

**ON NON-WELLFOUNDED
CONSTRUCTIVE SET THEORY**

S. TUPAILO

REPORT No. 37, 2000/2001

ISSN 1103-467X

ISRN IML-R- -37-00/01- -SE



INSTITUT MITTAG-LEFFLER
THE ROYAL SWEDISH ACADEMY OF SCIENCES

On Non-wellfounded Constructive Set Theory

Construction of non-wellfounded sets in Explicit Mathematics

Sergei Tupailo

We analyze the proof-theoretic strength of Constructive Set Theory without Foundation, NCZF^- , with natural numbers as urelements. The upper bounds are established by a realizability interpretation into Explicit Mathematics, which uses the same method as (14). An important feature is building bisimulation between sets “in stages”, in a way similar to (10). As a corollary we obtain that $|\text{NCZF}^-|$ is bounded by $\varphi(\varepsilon_0, 0)$.

1 Constructive Set Theory with Natural Numbers

The language $\mathcal{L}_{\in, \mathbb{N}}$ of Set Theory with natural numbers as urelements is two-sorted with **number variables** a, b, c, \dots , **set variables** A, B, C, \dots , and two **predicate constants** $=$ and \in . $=$ and \in accept arguments of either sort, that is, we are allowed to write $a = b$, $a = B$, $a \in B$, $B \in a$, etc. Additionally we have the following **function constants** which act from numbers to numbers: zero 0 , successor $'$, as well as countably many f_1, f_2, \dots for primitive recursive functions. Number **terms** are built from these in the standard way. The set of free variables of a formula F is denoted by $\text{FV}(F)$, and by $\text{FV}_0(F)$ and $\text{FV}_1(F)$ we denote correspondingly the sets of free number and set variables of F .

In free occurrences, we will use o, p, q, r as metavariables for both sorts. In bound occurrences,

* Research supported by the Swiss National Science Foundation and Estonian Science Foundation. The final version of this paper was prepared during the author's stay at the Institut Mittag-Leffler, Stockholm, Sweden.

I am grateful to the anonymous referee for careful reading of the paper.

Submitted for
Proceedings of LLC9

Ingrid van Loon, Grigori Mints, Reinhard Muskens (eds.)

Copyright ©2000, CSLI Publications

$\forall pF[p]$ will denote $\forall X F[X] \wedge \forall x F[x]$, and
 $\exists pF[p]$ will denote $\exists X F[X] \vee \exists x F[x]$.

As usual, $\forall X \in AF[X]$ and $\exists X \in AF[X]$ stand for $\forall X (X \in A \rightarrow F[X])$ and $\exists X (X \in A \wedge F[X])$, resp., and similarly for $\forall x \in AF[x]$ and $\exists x \in AF[x]$. By F^A we denote the result of replacing each quantifier QX in F by $QX \in A$. *Bounded* (or Δ_0) formulas of $\mathcal{L}_{\in, \mathbb{N}}$ are those built from atoms by means of $\wedge, \vee, \rightarrow, \forall x, \exists x, \forall X \in A$ and $\exists X \in A$.

The logic is *intuitionistic two-sorted with equality*. We take \perp (falsity) as a propositional constant with standard axioms pertaining to it.

The axioms of *Constructive Set Theory with natural numbers NCZF* are of three groups: *ontological ones, axioms for natural numbers and axioms for sets*.

Ontological axioms just express the basic features of the set universe with urelements: sets and urelements are different things, and urelements cannot contain anything. These axioms are the following:

$$\begin{aligned} a = A &\leftrightarrow \perp; \\ A = a &\leftrightarrow \perp; \\ a \in b &\leftrightarrow \perp; \\ A \in b &\leftrightarrow \perp. \end{aligned}$$

Arithmetical axioms are the standard axioms of Heyting's arithmetic, where *Induction schema* is taken in the form $G[0] \wedge \forall x (G[x] \rightarrow G[x']) \rightarrow \forall x G[x]$, for all formulas $G[x]$.

Set axioms are the following:

Extensionality:

$$\forall X \forall Y (\forall p \in X (p \in Y) \wedge \forall p \in Y (p \in X) \rightarrow X = Y);$$

Foundation:

$$\forall p (\forall q \in p G[q] \rightarrow G[p]) \rightarrow \forall p G[p], \text{ for all formulas } G[p];$$

Pair:

$$\forall p \forall q \exists Z (p \in Z \wedge q \in Z);$$

Union:

$$\forall X \exists Y \forall p \in X \forall q \in p (q \in Y);$$

Infinity:

$$\exists X \forall x (x \in X);$$

Δ_0 *Separation:*

$$\forall X \exists Y \forall p (p \in Y \leftrightarrow p \in X \wedge F[p]), \text{ for all } \Delta_0 \text{ formulas } F[p];$$

Strong Collection:

$$\forall X (\forall p \in X \exists q G[p, q] \rightarrow \exists W (\forall p \in X \exists q \in W G[p, q] \wedge \forall q \in W \exists p \in X G[p, q])),$$

for all formulas $G[p, q]$;

Subset Collection:

$\forall X \forall X' \exists Z (\forall p \in X \exists q \in X' G[p, q] \rightarrow \exists W \in Z (\forall p \in X \exists q \in W G[p, q] \wedge \forall q \in W \exists p \in X G[p, q]))$, for all formulas $G[p, q]$.

Definition 1 \mathbf{NCZF}^- denotes \mathbf{NCZF} without *Foundation*.

The theory \mathbf{NCZF}^- obviously contains Heyting arithmetic, intuitionistic $\Pi_0^1\text{-CA}$, and even more. We expect that its strength is bounded below by $\varphi(\varepsilon_0, 0)$.

2 Explicit Mathematics: a reminder

We want to give a realizability interpretation of \mathbf{NCZF}^- in the system **EETJ** of Explicit Mathematics, which would imply that \mathbf{NCZF}^- has a pretty low proof-theoretic strength, namely, at most $\varphi(\varepsilon_0, 0)$ (see Feferman (8, Ch.V, 1)). As was shown in (14), Explicit Mathematics is also an appropriate framework for interpreting Constructive Set Theory (other popular option: Intuitionistic Type Theory, cf. (1; 2), and, most recently, (6)). We have learned in (14) that *Elementary Comprehension* of **EM** suffices for everything, except *Strong Collection* and *Foundation*. In \mathbf{NCZF}^- we don't have *Foundation*; *Strong Collection* requires, in addition to *Elementary Comprehension*, only the axiom of *Join*.

Before we can get to the job in Section 3, we fix the formulation of Explicit Mathematics we prefer to work with. It's very close to the original formulation of (7), but takes some small deviations on a technical level.

Language \mathcal{L}_{EM} . All theories of Explicit Mathematics, considered in this paper, are formulated in a two-sorted language, containing variables for operations (individuals) and names, along with operation constants. Names are thought of as a special kind of operations, coding types (sets) of operations. We use **variables** $a, b, c, \dots, \mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$ as ranging over operations, and $\alpha, \beta, \gamma, \dots$ as ranging over names. The **constants** are the following: combinators \mathbf{k}, \mathbf{s} , pairing \mathbf{p} and projections $\mathbf{p}_0, \mathbf{p}_1$, zero $\mathbf{0}$, successor $\mathbf{s}_\mathbf{N}$ and predecessor $\mathbf{p}_\mathbf{N}$, distinction by cases on natural numbers Δ_0 and join \mathbf{j} . As a difference from the canonical system \mathbf{T}_0 , we omit the inductive generator \mathbf{i} . Additionally we have the following 9 **constants** called *name generators*: \mathbf{nat} , \mathbf{id} , \mathbf{inv} , \mathbf{emp} , \mathbf{and} , \mathbf{or} , \mathbf{imp} , \mathbf{all} , \mathbf{ex} . **Terms** are built from variables and constants by the following application clause: if s and t are *terms* then $s \cdot t$ is a *term*, so that the *application* function symbol \cdot accepts arguments of both sorts and returns an operation. **Atomic formulas** are $s = t$ (s coincides with t) and $s \varepsilon t$ (s belongs to the type named by t , s is classified under

t), where s and t are terms. **Formulas** are built from atomic formulas by $\wedge, \vee, \rightarrow$ and two kinds of quantifiers, over operations and over names, e.g. $\forall a, \exists a, \forall \alpha, \exists \alpha$. Finally, **expression** is a term or a formula.

Syntactical conventions.

1. We use $e[x]$ for an expression e , possibly containing occurrences of a variable x . In this context by $e[t]$ we mean the result of substituting expression t for all occurrences of x in e .
2. Parentheses in terms are assumed to be associated to the left: e.g., $s \cdot t \cdot u$ is read as $(s \cdot t) \cdot u$.
3. We adopt the following priority among propositional connectives and their abbreviations: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. For example, $F_1 \vee \neg F_2 \wedge F_3 \rightarrow F_4 \leftrightarrow F_5$ has to be read as $((F_1 \vee ((\neg F_2) \wedge F_3)) \rightarrow F_4) \leftrightarrow F_5$.

Abbreviations. We use the following shortcuts:

$\neg F :\leftrightarrow F \rightarrow \perp$;
 $F_0 \leftrightarrow F_1 :\leftrightarrow (F_0 \rightarrow F_1) \wedge (F_1 \rightarrow F_0)$;
 $t \downarrow :\leftrightarrow \exists x(t = x)$;
 $\mathcal{N}[t] :\leftrightarrow \exists \alpha(t = \alpha)$;
 $F[t \downarrow] :\leftrightarrow t \downarrow \wedge F[t]$;
 $t \doteq \{s[x_1, \dots, x_n] \mid F[x_1, \dots, x_n]\} :\leftrightarrow \mathcal{N}[t] \wedge \forall x(x \varepsilon t \leftrightarrow \exists x_1 \dots \exists x_n(x = s[x_1, \dots, x_n] \wedge F[x_1, \dots, x_n]))$;
 $s \simeq t :\leftrightarrow (s \downarrow \vee t \downarrow) \rightarrow s = t$;
 $s \dot{\subset} t :\leftrightarrow \forall x \varepsilon s(x \varepsilon t)$; $s \doteq t :\leftrightarrow s \dot{\subset} t \wedge t \dot{\subset} s$;
 $r : s \mapsto t$ for $\forall x \varepsilon s(r x \varepsilon t)$;
 $r : s^1 \mapsto t$ for $r : s \mapsto t$, $r : s^{m+1} \mapsto t$ for $\forall x \varepsilon s(r x : s^m \mapsto t)$;
 $\mathfrak{p}_{ij \dots k} t$ for $\mathfrak{p}_k(\dots (\mathfrak{p}_j(\mathfrak{p}_i t)) \dots)$, $i, j, \dots, k = 0, 0'$;
 t' for $s_{\mathbb{N}} \cdot t$; 1 for $0'$; 2 for $1'$; st for $s \cdot t$; $t(s_1, \dots, s_n)$ for $(\dots (ts_1) \dots s_n)$;
 $\langle s, t \rangle$ for $\mathfrak{p}st$; $s \neq t$ for $\neg s = t$, etc.

The logic of the theory is *intuitionistic 2-sorted logic of partial terms with equality*. See, e.g., (5, Ch.VI, 1) or (11, 1.3).

The axioms of Explicit Mathematics are divided in several groups. The basic are *applicative axioms, induction on natural numbers, explicit representation, elementary comprehension ECA* and *join J* (see, e.g., (14)). **EET** is the theory containing all the listed axioms except join, and **EETJ** is the one containing all of them.

EET will be our default theory for reasoning in Explicit Mathematics.

3 Realization of NCZF^- into **EETJ**

For each set variable $A \in \mathcal{L}_{\varepsilon, \mathbb{N}}$ we assume a name variable $\alpha_A \in \mathcal{L}_{\text{EM}}$. Sets are interpreted as (names of) trees, whose leaves may be labelled

by natural numbers. To begin, we need to set the stage.

We can define a name seq of the type of sequences so that

$$\text{seq} \doteq \{\langle x, y \rangle \mid (x = 0 \wedge y = 0) \vee (x = 1 \wedge y = \langle p_0 y, p_1 y \rangle \wedge p_0 y \varepsilon \text{seq})\}. \quad (1)$$

To do this, by **ECA** one defines a name seq_0 s.t.

$$\text{seq}_0 \doteq \{\langle 0, 0 \rangle\}, \quad (2)$$

and an operation seq_S s.t.

$$\text{seq}_S \alpha \doteq \{\langle 1, \langle y, z \rangle \rangle \mid y \varepsilon \alpha\}. \quad (3)$$

Then by primitive recursion one defines an operation sq s.t.

$$\begin{cases} \text{sq } 0 = \text{seq}_0, \\ \text{sq } n' = \text{seq}_S(\text{sq } n). \end{cases} \quad (4)$$

Finally by **J** and **ECA** one sets

$$\text{seq} \doteq \{x \mid \exists n \varepsilon \text{nat}(x \varepsilon \text{sq } n)\}. \quad (5)$$

We abbreviate

$$\text{nil} := \langle 0, 0 \rangle. \quad (6)$$

Now a name lseq of the type of labelled sequences is defined by

$$\text{lseq} := \{\langle 2, \langle y, n \rangle \rangle \mid y \varepsilon \text{seq} \wedge n \varepsilon \text{nat}\}. \quad (7)$$

The *label* of a labelled sequence is defined by

$$\text{lb} := \lambda x. p_1(p_1 x). \quad (8)$$

In order to work with (labelled) sequences, we define subsidiary operations *length* ln and *concatenation* conc (also written as $*$) by the following equations:

$$\begin{cases} \text{ln } \text{nil} \simeq 0, \\ \text{ln } \langle 1, \langle a, b \rangle \rangle \simeq s_{\mathbb{N}}(\text{ln } a), \\ \text{ln } \langle 2, \langle a, n \rangle \rangle \simeq \text{ln } a; \end{cases} \quad (9)$$

$$\begin{cases} \text{conc}(a, \text{nil}) \simeq a, \\ \text{conc}(a, \langle 1, \langle c, d \rangle \rangle) \simeq \langle 1, \langle \text{conc}(a, c), d \rangle \rangle, \\ \text{conc}(a, \langle 2, \langle c, n \rangle \rangle) \simeq \langle 2, \langle \text{conc}(a, c), n \rangle \rangle. \end{cases} \quad (10)$$

These operations have expected properties; see (14, Prop. 2.1 & 2.2) for details. We note that, in order $\text{conc}(s, t)$ to be defined, the first argument s of conc must be a sequence, while the second argument t may be labelled.

Definition 2 (\sqsupset)

By **ECA** a name \sqsupset is defined so that

$$\sqsupset \doteq \{(x, y) \mid x \varepsilon \text{seq} \cup \text{lseq} \wedge y \varepsilon \text{seq} \wedge \exists z \varepsilon \text{seq} \cup \text{lseq} (z \neq \text{nil} \wedge y * z = x)\}. \quad (11)$$

We will use $x \sqsupset y$, $x \sqsupseteq y$ in place of $\langle x, y \rangle \varepsilon \sqsupset$ and $(x \varepsilon \text{seq} \cup \text{lseq} \wedge x = y) \vee x \sqsupset y$, resp.

A *set* is a *tree*, i.e. non-empty type of (labelled) sequences downwards closed with respect to \sqsupset -relation:

Definition 3 (t names a set, $\text{Set}[t]$)

$\text{Set}[t]$ is defined as

$$\mathcal{N}[t] \wedge t \dot{\subseteq} \text{seq} \cup \text{lseq} \wedge \text{nil} \varepsilon t \wedge \forall x \varepsilon t \forall y (x \sqsupset y \rightarrow y \varepsilon t). \quad (12)$$

Note. Sets can be non-wellfounded with respect to \sqsupset , i.e. a set may contain an infinite sequence $\dots \sqsupset x_1 \sqsupset x_0$.

Definition 4 (Subtree operation, str)

By **ECA** we define an operation str in such a way that

$$\mathcal{N}[\text{str}(\alpha, z)] \wedge (x \varepsilon \text{str}(\alpha, z) \leftrightarrow x \varepsilon \text{seq} \cup \text{lseq} \wedge z * x \varepsilon \alpha). \quad (13)$$

Lemma 1 In **EETJ** we have

$$\text{Set}[\alpha] \wedge z \varepsilon \alpha \wedge z \varepsilon \text{seq} \rightarrow \text{Set}[\text{str}(\alpha, z)]. \quad (14)$$

Proof. Obvious from the definition. \square

As usual, equality between sets is interpreted as *bisimulation*. In order to stay low in proof-theoretic strength, we build bisimulation “in stages”, following the way it was done in (10). For any sets α and β we want to have an elementary relation $BS[\tau, \alpha, \beta]$ (“ τ is a proof that α and β are bisimulable”) and then to check that $BS[\tau, \alpha, \beta]$ has necessary properties. First, by **ECA** we define:

Definition 5 (st , ur)

$$\begin{aligned} \text{st } \alpha &:\doteq \{x \mid x \varepsilon \alpha \wedge x \varepsilon \text{seq} \wedge \text{ln } x = 1\}, \\ \text{ur } \alpha &:\doteq \{x \mid x \varepsilon \alpha \wedge x \varepsilon \text{lseq} \wedge \text{ln } x = 0\}. \end{aligned} \quad (15)$$

Definition 6 (Bisimulation)

Bisimulation is a formula $R[\tau, \alpha, \beta]$ together with an operation \mathfrak{f} such that the following holds:

if $\text{Set}[\alpha] \wedge \text{Set}[\beta] \wedge R[\tau, \alpha, \beta]$ then

$$\begin{aligned}
 & \forall x \varepsilon \text{st } \alpha \left(\text{p}_0(\text{p}_{00}(\text{fr})x) \varepsilon \text{st } \beta \wedge \right. \\
 & \quad \left. R[\text{p}_1(\text{p}_{00}(\text{fr})x), \text{str}(\alpha, x), \text{str}(\beta, \text{p}_0(\text{p}_{00}(\text{fr})x))] \right) \wedge \\
 & \quad \forall x \varepsilon \text{ur } \alpha \forall \mathfrak{s}(\text{p}_{01}(\text{fr}) \text{lb}(x) \mathfrak{s} \downarrow \wedge x \varepsilon \text{ur } \beta) \\
 & \quad \wedge \\
 & \forall y \varepsilon \text{st } \beta \left(\text{p}_0(\text{p}_{10}(\text{fr})y) \varepsilon \text{st } \alpha \wedge \right. \\
 & \quad \left. R[\text{p}_1(\text{p}_{10}(\text{fr})y), \text{str}(\alpha, \text{p}_0(\text{p}_{10}(\text{fr})y)), \text{str}(\beta, y)] \right) \wedge \\
 & \quad \forall y \varepsilon \text{ur } \beta \forall \mathfrak{s}(\text{p}_{11}(\text{fr}) \text{lb}(y) \mathfrak{s} \downarrow \wedge y \varepsilon \text{ur } \alpha).
 \end{aligned} \tag{16}$$

Remark. As long as we consider only bisimulations given by elementary (in parameters) formulas, a bisimulation can be defined as a pair $\langle n, f \rangle$, where $\mathcal{N}[n]$ (meaning: $n := \{\tau \mid R[\tau, \alpha, \beta]\}$) and the shown condition holds. This in fact suffices for purposes of the present paper.

Now we are interested to build a maximal bisimulation BS . We set

$$\text{bs}_0 := \lambda a \lambda b. n, \tag{17}$$

where $n := \{0\}$. Having in mind that we will have

$$\text{Set}[\alpha] \wedge \text{Set}[\beta] \wedge c : \text{Set}^2 \mapsto \mathcal{N},$$

where $c : \text{Set}^2 \mapsto \mathcal{N}$ stands for $\forall \gamma \forall \delta (\text{Set}[\gamma] \wedge \text{Set}[\delta] \rightarrow \mathcal{N}[c\gamma\delta])$, we want to define a name $t[\alpha, \beta, c]$ for

$$\begin{aligned}
 & \{ \tau \mid \\
 & \forall x \varepsilon \text{st } \alpha \left(\text{p}_0(\text{p}_{00}\tau x) \varepsilon \text{st } \beta \wedge \right. \\
 & \quad \left. \text{p}_1(\text{p}_{00}\tau x) \varepsilon c \cdot \text{str}(\alpha, x) \cdot \text{str}(\beta, \text{p}_0(\text{p}_{00}\tau x)) \right) \wedge \\
 & \quad \forall x \varepsilon \text{ur } \alpha \forall \mathfrak{s}(\text{p}_{01}\tau \text{lb}(x) \mathfrak{s} \downarrow \wedge x \varepsilon \text{ur } \beta) \\
 & \quad \wedge \\
 & \forall y \varepsilon \text{st } \beta \left(\text{p}_0(\text{p}_{10}\tau y) \varepsilon \text{st } \alpha \wedge \right. \\
 & \quad \left. \text{p}_1(\text{p}_{10}\tau y) \varepsilon c \cdot \text{str}(\alpha, \text{p}_0(\text{p}_{10}\tau y)) \cdot \text{str}(\beta, y) \right) \wedge \\
 & \quad \forall y \varepsilon \text{ur } \beta \forall \mathfrak{s}(\text{p}_{11}\tau \text{lb}(y) \mathfrak{s} \downarrow \wedge y \varepsilon \text{ur } \alpha) \}.
 \end{aligned} \tag{18}$$

To do this, first a name $t_1[\alpha, \beta]$ for

$$\{ \tau \mid \forall x \varepsilon \text{st } \alpha (\text{p}_0(\text{p}_{00}\tau x) \varepsilon \text{st } \beta) \}$$

is defined by **ECA**. Now we consider a function

$$f := \lambda u. c \cdot \text{str}(\alpha, \text{p}_1 u) \cdot \text{str}(\beta, \text{p}_0(\text{p}_{00}(\text{p}_0 u)(\text{p}_1 u)))$$

and form $t'_1 := j(t_1 \times \text{st } \alpha, f)$. A name $t_2[\alpha, \beta, c]$ for

$$\{ \tau \mid \forall x \varepsilon \text{st } \alpha (\langle \langle \tau, x \rangle, \text{p}_1(\text{p}_{00}\tau x) \rangle \varepsilon t'_1) \},$$

defined by **ECA**, is then also a name for

$$\{\tau \mid \forall x \varepsilon \text{st } \alpha \ (p_0(p_{00}tx) \varepsilon \text{st } \beta \ \wedge \ p_1(p_{00}tx) \varepsilon c \cdot \text{str}(\alpha, x) \cdot \text{str}(\beta, p_0(p_{00}tx)))\}.$$

The second part,

$$\{\tau \mid \forall x \varepsilon \text{ur } \alpha \forall s (\rho_{01}\tau \text{lb}(x) \varepsilon \downarrow \wedge x \varepsilon \text{ur } \beta)\},$$

has a name $t_3[\alpha, \beta]$ by **ECA**.

Symmetrically we construct names t_4 , t_5 and t_6 having to do with conjuncts $\forall y \in \beta \dots$ of 18 and finally build

$$t := \text{and}(\text{and}(t_2, t_3), \text{and}(t_5, t_6)).$$

Now we define an operation bs_S by

$$\text{bs}_S c = \lambda a \lambda b. t[a, b, c] \quad (19)$$

and formally verify in **EETJ**

$$c : \text{Set}^2 \mapsto \mathcal{N} \rightarrow \text{bs}_S c : \text{Set}^2 \mapsto \mathcal{N}.$$

By primitive recursion we define an operation bs s.t.

$$\begin{cases} \text{bs } 0 = \text{bs}_0, \\ \text{bs } n' = \lambda a \lambda b. \text{bs}_S(\text{bs } n)ab \end{cases} \quad (20)$$

and by induction on nat prove

$$\forall n \varepsilon \text{nat} (\text{bs } n : \text{Set}^2 \mapsto \mathcal{N}).$$

Now, by primitive recursion, we define projection functions $\text{pr } n$ by

$$\begin{cases} \text{pr } 0 \tau = 0, \\ \text{pr } n' \tau = \langle f, g \rangle, \end{cases} \quad (21)$$

where

$$\begin{cases} f := \langle \lambda x. \langle p_0(p_{00}tx), \text{pr } n(p_1(p_{00}tx)) \rangle, p_{01}\tau \rangle, \\ g := \langle \lambda y. \langle p_0(p_{01}\tau y), \text{pr } n(p_1(p_{01}\tau y)) \rangle, p_{11}\tau \rangle, \end{cases} \quad (22)$$

and by induction on nat prove

$$\forall n \varepsilon \text{nat} \forall \alpha \forall \beta (\text{Set}[\alpha] \wedge \text{Set}[\beta] \rightarrow \text{pr } n : \text{bs } n' \alpha \beta \mapsto \text{bs } n \alpha \beta).$$

Definition 7 (Maximal bisimulation $BS[\tau, \alpha, \beta]$)

A formula $BS[\tau, \alpha, \beta]$ is defined as

$$\forall n \varepsilon \text{nat} (\langle n, \tau n \rangle \varepsilon \text{j}(\text{nat}, \lambda m. \text{bs } m \alpha \beta) \wedge \text{pr } n(\tau n') = \tau n). \quad (23)$$

An τ such that $BS[\tau, \alpha, \beta]$ is called bisimulator for α and β and in this case α and β are called bisimulable by τ .

We **note** that if $\text{Set}[\alpha] \wedge \text{Set}[\beta]$ then $\mathcal{N}[\mathbf{j}(\text{nat}, \lambda m. \text{bs } m\alpha\beta)]$ and $BS[\mathbf{r}, \alpha, \beta]$ is elementary in $\mathbf{j}(\text{nat}, \lambda m. \text{bs } m\alpha\beta)$. Also, in this case, $BS[\mathbf{r}, \alpha, \beta]$ is equivalent to

$$\forall n \in \text{nat} (\mathbf{r}n \varepsilon \text{bs } n\alpha\beta \wedge \text{pr } n(\mathbf{r}n') = \mathbf{r}n).$$

Lemma 2 (BS is a bisimulation)

There is an operation ex_0 s.t. $\langle BS, \text{ex}_0 \rangle$ is a bisimulation.

Proof. Assume $\text{Set}[\alpha] \wedge \text{Set}[\beta] \wedge BS[\mathbf{r}, \alpha, \beta]$.

If $x \varepsilon \text{ur } \alpha$, we simply put $\mathbf{p}_{01}(\text{ex}_0\mathbf{r}) := \mathbf{p}_{01}(\mathbf{r}1)$; symmetrically we put $\mathbf{p}_{11}(\text{ex}_0\mathbf{r}) := \mathbf{p}_{11}(\mathbf{r}1)$. Given $x \varepsilon \text{st } \alpha$, we will find $y \varepsilon \text{st } \beta$ and a bisimulator for $(\text{str}(\alpha, x), \text{str}(\beta, y))$. A similar construction given $y \varepsilon \text{st } \beta$ will determine $x \varepsilon \text{st } \alpha$ and a bisimulator for $(\text{str}(\alpha, x), \text{str}(\beta, y))$. So let $x \varepsilon \text{st } \alpha$. For $n \varepsilon \text{nat}$ let

$$y_n := \mathbf{p}_0(\mathbf{p}_{00}(\mathbf{r}n')x) \varepsilon \text{st } \beta$$

and

$$\mathbf{t}_n := \mathbf{p}_1(\mathbf{p}_{00}(\mathbf{r}n')x) \varepsilon \text{bs } n \cdot \text{str}(\alpha, x) \cdot \text{str}(\beta, y_n).$$

Now,

$$\begin{aligned} \mathbf{r}n' &= \text{pr } n'(\mathbf{r}n''), \\ \mathbf{p}_{00}(\mathbf{r}n')x &= \mathbf{p}_{00}(\text{pr } n'(\mathbf{r}n''))x = \\ &\langle \mathbf{p}_0(\mathbf{p}_{00}(\mathbf{r}n'')x), \text{pr } n(\mathbf{p}_1(\mathbf{p}_{00}(\mathbf{r}n'')x)) \rangle. \end{aligned} \quad (24)$$

From this it follows that $y_{n'} = y_n$ for all $n \varepsilon \text{nat}$, and thus $y_n = y_0$ for all $n \varepsilon \text{nat}$. So, $\mathbf{t}_n \varepsilon \text{bs } n \cdot \text{str}(\alpha, x) \cdot \text{str}(\beta, y_0)$. Continuing 24, we have

$$\begin{aligned} \mathbf{p}_1(\mathbf{p}_{00}(\mathbf{r}n')x) &= \text{pr } n(\mathbf{p}_1(\mathbf{p}_{00}(\mathbf{r}n'')x)), \\ \mathbf{t}_n &= \text{pr } n\mathbf{t}_{n'}, \end{aligned} \quad (25)$$

so that $BS[\lambda n. \mathbf{t}_n, \text{str}(\alpha, x), \text{str}(\beta, y_0)]$.

Summarizing the construction, we set

$$\begin{aligned} \text{ex}_0 &:= \lambda \mathbf{r}. \\ &\langle \langle \lambda x. \langle \mathbf{p}_0(\mathbf{p}_{00}(\mathbf{r}1)x), \lambda n. \mathbf{p}_1(\mathbf{p}_{00}(\mathbf{r}n')x) \rangle, \mathbf{p}_{01}(\mathbf{r}1) \rangle, \\ &\langle \lambda y. \langle \mathbf{p}_0(\mathbf{p}_{10}(\mathbf{r}1)y), \lambda n. \mathbf{p}_1(\mathbf{p}_{10}(\mathbf{r}n')y) \rangle, \mathbf{p}_{11}(\mathbf{r}1) \rangle \rangle. \end{aligned} \quad (26)$$

□

Lemma 3 (BS is a maximal bisimulation)

There is an operation ex_1 s.t. if $\langle R, \mathbf{f} \rangle$ is a bisimulation then

$$\forall \alpha \forall \beta \forall \mathbf{r} (\text{Set}[\alpha] \wedge \text{Set}[\beta] \wedge R[\mathbf{r}, \alpha, \beta] \rightarrow BS[\text{ex}_1\mathbf{f}\mathbf{r}, \alpha, \beta]).$$

Proof. Let $\langle R, f \rangle$ be a bisimulation and $\text{Set}[\alpha] \wedge \text{Set}[\beta] \wedge R[\mathfrak{t}, \alpha, \beta]$. We let

$$f := \lambda \mathfrak{s}. p_{00}(\mathfrak{f}\mathfrak{s}), \quad g := \lambda \mathfrak{s}. p_{10}(\mathfrak{f}\mathfrak{s}).$$

Then, by primitive recursion, we define \mathfrak{t} as follows:

$$\begin{cases} \mathfrak{t}0 = \lambda \mathfrak{s}. 0, \\ \mathfrak{t}n' = \lambda \mathfrak{s}. \langle \lambda x. \langle p_0(\mathfrak{f}\mathfrak{s}x), \mathfrak{t}n p_1(\mathfrak{f}\mathfrak{s}x) \rangle, p_{01}(\mathfrak{f}\mathfrak{s}) \rangle, \\ \quad \langle \lambda y. \langle p_0(\mathfrak{g}\mathfrak{s}y), \mathfrak{t}n p_1(\mathfrak{g}\mathfrak{s}y) \rangle, p_{11}(\mathfrak{f}\mathfrak{s}) \rangle \rangle. \end{cases}$$

By induction on nat we prove

$$\forall n \in \text{nat} \forall \gamma \forall \delta \forall \mathfrak{s} (\text{Set}[\gamma] \wedge \text{Set}[\delta] \wedge R[\mathfrak{s}, \gamma, \delta] \rightarrow \mathfrak{t}n \mathfrak{s} \varepsilon \text{bs } n \gamma \delta \wedge \text{pr } n(\mathfrak{t}n' \mathfrak{s}) = \mathfrak{t}n \mathfrak{s}).$$

Now, let $\text{ex}_1 := \lambda f \lambda \mathfrak{s} \lambda n. \mathfrak{t}[f]n \mathfrak{s}$. Then from the previous we have

$$\forall n \in \text{nat} (\text{ex}_1 f \mathfrak{t}n \varepsilon \text{bs } n \alpha \beta) \wedge \text{pr } n(\text{ex}_1 f \mathfrak{t}n') = \text{ex}_1 f \mathfrak{t}n.$$

Thus, $BS[\text{ex}_1 f \mathfrak{t}, \alpha, \beta]$. □

Lemma 4 *There is an operation ex_2 s.t. if $\text{Set}[\alpha] \wedge \text{Set}[\beta] \wedge BS'[\mathfrak{t}, \alpha, \beta]$, where $BS'[\mathfrak{t}, \alpha, \beta]$ is a formula*

$$\begin{aligned} & \forall x \varepsilon \text{st } \alpha \left(p_0(p_{00}\mathfrak{t}x) \varepsilon \text{st } \beta \wedge \right. \\ & \quad \left. BS[p_1(p_{00}\mathfrak{t}x), \text{str}(\alpha, x), \text{str}(\beta, p_0(p_{00}\mathfrak{t}x))] \right) \wedge \\ & \quad \forall x \varepsilon \text{ur } \alpha \forall \mathfrak{s} (p_{01}\mathfrak{t} \text{lb}(x)\mathfrak{s} \downarrow \wedge x \varepsilon \text{ur } \beta) \\ & \quad \wedge \\ & \forall y \varepsilon \text{st } \beta \left(p_0(p_{10}\mathfrak{t}y) \varepsilon \text{st } \alpha \wedge \right. \\ & \quad \left. BS[p_1(p_{10}\mathfrak{t}y), \text{str}(\alpha, p_0(p_{10}\mathfrak{t}y)), \text{str}(\beta, y)] \right) \wedge \\ & \quad \forall y \varepsilon \text{ur } \beta \forall \mathfrak{s} (p_{11}\mathfrak{t} \text{lb}(y)\mathfrak{s} \downarrow \wedge y \varepsilon \text{ur } \alpha), \end{aligned} \tag{27}$$

then $BS[\text{ex}_2 \mathfrak{t}, \alpha, \beta]$.

Proof. It follows from Lemma 2 that there is an operation ex'_0 s.t. if $\text{Set}[\gamma] \wedge \text{Set}[\delta] \wedge BS'[\mathfrak{s}, \gamma, \delta]$, then

$$\begin{aligned} & \forall x \varepsilon \text{st } \gamma \left(p_0(p_{00}(\text{ex}'_0 \mathfrak{s})x) \varepsilon \text{st } \delta \wedge \right. \\ & \quad \left. BS'[p_1(p_{00}(\text{ex}'_0 \mathfrak{s})x), \text{str}(\gamma, x), \text{str}(\delta, p_0(p_{00}(\text{ex}'_0 \mathfrak{s})x))] \right) \wedge \\ & \quad \forall x \varepsilon \text{ur } \gamma \forall \mathfrak{s} (p_{01}\mathfrak{t} \text{lb}(x)\mathfrak{s} \downarrow \wedge x \varepsilon \text{ur } \delta) \\ & \quad \wedge \\ & \forall y \varepsilon \text{st } \delta \left(p_0(p_{10}(\text{ex}'_0 \mathfrak{s})y) \varepsilon \text{st } \gamma \wedge \right. \\ & \quad \left. BS'[p_1(p_{10}(\text{ex}'_0 \mathfrak{s})y), \text{str}(\gamma, p_0(p_{10}(\text{ex}'_0 \mathfrak{s})y)), \text{str}(\delta, y)] \right) \wedge \\ & \quad \forall y \varepsilon \text{ur } \delta \forall \mathfrak{s} (p_{11}\mathfrak{t} \text{lb}(y)\mathfrak{s} \downarrow \wedge y \varepsilon \text{ur } \gamma), \end{aligned} \tag{28}$$

so that $\langle BS', \text{ex}'_0 \rangle$ is a bisimulation. By Lemma 3 we have $BS[\text{ex}_1 \text{ex}'_0 \mathfrak{t}, \alpha, \beta]$, so we set $\text{ex}_2 := \text{ex}_1 \text{ex}'_0$. □

Lemma 5 *BS is an equivalence relation, i.e. there are operations eq₀, eq₁ and eq₂ s.t. if Set[α] ∧ Set[β] ∧ Set[γ] then:*

- a) $BS[eq_0, \alpha, \alpha]$;
- b) $BS[\tau, \alpha, \beta] \rightarrow BS[\lambda n. eq_1 n \tau, \beta, \alpha]$;
- c) $BS[\tau, \alpha, \beta] \wedge BS[s, \beta, \gamma] \rightarrow BS[\lambda n. eq_2 n \tau s, \alpha, \gamma]$.

Proof. eq₀, eq₁ and eq₂ are defined by primitive recursion and then by induction on nat necessary properties are proved. Recursive equations are the following:

$$\begin{cases} eq_0 0 = 0, \\ eq_0 n' = \langle \langle \lambda x. \langle x, eq_0 n \rangle, \lambda x \lambda u. 0 \rangle, \langle \lambda x. \langle x, eq_0 n \rangle, \lambda x \lambda u. 0 \rangle \rangle; \end{cases}$$

$$\begin{cases} eq_1 0 = \lambda \tau. 0, \\ eq_1 n' = \begin{cases} \lambda \tau. \\ \langle \langle \lambda y. \langle p_0(p_{10}(\tau n') y), eq_1 n \cdot p_1(p_{10}(\tau n') y) \rangle, \lambda y \lambda u. 0 \rangle, \\ \langle \lambda x. \langle p_0(p_{00}(\tau n') x), eq_1 n \cdot p_1(p_{00}(\tau n') x) \rangle, \lambda x \lambda u. 0 \rangle \rangle; \end{cases} \end{cases}$$

$$\begin{cases} eq_2 0 = \lambda \tau s. 0, \\ eq_2 n' = \begin{cases} \lambda \tau s. \\ \langle \langle \lambda x. \langle p_0(p_{00}(s n') (p_0(p_{00}(\tau n') x))), \\ eq_2 n \cdot p_1(p_{00}(\tau n') x) \rangle \cdot p_1(p_{00}(s n') (p_0(p_{00}(\tau n') x))), \\ \lambda x \lambda u. 0 \rangle, \\ \langle \lambda z. \langle p_0(p_{10}(\tau n') (p_0(p_{10}(s n') z))), \\ eq_2 n \cdot p_1(p_{10}(s n') z) \rangle \cdot p_1(p_{10}(\tau n') (p_0(p_{10}(s n') z))), \\ \lambda z \lambda u. 0 \rangle \rangle. \end{cases} \end{cases}$$

□

Remark. The main distinction of bisimulation defined here from the ones which are often used within set theory setting (see (9)), based on the idea of existence of a set \sim s.t.

$$A \sim B \leftrightarrow \forall X \in A \exists Y \in B (X \sim Y) \wedge \forall Y \in B \exists X \in A (X \sim Y),$$

where $U \sim V$ means $\langle U, V \rangle \in \sim$, is that we are actually *building* such a type \sim , not merely requiring its *existence*. The advantage is that such a construction requires proof-theoretically only very weak means (namely, in **EM**, only elementary comprehension, join, and induction on natural numbers), while, when standard bisimulation is used, for such an equality interpretation one needs some additional axioms, e.g. Σ_1 *Foundation* (see, for example, (3, S. 4)), which raise proof-theoretic strength compared to the non-wellfounded version of the theory.

For each function constant $f \in \mathcal{L}_{\in, \mathbb{N}}$ we can define an operation $\mathbf{N}(f)$ representing the same primitive recursive function as f and having the following property: if n is the arity of f then **EET** proves

$$\bigwedge_{i=1}^n x_i \varepsilon \text{nat} \rightarrow \mathbf{N}(f)x_1 \dots x_n \varepsilon \text{nat}.$$

Now terms of $\mathcal{L}_{\in, \mathbb{N}}$ are translated as follows:

Definition 8 ($\mathbf{N}(t)$)

$$\begin{cases} \mathbf{N}(x) := x; \\ \mathbf{N}(f(t_1, \dots, t_n)) := \mathbf{N}(f)\mathbf{N}(t_1) \dots \mathbf{N}(t_n). \end{cases}$$

Definition 9 (τ realizes F , $\tau \underline{\mathbf{rn}} F$)

For each formula $F \in \mathcal{L}_{\in, \mathbb{N}}$ we define a formula $(\tau \underline{\mathbf{rn}} F) \in \mathcal{L}_{\text{EM}}$ with a new free individual variable τ . The definition is given by the table below:

$F \in \mathcal{L}_{\in, \mathbb{N}}$	$(\tau \underline{\mathbf{rn}} F) \in \mathcal{L}_{\text{EM}}$
\perp	\perp
$s = t$	$\mathbf{N}(s) = \mathbf{N}(t)$
$s = A$	\perp
$A = s$	\perp
$A = B$	$BS[\tau, \alpha_A, \alpha_B]$
$s \in t$	\perp
$s \in A$	$\langle 2, \langle \text{nil}, \mathbf{N}(s) \rangle \rangle \varepsilon \alpha_A$
$A \in s$	\perp
$A \in B$	$p_0\tau \varepsilon \text{st } \alpha_B \wedge BS[p_1\tau, \alpha_A, \text{str}(\alpha_B, p_0\tau)]$
$F_0 \wedge F_1$	$p_0\tau \underline{\mathbf{rn}} F_0 \wedge p_1\tau \underline{\mathbf{rn}} F_1$
$F_0 \vee F_1$	$p_0\tau \varepsilon \text{nat} \wedge (p_0\tau = 0 \rightarrow p_1\tau \underline{\mathbf{rn}} F_0) \wedge (p_0\tau \neq 0 \rightarrow p_1\tau \underline{\mathbf{rn}} F_1)$
$F_0 \rightarrow F_1$	$\forall \tau (\tau \underline{\mathbf{rn}} F_0 \rightarrow \tau \downarrow \wedge \tau \underline{\mathbf{rn}} F_1)$
$\forall x G[x]$	$\forall x \varepsilon \text{nat} (\tau x \downarrow \wedge \tau x \underline{\mathbf{rn}} G[x])$
$\exists x G[x]$	$p_0\tau \varepsilon \text{nat} \wedge p_1\tau \underline{\mathbf{rn}} G[p_0\tau]$
$\forall X G[X]$	$\forall \alpha (\text{Set}[\alpha] \rightarrow \tau \alpha \downarrow \wedge \tau \alpha \underline{\mathbf{rn}} G[\alpha])$
$\exists X G[X]$	$\text{Set}[p_0\tau] \wedge p_1\tau \underline{\mathbf{rn}} G[p_0\tau]$

Remark. According to our notation for substitution, p. 4, in this definition $\rho_1 \underline{\mathbf{r}} \underline{\mathbf{r}} G[\rho_0 \mathbf{t}]$ in the last clause, for example, stands for $(\underline{\mathbf{r}} \underline{\mathbf{r}} G[X])[\mathbf{t}/\rho_1 \mathbf{t}][\alpha_X/\rho_0 \mathbf{t}]$.

Definition 10 (Realization, realizable)

1. A term $t \in \mathcal{L}_{\text{EM}}$ is called realization of a formula $F \in \mathcal{L}_{\in, \mathbb{N}}$ in a theory $\mathbf{T} \in \mathcal{L}_{\text{EM}}$, iff

$$\text{FV}_0(t) \subseteq \{a \mid a \in \text{FV}_0(F)\} \quad \bigwedge \quad \text{FV}_1(t) \subseteq \{\alpha_A \mid A \in \text{FV}(F)\}$$

and

$$\mathbf{T} \vdash \left(\bigwedge_{a \in \text{FV}_0(F)} (a \varepsilon \text{nat}) \wedge \bigwedge_{A \in \text{FV}_1(F)} \text{Set}[\alpha_A] \right) \rightarrow t \underline{\mathbf{r}} F.$$

2. If there exists such a term t then F is called realizable in \mathbf{T} . We call a theory \mathbf{T}_S realizable in \mathbf{T} iff every theorem of \mathbf{T}_S is realizable in \mathbf{T} . \mathbf{T}_S is realizable iff it's realizable in **EET**.

Theorem 1 Each theorem of intuitionistic two-sorted predicate calculus with equality is realizable in **EETJ**.

Proof is standard except for the case of equality axioms. We have to analyze three types of equalities: between numbers, between a number and a set, and between sets. Realizations of equalities between numbers follow from the law $\underline{\mathbf{r}} \underline{\mathbf{r}} (s = t) :\Leftrightarrow \mathbf{N}(s) = \mathbf{N}(t)$ (second line of Definition 9) and equality axioms of logic. In particular, $\underline{\mathbf{r}} \underline{\mathbf{r}} (s = t \wedge s \in A) \equiv \mathbf{N}(s) = \mathbf{N}(t) \wedge \langle 2, \langle \text{nil}, \mathbf{N}(s) \rangle \rangle \varepsilon \alpha_A \Rightarrow \langle 2, \langle \text{nil}, \mathbf{N}(t) \rangle \rangle \varepsilon \alpha_A \equiv \underline{\mathbf{r}} \underline{\mathbf{r}} (t \in A)$. Realizations of equalities of the kind $s = A$ follow from the fact that $\underline{\mathbf{r}} \underline{\mathbf{r}} (s = A)$ is \perp , and the logical axiom “ $\perp \rightarrow$ anything”. Finally, when considering equalities between sets, the main cases are:

(Eq1) $A = B \wedge A \in C \rightarrow B \in C$;

(Eq2) $A = B \wedge C \in A \rightarrow C \in B$.

For (Eq1), if $\underline{\mathbf{r}} \underline{\mathbf{r}} \alpha = \beta$ and $\mathbf{s} \underline{\mathbf{r}} \underline{\mathbf{r}} \alpha \in \gamma$, then \mathbf{s} gives us an appropriate point $\rho_0 \mathbf{s}$ in γ , and a bisimulator between β and $\text{str}(\gamma, \rho_0 \mathbf{s})$ is obtained by commutativity and transitivity (Lemma 5).

For (Eq2), if $\underline{\mathbf{r}} \underline{\mathbf{r}} \alpha = \beta$ and $\mathbf{s} \underline{\mathbf{r}} \underline{\mathbf{r}} \gamma \in \alpha$, then $\rho_0 \mathbf{s} \varepsilon \text{st} \alpha \wedge BS[\rho_1 \mathbf{s}, \gamma, \text{str}(\alpha, \rho_0 \mathbf{s})]$. By \mathbf{r} and Lemma 2 we obtain $t \varepsilon \text{st} \beta \wedge BS[\mathbf{q}, \text{str}(\alpha, \rho_0 \mathbf{s}), \text{str}(\beta, t)]$ for some t and \mathbf{q} and by transitivity follows the assertion. □

Now we turn to realizing mathematical axioms of \mathbf{NCZF}^- in \mathbf{EETJ} . According to Theorem 1, this is sufficient to claim realizability of every theorem.

We have to proceed by different groups of axioms of \mathbf{NCZF}^- . *Ontological* ones are obviously realized by $\langle \lambda x.x, \lambda y.y \rangle$, since $\tau \underline{\mathbf{rn}} F \Leftrightarrow \tau \underline{\mathbf{rn}} \perp \Leftrightarrow \perp$, for F of any of the forms $a = A$, $A = a$, $a \in b$, or $A \in a$. *Arithmetical* axioms are realized as in (12, Ch.IV, Section 4). We note that realizing *Induction* requires induction on natural numbers of Explicit Mathematics.

What remains are proofs of the *set* axioms. Those are again realized exactly as in (14, S. 3); the only thing to note is that since Δ_0 formulas now may contain quantifiers over urelements, we have to make sure that we still can provide *elementary realization* $\underline{\mathbf{rn}}^\varepsilon$ for those formulas. But this is indeed so, since the clauses for $\tau \underline{\mathbf{rn}} \forall x G[x]$ and $\tau \underline{\mathbf{rn}} \exists x G[x]$ in the Definition 9 are elementary in $\tau \underline{\mathbf{rn}} G[x]$:

Definition 11 τ elementarily realizes F , $\tau \underline{\mathbf{rn}}^\varepsilon F$

For each bounded formula $F \in \mathcal{L}_{\in, \mathbb{N}}$ we define an elementary formula $(\tau \underline{\mathbf{rn}}^\varepsilon F) \in \mathcal{L}_{\text{EM}}$ with a new free individual variable τ . The definition is obtained from the Definition 9 of $\underline{\mathbf{rn}}$ by replacing $\underline{\mathbf{rn}}$ by $\underline{\mathbf{rn}}^\varepsilon$ everywhere and taking clauses for $\forall X \in AG[X]$ and $\exists X \in AG[X]$ as follows:

F	$\tau \underline{\mathbf{rn}}^\varepsilon F$
$\forall X \in AG[X]$	$\forall x \varepsilon \text{st } \alpha_A (\tau x \downarrow \wedge \tau x \underline{\mathbf{rn}}^\varepsilon G[\text{str}(\alpha_A, x)])$
$\exists X \in AG[X]$	$\rho_0 \tau \varepsilon \text{st } \alpha_A \wedge \rho_1 \tau \underline{\mathbf{rn}}^\varepsilon G[\text{str}(\alpha_A, \rho_0 \tau)]$

One establishes correspondence between realizations $\underline{\mathbf{rn}}$ and $\underline{\mathbf{rn}}^\varepsilon$ in the same way as in Definition 2.11 and Lemma 2.5 of (14):

Lemma 6 Δ_0 -lemma, cf. (14, L. 2.5)

If $F \in \mathcal{L}_{\in, \mathbb{N}}$ is bounded then there is a term $\text{eq}_F^\varepsilon \in \mathcal{L}_{\text{EM}}$ s.t. $\text{FV}(\text{eq}_F^\varepsilon) \subseteq \{a \mid a \in \text{FV}_0(F)\} \cup \{\alpha_A \mid A \in \text{FV}_1(F)\}$, $\text{eq}_F^\varepsilon = \langle \rho_0 \text{eq}_F^\varepsilon, \rho_1 \text{eq}_F^\varepsilon \rangle$ and the following holds:

$$\tau \underline{\mathbf{rn}} F \rightarrow \rho_0 \text{eq}_F^\varepsilon \tau \underline{\mathbf{rn}}^\varepsilon F \quad (29)$$

and

$$\eta \underline{\mathbf{rn}}^\varepsilon F \rightarrow \rho_1 \text{eq}_F^\varepsilon \eta \underline{\mathbf{rn}} F. \quad (30)$$

Extensionality is the main axiom for which bisimulation is responsible. However, everything is already contained in Lemma 4.

Lemma 7 (*Extensionality*)

The Extensionality axiom is realizable in **EETJ**.

Proof. Indeed, given $\text{Set}[\alpha] \wedge \text{Set}[\beta]$,
 $\underline{\text{rn}}^\varepsilon ((\forall X \in \alpha (X \in \beta) \wedge \forall x \in \alpha (x \in \beta)) \wedge (\forall Y \in \beta (Y \in \alpha) \wedge \forall y \in \beta (y \in \alpha)))$
 reads as

$$\begin{aligned} & \forall x \varepsilon \text{st } \alpha \left(\text{p}_0(\text{p}_{00}\text{t}x) \varepsilon \text{st } \beta \wedge \right. \\ & \quad \left. BS[\text{p}_1(\text{p}_{00}\text{t}x), \text{str}(\alpha, x), \text{str}(\beta, \text{p}_0(\text{p}_{00}\text{t}x))] \right) \wedge \\ & \quad \text{p}_{01}\text{t } \underline{\text{rn}}^\varepsilon \forall x \in \alpha (x \in \beta) \\ & \quad \quad \quad \wedge \\ & \forall y \varepsilon \text{st } \beta \left(\text{p}_0(\text{p}_{10}\text{t}y) \varepsilon \text{st } \alpha \wedge \right. \\ & \quad \left. BS[\text{p}_1(\text{p}_{10}\text{t}y), \text{str}(\alpha, \text{p}_0(\text{p}_{10}\text{t}y)), \text{str}(\beta, y)] \right) \wedge \\ & \quad \text{p}_{11}\text{t } \underline{\text{rn}}^\varepsilon \forall y \in \beta (y \in \alpha), \end{aligned} \tag{31}$$

$$\begin{aligned} & \text{p}_{01}\text{t } \underline{\text{rn}}^\varepsilon \forall x \in \alpha (x \in \beta) \equiv \\ & \forall x \varepsilon \text{nat} \forall \mathfrak{s} (\mathfrak{s } \underline{\text{rn}} (x \in \alpha) \rightarrow \text{p}_{01}\text{t}x\mathfrak{s} \downarrow \underline{\text{rn}} (x \in \beta)) \equiv \\ & \forall x \varepsilon \text{ur } \alpha \forall \mathfrak{s} (\text{p}_{01}\text{t } \text{lb}(x)\mathfrak{s} \downarrow \wedge x \varepsilon \text{ur } \beta), \\ & \quad \quad \quad \equiv \\ & \text{p}_{11}\text{t } \underline{\text{rn}}^\varepsilon \forall y \in \beta (y \in \alpha) \equiv \\ & \forall y \varepsilon \text{nat} \forall \mathfrak{s} (\mathfrak{s } \underline{\text{rn}} (y \in \beta) \rightarrow \text{p}_{11}\text{t}y\mathfrak{s} \downarrow \underline{\text{rn}} (y \in \alpha)) \equiv \\ & \forall y \varepsilon \text{ur } \beta \forall \mathfrak{s} (\text{p}_{11}\text{t } \text{lb}(y)\mathfrak{s} \downarrow \wedge y \varepsilon \text{ur } \alpha), \end{aligned} \tag{32}$$

and then a bisimulator between α and β is obtained by Lemma 4. \square

The only axiom which requires a new proof is the axiom of *Infinity*. This is given below.

Lemma 8 (*Infinity*)

Axiom *Infinity* is realizable in **EETJ**.

Proof. An infinite tree is constructed as

$$\{x \mid x = \text{nil} \vee \exists n \varepsilon \text{nat} (x = \langle 2, \langle \text{nil}, n \rangle \rangle)\}. \tag{33}$$

These facts give us \square

Theorem 2 NCZF^- is realizable in **EETJ**; therefore its strength is bounded above by $\varphi(\varepsilon_0, 0)$.

Proof. Realizability has been shown. The fact about $\varphi(\varepsilon_0, 0)$ follows from $|\mathbf{EETJ}| = \varphi(\varepsilon_0, 0)$ (8, Ch.V, 1). \square

References

- [1] P. Aczel. The Type Theoretic Interpretation of Constructive Set Theory. In: A. MacIntyre, L. Pacholski, J. Paris (eds.), *Logic Colloquium '77*: 55–66, 1978
- [2] P. Aczel. The Type Theoretic Interpretation of Constructive Set Theory: inductive definitions. In: R.B. Marcus et al. (eds), *Logic, Methodology and Philosophy of Science VII*: Amsterdam, 1986
- [3] J. Avigad. Interpreting classical theories in constructive ones. To appear in the *Journal of Symbolic Logic*
- [4] J. Barwise. Admissible sets and structures. Springer, 1975
- [5] M. Beeson. Foundations of Constructive Mathematics. Springer, 1985
- [6] L. Crosilla. Realizability models for constructive set theories with restricted induction. *PhD Thesis*, University of Leeds, 2000
- [7] S. Feferman. A language and axioms for explicit mathematics. In: Algebra and Logic, *LNM* 450: 87–139, 1975
- [8] S. Feferman. Constructive theories of functions and classes. In: *Logic Colloquium '78*, 159–224, 1979
- [9] H. Friedman. The consistency of classical set theory relative to a set theory with intuitionistic logic. *Journal of Symbolic Logic*, (2): 315–319, 1973
- [10] I. Lindström. A construction of non-well-founded sets within Martin-Löf's type theory. *Journal of Symbolic Logic*, Vol. 54, No. 1, pp. 57–64, 1989
- [11] A. Troelstra. Realizability. In: S. Buss, ed., *Handbook of Proof Theory*. North Holland, 1998: 407–474
- [12] A. Troelstra, D. van Dalen. *Constructivism in Mathematics*, v. I–II. North Holland, 1988
- [13] S. Tupailo. Realization of analysis into Explicit Mathematics. Technical report IAM-00-003, University of Bern, Switzerland, 15 p. To appear in the *Journal of Symbolic Logic*
- [14] S. Tupailo. Realization of Constructive Set Theory into Explicit Mathematics: a lower bound for impredicative Mahlo universe. Technical report IAM-00-004, University of Bern, Switzerland, 25 p. *Submitted for publication*