* \) and \( \ddot{e} : V(Y^*) \times V(Y^*) \rightarrow Y^* \) are defined by recursion as follows:

\[
\dot{=} (\alpha, \beta) = \prod_{x \in a} \sum_{y \in \beta} \dot{=} (\tilde{\alpha}(x), \tilde{\beta}(y)) \times \prod_{y \in \beta} \sum_{x \in \tilde{\alpha}} \dot{=} (\tilde{\alpha}(x), \tilde{\beta}(y)),
\]

(5)

\[
\ddot{e} (\alpha, \beta) = \sum_{y \in \beta} \dot{=} (\alpha, \tilde{\beta}(y)).
\]

(6)
**Definition 3.3.** (CZF + REA)
In the same vein as in Definitions 3.1 and 3.2 we define a class functions \( \mathcal{V}(Y^*_w) \times \mathcal{V}(Y^*_w) \to Y^*_w \) and \( \dot{\varepsilon} : \mathcal{V}(Y^*_w) \times \mathcal{V}(Y^*_w) \to Y^*_w \) by replacing \( Y^* \) with \( Y^*_w \).

**Convention.** We will write \( \alpha \mathrel{\dot{\varepsilon}} \beta \) and \( \alpha \mathrel{\dot{\varepsilon}} \beta \) for \( (\alpha, \beta) \) and \( (\alpha, \beta) \), respectively.

**Definition 3.4.** (CZF)
For any set \( A \) and class \( B \) we define:
\[
A \to B = \prod_{x \in A} B. \tag{7}
\]
For any classes \( A \) and \( B \) we define:
\[
A \times B = \sum_{x \in A} B, \quad A + B = \sum_{x \in 2} C_x, \quad \text{where } C_0 = A \text{ and } C_1 = B. \tag{8}
\]

**Definition 3.5.** A \( \mathcal{V}(Y^*) \)-assignment is a mapping \( M : \text{Var} \to \mathcal{V}(Y^*) \), where \( \text{Var} \) is the set of variables of the language. \( M(a) \) will also be denoted by \( a_M \).

**Definition 3.6.** (CZF)
To any bounded formula \( \theta \in L \) and \( \mathcal{V}(Y^*) \)-assignment \( M \) we shall assign a set \( \parallel \theta \parallel_M \). Since we have already used the symbol “\( \to \)” for function spaces we shall denote the conditional by “\( \supset \)” and the bi-conditional by “\( \text{iff} \)”.

The recursive definition of \( \parallel \theta \parallel_M \) is given in the table below:

<table>
<thead>
<tr>
<th>( \theta \in L )</th>
<th>( \parallel \theta \parallel_M )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \perp )</td>
<td>0</td>
</tr>
<tr>
<td>( a = b )</td>
<td>( a_M \mathrel{\dot{\varepsilon}} b_M )</td>
</tr>
<tr>
<td>( a \in b )</td>
<td>( a_M \mathrel{\dot{\varepsilon}} b_M )</td>
</tr>
<tr>
<td>( \theta_0 \land \theta_1 )</td>
<td>( \parallel \theta_0 \parallel_M \times \parallel \theta_1 \parallel_M )</td>
</tr>
<tr>
<td>( \theta_0 \lor \theta_1 )</td>
<td>( \parallel \theta_0 \parallel_M + \parallel \theta_1 \parallel_M )</td>
</tr>
<tr>
<td>( \theta_0 \supset \theta_1 )</td>
<td>( \parallel \theta_0 \parallel_M \to \parallel \theta_1 \parallel_M )</td>
</tr>
<tr>
<td>( \forall v \in a \psi )</td>
<td>( \prod_{x \in a_M} \parallel \psi \parallel_M(v\mathrel{\dot{\varepsilon}}a_M(x)) )</td>
</tr>
<tr>
<td>( \exists v \in a \psi )</td>
<td>( \sum_{x \in a_M} \parallel \psi \parallel_M(v\mathrel{\dot{\varepsilon}}a_M(x)) )</td>
</tr>
</tbody>
</table>

**Lemma 3.7.** (CZF)
For every bounded \( \theta \in L \) and \( \mathcal{V}(Y^*) \)-assignment \( M \), \( \parallel \theta \parallel_M \in Y^* \).

**Proof.** This is proved by induction on \( \theta \) using Lemma 2.8 and Definitions 3.2 and 3.4. \( \square \)
Lemma 3.8. (CZF + REA)

A $V(Y_w^*)$-assignment is defined similarly as a $V(Y^*)$-assignment in Definition 3.5. Likewise, as in Definition 3.6, to any bounded formula $\theta \in \mathcal{L}_c$ and $V(Y_w^*)$-assignment $\mathcal{M}$ we assign a set $\llbracket \theta \rrbracket_M$. We then have, for every bounded $\theta \in \mathcal{L}_c$ and $V(Y_w^*)$-assignment $\mathcal{M}$, $\llbracket \theta \rrbracket_M \in Y_w^*$.

Proof. This is proved as Lemma 3.7. \hfill $\square$

4 The formulae-as-classes interpretation for arbitrary formulae

In order to reflect within CZF the formulae-as-classes interpretation for arbitrary set-theoretic formulae, we would need to represent large types $\Pi\Sigma$-generated on top of $V(Y^*)$. The language of CZF is not rich enough to do it in a straightforward way. To remedy this we introduce a notion of set recursive partial functions.

4.1 Extended $E$-recursive functions

We would like to have unlimited application of sets to sets, i.e. we would like to assign a meaning to the symbol $\{a\}(x)$ where $a$ and $x$ are sets. In generalized recursion theory this is known as $E$-recursion or set recursion (see, e.g., [17] or [20, Ch.X]). However, we shall introduce an extended notion of $E$-computability, christened $E_\nu$-computability, rendering the function $\exp(a, b) = ^a b$ is computable as well, (where $^ab$ denotes the set of all functions from $a$ to $b$). Moreover, the constant function with value $\omega$ is taken as an initial function in $E_\nu$-computability. From a classical standpoint, $E_\nu$-computability is related to power recursion, where the power set operation is regarded to be an initial function. The latter notion has been studied by Moschovakis [14] and Moss [15].

There is a lot of leeway in setting up $E_\nu$-recursion. The particular schemes we use are especially germane to our situation. Our construction will provide a specific set-theoretic model for the elementary theory of operations and numbers EON (see, e.g., [7, VI.2], or the theory APP as described in [21, Ch.9, Sect.3]). We utilize encoding of finite sequences of sets by the pairing function $\langle, \rangle$.

Definition 4.1. (CZF)

First, we select distinct non-zero natural numbers $k$, $s$, $p$, $p_0$, $p_1$, $s_N$, $p_N$, $d_N$, $\bar{0}$, $\bar{\omega}$, $\pi$, $\sigma$, $\bar{pl}$, $i$, $\bar{Fa}$, and $\bar{AB}$ which will provide indices for special $E_\nu$-recursive partial (class) functions. Inductively we shall define a class $\mathbb{E}$ of triples $\langle e, x, y \rangle$. Rather than "$\langle e, x, y \rangle \in \mathbb{E}$", we shall write "$\{e\}(x) \supseteq y$", and moreover, if $n > 0$, we shall use $\{e\}(x_1, \ldots, x_n) \supseteq y$ to convey that

$$\{e\}(x_1) \supseteq \langle e, x_1 \rangle \land \{e, x_1\}(x_2) \supseteq \langle e, x_1, x_2 \rangle \land \cdots \land \{e, x_1, \ldots, x_{n-1}\}(x_n) \supseteq y.$$  

We shall say that $\{e\}(x)$ is defined, written $\{e\}(x) \downarrow$, if $\{e\}(x) \supseteq y$ for some $y$. Let $\mathbb{N} := \omega$. 
$\mathbb{E}$ is defined by the following clauses (inference steps):

\[
\begin{align*}
\{k\}(x, y) & \vdash x \\
\{s\}(x, y, z) & \vdash \{\{x\}(z)\}\{\{y\}(z)\} \\
\{p\}(x, y) & \vdash \langle x, y \rangle \\
\{p_0\}(x) & \vdash \langle x \rangle_0 \\
\{p_1\}(x) & \vdash \langle x \rangle_1 \\
\{s_N\}(n) & \vdash n + 1 \text{ if } n \in \mathbb{N} \\
\{p_N\}(0) & \vdash 0 \\
\{p_N\}(n + 1) & \vdash n \text{ if } n \in \mathbb{N} \\
\{d_N\}(n, m, x, y) & \vdash x \text{ if } n, m \in \mathbb{N} \text{ and } n = m \\
\{d_N\}(n, m, x, y) & \vdash y \text{ if } n, m \in \mathbb{N} \text{ and } n \neq m \\
\{0\}(x) & \vdash 0 \\
\{\omega\}(x) & \vdash \omega \\
\{\pi\}(x, y) & \vdash \prod_{z \in x} g(z) \text{ if } g \text{ is a (set-)function with } \text{dom}(g) = x \\
\{\sigma\}(x, y) & \vdash \sum_{z \in x} g(z) \text{ if } g \text{ is a (set-)function with } \text{dom}(g) = x \\
\{p_l\}(x, y) & \vdash x + y \\
\{i\}(x, y, z) & \vdash I(x, y, z) \\
\{f_\alpha\}(g, x) & \vdash g(x) \text{ if } g \text{ is a (set-)function and } x \in \text{dom}(g) \\
\{a\bar{B}\}(e, a) & \vdash h \text{ if } h \text{ is a (set-)function with } \text{dom}(h) = a \\
& \quad \text{ and } \forall x \in a \{e\}(x) \supseteq h(x). 
\end{align*}
\]

Note that for $\{s\}(x, y, z)$ to be defined it is required that $\{x\}(z), \{y\}(z)$ and $\{\{x\}(z)\}\{\{y\}(z)\}$ be defined. The clause for $s$ is thus to be read as a conjunction of the following clauses: $\{s\}(x) \supseteq \langle s, x \rangle$, $\{\{s\}(x)\}(y) \supseteq \langle s, x, y \rangle$ and, if there exist $a, b, c$ such that $\{x\}(z) \supseteq a$, $\{y\}(z) \supseteq b$, $\{a\}(b) \supseteq c$, then $\{\{s\}(x)\}(z) \supseteq c$.

The constants $f_\alpha$ and $a\bar{B}$ stand for function application and function abstraction, respectively.

** Lemma 4.2. (CZF)**

$\mathbb{E}$ is an inductively defined class and $\mathbb{E}$ is functional in that for all $e, x, y, y'$,

\[\langle e, x, y \rangle \in \mathbb{E} \wedge \langle e, x, y' \rangle \in \mathbb{E} \implies y = y'.\]

**Proof.** The inductive definition of $\mathbb{E}$ falls under the heading of Lemma 2.7. If $\{e\}(x) \supseteq y$ the uniqueness of $y$ follows by induction on the stages (see Lemma 2.7) of that inductive definition. $\square$

**Definition 4.3.** Application terms are defined inductively as follows:

(i) The constants $k, s, p_0, p_1, s_N, p_N, d_N, 0, \omega, \pi, \sigma, p_l, i, f_\alpha, a\bar{B}$ singled out in Definition 4.1 are application terms;

(ii) variables are application terms;

(iii) if $s$ and $t$ are application terms then $(st)$ is an application term.

**Definition 4.4.** Application terms are easily formalized in CZF. However, rather than translating application terms into the set-theoretic language of CZF, we define the translation of expressions of the form $t \simeq u$, where $t$ is an application term and $u$ is a variable. The translation proceeds along the way that $t$ was built up:

\[
\begin{align*}
[c \simeq u] & \text{ is } c = u \text{ if } c \text{ is a constant or a variable; } \\
[(st) \simeq u] & \text{ is } \exists x \exists y([s \simeq x] \land [t \simeq y] \land \langle x, y, u \rangle \in \mathbb{E}).
\end{align*}
\]
Abbreviations. For application terms $s, t, t_1, \ldots, t_n$ we will use:

- $s(t_1, \ldots, t_n)$ as a shortcut for $(\ldots((s(t_1))\ldots)t_n)$; (parentheses associated to the left);
- $st_1 \ldots t_n$ as a shortcut for $s(t_1, \ldots, t_n)$;
- $\downarrow$ as a shortcut for $\exists x(t \simeq x)^\wedge$; (t is defined);
- $(s \simeq t)^\wedge$ as a shortcut for $s \downarrow \lor t \downarrow \lor \exists x((s \simeq x)^\wedge(t \simeq x)^\wedge)$;
- $\{x\}(y) = z$ as a shortcut for $\langle x, y, z \rangle \in E$.

A closed application term is an application term that does not contain variables. If $t$ is a closed application term and $a_1, \ldots, a_n, b$ are sets we use the abbreviation

$$t(a_1, \ldots, a_n) \simeq b \quad \forall x_1 \ldots x_n \exists y \left( x_1 = a_1 \land \ldots \land x_n = a_n \land y = b \land [t(x_1, \ldots, x_n) \simeq y]^\wedge \right).$$

**Definition 4.5.** Every closed application term gives rise to a partial class function. A partial $n$-place (class) function $\Upsilon$ is said to be an $E_\psi$-recursive partial function if there exists a closed application term $t_\Upsilon$ such that

$$\text{dom}(\Upsilon) = \{(a_1, \ldots, a_n) \mid t_\Upsilon(a_1, \ldots, a_n) \downarrow\}$$

and for all for all sets $(a_1, \ldots, a_n) \in \text{dom}(\Upsilon)$,

$$t_\Upsilon(a_1, \ldots, a_n) \simeq \Upsilon(a_1, \ldots, a_n).$$

In the latter case, $t_\Upsilon$ is said to be an index for $\Upsilon$.

If $\Upsilon_1, \Upsilon_2$ are $E_\psi$-recursive partial functions, then $\Upsilon_1(\vec{a}) \simeq \Upsilon_2(\vec{a})$ iff neither $\Upsilon_1(\vec{a})$ nor $\Upsilon_2(\vec{a})$ are defined, or $\Upsilon_1(\vec{a})$ and $\Upsilon_2(\vec{a})$ are defined and equal.

The next two results can be proved in the theory $\text{APP}$ and thus hold true in any applicative structure. Hence the particular applicative structure considered here satisfies the Abstraction Lemma andRecursion Theorem (see e.g. [10] or [7]).

**Lemma 4.6.** (Abstraction Lemma, cf. [7, VI.2.2])

For every application term $t[x]$ there exists an application term $\lambda x.t[x]$ with $\text{FV}(\lambda x.t[x]) := \{x_1, \ldots, x_n\} \subseteq \text{FV}(t[x]) \setminus \{x\}$ such that the following holds:

$$\forall x_1 \ldots \forall x_n.(\lambda x.t[x] \downarrow \land \forall y.(\lambda x.t[x])y \simeq t[y]).$$

**Proof.** (i) $\lambda x.x$ is $\text{skk}$; (ii) $\lambda x.t$ is $kt$ for $t$ a constant or a variable other than $x$; (iii) $\lambda x.uv$ is $(s(\lambda x.u))(\lambda x.v)$. \hfill $\Box$

**Lemma 4.7.** (Recursion Theorem, cf. [7, VI.2.7])

There exists a closed application term $\text{rec}$ such that for any $f, x$,

$$\text{rec}f \downarrow \land \text{rec}fx \simeq f(\text{rec}f)x.$$

**Proof.** Take $\text{rec}$ to be $\lambda f.tt$, where $t$ is $\lambda y.\lambda x.f(yy)x$. \hfill $\Box$

**Corollary 4.8.** For any $E_\psi$-recursive partial function $\Upsilon$ there exists a closed application term $\tau_{fix}$ such that $\tau_{fix} \downarrow$ and for all $\vec{a}$,

$$\Upsilon(\vec{e}, \vec{a}) \simeq \tau_{fix}(\vec{a}),$$

where $\tau_{fix} \simeq \vec{e}$. Moreover, $\tau_{fix}$ can be effectively (e.g. primitive recursively) constructed from an index for $\Upsilon$. \hfill $\Box$
4.2 Arbitrary formulae

**Definition 4.9.** (CZF)
If $B$ is a class and $a$, $x$ are sets, we write $\{a\}(x) \in B$ with the following meaning:
\[
\{a\}(x) \in B :\iff \exists y (\{a\}(x) = y \land y \in B).
\]
(9)

If $A$ is a class and $B_x$ are classes for all $x \in A$, then we define a class $\prod_{x \in A} B_x$ in the following way:
\[
\prod_{x \in A} B_x := \{a | \forall x \in A(\{a\}(x) \in B_x)\}.
\]
(10)

For any classes $A$ and $B$ we define a class $A \cong B$ by
\[
A \cong B := \{a | \forall x \in A(\{a\}(x) \in B)\} = \prod_{x \in A} B.
\]
(11)

**Definition 4.10.** (CZF)
For every formula $\theta \in \mathcal{L}_e$ and $\mathbf{V}(\mathbf{Y}^*)$-assignment $\mathcal{M}$, we define a class $[\theta]_\mathcal{M}$. The definition is given by the table below:

<table>
<thead>
<tr>
<th>$\theta \in \mathcal{L}_e$</th>
<th>$[\theta]_\mathcal{M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bot$</td>
<td>$0$</td>
</tr>
<tr>
<td>$a = b$</td>
<td>$a_\mathcal{M} \doteq b_\mathcal{M}$</td>
</tr>
<tr>
<td>$a \in b$</td>
<td>$a_\mathcal{M} \in b_\mathcal{M}$</td>
</tr>
<tr>
<td>$\theta_0 \land \theta_1$</td>
<td>$[\theta_0]<em>\mathcal{M} \times [\theta_1]</em>\mathcal{M}$</td>
</tr>
<tr>
<td>$\theta_0 \lor \theta_1$</td>
<td>$[\theta_0]<em>\mathcal{M} + [\theta_1]</em>\mathcal{M}$</td>
</tr>
<tr>
<td>$\theta_0 \supset \theta_1$</td>
<td>$[\theta_0]<em>\mathcal{M} \rightarrow [\theta_1]</em>\mathcal{M}$ if $\theta_0$ is bounded</td>
</tr>
<tr>
<td>$\theta_0 \supset \theta_1$</td>
<td>$[\theta_0]<em>\mathcal{M} \cong [\theta_1]</em>\mathcal{M}$ if $\theta_0$ is not bounded</td>
</tr>
<tr>
<td>$\forall v \in a \psi$</td>
<td>$\prod_{x \in aV} [\psi]_\mathcal{M}(v</td>
</tr>
<tr>
<td>$\exists v \in a \psi$</td>
<td>$\sum_{x \in aV} [\psi]_\mathcal{M}(v</td>
</tr>
<tr>
<td>$\forall v \psi$</td>
<td>$\prod_{x \in \mathbf{V}(\mathbf{Y}^*)} [\psi]_\mathcal{M}(v</td>
</tr>
<tr>
<td>$\exists v \psi$</td>
<td>$\sum_{x \in \mathbf{V}(\mathbf{Y}^*)} [\psi]_\mathcal{M}(v</td>
</tr>
</tbody>
</table>

**Lemma 4.11.** (CZF)
For every bounded formula $\theta$ and a $\mathbf{V}(\mathbf{Y}^*)$-assignment $\mathcal{M}$, $[\theta]_\mathcal{M} = [\theta]_{\mathcal{M}'}$.

**Proof.** This follows by induction on $\theta$ by comparing Definitions 3.6 and 4.10. \qed

**Definition 4.12.** If $\theta(\alpha_1, \ldots, \alpha_r)$ is a formula of $\mathcal{L}_e$ all of whose free variables are among $u_1, \ldots, u_r$ and $\alpha_1, \ldots, \alpha_r \in \mathbf{V}(\mathbf{Y}^*)$, we shall use the shorthand $[\theta(\alpha_1, \ldots, \alpha_r)]$ rather than $[\theta]_{\mathcal{M}'}$, whenever $\mathcal{M}$ is an assignment satisfying $\mathcal{M}(u_i) = \alpha_i$ for $1 \leq i \leq r$. In the special case when $\theta$ is a sentence we will simply write $[\theta]$. We will also use the following abbreviations:
\[
e \models_\mathcal{M} \theta(\alpha_1, \ldots, \alpha_r) \iff e \in [\theta(\alpha_1, \ldots, \alpha_r)]
\]

$\mathbf{V}(\mathbf{Y}^*) \models \theta(\alpha_1, \ldots, \alpha_r) \iff e \models_\mathcal{M} \theta(\alpha_1, \ldots, \alpha_r)$ for some $e$

$\models^* \theta(\alpha_1, \ldots, \alpha_r) \iff \mathbf{V}(\mathbf{Y}^*) \models \theta(\alpha_1, \ldots, \alpha_r)$.

For a set-theoretic formula $\theta(\bar{\alpha})$ we say that $\theta(\alpha_1, \ldots, \alpha_r)$ is validated in $\mathbf{V}(\mathbf{Y}^*)$ if we have produced a closed application term $t$ such that $t(\bar{\alpha}) \models_\mathcal{M} \theta(\bar{\alpha})$ holds for all $\bar{\alpha} \in \mathbf{V}(\mathbf{Y}^*)$. 

\[\text{\normalfont The Formulae-as-Classes Interpretation of Constructive Set Theory}\]
4.3 The formulae-as-classes interpretation for CZF

The objective of the remainder of this section is to show the following interpretation.

**Theorem 4.13.** Let \( \theta(u_1, \ldots, u_r) \) be a formula of \( \mathcal{L}_e \) all of whose free variables are among \( u_1, \ldots, u_r \). If

\[
\text{CZF} + \text{RDC} + \Pi \Sigma - \text{AC} \vdash \theta(u_1, \ldots, u_r),
\]

then one can effectively construct an index of an \( E_v \)-recursive partial function \( g \) such that

\[
\text{CZF}_{Exp} \vdash \forall \alpha_1, \ldots, \alpha_r \in V(Y^*) \ g(\alpha_1, \ldots, \alpha_r) \in \llbracket \theta(\alpha_1, \ldots, \alpha_r) \rrbracket,
\]

where \( \text{CZF}_{Exp} \) denotes the modification of \( \text{CZF} \) with Exponentiation in place of Subset Collection.

**Proof.** This will follow from 4.16, 4.17, 4.26 and 4.31 below. \( \Box \)

The proof of 4.13 is rather long and requires close attention to the definition of indices of \( E_v \)-recursive functions. The details of an encoding are fascinating to work out and boring to read. The author wrote the present section for his own benefit and his feelings will not be hurt if the reader chooses to skip some of it.

**Lemma 4.14.** For every bounded \( \theta(u_1, \ldots, u_r) \) of \( \mathcal{L}_e \) all of whose free variables are among \( u_1, \ldots, u_r \) one can effectively construct an application term \( t_\theta \) such that

\[
\text{CZF}_{Exp} \vdash \forall \alpha_1, \ldots, \alpha_r \in V(Y^*) \ t_\theta(\alpha_1, \ldots, \alpha_r) \approx \llbracket \theta(\alpha_1, \ldots, \alpha_r) \rrbracket.
\]

**Proof.** We proceed by induction on the build-up of \( \theta \). The atomic cases amount to showing that the functions \( \alpha, \beta \mapsto \dot{x}(\alpha, \beta) \) and \( \alpha, \beta \mapsto \dot{\varepsilon}(\alpha, \beta) \) of Definition 3.2 are \( E_v \)-recursive partial functions defined on \( V(Y^*) \times V(Y^*) \).

1. Let \( \theta(u, v) \) be the atomic formula \( u = v \). We define an application term \( \ell \) by

\[
\ell(e, \alpha, \beta) \approx \pi(\bar{a}, \bar{b}(\bar{a}, \lambda x. \sigma(\bar{b}, \lambda y.e(\bar{a}(x), \bar{b}(y))))))
\]

\[
\approx \pi(\bar{p}_0\alpha, \bar{a}(\bar{p}_0\alpha, \lambda x. \sigma(\bar{p}_0\beta, \bar{a}(\bar{p}_0\beta, \lambda y.e(\bar{a}(p_1\alpha, x), \bar{a}(p_1\beta, y))))))).
\]

By the recursion theorem 4.7, we find an application term \( e \) such that

\[
e(\alpha, \beta) \approx \sigma(\ell(e, \alpha, \beta), \bar{a}(\ell(e, \alpha, \beta), \lambda z.\ell(e, \beta, \alpha))).
\]

Now put \( t_{u=v} := e \). Let \( < \) be the well-founded ordering on \( V(Y^*) \times V(Y^*) \) defined by:

\( \langle \alpha, \beta \rangle < \langle \gamma, \delta \rangle \) iff \( \alpha = \dot{\gamma}(x) \) and \( \beta = \dot{\delta}(y) \) for some \( x \in \dot{\gamma} \) and \( y \in \dot{\delta} \). In view of Definition 3.2, one shows by induction on \( < \) that for all \( \alpha, \beta \in V(Y^*) \), \( t_{u=v}(\alpha, \beta) \downarrow \) and \( t_{u=v}(\alpha, \beta) \approx \dot{x}(\alpha, \beta) \), i.e. \( t_{u=v}(\alpha, \beta) \approx \llbracket \alpha = \beta \rrbracket \).

2. For the other type of atomic formula, where \( \theta(u, v) \) is \( u \in v \), we put

\[
t_{u \in v} := \lambda x.y.\sigma(\bar{p}_0\gamma, \bar{a}(\bar{p}_0\gamma, \lambda z.t_{u=v}(x, \bar{a}(p_1\gamma, \gamma)))),
\]

so that \( t_{u \in v}(\alpha, \beta) \approx \dot{x}(\alpha, \beta) \), whence \( t_{u \in v}(\alpha, \beta) \approx \llbracket \alpha \in \beta \rrbracket \) for all \( \alpha, \beta \in V(Y^*) \).

3. If \( \theta(\bar{u}) = \theta_0(\bar{u}) \land \theta_1(\bar{u}) \), put

\[
t_{\theta(\bar{u})} := \lambda \bar{x}.\sigma(\bar{a}(\theta_0(\bar{u}), \bar{x}, \bar{a}(\theta_0(\bar{u}), \bar{x}, \lambda z.t_{\theta_0(\bar{u})}(\bar{x}))), \lambda z.(t_{\theta_0(\bar{u})}(\bar{x}))).
\]
Proof. If $\theta(\vec{u})$ is $\theta_0(\vec{u}) \lor \theta_1(\vec{u})$, put $t_{\theta(\vec{u})} := \lambda \vec{x}.\vec{p}(t_{\theta_0(\vec{u})}(\vec{x}), t_{\theta_1(\vec{u})}(\vec{x})).$

5. If $\theta(\vec{u})$ is $\theta_0(\vec{u}) \supset \theta_1(\vec{u})$, put

$$t_{\theta(\vec{u})} := \lambda \vec{x}.\pi(t_{\theta_0(\vec{u})}(\vec{x}), \overline{\alpha \beta}(t_{\theta_0(\vec{u})}(\vec{x}), \lambda z.(t_{\theta_1(\vec{u})}(\vec{x}))))).$$

6. If $\theta(v, \vec{u})$ is $\forall v \in \theta_0(w, v, \vec{u})$, put

$$t_{\theta(v, \vec{u})} := \lambda y \vec{x}.\pi(p_{00y}, \overline{\alpha \beta}(p_{00y}, \lambda z.t_{\theta_0(w,v,\vec{u})}(\vec{z}, y, \vec{x}))).$$

7. If $\theta(v, \vec{u})$ is $\exists w \in \theta_0(w, v, \vec{u})$, put

$$t_{\theta(v, \vec{u})} := \lambda y \vec{x}.\sigma(p_{00y}, \overline{\alpha \beta}(p_{00y}, \lambda z.t_{\theta_0(w,v,\vec{u})}(\vec{z}, y, \vec{x}))).$$

Lemma 4.15. There are closed application terms $id_r, id_s, id_t, id_{t_1}, id_{t_2}$ such that for all $\alpha, \beta, \gamma \in V(Y^*)$,

$$\begin{align*}
id_r(\alpha) &\vdash \alpha = \alpha. \\
id_s(\alpha, \beta) &\vdash \alpha = \beta \supset \beta = \alpha. \\
id_t(\alpha, \beta, \gamma) &\vdash \alpha = \beta \land \beta = \gamma \supset \alpha = \gamma. \\
id_{t_1}(\alpha, \beta, \gamma) &\vdash (\alpha = \beta \land \beta = \gamma) \supset \alpha = \gamma. \\
id_{t_2}(\alpha, \beta, \gamma) &\vdash (\alpha = \beta \land \beta \in \gamma) \supset \alpha \in \gamma. \\
\end{align*}$$

Proof. What do we need to construct $id_r$? Suppose we have

$$e(\hat{a}(j)) \in \llbracket \hat{a}(j) = \hat{a}(j) \rrbracket$$

for all $j \in \hat{a}$. Then, letting $f$ with domain $\hat{a}$ be defined by $f(j) \simeq \langle j, e(\hat{a}(j)) \rangle$, we get $\langle f, f \rangle \in \llbracket \alpha = \alpha \rrbracket$ (see Definition 3.2), thence we have

$$\begin{align*}
p(\overline{\alpha \beta}(p_{00\alpha}, \lambda x.p(x, e(\overline{\alpha \beta}(p_{01\alpha}, x))))) &\in \llbracket \alpha = \alpha \rrbracket. \\
\end{align*}$$

Now choose an index $d$ by the recursion theorem so that, for all $\alpha$,

$$d\alpha \simeq p(\overline{\alpha \beta}(p_{00\alpha}, \lambda x.p(x, d(\overline{\alpha \beta}(p_{01\alpha}, x))))) \overline{\alpha \beta}(p_{00\alpha}, \lambda x.p(x, d(\overline{\alpha \beta}(p_{01\alpha}, x)))))$$

and put $id_r := d$. By $\varepsilon$-induction on $\alpha$ one easily verifies that $id_r, \alpha \vdash$ and that $id_r$ has the desired property.

Intuitively, $id_s$ is the application term that interchanges the left and right member of any pair. However, a witness for a bounded conditional has to be a function. This is achieved by letting

$$id_s := \lambda \alpha \beta.\overline{\alpha \beta}(\llbracket \alpha = \beta \rrbracket, \lambda x.p(p_{10\alpha}, p_{00\alpha})) := \lambda y z.\overline{\alpha \beta}(\overline{t_{x=w}(y, z)}, \lambda x.p(p_{10\alpha}, p_{00\alpha})).$$
noting that the latter is an application term owing to Lemma 4.14.

To construct \( \text{id}_t \), suppose \( e \in \semantics{\alpha = \beta \land \beta = \gamma} \). Then \( p_0 e \in \semantics{\alpha = \beta} \) and \( p_1 e \in \semantics{\beta = \gamma} \). Thus \( p_0 e \simeq \langle f, g \rangle \) for functions \( f, g \) with domain \( \bar{\alpha} \) and \( \bar{\beta} \), respectively. Likewise, \( p_1 e \simeq \langle f', g' \rangle \) for functions \( f', g' \) with domain \( \bar{\beta} \) and \( \bar{\gamma} \), respectively. In view of the definition of \( \semantics{\alpha = \gamma} \), one easily sees that \( \langle f' \circ f, g \circ g' \rangle \in \semantics{\alpha = \gamma} \). We have to show that \( \langle f' \circ f, g \circ g' \rangle \) can be computed \( E_\nu \)-recursively from \( e, \alpha, \beta, \gamma \). We have

\[
\begin{align*}
\overline{ab}(p_0 \alpha, \lambda w. (\lambda u. \overline{fa}(f', u))((\lambda v. \overline{fa}(f, v))(w))) &\simeq f' \circ f \\
\overline{ab}(p_0 \gamma, \lambda w. (\lambda u. \overline{fa}(g, u))((\lambda v. \overline{fa}(g', v))(w))) &\simeq g \circ g',
\end{align*}
\]

thus letting

\[
\begin{align*}
\tau_1 &:= \lambda x y z. \overline{ab}(p_0 x, \lambda w. (\lambda u. \overline{fa}(f', u))((\lambda v. \overline{fa}(f, v))(w))), \\
\tau_2 &:= \lambda x'y' z'. \overline{ab}(p_0 x' \lambda w. (\lambda u. \overline{fa}(g', u))((\lambda v. \overline{fa}(g', v))(w))),
\end{align*}
\]

we get \( \tau_1 \alpha f f' \simeq f' \circ f \) and \( \tau_2 \gamma g g' \simeq g \circ g' \), so that

\[
\begin{align*}
\pi(p_1 \alpha (p_0 (p_0 e))(p_1 10 e)) (p_2 \gamma (p_0 1 e)(p_1 11 e)) &\simeq \langle f' \circ f, g \circ g' \rangle,
\end{align*}
\]

where \( p_{i,j} := \lambda v. p_j(p_i v) \). Thus letting

\[
\text{id}_t := \lambda \alpha \beta \gamma. \overline{ab}(\semantics{\alpha = \beta \land \beta = \gamma}, \lambda y p_1 \alpha (p_0 0 y)(p_1 10 y)) (\tau_2 \gamma (p_0 1 y)(p_1 11 y))
\]

\[
:= \lambda z_1 z_2 z_3. \overline{ab}(\theta_{\alpha, \beta, \gamma}(z_1, z_2, z_3), \lambda y p_1 \alpha (p_0 0 y)(p_1 10 y)) (\tau_2 z_3 (p_0 1 y)(p_1 11 y)),
\]

where \( \theta(u_1, u_2, u_3) \) stands for \( u_1 = u_2 \land u_2 = u_3 \), we get

\[
\text{id}_t(\alpha, \beta, \gamma) \Vdash (\alpha = \beta \land \beta = \gamma) \supset \alpha = \gamma.
\]

Next we show how to construct \( \text{id}_{t_1} \). To this end suppose \( d_0 \in \semantics{\alpha = \beta} \) and \( d_1 \in \semantics{\beta \in \gamma} \). Then \( d_1 = \langle i, e' \rangle \) for some \( i, e' \) with \( i, e' \Vdash \beta = \bar{\gamma}(i) \). Hence we get

\[
\text{id}_t(\alpha, \beta, \bar{\gamma}(i), \langle d, e' \rangle) \Vdash \alpha = \bar{\gamma}(i),
\]

which yields \( \langle i, \text{id}_t(\alpha, \beta, \bar{\gamma}(i), \langle d, e' \rangle) \rangle \Vdash \alpha \in \gamma \).

In view of previous constructions it is therefore clear that we can cook up an application term \( t_\ast \) such that whenever \( d \Vdash \alpha = \beta \land \beta \in \gamma \) then \( t_\ast(\alpha, \beta, \gamma, d) \Vdash \alpha \in \gamma \). Whence we can put

\[
\text{id}_{t_1} := \lambda \alpha \beta \gamma. \overline{ab}(\semantics{\alpha = \beta \land \beta \in \gamma}, \lambda z. t_\ast(\alpha, \beta, \gamma, z)).
\]

For \( \text{id}_{t_2} \) suppose that \( d_0 \in \semantics{\alpha = \beta} \) and \( d_1 \in \semantics{\gamma \in \alpha} \). Then \( d_1 = \langle i, e' \rangle \) for some \( i, e' \) with \( i \in \bar{\alpha} \) and \( e' \Vdash \gamma = \bar{\alpha}(i) \). Moreover, \( d_0 = \langle f, g \rangle \) for functions \( f, g \), where the domain of \( f \) is \( \bar{\alpha} \), and \( f(i) = \langle j, c \rangle \) for some \( j \in \bar{\beta} \) and \( c \Vdash \bar{\alpha}(i) = \bar{\beta}(j) \). It follows that

\[
\langle j, \text{id}_t(\gamma, \bar{\alpha}(i), \bar{\beta}(j), \langle e', c \rangle) \rangle \Vdash \gamma = \bar{\beta}(j),
\]

and hence

\[
\langle j, \text{id}_t(\gamma, \bar{\alpha}(i), \bar{\beta}(j), \langle e', c \rangle) \rangle \Vdash \gamma \in \beta.
\]

It is evident, by previous application term constructions that \( \langle j, \text{id}_t(\gamma, \bar{\alpha}(i), \bar{\beta}(j), \langle e', c \rangle) \rangle \) can be obtained \( E_\nu \)-recursively from \( d_0, d_1, \alpha, \beta, \gamma \).

\[\square\]
Lemma 4.16. If the formula $\theta(\vec{u})$ is derivable in intuitionistic predicate logic with identity, then one can effectively construct an application term $t$ from the deduction such that

$$t(\vec{\alpha}) \vdash \theta(\vec{\alpha})$$

holds for all $\vec{\alpha} \in V(Y^*)$.

We have already shown this for the axioms pertaining to equality. We illustrate the ideas for logical axioms by carrying out a few examples. If $\varphi(\vec{u})$ and $\psi(\vec{u})$ are unbounded formulas, then one easily checks that

$$k \vdash \varphi(\vec{\alpha}) \supset (\psi(\vec{\alpha}) \supset \varphi(\vec{\alpha})), \quad (12)$$

and

$$s \vdash (\varphi(\vec{\alpha}) \supset (\psi(\vec{\alpha}) \supset \chi(\vec{\alpha}))) \supset ((\varphi(\vec{\alpha}) \supset \psi(\vec{\alpha})) \supset (\varphi(\vec{\alpha}) \supset \chi(\vec{\alpha}))). \quad (13)$$

In fact, this justifies the combinators $s$ and $k$. Combinatory completeness of these two combinators is equivalent to the fact that the deductive closure under modus ponens of (12) and (13) is the full set of theorems of propositional implicational intuitionistic logic.

In case that $\varphi$ or $\psi$ are bounded we need to make crucial changes to the above. In case that both $\varphi$ and $\psi$ are bounded, we get that

$$ab((\varnothing(\vec{\alpha})), \lambda x. ab((\varnothing(\vec{\alpha})), \lambda y. kxy)) \vdash \varphi(\vec{\alpha}) \supset (\psi(\vec{\alpha}) \supset \varphi(\vec{\alpha})).$$

It is pivotal to note here that according to Lemma 4.14, both $\varnothing(\vec{\alpha})$ and $\varnothing(\vec{\alpha})$ are $E_\psi$-computable from $\vec{\alpha}$, yielding that $ab((\varnothing(\vec{\alpha})), \lambda x. ab((\varnothing(\vec{\alpha})), \lambda y. kxy))$ is $E_\psi$-computable from $\vec{\alpha}$.

In the case that $\varphi$ is bounded but $\psi$ is unbounded we get

$$\lambda x. ab((\varnothing(\vec{\alpha})), \lambda xy. kxy) \vdash \varphi(\vec{\alpha}) \supset (\psi(\vec{\alpha}) \supset \varphi(\vec{\alpha})).$$

In the case that $\varphi$ is unbounded but $\psi$ is bounded we get

$$\lambda x. ab((\varnothing(\vec{\alpha})), \lambda y. kxy) \vdash \varphi(\vec{\alpha}) \supset (\psi(\vec{\alpha}) \supset \varphi(\vec{\alpha})).$$

By the foregoing it should also be obvious what modifications have to be made to (13) if one of the formulas $\varphi$ or $\psi$ is bounded.

Assuming all formulas are unbounded, one can easily check that:

$$\lambda xy. pxy \vdash \varphi(\vec{\alpha}) \supset (\psi(\vec{\alpha}) \supset \varphi(\vec{\alpha}) \land \psi(\vec{\alpha})), \lambda x. x \vdash \forall y (\forall x \theta(x, \vec{\alpha}) \supset \theta(y, \vec{\alpha})))$$

and that the term

$$\lambda x. \lambda y. d_N(y(p_1x))(z(p_1))(p_0x)$$

witnesses $\lor$-elimination, that is to say, the axiom scheme

$$\varphi \lor \psi \supset ((\varphi \lor \chi) \supset [(\psi \lor \chi) \lor \chi]).$$

Minor modifications yield witnessing terms if one or more of the formulas are bounded.
From Lemma 4.15 in conjunction with the usual laws of intuitionistic connectives and quantifiers one can deduce that \( \forall x[\theta(x, \vec{y}) \supset \forall z(x = x \supset \theta(z, \vec{y}))] \) and hence one can effectively construct a term \( t_{\equiv, \theta} \) such that

\[
t_{\equiv, \theta} \vdash \forall x[\theta(x, \vec{a}) \supset \forall z(x = x \supset \theta(z, \vec{a}))]. \tag{14}
\]

Since the quantifier logic employed here is slightly unusual in that bounded quantifiers are treated as quantifiers in their own right, we should check that the usual logical connections obtain between bounded and unbounded quantifiers. The claim is that one can construct terms \( t_{\forall, \varphi} \) and \( t_{\exists, \varphi} \) such that

\[
t_{\forall, \varphi}(\beta, \vec{a}) \vdash \forall x \in \beta \varphi(x, \beta, \vec{a}) \iff \forall x[\varphi(x, \beta, \vec{a})] \tag{15}
\]

\[
t_{\exists, \varphi}(\beta, \vec{a}) \vdash \exists x \in \beta \varphi(x, \beta, \vec{a}) \iff \exists x[\varphi(x, \beta, \vec{a})]. \tag{16}
\]

To prove (15) first suppose that \( f \vdash \forall x \in \beta \varphi(x, \beta, \vec{a}) \). Then for all \( i \in \vec{\beta} \), \( f(i) \vdash \varphi(\vec{\beta}(i), \beta, \vec{a}) \). Now suppose in addition that \( \delta \in V(Y^*) \) and \( d \vdash \delta \in \beta \). Then \( d = \langle j, d_1 \rangle \) with \( j \in \vec{\beta} \) and \( d_1 \vdash \delta = \beta(i) \). Utilizing (14), we can design an application term \( \tau \) such that

\[
\tau(f, d, j, \vec{\beta}(j), \vec{\alpha}, \beta, \delta) \vdash \varphi(\delta, \beta, \vec{a}).
\]

Moreover, since \( j \) and \( \vec{\beta}(j) \) are actually computable from \( d \) and \( \beta \), there is a closed application term \( \tau' \) such that

\[
\tau'(f, d, \vec{\alpha}, \beta, \delta) \vdash \varphi(\delta, \beta, \vec{a}). \tag{17}
\]

As a consequence of (17), we get

\[
\overline{\text{ab}}(\prod \delta \in \beta, \lambda y. \tau'(f, y, \vec{\alpha}, \beta, \delta)) \vdash \delta \in \beta \supset \varphi(\delta, \beta, \vec{a}). \tag{18}
\]

Thus we have

\[
\lambda \delta. \overline{\text{ab}}(\prod \delta \in \beta, \lambda y. \tau'(f, y, \vec{\alpha}, \beta, \delta)) \vdash \forall x[\varphi(x, \beta, \vec{a})]. \tag{19}
\]

For the other direction of (15), suppose

\[
e \vdash \forall x[\varphi(x, \beta, \vec{a})]. \tag{20}
\]

We need to effectively construct a function \( f \) with domain \( \vec{\beta} \) such that \( f(i) \vdash \varphi(\vec{\beta}(i), \beta, \vec{a}) \). Let \( i \in \vec{\beta} \). By Lemma 4.15, we have \( \text{id}_\lambda(\vec{\beta}(i)) \in \prod \vec{\beta}(i) = \vec{\beta}(i) \), thus

\[
\langle i, \text{id}_\lambda(\vec{\beta}(i)) \rangle \vdash \vec{\beta}(i) \in \beta,
\]

so that by (20), \( e(p(i, \text{id}_\lambda(\vec{\beta}(i)))) \vdash \varphi(\vec{\beta}(i), \beta, \vec{a}) \). Letting

\[
f := \overline{\text{ab}}(\vec{\beta}, \lambda x. e(p(x, \text{id}_\lambda(\vec{\beta}(x)))))
\]

we get \( f \vdash \forall x \in \beta \varphi(x, \beta, \vec{a}) \). Also, by now it is plain how to design an application term which shows that \( f \) is \( E_\psi \)-computable from \( e, \beta \).

The proof of (16) is very similar to (15). \qed
Lemma 4.17. For each axiom $\theta(\vec{a})$ of CZF one can effectively construct an application term $t$ from the deduction such that

$$t(\vec{a}) \vdash \theta(\vec{a})$$

holds for all $\vec{a} \in V(Y^*)$.

Proof. 1. Extensionality: Suppose

$$e \vdash \forall x [(x \in \alpha \supset x \in \beta) \land (x \in \beta \supset x \in \alpha)].$$

By lemma 4.16, in particular (15), there is an application term $t_{ex}$ such that

$$t_{ex}(\alpha, \beta, e) \vdash \forall x \in \alpha x \in \beta \land \forall x \in \beta x \in \alpha.$$

The latter actually amounts to $t_{ex}(\alpha, \beta, e) \vdash \alpha = \beta$, so that

$$\lambda x. t_{ex}(\alpha, \beta, x) \vdash \forall x [(x \in \alpha \supset x \in \beta) \land (x \in \beta \supset x \in \alpha)] \supset \alpha = \beta.$$

2. Pair: Let $\alpha, \beta \in V(Y^*)$. Put $\gamma := \sup(2, f)$, where $\text{dom}(f) = 2$, $f(0) = \alpha$, and $f(1) = \beta$. As $2 \in Y^*$, we get $\gamma \in V(Y^*)$. Also $\gamma$ is computable from $\alpha$ and $\beta$ as $0(\alpha) \simeq 0$, $s_0(0) \simeq 1$, $s_1(1) \simeq 2$, $\overline{\text{ab}}(2, \lambda n. \overline{d}_N(0, n, \alpha, \beta)) \simeq f$ and $p^2 f \simeq \gamma$. Moreover, one easily constructs a term $t_p$ such that $t_p(\alpha, \beta) \vdash \forall x \in \gamma \text{ iff } [x = \alpha \lor x = \beta]$ and hence

$$p(\gamma, t_p(\alpha, \beta)) \vdash \exists y \forall x (x \in y \text{ iff } [x = \alpha \lor x = \beta]),$$

from which we can distill a term witnessing the pairing axiom.

Union: Let

$$\gamma := \sup(a, f) = paf,$$

$$a := \sigma(\bar{a}, \overline{\text{ab}}(\bar{a}, \lambda z. \bar{a}(z))),$$

$$f := \overline{\text{ab}}(a, \lambda x. (\bar{a}(\bar{p}_0 x))(p_1(x))).$$

One readily checks that $a \in Y^*$ and $f : a \rightarrow V(Y^*)$, thus $\gamma \in V(Y^*)$. Also, it is clear that $\gamma$ is $E_\gamma$-computable from $\alpha$. We leave it to the reader to construct a term $t_\cup$ such that $t_\cup(\alpha) \vdash \forall v \in \gamma \text{ iff } \exists u \in v \in u]$, so that

$$p(\gamma, t_\cup(\alpha)) \vdash \exists y \forall u [v \in y \text{ iff } \exists u \in \alpha v \in u].$$

Bounded Separation: Let $\theta(u, v, \vec{w})$ be a bounded formula and $\alpha, \vec{\beta} \in V(Y^*)$. Let $A := \llbracket \exists x \in \alpha, \theta(x, \alpha, \vec{\beta}) \rrbracket$ and $g : A \rightarrow V(Y^*)$ be defined by $g(z) := \bar{a}(p_0 z)$. Put $\gamma := \sup(A, g)$. Note that $g = \overline{\text{ab}}(A, \lambda z. \bar{a}(p_0 z))$. We have $A \in Y^*$ by Lemma 3.7 so that $\gamma \in V(Y^*)$. By Lemma 4.14, $A$ and $g$ and hence $\gamma$ are $E_\gamma$-computable from $\alpha, \vec{\beta}$. We also have

$$\langle i, x \rangle \in \llbracket \eta \in \gamma \rrbracket \text{ iff } i \in A \land x \in \llbracket \eta = \bar{\gamma}(i) \rrbracket \land p_0 i \in \llbracket \theta(\bar{\gamma}(i), \alpha, \vec{\beta}) \rrbracket$$

$$\text{iff } i \in A \land x \in \llbracket \eta = \bar{\gamma}(i) \rrbracket \land t_\vec{0}(i, \gamma, \alpha, \vec{\beta}) \in \llbracket \theta(\eta, \alpha, \vec{\beta}) \rrbracket$$

for some term $t_\vec{0}$ obtainable from (14). As a result we can compose a term $t_s$ such that

$$p(\gamma, t_s(\alpha, \vec{\beta})) \vdash \forall u (u \in \gamma \text{ iff } [u \in \alpha \land \theta(u, \alpha, \vec{\beta})]).$$
Strong Collection: Suppose
\[ f \vdash \forall u \in \alpha \exists v \, \theta(u, v, \alpha, \vec{\beta}). \]
Then \( f \) is a function with domain \( \vec{\alpha} \) such that, for all \( i \in \vec{\alpha} \),
\[ \mathbf{P}_1(f(i)) \vdash \theta((\vec{\alpha}(i)), \mathbf{P}_0(f(i)), \vec{\beta}). \]
Let \( g \) be the result of \( \overline{\alpha \beta}(\vec{\alpha}, \lambda x. \mathbf{P}_0(f(x))) \) and put \( \gamma = \mathbf{P}(\vec{\alpha}, g) \). By definition, \( \gamma \) is \( E_\varphi \)-computable from \( \alpha \) and \( f \). Furthermore, one readily constructs a term \( t_1 \) satisfying
\[ t_1(\alpha, f) \vdash \forall u \in \alpha \exists v \in \gamma \, \theta(u, v, \vec{\beta}) \land \forall v \in \gamma \exists u \in \alpha \, \theta(u, v, \vec{\beta}), \]
from which we can glean a term \( t_{++} \) satisfying
\[ t_{++}(\alpha, \vec{\beta}) \vdash \forall u \in \alpha \exists v \in \gamma \, \theta(u, v, \vec{\beta}) \supset \exists w \theta'(u, w, \vec{\beta}), \]
where \( \theta'(\alpha, w, \vec{\beta}) \) denotes the formula \( \forall u \in \alpha \exists v \in w \, \theta(u, v, \vec{\beta}) \land \forall v \in w \exists u \in \alpha \, \theta(u, v, \vec{\beta}) \).

Subset Collection: Let \( \alpha, \beta \in \mathbf{V}(Y^*) \) and define \( A := (\vec{\alpha} \rightarrow \vec{\beta}) \) and a function \( h \) with domain \( A \) by
\[ h(z) := \sup(\vec{\alpha}, \text{ab}(\vec{\alpha}, \lambda x. \tilde{\beta}(z(x)))) \].
Put \( \gamma := \sup(A, h) \). It is easy to see that \( \gamma \in \mathbf{V}(Y^*) \) and that \( \gamma \) is \( E_\varphi \)-computable from \( \alpha \) and \( \beta \). Now assume that
\[ f \vdash \forall u \in \alpha \exists v \in \beta \, \varphi(u, v, \alpha, \vec{\beta}). \]
Let \( \theta(u, v) \) stand for \( \varphi(u, v, \alpha, \vec{\beta}, \vec{\xi}) \). Then we have
\[ \forall i \in \vec{\alpha} \left[ \mathbf{P}_0(f(i)) \in \vec{\beta} \land \mathbf{P}_1(f(i)) \vdash \theta(\vec{\alpha}(i), \tilde{\beta}(\mathbf{P}_0(f(i)))) \right]. \]
Thus we have \( g \in A \), where \( g \) is the function with domain \( \vec{\alpha} \) and \( g(i) := \mathbf{P}_0(f(i)) \). Let
\[ \delta := \sup(\vec{\alpha}, \text{ab}(\vec{\alpha}, \lambda x. \tilde{\beta}(g(x)))) \].
Then \( \delta \in \mathbf{V}(Y^*) \) and \( \delta \) is clearly \( E_\varphi \)-computable from \( \alpha \) and \( f \). Moreover, \( \delta = h(g) \) so that
\[ \mathbf{P}(g, \text{id}_\delta(\delta)) \vdash \delta \in \gamma. \] (21)
Also \( \vec{\alpha} = \vec{\delta} \) and
\[ \forall i \in \vec{\alpha} \left[ \mathbf{P}_1(f(i)) \vdash \theta(\vec{\alpha}(i), \vec{\delta}(i)) \right]. \]
The latter yields
\[ \mathbf{P}(\vec{\alpha}, \text{ab}(\vec{\alpha}, \lambda x. \mathbf{P}(x, \mathbf{P}_1(f(x)))) \) \vdash \forall u \in \alpha \exists v \in \delta \, \theta(u, v) \quad (22) \]
\[ \mathbf{P}(\vec{\alpha}, \text{ab}(\vec{\alpha}, \lambda x. \mathbf{P}(x, \mathbf{P}_1(f(x)))) \) \vdash \forall v \in \delta \exists u \in \alpha \, \theta(u, v). \]
In view of (21) and (22), we can construct a term \( t_c \) such that
\[ t_c(\alpha, \beta, f) \vdash \exists w \in \gamma \left[ \forall u \in \alpha \exists v \in w \, \theta(u, v) \land \forall v \in w \exists u \in \alpha \, \theta(u, v) \right]. \]
Thence we can construct a term \( t_{sac} \) such that
\[
t_{sac}(\alpha, \beta, \xi) \models \exists z \left( \forall u \in \alpha \exists v \in \beta \theta(u, v) \supset \exists w \in \gamma \left[ \forall u \in \alpha \exists v \in w \theta(u, v) \land \forall v \in u \exists u \in \alpha \theta(u, v) \right] \right),
\]
verifying Subset Collection.

**\( \in \)-Induction:** By the recursion theorem we can effectively construct an index of an \( E_\varphi \)-recursive partial function \( h \) satisfying
\[
h(e, \alpha) \simeq (e\alpha)(\overline{aB}(\overline{\alpha}, \lambda x. h(e, x))).
\]
Now suppose \( e \models \forall u[\forall v \in u \theta(v) \supset \theta(u)] \). By induction on \( \alpha \), we shall prove that
\[
h(e, \alpha) \models \theta(\alpha).
\]
Toward this end suppose that for all \( i \in \overline{\alpha} \), \( h(e, \alpha) \models \theta(\overline{\alpha}(i)) \). Then we have \( \overline{aB}(\overline{\alpha}, \lambda x. h(e, x)) \]
\[
\models \forall v \in \alpha \theta(v) \quad \text{so that} \quad (e\alpha)(\overline{aB}(\overline{\alpha}, \lambda x. h(e, x))) \models \theta(\alpha),
\]
yielding (23). Finally from (23) we arrive at
\[
\lambda y z. h(y, z) \models \forall u[\forall v \in u \theta(v) \supset \theta(u)] \supset \forall u \theta(u).
\]

**Infinity:** First note that the formula \( 0 \in y \land \forall w \in y \; w + 1 \in y \) written out in full detail reads
\[
(\exists z \in y \forall u \in z \perp) \land \forall w \in y \exists v \in y \forall u \in v[u \in w \lor u = w].
\]
Using the recursion theorem, one effectively constructs an index \( \hat{e} \) such that
\[
\hat{e} x \simeq \sup\left( x, \overline{aB}(x, \lambda y. d_N(p_N x, y, \sup(p_N x, \overline{aB}(p_N x, \lambda z. \hat{e} z)), \hat{e} y)) \right).
\]
By induction on \( n \in \omega \), we show \( \hat{e} k \downarrow \) and \( \hat{e} k \in V(Y^*) \) for all \( k \leq n \). Note that \( n \in Y^* \) for all \( n \in \omega \). Clearly we have \( \hat{e} 0 \downarrow \) and \( \hat{e} 0 \in V(Y^*) \). Assume that \( \hat{e} k \downarrow \) and \( \hat{e} k \in V(Y^*) \) for all \( k \leq n \). As \( p_N(n + 1) \simeq n \) we then have \( \overline{aB}(p_N(n + 1), \lambda z. \hat{e} z)(0) \downarrow \), which yields
\[
d_N(p_N(n + 1), m, \sup(p_N(n + 1), \overline{aB}(p_N(n + 1), \lambda z. \hat{e} z)), \hat{e} m) \downarrow
\]
for all \( m \leq n \). The latter yields \( \hat{e}(n + 1) \downarrow \) and also \( \hat{e}(n + 1) \in V(Y^*) \).

By the above we may define a function \( h_\omega \) with domain \( \omega \) such that \( \hat{e} n \simeq h_\omega(n) \). As \( \omega \in V(Y^*) \) we get
\[
\omega^* := \sup(\omega, h_\omega) \in V(Y^*).
\]
By construction, \( \omega^* \) is \( E_\varphi \)-computable. Note that
\[
h_\omega(n + 1) \simeq \sup(n + 1, \overline{aB}(n + 1, \lambda y. d_N(n, y, \sup(n, h_\omega \upharpoonright n), h_\omega(y))))
\]
where \( h_\omega \upharpoonright n \) denotes the restriction of \( h_\omega \) to \( n \). Thus, inductively we have \( \overline{h_\omega(0)} = 0 \), \( \overline{h_\omega(n + 1)} = n + 1 \), and
\[
h_\omega(n) = \sup(n, h_\omega \upharpoonright n)
\]
for all \( n \in \omega \), so that
\[
\forall i \in h_\omega(n) \ (h_\omega(n))(i) = h_\omega(i).
\]
We then get
\[
\forall i \in h_\omega(n + 1) \ [(h_\omega(n + 1))(i) = (h_\omega(n))(i) \lor (h_\omega(n + 1))(i) = h_\omega(n)].
\]
Defining a function \( f_n \) by \( \text{dom}(f) = n + 1 \) and
\[
f_n(i) := d_N(n, i, \langle i, d_{id_\omega}(h_\omega(n))(i) \rangle, \langle 1, id_\omega(h_\omega(n)) \rangle),
\]
we obtain \( f_n \models \forall u \in h_\omega(n + 1) \ [u \in h_\omega(n) \lor u = h_\omega(n)] \), that is
\[
f_n \models h_\omega(n + 1) = h_\omega(n) + 1 \quad (26)
\]
in short hand. As a result,
\[
\langle n + 1, f_n \rangle \models \exists v \in \omega^* \forall u \in v \ [u \in h_\omega(n) \lor u = h_\omega(n)]
\]
and consequently
\[
f^* \models \forall w \in \omega^* \exists v \in \omega^* \forall u \in v \ [u \in w \lor u = w], \quad (27)
\]
where \( f^* \) is the function with domain \( \omega \) and \( f^*(n) = \langle n + 1, f_n \rangle \). We also have \( \overline{aB}(0, \lambda x.x) \models \forall u \in h_\omega(0) \perp \), and whence
\[
\langle 0, \overline{aB}(0, \lambda x.x) \rangle \models \exists z \in \omega^* \forall u \in z \perp. \quad (28)
\]
Since \( f^* \) is \( E_0 \)-computable we can combine (27) and (28) to extract a term \( t^*_0 \) such that
\[
t^*_0 \models \exists z \in \omega^* \forall u \in z \perp \land \forall w \in \omega^* \exists v \in \omega^* \forall u \in v \ [u \in w \lor u = w],
\]
in other words
\[
t^*_0 \models 0 \in \omega^* \land \forall w \in \omega^* w + 1 \in \omega^*. \quad (29)
\]
Let \( \theta(\alpha) \) abbreviate the formula
\[
\exists z \in \alpha \forall u \in z \perp \land \forall w \in \alpha \exists v \in \alpha \forall u \in v \ [u \in w \lor u = w]
\]
and suppose
\[
e \models \theta(\alpha), \quad (30)
\]
for some \( \alpha \in V(Y^*) \). In view of (26) one then constructs a term \( t_\omega \) such that for all \( n \in \omega \),
\[
t_\omega(n, \alpha, e) \models h_\omega(n) \in \alpha
\]
so that
\[
\overline{aB}(\omega, \lambda x.t_\omega(x, \alpha, e)) \models \forall u \in \omega^* u \in \alpha.
\]
Hence
\[
\lambda y z. \overline{aB}(\omega, \lambda x.t_\omega(x, y, z)) \models \forall a \ [\theta(\alpha) \supset \omega^* \subseteq a]. \quad (31)
\]
Using the witnessing terms from (29) and (31) and the fact that \( \omega^* \) is \( E_\omega \)-computable we then get a term \( t^* \) such that
\[
t^* \models \exists y (\theta(y) \land \forall a \ [\theta(\alpha) \supset y \subseteq a]).
\]
\( \square \)
Remark 4.18. In his interpretation of set theory in Martin-Löf type theory (cf. [1, 2, 3]), Aczel frequently invokes the axiom of choice in type theory as for instance in the proofs of the validity of Strong Collection and Subset Collection. To the author’s mind, this casts a shroud of mystery over these proofs. The axiom of choice is usually invoked, as it were, to pull a function out of thin air, but due to the propositions-as-types doctrine of Martin-Löf’s type theory, the axiom of choice is deducible therein since a judgement that a proposition of the form \( \forall x : A \exists y : B \varphi(x, y) \) holds true provides all the information needed to construct a function \( f : A \to B \) such that \( \forall x : A \varphi(x, f(x)) \). The formulae-as-classes interpretation via \( E \)-\( \mathbb{P} \)-recursive partial functions bears out this fact very lucidly. The latter remarks are even more appropriate to the proofs of the subsection to come.

4.4 Choice principles in \( V(Y^*) \)

In this subsection we show that \( V(Y^*) \) validates \( \text{RDC} \) and \( \Pi \Sigma I – \text{AC} \). We will be retracing the ground of [2] §5–6.

Definition 4.19. Let \( \text{OP}(x, y) = z \) be an abbreviation for the set-theoretic formula expressing that \( z \) is the ordered pair of \( x \) and \( y \). By Lemma 4.17 there exist \( E_\varphi \)-recursive functions \( \alpha, \beta \mapsto \langle \alpha, \beta \rangle \) and \( \widehat{\varphi} \) such that for all \( \alpha, \beta \in V(Y^*) \), \( \widehat{\varphi}(\alpha, \beta) \downarrow \) and \( \langle \alpha, \beta \rangle \in V(Y^*) \) and

\[
\widehat{\varphi}(\alpha, \beta) \vdash \text{OP}(\alpha, \beta) = \langle \alpha, \beta \rangle.
\]  

(32)

If \( A \in Y^* \) and \( f \) is an \( E_\varphi \)-recursive function which is total on \( A \) then we shall use the notation

\[
\left( \sup i \in A \right) f(i) := \sup \left( A, \text{aB}(A, \lambda x.f(x)) \right).
\]

For \( \alpha, \beta \in V(Y^*) \) let

\[
S(\alpha, \beta) := \left( \sup i \in \bar{\alpha} \right) \langle \bar{\alpha}(i), \bar{\beta}(i) \rangle = \langle \alpha, \beta \rangle.
\]  

(33)

Note that \( S \) is an \( E_\varphi \)-partial recursive function such that if \( \bar{\alpha} = \bar{\beta} \) then \( S(\alpha, \beta) \downarrow \) and, moreover, \( S(\alpha, \beta) \in V(Y^*) \).

Lemma 4.20. There are \( E_\varphi \)-recursive partial functions \( g_0, g_1, g_2 \) such that the following hold:

(i) If \( \alpha, \beta \in V(Y^*) \) with \( \bar{\alpha} = \bar{\beta} \), then

\( g_0(\alpha, \beta) \vdash S(\alpha, \beta) \) is a relation with domain \( \alpha \) and range \( \beta \).

(ii) If \( \alpha, \gamma \in V(Y^*) \) such that

\( e \vdash \gamma \) is a relation with domain \( \alpha \)

then \( g_1(e, \alpha, \gamma) \in V(Y^*) \) and, letting \( \delta := g_1(e, \alpha, \gamma) \), we have \( \bar{\delta} = \bar{\alpha} \) and

\( g_2(e, \alpha, \gamma) \vdash S(\alpha, \delta) \subseteq \gamma \).
Proof. (i): Let $\alpha, \beta \in V(Y^*)$ such that $\bar{\alpha} = \bar{\beta}$ and let $\gamma = S(\alpha, \beta)$. Also let $A = \bar{\alpha}$. Then, by choosing $x = y = u$,
\begin{align*}
\forall u \in A \exists x \in A \exists y \in A \models ^* \gamma = \langle \alpha, \beta \rangle_V,
\end{align*}
so that
\begin{align*}
h_0(\alpha, \beta) \models \forall u \in \gamma \exists x \in \alpha \exists y \in \beta \ OP(x, y) = u,
\end{align*}
where $h_0$ stands for the obvious $E_\ell$-recursive partial function. Also, by choosing $u = y = x$, we get
\begin{align*}
h_1(\alpha, \beta) \models \forall x \in \alpha \exists u \in \gamma \exists y \in \beta \ OP(x, y) = u,
\end{align*}
where $h_1$ stands for the obvious $E_\ell$-recursive partial function. As the the set-theoretic statements in (34) and (35) (after the “$\models$” symbol) provably in CZF entail that
\begin{align*}
S(\alpha, \beta) \text{ is a relation with domain } \alpha \text{ and range } \beta,
\end{align*}
we can construct the desired function $g_0$ from $h_0$ and $h_1$.

(ii): Suppose that $e \models \gamma$ is a relation with domain $\alpha$. Then we obtain $E_\ell$-recursive partial functions $\ell_0, \ell_1$ such that
\begin{align*}
\forall i \in A \ell_0(\alpha, i, \gamma, e) \models \exists z \in S(\alpha, \beta) \ OP(\bar{\alpha}(i), \delta) = z,
\end{align*}
where $\delta = \ell_1(\alpha, i, \gamma, e)$. Now let $f : A \to V(Y^*)$ be defined by $\text{ad}(A, \lambda v. \ell_1(\alpha, v, \gamma, e))$ and put $\beta = \sup(A, f)$. Then $\beta \in V(Y^*)$ and $\bar{\beta} = \bar{\alpha}$. Clearly, $\beta$ can be computed from $\alpha, \gamma, e$.

From the above we construct $g_2$ such that $g_2(e, \alpha, \gamma) \models S(\alpha, \delta) \subseteq \gamma$.

\begin{proof}
\end{proof}

Definition 4.21. $\alpha \in V(Y^*)$ is injectively presented if for all $i, j \in \bar{\alpha}$, whenever
\begin{align*}
e \models \bar{\alpha}(i) = \bar{\alpha}(j)
\end{align*}
for some $e$, then $i = j$.

Lemma 4.22. There are $E_\ell$-recursive partial functions $\ell_0^*, \ldots, \ell_5^*$ such that for any injectively presented $\alpha \in V(Y^*)$ the following hold:

(i) If $\beta \in V(Y^*)$, such that $\bar{\beta} = \bar{\alpha}$ and $\delta \in V(Y^*)$ then for all $i \in \bar{\alpha}$,
\begin{align*}
\ell_0^*(\alpha, i, \beta, \delta) \models \exists z \in S(\alpha, \beta) \ OP(\bar{\alpha}(i), \delta) = z \text{ iff } \delta = \bar{\beta}(i),
\end{align*}

(ii) If $\gamma \in V(Y^*)$ and $e \models \gamma$ is a function with domain $\alpha$

then
\begin{align*}
\ell_1^*(\alpha, \gamma, e) \models \gamma = S(\alpha, \beta),
\end{align*}
with $\beta = \ell_2^*(\alpha, \gamma, e) \in V(Y^*)$ and $\bar{\beta} = \bar{\alpha}$.

Conversely, if $\bar{\beta} = \bar{\alpha}$ and $e' \models \gamma = S(\alpha, \beta)$, then
\begin{align*}
\ell_3^*(\alpha, \beta, \gamma, e') \models \gamma \text{ is a function with domain } \alpha.
\end{align*}
(iii) If $\beta_1, \beta_2 \in V(Y^*)$ such that $\beta_1 = \beta_2 = \bar{\alpha}$, then $e \models S(\alpha, \beta) = S(\alpha, \beta_2)$ implies

$$\forall i \in \bar{\alpha} \ \ell_i'(\alpha, i, \beta, \beta_2) \models \beta_1(i) = \bar{\beta}_2(i),$$

and conversely, if $\forall i \in \bar{\alpha} \ e' i \models \beta_1(i) = \bar{\beta}_2(i)$, then

$$\ell_i'(\alpha, \beta, \beta_2, e') \models S(\alpha, \beta_1) = S(\alpha, \beta_2).$$

**Proof.** Let $\alpha$ be injectively presented.

(i): Let $\beta \in V(Y^*)$ such that $\bar{\beta} = \bar{\alpha}, \delta \in V(Y^*)$ and $i \in \bar{\alpha}$. Note that $S(\alpha, \bar{\beta}) = \bar{\alpha}$. Let $\mu := S(\alpha, \beta)$.

First assume $e \models \exists z \in \mu \ \text{OP}(\bar{\alpha}(i), \delta) = z$. Then $p_0 e = j$ for some $j \in \bar{\alpha}$ and $p_1 e \models \text{OP}(\bar{\alpha}(i), \delta) = \bar{\gamma}(j)$, so that $p_1 e \models \text{OP}(\bar{\alpha}(i), \delta) = (\bar{\alpha}(j), \bar{\beta}(j))_Y$. The latter yields

$$h^r(\alpha, \beta, \delta, i, e) \models \bar{\alpha}(i) = \bar{\alpha}(j) \land \delta = \bar{\beta}(j)$$

for some $E_\varphi$-recursive partial function $h^r$. Since $\alpha$ is injectively presented the latter implies $i = j$, and hence $p_1(h^r(\alpha, \beta, \delta, i, e)) \models \delta = \bar{\beta}(i)$.

Conversely assume $e' \models \delta = \bar{\beta}(i)$. Then

$$p(i, p(\text{id}_s(\bar{\alpha}(i)))) \models \bar{\alpha}(i) = \bar{\alpha}(i) \land \delta = \bar{\beta}(i).$$

The latter yields

$$h^+(\alpha, \beta, \delta, i, e') \models \text{OP}(\bar{\alpha}(i), \delta) = (\bar{\alpha}(i), \bar{\beta}(i))_Y,$$

for some $E_\varphi$-recursive partial function $h^+$, and hence

$$p(i, h^+(\alpha, \beta, \delta, i, e')) \models \exists z \in \mu \ \text{OP}(\bar{\alpha}(i), \delta) = z.$$

It is now obvious how to distill the desired function $\ell_0$ from $h^r$ and $h^+$.

Let $\gamma \in V(Y^*)$. If $e \models \gamma$ is a function with domain $\alpha$,

then by 4.20 (ii), $g_1(e, \alpha, \gamma) \in V(Y^*)$ and, letting $\delta := g_1(e, \alpha, \gamma)$, we have $\bar{\delta} = \bar{\alpha}$ and

$$g_2(e, \alpha, \gamma) \models S(\alpha, \delta) \subseteq \gamma.$$

It is now obvious how to define $\ell_1'$ and $\ell_2'$.

For the converse let $e' \models \gamma = S(\alpha, \beta)$, where $\beta \in V(Y^*)$ such that $\bar{\beta} = \bar{\alpha}$. By 4.20

$$g_0(\alpha, \beta) \models S(\alpha, \beta)$$

is a relation with domain $\alpha$ and range $\beta$.

Also, for $i, j \in \bar{\alpha}$, if $\models^* (\bar{\alpha}(i), \bar{\beta}(j))_Y \in \gamma$ then, owing to (i), $\models^* \bar{\alpha}(i) = \bar{\alpha}(j)$ since $\models^* (\bar{\alpha}(i), \bar{\beta}(i))_Y \in S(\alpha, \beta)$, so that $i = j$ because $\alpha$ is injectively presented. Hence, there is a $E_\varphi$-recursive partial function $\ell_3^*$ such that

$$\ell_3^*(\alpha, \beta, \gamma, e') \models \gamma$ is a function with domain $\alpha$.\]
(iii): Let \( \beta_1, \beta_2 \in \mathbf{V}(\mathbf{Y}^*) \) such that \( \overline{\beta_1} = \overline{\beta_2} = \alpha \). Then \( f \vdash_{\mathbf{S}} (\alpha, \beta_1) \subseteq \mathbf{S}(\alpha, \beta_2) \) is equivalent to
\[
\forall i \in \alpha \ f(i) \vdash \langle \tilde{\alpha}(i), \tilde{\beta}_1(i) \rangle \mathbf{V} \in \mathbf{S}(\alpha, \beta_2).
\]
By (ii) there is an \( E_\varnothing \)-recursive partial function \( h^s \) such that the latter implies
\[
h^s(\alpha, \beta_1, \beta_2, f, i) \vdash \tilde{\beta}_1(i) = \tilde{\beta}_2(i)
\]
for all \( i \in \alpha \). On the other hand, one easily constructs an \( E_\varnothing \)-recursive partial function \( h^q \) such that \( e i \vdash \tilde{\beta}_1(i) = \tilde{\beta}_2(i) \) for all \( i \in \alpha \) implies
\[
h^q(\alpha, \beta_1, \beta_2, e) \vdash \mathbf{S}(\alpha, \beta_1) \subseteq \mathbf{S}(\alpha, \beta_2).
\]
As the roles of \( \beta_1 \) and \( \beta_2 \) can be interchanged, by the above it is obvious how to construct the desired functions \( \ell_1^s \) and \( \ell_2^q \).

**Theorem 4.23.** There is an \( E_\varnothing \)-recursive partial function \( \iota \) such that whenever \( \alpha \in \mathbf{V}(\mathbf{Y}^*) \) is injectively presented,
\[
\iota(\alpha) \vdash \alpha \text{ is a base.}
\]

**Proof.** Let \( \alpha \in \mathbf{V}(\mathbf{Y}^*) \) be injectively presented and let \( \gamma \in \mathbf{V}(\mathbf{Y}^*) \) satisfy
\[
e \vdash \gamma \text{ is a relation with domain } \alpha.
\]
Then by 4.20 (ii), \( g_1(e, \alpha, \gamma) \in \mathbf{V}(\mathbf{Y}^*) \) and, letting \( \delta := g_1(e, \alpha, \gamma) \), we have \( \overline{\delta} = \overline{\alpha} \) and
\[
g_2(e, \alpha, \gamma) \vdash \mathbf{S}(\alpha, \delta) \subseteq \gamma.
\]
Let \( \eta := \mathbf{S}(\alpha, \delta) \). By 4.22 (ii) we also have
\[
\ell_3^s(\alpha, \beta, \eta, \mathbf{id}_s(\eta)) \vdash \eta \text{ is a function with domain } \alpha.
\]
Hence, in view of the foregoing we can compose a function \( \iota^* \) such that
\[
\iota^*(\alpha, \gamma, e) \vdash \exists f [\mathbf{Fun}(f) \land \text{dom}(f) = \alpha \land f \subseteq \gamma]
\]
so that
\[
\iota(\alpha) \vdash \alpha \text{ is a base},
\]
where \( \iota := \lambda xyz.\iota^*(x, y, z) \).

**Lemma 4.24.** \( \omega^* \) and \( h_\omega(n) \) for \( n \in \omega \) are injectively presented.

**Proof.** Recall that \( \omega^* = \sup(\omega, h_\omega) \).
Suppose \( \models^* h_\omega(n) = h_\omega(k) \) for some \( n, k \in \omega \). If \( n \in k \), then, by (25), \( \models^* h_\omega(n) \in h_\omega(k) \), so that \( \models^* h_\omega(n) \in h_\omega(n) \). But by (4.17) we have \( \models^* \forall x x \notin x \), and hence, as \( \not\models^* \perp \), we conclude \( n \notin k \). Likewise we can conclude \( k \notin n \). Since
\[
\forall x, y \in \omega (x \in y \lor x = y \lor y \in x),
\]
we arrive at \( n = k \).
Note that by (25), \( \overline{h_\omega(n)} = n \) and
\[
(\overline{h_\omega(n)})(k) = h_\omega(k)
\]
for \( k \in n \), so that from
\[
\models^* (\overline{(h_\omega(n)})(k) = (\overline{(h_\omega(n)})(k')
\]
for \( k, k' \in n \) we obtain \( h_\omega(k) = h_\omega(k') \), and thus \( k = k' \) by the foregoing. \( \square \)
**Lemma 4.25.** Let $\theta_0(x, \vec{u})$ and $\psi_0(x, y, \vec{u})$ be formulas of set theory. There is an $E^\omega_\omega$-recursive partial function $h^d$ such that whenever $\bar{\xi} \in V(Y^*)$ and

$$e \models \forall x (\theta_0(x, \bar{\xi}) \supset \exists y [\theta_0(y, \bar{\xi}) \land \psi_0(x, y, \bar{\xi})])$$

and

$$e' \models \theta_0(\alpha, \bar{\xi})$$

then

$$h^d(e, e', \xi, \alpha) \models \exists f \left( f \text{ is a function with domain } \omega \land f(0) = \alpha \land \forall n \in \omega [\theta_0(f(n), \bar{\xi}) \land \psi_0(f(n), f(n+1), \bar{\xi})] \right).$$

**Proof.** Let $\theta(x)$ and $\psi(x, y)$ be $\theta_0(x, \bar{\xi})$ and $\psi_0(x, y, \bar{\xi})$, respectively. Suppose that (36) and (37) hold true. For the sake of simplicity we shall also assume that $\theta(x)$ is an unbounded formula.

Let $a_i$ be $p_i a$ and $a_{ij}$ be $p_j(p_i a)$. By the recursion theorem we can construct an index $c$ such that

$$c(u, x, y, z) \simeq d_N(0, u, p_x(p_xy), z((c(p_N u, x, y, z))_0, (c(p_N u, x, y, z))_{11})).$$

For $n \in \omega$ put

$$g(n) := c(n, \alpha, e', e)$$

and $f(n) := (g(n))_0$. Note that $f(0) = \alpha$ and $(g(0))_{11} = e'$, so that by (37),

$$(g(0))_{11} \models \theta(f(0)).$$

By induction on $n > 0$ we shall show that $g(n) \downarrow$, $f(n) \downarrow$, and

$$g(n) \models \exists y [\theta(y) \land \psi(f(n-1), y)].$$

By (39),

$$g(1) \simeq e((c(0, \alpha, e', e))_0, (c(0, \alpha, e', e))_{11})$$

$$\simeq e((g(0))_0, (g(0))_{11})$$

$$\simeq e(f(0), (g(0))_{11})$$

so that by (41) and (36) we have $g(1) \downarrow$, $f(1) \downarrow$, and

$$g(1) \models \exists y [\theta(y) \land \psi(f(0), y)].$$

Now assume (41) to be true, where $n > 0$. By (39),

$$g(n + 1) \simeq e((c(n, \alpha, e', e))_0, (c(n, \alpha, e', e))_{11})$$

$$\simeq e((g(n))_0, (g(n))_{11})$$

$$\simeq e(f(n), (g(n))_{11})$$

so that by (38) and (37) we have

$$h^d(e, e', \xi, \alpha) \models \exists f \left( f \text{ is a function with domain } \omega \land f(0) = \alpha \land \forall n \in \omega [\theta_0(f(n), \bar{\xi}) \land \psi_0(f(n), f(n+1), \bar{\xi})] \right).$$
so that by (41) and (36) we have \( g(n + 1) \downarrow, f(n + 1) \downarrow \), and

\[
g(n + 1) \models \exists y \left[ \theta(y) \land \psi(f(n), y) \right].
\]

Hence by induction on \( n \), (41) holds. Note that (41) also entails that

\[
(g(n))_1 \models \theta(f(n)) \land \psi(f(n - 1), f(n))).
\]

Let \( \eta := \sup(\omega, f) \). Then \( \eta \in \text{V}(\text{Y}^*) \) with \( \tilde{\eta} = \omega \), so that if \( \delta = S(\omega^*, \eta) \) then \( \delta \in \text{V}(\text{Y}^*) \) and by 4.22 (ii), as \( \omega^* \) is injectively presented, we get

\[
|^{*} \delta \ \text{is a function with domain } \omega^*.
\]

As \( \tilde{\delta}(0) = \langle h_\omega(0), \tilde{\eta}(0) \rangle \rangle = \langle h_\omega(0), f(0) \rangle \rangle \) it follows that \( |^{*} \langle h_\omega(0), \alpha \rangle \rangle \rho \in \delta \). Finally, let \( k \in \omega \) and \( \gamma, \rho \in \text{V}(\text{Y}^*) \) such that \( |^{*} \langle h_\omega(k), \rho \rangle \rangle \rho \in \delta \) and \( |^{*} \langle h_\omega(k + 1), \gamma \rangle \rangle \rho \in \delta \). Then by 4.22 (i), \( |^{*} \rho = f(k) \) and \( |^{*} \gamma = f(k + 1) \) and hence, from (40) and (42),

\[
|^{*} \theta(\rho) \land \psi(\rho, \gamma).
\]

Note also that via its definition, \( \delta \) is \( E_\rho \)-computable from \( \alpha, e, e' \) and, moreover, all transformations involving "\( |^{*} \)" can be made \( E_\rho \)-computable, so that we can extract \( h^d \) from the above.

**Theorem 4.26.** \( \text{RDC} \) is validated in \( \text{V}(\text{Y}^*) \).

**Proof.** This follows from Lemma 4.25. □

The aim of the remainder of this section is to show that internally in \( \text{V}(\text{Y}^*) \), every \( \Pi \Sigma I \) -generated set is a base. In the following definitions we shall define \( \Sigma(\alpha, \beta) \in \text{V}(\text{Y}^*) \) and \( \Pi(\alpha, \beta) \in \text{V}(\text{Y}^*) \) for \( \alpha, \beta \in \text{V}(\text{Y}^*) \) such that \( \bar{\alpha} = \bar{\beta} \). We then show that when \( \alpha \) is injectively presented then in \( \text{V}(\text{Y}^*) \) these are the disjoint union and cartesian product, respectively, of the family of sets \( S(\alpha, \beta) \) indexed by \( \alpha \).

**Definition 4.27.** Let \( \alpha, \beta \in \text{V}(\text{Y}^*) \) with \( \bar{\alpha} = \bar{\beta} \). As \( \bar{\alpha} \in \text{Y}^* \) and \( \bar{\beta}(i) \in \text{Y}^* \) for \( i \in \bar{\alpha} \) it follows that \( C \in \text{Y}^* \), where

\[
C := \Sigma_{i \in \bar{\alpha}} \bar{\beta}(i).
\]

For \( z \in C \), \( p_0 z \in \bar{\alpha} \) so that \( f(z) \in \text{V}(\text{Y}^*) \) where

\[
f(z) := (\bar{\beta}(p_0 z))(p_1 z).
\]

Now define

\[
\Sigma(\alpha, \beta) := (\sup z \in C)\langle \bar{\alpha}(p_0 z), f(z) \rangle.
\]

Note that

\[
D := \Pi_{i \in \bar{\alpha}} \bar{\beta}(i)
\]

is in \( \text{Y}^* \). Let \( z \in D \). For \( i \in \bar{\alpha} \) we have \( \bar{\alpha}(i) \in \text{V}(\text{Y}^*) \), \( \bar{\beta}(i) \in \text{V}(\text{Y}^*) \) and \( z(i) \in \bar{\beta}(i) \) so that

\[
(\bar{\beta}(i))(z(i)) \in \text{V}(\text{Y}^*).\]

Hence \( g(z) \in \text{V}(\text{Y}^*) \) and \( g(z) = \bar{\alpha} \) where

\[
g(z) := (\sup i \in \bar{\alpha})(\bar{\beta}(i))(z(i)).
\]
Hence $S(\alpha, g(z)) \in V(Y^*)$. Now define
\[ \Pi(\alpha, \beta) := (\sup z \in D)S(\alpha, g(z)). \]

Then $\Pi(\alpha, \beta) \in V(Y^*)$.

**Lemma 4.28.** The partial functions $\alpha, \beta \mapsto \Sigma(\alpha, \beta)$ and $\alpha, \beta \mapsto \Pi(\alpha, \beta)$ are $E_\nu$-recursive partial functions. Also, whenever $\bar{\alpha} = \bar{\beta}$ then $\Sigma(\alpha, \beta) \downarrow$ and $\Pi(\alpha, \beta) \downarrow$.

**Proof.** This is obvious by their definition. ☐

**Proposition 4.29.** There are $E_\nu$-recursive partial functions $h_s, h_p$ such that whenever $\alpha$ is injectively presented and $\bar{\alpha} = \bar{\beta}$ then the following hold:
\[ h_s(\alpha, \beta) \vdash \Sigma(\alpha, \beta) \text{ is the disjoint union of the family of sets } S(\alpha, \beta). \tag{43} \]
\[ h_p(\alpha, \beta) \vdash \Pi(\alpha, \beta) \text{ is the cartesian product of the family of sets } S(\alpha, \beta). \tag{44} \]

Moreover, if $\tilde{\beta}(i)$ is injectively presented for all $i \in \bar{\alpha}$ then both $\Sigma(\alpha, \beta)$ and $\Pi(\alpha, \beta)$ are injectively presented.

**Proof.** By 4.22, $h^a(\alpha, \beta) \vdash S(\alpha, \beta)$ is a function with domain $\alpha$ for a $E_\nu$-recursive partial function $h^a$. Let $\eta \in V(Y^*)$. Suppose
\[ e \vdash \eta \text{ is in the disjoint union of } S(\alpha, \beta). \]

There is an $E_\nu$-recursive partial function $h_0$ such that the latter yields
\[ h_0(\alpha, \beta, \eta, e) \vdash \exists x \in \alpha \exists v \left[ \exists u \in S(\alpha, \beta) \langle x, v \rangle_V = u \land \exists y \in v OP(x, y) = \eta \right] \]
so that
\[ (h_0(\alpha, \beta, \eta, e))_1 \vdash \exists v \left[ \exists u \in S(\alpha, \beta) \langle \tilde{\alpha}(i), v \rangle_V = u \land \exists y \in v OP(\tilde{\alpha}(i), y) = \eta \right], \]
where $i = (h_0(\alpha, \beta, \eta, e))_0$. Hence by 4.22 (ii),
\[ h_1(\alpha, \beta, \eta, e) \vdash \delta = \tilde{\beta}(i) \land \exists y \in \delta OP(\tilde{\alpha}(i), y) = \eta, \]
where $\delta = (h_0(\alpha, \beta, \eta, e))_1$ and $h_1$ is an $E_\nu$-recursive partial function. Thus
\[ h_2(\alpha, \beta, \eta, e) \vdash \exists y \in \tilde{\beta}(i) OP(\tilde{\alpha}(i), y) = \eta, \]
with another $E_\nu$-recursive partial function $h_2$. The latter yields
\[ h_3(\alpha, \beta, \eta, e) \vdash \langle \tilde{\alpha}((z)_0), f(z) \rangle_V = \eta, \]
where $z \in C$ is defined by $z = \langle i, (h_2(\alpha, \beta, \eta, e))_0 \rangle$ and $h_3$ is yet another $E_\nu$-recursive partial function. Therefore
\[ h_4(\alpha, \beta, \eta, e) \vdash \eta \in \Sigma(\alpha, \beta) \]
for a final $E_\nu$-recursive partial function $h_4$.

Going in the other direction and starting from

$$e' \models \eta \in \Sigma(\alpha, \beta),$$

we can reverse the above implication so that we get

$$h'(\alpha, \beta, \eta, e') \models \eta \text{ is in the disjoint union of } S(\alpha, \beta)$$

for some $E_\nu$-recursive partial function $h'$. As a consequence of the above we then can compose an $E_\nu$-recursive partial function $h_s$ such that (43) holds.

For the cartesian product suppose

$$e \models \eta \text{ is in the cartesian product of the family of sets } S(\alpha, \beta). \quad (45)$$

Then in particular

$$h^*_0(\alpha, \beta, \eta) \models \eta \text{ is a function with domain } \alpha$$

for some $E_\nu$-recursive partial function $h^*_0$, and by 4.22 (ii) we get

$$\ell^*_1(\alpha, \eta, h^*_0(\alpha, \beta, \eta)) \models \eta = S(\alpha, \gamma) \quad (46)$$

with $\gamma = \ell^*_1(\alpha, \eta, h^*_0(\alpha, \beta, \eta))$ and $\bar{\gamma} = \bar{\alpha}$. Hence from (45) we obtain

$$h^*_1(\alpha, \beta, \eta, e) \models \eta = S(\alpha, \gamma) \land \theta(\alpha, \beta, \gamma) \quad (47)$$

for some $E_\nu$-recursive partial function $h^*_1$, where $\theta(\alpha, \beta, \gamma)$ stands for

$$\forall x \in \alpha \exists u \left[ \exists v \in S(\alpha, \beta) \, \text{OP}(x, u) = v \land \exists y \in u \exists v \in \Sigma(\alpha, \gamma) \, \text{OP}(x, u) = v \right].$$

Using 4.22 (i) twice, from (47) we obtain that for all $i \in \bar{\alpha}$,

$$h^*_2(\alpha, \beta, \eta, i) \models \exists y \in \bar{\beta}(i) \, y = \hat{\gamma}(i)$$

so that

$$h^*_3(\alpha, \beta, \eta, i) \models (\bar{\beta}(i))(j) = \hat{\gamma}(i), \quad (48)$$

where

$$j = \left( h^*_2(\alpha, \beta, \eta, i) \right)_0$$

and $h^*_2, h^*_3$ are appropriate $E_\nu$-recursive partial functions. Now let $z$ be the function with domain $\bar{\alpha}$ and

$$z(i) = \left( h^*_2(\alpha, \beta, \eta, i) \right)_0$$

so that $z \in D$. Then (48) yields

$$h^*_2(\alpha, \beta, \eta, i) \models (\bar{\beta}(i))(z(i)) = \hat{\gamma}(i), \quad (49)$$

whence, owing to the definition of $g(z)$,

$$h^*_2(\alpha, \beta, \eta) \models \Sigma(\alpha, g(z)) = \Sigma(\alpha, \gamma) \quad (50)$$
for an \( E_\nu \)-recursive partial function \( h_\xi^\xi \). It follows from (46) and (50) that

\[
h_\xi^\xi(\alpha, \beta, \eta, e) \models \eta \in \Pi(\alpha, \beta)
\]

for an \( E_\nu \)-recursive partial function \( h_\xi^\xi \).

By inspecting the implications leading from (45) to (51) one realizes that they can be reversed so that if

\[
e' \models \eta \in \Pi(\alpha, \beta)
\]

we can construct an \( E_\nu \)-recursive partial function \( h_\xi^\xi \) such that

\[
h_\xi^\xi(\alpha, \beta, \eta, e') \models \eta \text{ is in the cartesian product of the family of sets } S(\alpha, \beta).
\]

As a consequence of the above we can compose an \( E_\nu \)-recursive partial function \( h_p \) such that (44) holds.

Now suppose that \( \tilde{\beta}(i) \) is injectively presented for all \( i \in \bar{\alpha} \). Let \( \gamma := \Sigma(\alpha, \beta) \) and let \( z_1, z_2 \in \mathcal{C} \) such that \( e \models \gamma(z_1) = \gamma(z_2) \). We must show that \( z_1 = z_2 \). Let \( x_i := p_0 z_i \) and \( y_i := p_1 z_i \) for \( i = 1, 2 \). As \( e \models \gamma(z_1) = \gamma(z_2) \) it follows that

\[
l_0(\alpha, \beta, z_1, z_2, e) \models \langle \tilde{\alpha}(x_1), (\tilde{\beta}(x_1))(y_1) \rangle_V = \langle \tilde{\alpha}(x_2), (\tilde{\beta}(x_2))(y_2) \rangle_V
\]

so that

\[
l_1(\alpha, \beta, z_1, z_2, e) \models \tilde{\alpha}(x_1) = \tilde{\alpha}(x_2)
\]

\[
l_2(\alpha, \beta, z_1, z_2, e) \models (\tilde{\beta}(x_1))(y_1) = (\tilde{\beta}(x_2))(y_2)
\]

for some \( E_\nu \)-recursive partial functions \( l_0, l_1, l_2 \). As \( \alpha \) is injectively presented \( x_1 = x_2 \) and hence

\[
l_2(\alpha, \beta, z_1, z_2, e) \models (\tilde{\beta}(x_1))(y_1) = (\tilde{\beta}(x_2))(y_2).
\]

Whence as \( \tilde{\beta}(x_1) \) is injectively presented, \( y_1 = y_2 \), so that \( z_1 = z_2 \). Thus \( \Sigma(\alpha, \beta) \) is injectively presented.

Next let \( \delta := \Pi(\alpha, \beta) \) and let \( z_1, z_2 \in D \) such that \( e \models \tilde{\delta}(z_1) = \tilde{\delta}(z_2) \). We must show that \( z_1 = z_2 \). Note that

\[
\tilde{\delta}(z_i) = S(\alpha, g(z_i)) = (\sup i \in \bar{\alpha})\langle \tilde{\alpha}(i), (g(z_2))(i) \rangle_V.
\]

Hence, as

\[
l_3(\alpha, \beta, e, z_1, z_2) \models \tilde{\beta}(z_1) \subseteq \tilde{\beta}(z_1)
\]

for some \( E_\nu \)-recursive partial function \( l_3 \), we conclude that for all \( i \in \bar{\alpha} \),

\[
l_4(\alpha, \beta, z_1, z_2, e, i) \models \langle \tilde{\alpha}(i), (g(z_1))(i) \rangle_V = \langle \tilde{\alpha}(j), (g(z_2))(j) \rangle_V
\]

where \( j := \langle \langle l_3(\alpha, \beta, e, z_1, z_2) \rangle \rangle_0 \) and \( l_4 \) is an \( E_\nu \)-recursive partial function. But if \( e' \models \tilde{\alpha}(i) = \tilde{\alpha}(j) \) for some \( e' \) then \( i = j \), so that if also \( e'' \models (g(z_1))(i) = (g(z_2))(j) \) for some \( e'' \) then

\[
e'' \models (\tilde{\beta}(i))(z_1(i)) = (\tilde{\beta}(i))(z_2(i)),
\]

and as \( \tilde{\beta}(i) \) is injectively presented, \( z_1(i) = z_2(i) \). Hence it follows from (53) that \( z_1(i) = z_2(i) \) for all \( i \in \bar{\alpha} \), so that \( z_1 = z_2 \).
Lemma 4.30. For $A \in Y^*$ and $a, b \in A$, $\bar{I}(A, a, b) \in V(Y^*)$, where

$$\bar{I}(A, a, b) := (\sup z \in I(A, a, b)) h_\omega(0),$$

and

1. $\bar{I}(A, a, b)$ is injectively presented,

2. there is an $E_\varphi$-recursive partial function $\ell_I$ such that if $\alpha \in V(Y^*)$ is injectively presented and $\bar{\alpha} = A$ then for all $\eta \in V(Y^*)$,

$$\ell_I(\alpha, \eta, a, b) \vdash \eta \in \bar{I}(A, a, b) \text{ iff } [\eta = h_\omega(0) \land \bar{\alpha}(a) = \bar{\alpha}(b)].$$

Proof. As $I(A, a, b) \in Y^*$ and $h_\omega(0) \in V(Y^*)$ it follows that $\bar{I}(A, a, b) \in V(Y^*)$.

(i): Let $\gamma := \bar{I}(A, a, b)$ and let $z_1, z_2 \in I(A, a, b)$ such that $\bar{\gamma}(z_1) = \bar{\gamma}(z_2)$. We must show that $z_1 = z_2$. But, as $z_1, z_2 \in I(A, a, b)$, $z_1 = 0 = z_2$ so that $z_1 = z_2$.

(ii): As $\alpha$ is injectively presented and $a, b \in A = \bar{\alpha}$, we have $\models^* \bar{\alpha}(a) = \bar{\alpha}(b)$ if, and only if $a = b$.

Let $\eta \in V(Y^*)$. Suppose $e \models \eta \in \bar{I}(A, a, b)$. Then $(e)_1 \models \eta = h_\omega(0)$ and $a = b$, yielding

$$(\langle e \rangle_1, \text{id}_a(\bar{\alpha}(a))) \models \eta = h_\omega(0) \land \bar{\alpha}(a) = \bar{\alpha}(b).$$

Conversely, if $e \models \eta = h_\omega(0) \land \bar{\alpha}(a) = \bar{\alpha}(b)$, then $(e)_0 \models \eta = h_\omega(0)$ and $a = b$, so that $0 \in I(A, a, b)$ and whence $(0, (e)_0) \models \eta \in \bar{I}(A, a, b)$.

In view of the above it is clear how to construct $\ell_I$. □

Theorem 4.31. $\Pi\Sigma - AC$ is validated in $V(Y^*)$.

Proof. Note that the $\Pi\Sigma$-generated sets are an inductively defined class $I(\Phi)$. By lemma 2.7 there is a formula $\theta(u, v)$ of set theory such that $I(\Phi) = \{x \mid \exists u \theta(u, x)\}$ and, moreover,

$$CZF \vdash \forall u \forall x [\theta(u, x) \text{ iff } \bigvee_{i=1}^{5} \theta_i(u, x)],$$

where

$$\begin{align*}
\theta_1(u, x) &:= x \in \omega \\
\theta_2(u, x) &:= x = \omega \\
\theta_3(u, x) &:= \exists v \in u \exists a(\theta(v, a) \land \exists f \left[\text{Fun}(f) \land \text{dom}(f) = a \land \forall i \in a \exists z \in u \theta(z, f(i))\right] \\
&\land \ x \text{ is the disjoint union of } f \\
\theta_4(u, x) &:= \exists v \in u \exists a(\theta(v, a) \land \exists f \left[\text{Fun}(f) \land \text{dom}(f) = a \land \forall i \in a \exists z \in u \theta(z, f(i))\right] \\
&\land \ x \text{ is the cartesian product of } f \\
\theta_5(u, x) &:= \exists v \in u \exists a(\theta(v, a) \land \exists w \in a \exists w' \in a \forall z \left[\exists x \text{ iff } (z = 0 \land w = w')\right]).
\end{align*}$$

We shall show that there are $E_\varphi$-recursive partial functions $\ell^a$ and $\ell^#$ such that whenever $\xi, \alpha \in V(Y^*)$ and

$$e \models \theta(\xi, \alpha)$$

and
The Formulae-as-Classes Interpretation of Constructive Set Theory

then

\[ \ell^k(e, \xi, \alpha) \in \mathbb{V}(Y^*), \]  
\[ \ell^k(e, \xi, \alpha) \text{ is injectively presented,} \]  
\[ \ell^\#(e, \xi, \alpha) \vdash \ell^k(e, \xi, \alpha) = \alpha. \]  

\( \ell^k(e, \xi, \alpha) \) and \( \ell^\#(e, \xi, \alpha) \) will be defined by recursion on \( \xi \) so that, ultimately, the \( E^\varphi \)-recursive partial functions \( \ell^\# \) and \( \ell^k \) will have to be defined by invoking the recursion theorem 4.7. Note that (54) and Theorem 4.17 entail that there are \( E^\varphi \)-recursive partial functions \( g_1, g_2, g_3, g_4, g_5 \) such that \( e \vdash \theta(\xi, \alpha) \) implies

\[ (g_i(e, \xi, \alpha))_0 = 0 \lor (g_i(e, \xi, \alpha))_0 = 1 \]  
\[ (g_i(e, \xi, \alpha))_0 = 0 \Rightarrow (g_i(e, \xi, \alpha))_1 \vdash \theta_i(\xi, \alpha) \]  
for \( i = 1, 2, 3, 4, 5. \)

Now assume that \( \alpha, \xi \in \mathbb{V}(Y^*), e \vdash \theta(\xi, \alpha) \) and that for all \( \beta \in \mathbb{V}(Y^*), j \in \overline{\xi} \) and \( c \), if \( c \vdash \theta(\xi(j), \beta) \), then \( \ell^\#(e', \xi(j), \beta) \downarrow \) and \( \ell^k(e', \xi(j), \beta) \downarrow \) and that the properties pertaining to (55) obtain. Owing to (56) we can proceed by division into cases. Let

\[ g_{ij}(e, \xi, \alpha) := (g_i(e, \xi, \alpha))_j. \]

**Case 1:** Let \( g_{10}(e, \xi, \alpha) = 0 \). Then \( g_{11}(e, \xi, \alpha) \vdash \alpha \in \omega \), so that by the proof of 4.17,

\[ (\Upsilon_1(g_{11}(e, \xi, \alpha), \alpha))_1 \vdash \alpha = h_\omega(n) \]

for some \( E^\varphi \)-recursive partial function \( \Upsilon_1 \), where

\[ n := (\Upsilon_1(g_{11}(e, \xi, \alpha), \alpha))_0 \in \omega. \]

Hence we let \( \ell^k(e, \xi, \alpha) := h_\omega(n) \) and

\[ \ell^\#(e, \xi, \alpha) := (\Upsilon_1(g_{11}(e, \xi, \alpha), \alpha))_1. \]

It follows from Lemma 4.24 that \( \ell^k(e, \xi, \alpha) \) is injectively presented.

**Case 2:** Suppose \( g_{10}(e, \xi, \alpha) = 1 \) and \( g_{20}(e, \xi, \alpha) = 0 \). Then \( g_{21}(e, \xi, \alpha) \vdash \alpha \in \omega \), so that by the proof of 4.17,

\[ \Upsilon_2(g_{21}(e, \xi, \alpha), \alpha) \vdash \alpha = \omega^* \]

for some \( E^\varphi \)-recursive partial function \( \Upsilon_2 \). This time let

\[ \ell^k(e, \xi, \alpha) := \omega^*, \]
\[ \ell^\#(e, \xi, \alpha) := \Upsilon_2(g_{21}(e, \xi, \alpha), \alpha). \]

It follows from Lemma 4.24 that \( \ell^k(e, \xi, \alpha) \) is injectively presented.

**Case 3:** Suppose \( g_{10}(e, \xi, \alpha) = g_{20}(e, \xi, \alpha) = 1 \) and \( g_{30}(e, \xi, \alpha) = 0 \). Then \( g_{31}(e, \xi, \alpha) \vdash
\( \theta_3(\xi, \alpha) \). Unpacking the definition of \( g_{31}(e, \xi, \alpha) \vdash \theta_3(\xi, \alpha) \), we can \( E_\psi \)-computably extract from \( e, \xi, \alpha \) sets \( \gamma, \beta \in V(Y^*) \) and \( j \in \xi \) such that

\[
Y_3^*(e, \xi, \alpha) \vdash \theta(\xi(j), \gamma) \land \text{Fun}(\beta) \land \text{dom}(\beta) = \gamma \\
\land \forall y \in \gamma \exists z \in \xi \exists w \in \beta \exists w'[\text{OP}(y, w') = w \land \theta(z, w)] \\
\land \alpha \text{ is the disjoint union of } \beta.
\]

for some \( E_\psi \)-recursive partial function \( Y_3^* \). From the first conjunct above we get

\[
(\text{Fun}(\beta_3(e, \xi, \alpha)_0) \vdash \theta(\ldots), \gamma).
\]

From (57) we also get

\[
(\text{Fun}(\beta_3(e, \xi, \alpha)_0) \vdash \forall w \in \beta \exists z \in \xi \theta(z, w),
\]

and hence we can \( E_\psi \)-computably extract a function \( l' \) with domain \( \bar{\beta} \) from \( e, \xi, \alpha \) such that for all \( i \in \beta \) we have \( l'(i) \in \xi \) and

\[
Y_3^1(e, \xi, \alpha, i) \vdash \theta(\ldots), \bar{\beta}(i)
\]

for some \( E_\psi \)-recursive partial function \( Y_3^1 \). Put

\[
\rho \ := \ \ell^k((Y_3^*(e, \xi, \alpha))_0, \xi(j), \gamma) \\
\tau_i \ := \ \ell^k(Y_3^1(e, \xi, \alpha, i), \xi(l'(i)), \bar{\beta}(i))
\]

for \( i \in \tilde{\xi} \). By the inductive hypotheses we get \( \rho \downarrow, \rho \in V(Y^*) \), \( \rho \) is injectively presented, \( \tau_i \downarrow, \tau_i \in V(Y^*) \), \( \tau_i \) is injectively presented and

\[
\ell^k((Y_3^*(e, \xi, \alpha))_0, \xi(j), \gamma) \vdash \rho = \gamma \\
\ell^k(Y_3^1(e, \xi, \alpha, i), \xi(l'(i)), \bar{\beta}(i)) \vdash \tau_i = \bar{\beta}(i)
\]

for all \( i \in \tilde{\beta} \). Put

\[
\tau := (\sup i \in \tilde{\beta}) \tau_i \\
e_\# := \ell^k((Y_3^*(e, \xi, \alpha))_0, \xi(j), \gamma) \\
f_\# := \text{ab}(\bar{\beta}, \lambda x. \ell^k(Y_3^1(e, \xi, \alpha, i), \xi(l'(i)), \bar{\beta}(i))).
\]

Then \( \tau \in V(Y^*) \). From (57), (61) and (62) we can conclude that

\[
Y_3^{**}(e_\#, f_\#, \xi, \alpha) \vdash \text{Fun}(\tau) \land \text{dom}(\tau) = \rho \\
\land \alpha \text{ is the disjoint union of } \tau
\]

for some \( E_\psi \)-recursive partial function \( Y_3^{**} \). Thus, by 4.22 we can \( E_\psi \)-computably extract \( \delta \in V(Y^*) \) from \( Y_3^{**}(e_\#, f_\#, \xi, \alpha) \), \( \rho, \tau \) such that \( \delta = \bar{\rho} \) and

\[
Y_3^{++}(Y_3^{**}(e_\#, f_\#, \xi, \alpha), \rho, \tau) \vdash \tau = \text{S}(\rho, \delta)
\]

for some \( E_\psi \)-recursive partial function \( Y_3^{++} \). Thence, from (63), (64) and Proposition 4.29 we get

\[
Y_3^3(Y_3^{**}(e_\#, f_\#, \xi, \alpha), \rho, \tau) \vdash \Sigma(\rho, \delta) = \alpha
\]
and also that \( \Sigma(\rho, \delta) \) is injectively presented. As a result, we define
\[
\ell^c(e, \xi, \alpha) := \Sigma(\rho, \delta) \\
\ell^+(e, \xi, \alpha) := \mathcal{Y}_3(\mathcal{Y}_3^*(e^\#, f^\#, e, \xi, \alpha), \rho, \tau).
\]

**Case 4:** Suppose \( g_{10}(e, \xi, \alpha) = g_{20}(e, \xi, \alpha) = g_{30}(e, \xi, \alpha) = 1 \) and \( g_{40}(e, \xi, \alpha) = 0 \). Then \( g_{41}(e, \xi, \alpha) \models \theta_4(\xi, \alpha) \). Here we proceed in the same vein as in the previous case, crucially utilizing 4.22 (ii) and 4.29.

**Case 5:** Suppose \( g_{10}(e, \xi, \alpha) = g_{20}(e, \xi, \alpha) = g_{30}(e, \xi, \alpha) = g_{40}(e, \xi, \alpha) = 1 \) and \( g_{50}(e, \xi, \alpha) = 0 \). Then \( g_{51}(e, \xi, \alpha) \models \theta_5(\xi, \alpha) \). Unpacking the definition of \( g_{51}(e, \xi, \alpha) \models \theta_5(\xi, \alpha) \), we can \( E_\psi \)-computably extract from \( e, \xi, \alpha \) sets \( \beta \in \mathcal{V}(Y^*) \), \( j \in \xi \) and \( i_0, i_1 \in \beta \) such that
\[
\mathcal{Y}_5^*(e, \xi, \alpha) \models \theta(\xi(j), \beta) \land \forall z(z \in \alpha \iff z = h_\omega(0) \land \beta(i_0) = \beta(i_1)).
\]
for some \( E_\psi \)-recursive partial function \( \mathcal{Y}_5^* \). From the first conjunct above we get
\[
(\mathcal{Y}_5^*(e, \xi, \alpha))_0 \models \theta(\xi(j), \beta).
\]
Put
\[
\rho := \ell^c((\mathcal{Y}_5^*(e, \xi, \alpha))_0, \xi(j), \beta).
\]
By the inductive hypotheses we get that \( \rho \upharpoonright, \rho \in \mathcal{V}(Y^*) \), and that \( \rho \) is injectively presented and
\[
\ell^+(((\mathcal{Y}_5^*(e, \xi, \alpha))_0, \xi(j), \gamma) \models \rho = \beta.
\]
Put
\[
e^\# := \ell^+(((\mathcal{Y}_5^*(e, \xi, \alpha))_0, \xi(j), \beta).
\]
From (66) and (68) we can extract \( j_0, j_1 \in \rho \) such that
\[
\mathcal{Y}_5^{*+}(e^\#, e, \xi, \alpha) \models \forall z(z \in \alpha \iff z = h_\omega(0) \land \rho(j_0) = \rho(j_1)).
\]
for some \( E_\psi \)-recursive partial function \( \mathcal{Y}_5^{*+} \). Thus, by 4.30 we get that
\[
\mathcal{Y}_5^{*++}(\mathcal{Y}_5^{*+}(e^\#, e, \xi, \alpha), \rho) \models \alpha = \mathfrak{I}(\rho, j_0, j_1)
\]
for some \( E_\psi \)-recursive partial function \( \mathcal{Y}_5^{*++} \), and, moreover, that \( \mathfrak{I}(\rho, j_0, j_1) \) is injectively presented. As a result, we define
\[
\ell^c(e, \xi, \alpha) := \mathfrak{I}(\rho, j_0, j_1) \\
\ell^+(e, \xi, \alpha) := \mathcal{Y}_5^{*++}(\mathcal{Y}_3^*(e^\#, e, \xi, \alpha), \rho).
\]
Finally, by combining (55) and Theorem 4.23, we can design an \( E_\psi \)-recursive partial function \( \Psi \) such that if
\[
e \models \exists z \theta(z, \alpha)
\]
then
\[
\Psi(\alpha, c) \models \alpha \text{ is a base},
\]
showing that \( \Pi \Sigma I \land AC \) is validated in \( \mathcal{V}(Y^*) \). \( \square \)
Remark 4.32. Aczel uses the notion of a strong base to show that $\Pi \Sigma I - AC$ is validated in Martin-Löf type theory. $\alpha \in V(Y^*)$ is said to be a strong base if $\models^* \alpha = \beta$ for some injectively presented $\beta \in V(Y^*)$. The proof of 4.31 does not use this notion. The main reason for its avoidance is that I could not see how to internally characterize the notion of a strong base in $V(Y^*)$, that is, via a formula $\psi(u)$ such that $\models^* \psi(\alpha)$ would hold iff $\alpha$ were a strong base. As Aczel also invokes the axiom of choice in type theory in connection with propositions involving the notion of a strong base there was a danger that dealing with it would require some forms of the axiom of choice to hold in the background theory.

4.5 The formulae-as-classes interpretation for CZF + REA

As the reader may expect, the formulae-as-classes interpretation given for CZF above can be extended to CZF + REA also. The first step is to add the following condition to the definition of $E_\nu$-recursive functions, giving rise to the $E_\nu^w$-recursive functions:

$$\{\bar{w}\}(x,g) = W_{z \in x} g(z) \text{ if } g \text{ is a (set-)function with } \text{dom}(g) = x,$$

where $\bar{w}$ is a “fresh” natural number.

One then defines for every formula $\theta \in L \in V(Y_w^*)$-assignment $M$, a class $\| \theta \|_M$ as in Definition 4.10, where, however, the definition of the product

$$\prod_{x \in A} B_x := \{a | \forall x \in A(\{a\}(x) \in B_x)\} \tag{71}$$

is to be understood in the sense of $E_\nu^w$-recursive functions. Correspondingly, we obtain the following result.

Theorem 4.33. Let $\theta(u_1, \ldots, u_r)$ be a formula of $L \in V(Y^*)$ all of whose free variables are among $u_1, \ldots, u_r$. If

$$\text{CZF + REA + RDC + } \Pi \Sigma W - AC \vdash \theta(u_1, \ldots, u_r),$$

then one can effectively construct an index of a $E_\nu^w$-recursive partial function $g$ such that

$$\text{CZF}_{Exp} + \text{REA} \vdash \forall \alpha_1, \ldots, \alpha_r \in V(Y^*) \ g(\alpha_1, \ldots, \alpha_r) \in [\prod \theta(\alpha_1, \ldots, \alpha_r)].$$

Proof. The proof builds on the proof of Theorem 4.13. First we have to deal with REA. Since CZF proves that every set is a subset of a transitive set it suffices to show that there is a term $t_{rea}$ such that if $\alpha \in V(Y^*)$ and $e \models \text{Tran}(\alpha)$ then

$$t_{rea}(\alpha, e) \models \exists x (\alpha \subseteq x \land x \text{ is regular}).$$

So suppose $e \models \text{Tran}(\alpha)$. Define a function $f_\alpha$ with domain $\bar{\alpha}$ by $f_\alpha(i) := \bar{\alpha}(i)$ and put

$$A := W_{i \in \bar{\alpha}} f_\alpha(i).$$

Note that $A$ is $E_\nu^w$-computable from $\alpha$ and that $A \in Y_w^*$. By the recursion theorem we can construct an index $c$ such that

$$c(x, y, z) \simeq p(\bar{f}(x, y), \bar{a}b(\bar{f}(x, y), \lambda u. c(x, y, \bar{f}(z, u)))) \tag{72}.$$
Note that the elements of \( A \) are of the form \( \sup(i, g) \) where \( i \in \bar{\alpha} \) and \( g : f_{\alpha}(i) \to A \). By \( \in \)-induction on the elements \( \sup(i, g) \) of \( A \) one shows that
\[
c(f_{\alpha}, i, g) \simeq \sup(f_{\alpha}(i), \bar{\alpha}b(f_{\alpha}(i), \lambda u. c(f_{\alpha}(i), i, g(u)))) \tag{73}
\]
so that \( c(f_{\alpha}, i, g) \downarrow \) and, moreover, \( c(f_{\alpha}, i, g) \in V(Y_\alpha^\ast) \). Now put
\[
h_{\alpha} := \bar{\alpha}b(A, \lambda x. c(f_{\alpha}, p_0x, p_1x))
\]
and let
\[
\beta := \sup(A, h_{\alpha}).
\]
Then \( \beta \in V(Y_\alpha^\ast) \) and by the above, \( \beta \) is \( E_\alpha^\ast \)-computable from \( \alpha \).
In what follows, we shall frequently suppress the witnessing information and write \( \models^* \theta(\tilde{\alpha}) \) rather than \( e \models^* \theta(\tilde{\alpha}) \) for a specific \( e \). In all cases the witnessing information could be supplied and is indeed \( E_\alpha^\ast \)-computable from the exhibited parameters. Having progressed thus far in the paper, we consider it unnecessary and too tedious to make witnesses explicit all of the time.
First we prove that
\[
\models^* \beta \text{ is transitive.} \tag{74}
\]
So assume \( \models^* \gamma \in \beta \). Then \( \models^* \gamma = h_{\alpha}(j) \) for some \( j \in A \). But \( j = \sup(i, g) \) for some \( i \in \bar{\alpha} \) and \( g : f_{\alpha}(i) \to A \). Hence, letting \( \delta := \sup(f_{\alpha}(i), g') \) and \( g' \) be the function with domain \( f_{\alpha}(i) \) satisfying \( g'(u) = h_{\alpha}(g(u)) \), we get \( \models^* \delta \subseteq \beta \) and \( \models^* \gamma = \delta \), whence \( \models^* \gamma \subseteq \beta \).
To demonstrate that \( \beta \) is regular we further need to show that if \( \models^* \gamma \in \beta \) and \( \models^* \forall u \in \gamma \exists v \in \beta \theta(u, v) \), then
\[
\models^* \exists x \in \beta \theta'(x, \gamma), \tag{75}
\]
where \( \theta'(x, \gamma) \) stands for
\[
\forall u \in \gamma \exists v \in x \theta(u, v) \land \forall v \in x \exists u \in \gamma \theta(u, v).
\]
So assume \( \models^* \gamma \in \beta \) and \( \models^* \forall u \in \gamma \exists v \in \beta \theta(u, v) \). Then \( \models^* \gamma = h_{\alpha}(j) \) for some \( j \in A \) and
\[
\ell \models^* \forall u \in h_{\alpha}(j) \exists v \in \beta \theta(u, v) \tag{76}
\]
for a function with domain \( \tilde{h}_{\alpha}(j) \). Then \( j = \sup(i, g) \) for some \( i \in \bar{\alpha} \) and \( g : f_{\alpha}(i) \to A \). Also \( \tilde{h}_{\alpha}(j) = f_{\alpha}(i) \). From (76) we conclude that
\[
\forall \nu \in f_{\alpha}(i) [p_0(\ell(\nu)) \in A \land p_1(\ell(\nu)) \models \theta(\tilde{h}_{\alpha}(j)(\nu), h_{\alpha}(p_0(\ell(\nu))))].
\]
Now let the function \( \sigma \) with domain \( f_{\alpha}(i) \) be defined by \( \sigma(\nu) = p_0(\ell(\nu)) \). So \( \sup(i, \sigma) \in A \) and if
\[
\delta := h_{\alpha}(\sup(i, \sigma))
\]
then \( \models^* \delta \in \beta \). Moreover, we have
\[
\forall \nu \in \delta p_1(\ell(\nu)) \models \theta(\tilde{h}_{\alpha}(j)(\nu), h_{\alpha}(\sigma(\nu))].
\]
From the latter we obtain \( \models^* \theta'(\delta, h_\alpha(j)) \) and hence \( \models^* \theta'(\delta, \gamma) \), so that \( \models^* \exists x \in \beta \theta'(x, \gamma) \).

It remains to prove that

\[ \models^* \alpha \subseteq \beta. \]

By induction on \( \eta \in V(Y^*) \) we shall show that \( \models^* \eta \in \alpha \supset \eta \in \beta \). So as induction hypothesis we assume that

\[ \models^* \forall u \in \eta \left( u \in \alpha \supset u \in \beta \right). \]

Now if \( \eta \in \alpha \) then \( \models^* \eta = \tilde{\alpha}(i) \) for some \( i \in \tilde{\alpha} \). Hence \( \models^* \forall u \in \tilde{\alpha}(i) \ u \in \beta \). The latter implies that we have a function \( \ell \) with domain \( f_\alpha(i) = \overline{\alpha}(i) \) such that \( \ell : f_\alpha(i) \rightarrow A \) and for all \( j \in f_\alpha(i) \)

\[ \models^* (\tilde{\alpha}(i))(j) = h_\alpha(\ell(j)). \]

Then \( \sup(i, \ell) \in A \). Let \( \delta = h_\alpha(\sup(i, \ell)) \). Then \( \models^* \delta \in \beta \) and for all \( j \in f_\alpha(i) \equiv \tilde{\delta}, \)

\[ \models^* (\tilde{\alpha}(i))(j) = \tilde{\delta}(j). \]

The latter entails \( \models^* \tilde{\alpha}(i) = \delta \), whence \( \models^* \eta = \delta \), so that \( \models^* \eta \in \beta \). This completes the proof that \( V(Y^*) \) validates REA.

Finally we turn to \( \Pi\Sigma WI - AC \). The proof of the validity of \( \Pi\Sigma I - AC \) was carried out in Theorem 4.31. Here we shall only discuss the additional constructions that are needed to accommodate the \( W \)-operation. In Definition 4.27, \( \Sigma(\alpha, \beta) \in V(Y^*_w) \) and \( \Pi(\alpha, \beta) \in V(Y^*_w) \) are defined for \( \alpha, \beta \in V(Y^*_w) \) such that \( \tilde{\alpha} = \tilde{\beta} \), and 4.30 they are related to the disjoint union and cartesian product operations when \( \alpha \) is injectively presented and \( \tilde{\beta}(i) \) is injectively presented for each \( i \in \tilde{\alpha} \). Moreover, \( \Sigma(\alpha, \beta) \) and \( \Pi(\alpha, \beta) \) are shown to be injectively presented under these conditions. Aczel has carried a similar construction for the operation \( W \).

Let \( \alpha, \beta \in V(Y^*_w) \) such that \( \tilde{\alpha} = \tilde{\beta} \). First note that \( A \in V(Y^*_w) \), where

\[ A = W_{i \in \tilde{\alpha}}(\tilde{\beta}(i)). \]

Define \( h_{\alpha, \beta} : A \rightarrow V(Y^*_w) \) by transfinite recursion on \( A \) so that for \( i \in \tilde{\alpha} \) and \( f : \tilde{\beta}(i) \rightarrow A \),

\[ h_{\alpha, \beta}(\sup(i, f)) := \langle \tilde{\alpha}(i), S(\tilde{\beta}(i), (\sup j \in \tilde{\beta}(i))h_{\alpha, \beta}(f(j)))_V \rangle. \]

Finally let

\[ W(\alpha, \beta) := \sup(A, h_{\alpha, \beta}). \]

\( A \) is obviously \( E^w_p \)-computable from \( \alpha, \beta \). Similarly as in (72) one can show with the aid of the recursion theorem that \( \alpha, \beta \mapsto h_{\alpha, \beta} \) is \( E^w_p \)-computable form \( \alpha, \beta \) so that \( W(\alpha, \beta) \) is \( E^w_p \)-computable from \( \alpha, \beta \) also.

Under the above conditions, if \( \alpha \) is injectively presented and \( \tilde{\beta}(i) \) is injectively presented for all \( i \in \tilde{\alpha} \), it can be shown that \( W(\alpha, \beta) \) is injectively presented, too. The proof consists in adapting the proof of [3] A2.2 to the present context. Furthermore, by using \( W(\alpha, \beta) \) and the techniques of the proof of Theorem 4.31 one can show that \( V(Y^*_w) \) validates \( \Pi\Sigma WI - AC \). Due to space limitations a proof with all the details cannot be included in this paper. However, as I trust that the reader has by now amassed enough experience from earlier proofs in this paper as to how to do things in an \( E^w_p \)-computable way, he or she can glean the details from the proof of [3] A2.2. \[ \square \]
References


