Monotone Inductive Definitions and consistency of *New Foundations*

Sergei Tupailo*
Institute of Cybernetics at TUT, Tallinn, Estonia
sergei@cs.ioc.ee

January 8, 2006
To appear in the ***Proceedings of Logic Colloquium 2005***

Abstract

In this paper we reduce the consistency problem for **NF** to consistency of a certain extension of Jensen's **NFU**. Working in **NFU**+*Pairing*, which is known to be consistent relative to Zermelo set theory, due to Jensen [19], we define a certain monotone operation **pw** and conclude that existence of its least fixpoint is sufficient to model **NF**.

1 Introduction. New Foundations

New Foundations, **NF**, is a system of set theory named after Quine's 1937 article [20] *"New foundations for mathematical logic"*, where it was introduced. The language $\mathcal{L}_\in$ of **NF** is the simple set-theoretic language, i.e. the usual first-order language with the only constants $=$ and $\in$. The logic is classical first-order with equality. The only non-logical axioms are *Extensionality* and *Stratified Comprehension* as described below.

Extensionality is an axiom

$$\textbf{Ext}: \qquad \forall x \forall y \left(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y\right).$$

Definition 1.1 Stratification *of a formula φ is an assignment of natural numbers to variables (both free and bound) in φ s.t. every atomic subformula $x = y$ of φ receives an assignment $x^n = y^n$, for some n, and every atomic subformula $x \in y$ of φ receives an assignment $x^m \in y^{m+1}$, for some m. A formula φ is* stratified *iff there exists a stratification of φ.*

Examples. The formula $x \in y \wedge y \in z$ is stratified, but the formula $x \in y \wedge y \in x$ is not.

Stratified Comprehension is an axiom scheme

$$\textbf{SCA}: \qquad \exists y \forall x \left(x \in y \leftrightarrow \varphi[x]\right),$$

for every stratified formula φ with y not free in φ.

It is known that **NF** is at least as strong as Simple Type Theory with Infinity, but **NF** is not known to be consistent, relative to any known extension of Zermelo-Fraenkel Set Theory, – see e.g. [23, 24, 26, 27, 19, 6, 11, 3, 15, 12, 13, 16, 17, 25, 10, 18].

*A part of this work was done during the author's visit to The Ohio State University, USA, whose support is gratefully acknowledged.

A thorough review by an anonymous referee is gratefully acknowledged.

There is a number of subsystems of **NF** which *are* known to be consistent. Perhaps the most famous of them is **NFU**, so called "**NF** with Urelements", introduced by Jensen 1969 [19]. **NFU** results from **NF** by restricting extensionality to non-empty sets, i.e. by replacing the axiom **Ext** by the following axiom

$$\mathbf{Ext}' : \qquad \forall x \forall y \left(\exists z (z \in x \vee z \in y) \wedge \forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y \right).$$

NFU, however, is surprisingly weak: a model of **NFU** can be constructed within Peano Arithmetic. One of the drawbacks of **NFU** is that, contrary to **NF**, it doesn't prove the axiom of *Infinity*. On the other hand, it was also shown by Jensen [19] that **NFU** is consistent with *Infinity*, as well as with *Infinity* and *Choice*, **AC**, notwithstanding **NF** refuting **AC**, according to Specker [26]. This time the consistency results are relative to a much stronger theory, Zermelo Set Theory with Separation restricted to Δ_0 formulae (also known as Mac Lane Set Theory), or, equivalently, Simple Type Theory with Infinity (see [19, Theorem 1 and Lemma 4]). There are further consistent extensions of **NFU**, forming a kind of "large cardinals program" in this set theory – see e.g. [19, 6, 17, 25]. It's worthwhile to note that appropriate **NF**- large cardinal axioms, when added to **NF**, or even to **NFU**, do allow one to model **ZF**: a good reference is [16].

This paper is an attempt to apply the so called *bisimulation* method in order to model **NF** in an appropriate extension of **NFU**. This method has been used in many different situations, when there was a need to satisfy *Extensionality* in a non-extensional, non-wellfounded, framework: basic references here are [1] and [2][1]. On the language part, in order to carry out necessary constructions, the only required addition to $\mathcal{L}_\in$ is a type-preserving *ordered pairing* function $\langle \cdot, \cdot \rangle$ built-in. The fact that this extension is equivalent (in **NF**) to having *Infinity* axiom was shown first by Rosser [24] (but see also Quine [21]), and in the context of **NFU** was employed by Holmes [15, 16]. When having this kind of pairing, it was easy to talk about finite *sequences*, *trees* and *bisimulations*, which are the key preparatory notions in the present paper. Working in **NFU** + *Pairing*, **NFUP**[2], we define a certain monotone operation **pw** acting on sets of trees and conclude the following:

Lemma 3.4 *Any set models all Equality axioms of* **NF**,

Lemma 3.19 *Any fixpoint of the* **pw** *operation models Stratified Comprehension,*

and

Lemma 3.18 *Any least fixpoint, in addition, models Extensionality.*

Thus, existence of a **pw**- least fixpoint is sufficient to model **NF**.

This connects us with the well-known **MID** principle, which asserts existence of least fixpoints of monotone operations and has been studied extensively in different areas of Mathematical Logic. For example, in Set Theory, many **ZF**- large cardinal axioms can be seen as the **MID** principle for particular monotone operations; in Proof Theory, much research has been done about the **MID** principle over Peano Arithmetic and subsystems of Analysis, for a start see [4]; in Computer Science, one manifestation of **MID** is various μ-calculi.

Related to all of the above, including *New Foundations*, is the study of **MID** in Feferman's Explicit Mathematics, **EM**: one can start from [9, 14, 22, 31]. Explicit Mathematics can be seen as an extension of the restriction of **NFUP** containing only two types, cf. [5]; for this reason the only set operations f one can talk about in **EM** are *type-preserving* (or *type level*), i.e. such that x and $f(x)$ must have the same type. However, since **EM** postulates many more set existence principles than just those provided for by *Stratified Comprehension*, the very question of consistency and strength of **MID** becomes very non-trivial; this question has been answered, positively. In **NFUP** in general, as well, **MID** for type level operations easily follows from *Stratified Comprehension*, but the consistency question seems to be much more difficult if the operation is not so. Anyway, for our operation **pw**, a positive answer would imply Consis(**NF**)[3].

[1] The referee has pointed out a similarity of our method to the development of the *isomorphism classes of well-founded extensional relations with top*, which was used more often in the **NF** literature, in particular for modelling **ZF** in extensions of **NFUP**. However, there are trees which are bisimilar but don't give rise to isomorphic relations in the above sense, so it's not clear to which extent these two methods realize "the same" structure.

[2] This theory apparently was first considered in Feferman [6], where its consistency, and of a strong extension thereof, was proved. Holmes [15] offered a way to implement Quine pair in the **NFU** environment, resulting in an interpretation of **NFUP** in **NFU** + *Infinity*. Conversely, *Infinity* is deducible from type level pair by the device due to Rosser [24].

[3] Observation made by the referee: For some other monotone operations, existence of least fixpoints is *inconsistent* with

2 Preliminary developments in NFUP: sequences, trees and bisimulations

Throughout this paper, **NFUP** will mean an extension of **NFU** as described in the Introduction by the *ordered pairing* operation built in. *Stratified Comprehension* **SCA** and restricted *Extensionality* $\mathbf{Ext}'$ axioms remain as above; now we describe a mechanism to include ordered pairing. To do this, we add to the language $\mathcal{L}_\in$ the ordered pairing $\langle\cdot,\cdot\rangle$ function constant and adjoin to the theory the following *Pairing* axiom:

$$\mathbf{Pair}: \qquad \langle x,u\rangle = \langle y,v\rangle \ \rightarrow\ x = y \wedge u = v.$$

Using *Pairing*, we can conservatively define projection functions p_0 and p_1. Namely, translate every atomic formula

$$\begin{aligned} \psi[\mathsf{p}_0(t)] \quad &:\Leftrightarrow \quad \exists x \exists y\,(t = \langle x,y\rangle \wedge \psi[x]),\\ \psi[\mathsf{p}_1(t)] \quad &:\Leftrightarrow \quad \exists x \exists y\,(t = \langle x,y\rangle \wedge \psi[y]). \end{aligned}$$

From this translation we see that p_0 and p_1 are inverses of $\langle\cdot,\cdot\rangle$:

$$\mathbf{Unpair}: \qquad \mathsf{p}_0(\langle x,y\rangle) = x \wedge \mathsf{p}_1(\langle x,y\rangle) = y.$$

The new extended language will be called $\mathcal{L}_{\mathbf{P}}$. The notion of *stratification* is adjusted in such a way that in the term $\langle s,t\rangle$ the components s and t must have the same type n, and then the whole term $\langle s,t\rangle$ is also assigned the type n. The requirements for $x^n = y^n$ and $x^m \in y^{m+1}$ of the Definition 1.1 are left intact, now relating to terms s, t instead of mere variables x, y. It follows that the type of $\mathsf{p}_0(t)$, $\mathsf{p}_1(t)$ must be the same as the type of t. Keep in mind that in the **SCA** axiom of **NFUP** the formula φ must be stratified in the new sense.

NFUP is formulated in $\mathcal{L}_{\mathbf{P}}$ and based on classical logic with equality. We set

$$\mathbf{NFUP} := \mathbf{Ext}' + \mathbf{SCA} + \mathbf{Pair}.$$

In this paper by default we will be reasoning in **NFUP**. V will denote the universal set $\{x \mid x = x\}$, and Λ the empty set $\{x \mid x \neq x\}$. We customarily define $\langle x_1,\ldots,x_n\rangle := \langle\langle x_1,\ldots,x_{n-1}\rangle, x_n\rangle$ for $n \geq 3$.

Having ordered pair at our disposal, we can define the *Cartesian product*, *relations* and *functions*. Namely,

Definition 2.1

$$\begin{aligned} x \times y \quad &:= \quad \{\langle u,v\rangle \mid u \in x \wedge v \in y\};\\ \mathrm{Rel} \quad &:= \quad \{R \mid \forall x{\in}R\, \exists y \exists z\, x = \langle y,z\rangle\};\\ \mathrm{dom}(R) \quad &:= \quad \{x \mid \exists y\, \langle x,y\rangle \in R\};\\ \mathrm{rng}(R) \quad &:= \quad \{y \mid \exists x\, \langle x,y\rangle \in R\};\\ \mathrm{Fun} \quad &:= \quad \{f \in \mathrm{Rel} \mid \forall x{\in}f \forall y{\in}f\,(\mathsf{p}_0 x = \mathsf{p}_0 y \rightarrow \mathsf{p}_1 x = \mathsf{p}_1 y)\};\\ f\!:\!x \mapsto y \quad &:\Leftrightarrow \quad f \in \mathrm{Fun} \wedge \mathrm{dom}(f) = x \wedge \mathrm{rng}(f) \subseteq y;\\ f(x) \quad &:= \quad \text{"the unique } y \text{ s.t. } \langle x,y\rangle \in f\text{" for } f \in \mathrm{Fun} \text{ and } x \in \mathrm{dom}(f). \end{aligned}$$

We define Frege integers in the standard way (see [16, p.79-80]). Namely, set

$$0 \quad := \quad \{\Lambda\}, \tag{1}$$
$$S(x) \quad := \quad \{y \cup \{z\} \mid y \in x \wedge z \notin y\}, \tag{2}$$

NFUP. For our operation **pw**, an inconsistency can be deduced if one adds unrestricted *Choice* to **NFUP**. Therefore, this could lead to a model of **NF** only after building very special models of **NFUP** which violate *Choice*.

and, finally,

$$\mathbb{N} \;:=\; \bigcap\{x \mid 0 \in x \wedge \forall y{\in}x\, S(y) \in x\}. \tag{3}$$

All Peano axioms hold for so defined $\mathbb{N}$. We use $1 := S(0)$. Addition $+$, subtraction $-$, etc., can be defined to satisfy the standard properties. For details of those developments, see e.g. [16, Ch.12]. Equally, we have access to (primitive) recursion and induction on $\mathbb{N}$:

Lemma 2.2 (Induction on $\mathbb{N}$, see [16, p.81])
If $X \subseteq \mathbb{N}$, $0 \in X$ and $\forall y{\in}X\, S(y) \in X$, then $X = \mathbb{N}$.

Lemma 2.3 (Recursion on $\mathbb{N}$, see [16, p.83])
If X is a set, x is an element of X, and $f{:}X \times \mathbb{N} \mapsto X$, then there exists a unique function $g{:}\mathbb{N} \mapsto X$ s.t. $g(0) = x$ and $g(S(k)) = f(g(k), k)$ for each $k \in \mathbb{N}$.

We can define a set Seq of sequences so that

$$\mathsf{Seq} = \{\langle x, y\rangle \mid (x = 0 \wedge y = 0) \vee (x = 1 \wedge y = \langle \mathsf{p}_0(y), \mathsf{p}_1(y)\rangle \wedge \mathsf{p}_0(y) \in \mathsf{Seq})\}.$$

To do this, by **SCA** one defines a set Seq_0 s.t.

$$\mathsf{Seq}_0 := \{\langle 0, 0\rangle\}, \tag{4}$$

and a function Seq_S s.t.

$$\mathsf{Seq}_\mathsf{S}(Y) := \{\langle 1, \langle y, z\rangle\rangle \mid y \in Y\}. \tag{5}$$

Then by recursion on $\mathbb{N}$ ($X := \{y \mid \exists z\, z \in y\}$, $x := \mathsf{Seq}_0$, $f(Y, n) := \mathsf{Seq}_\mathsf{S}(Y)$) one defines a function sq s.t.

$$\begin{cases} \mathsf{sq}(0) := \mathsf{Seq}_0, \\ \mathsf{sq}(S(n)) := \mathsf{Seq}_\mathsf{S}(\mathsf{sq}(n)). \end{cases} \tag{6}$$

Finally by **SCA** one sets

$$\mathsf{Seq} \;:=\; \bigcup \mathrm{rng}(\mathsf{sq}). \tag{7}$$

We abbreviate

$$\mathsf{nil} := \langle 0, 0\rangle.$$

Since the definition of Seq is inductive, we have the standard principles of induction and recursion on Seq:

Lemma 2.4 (Induction on Seq)
If $X \subseteq \mathsf{Seq}$, $\mathsf{nil} \in X$ and $\forall y{\in}X \forall u\, \langle 1, \langle y, u\rangle\rangle \in X$, then $X = \mathsf{Seq}$.

Proof. By induction on n we prove $\mathsf{sq}(n) \subseteq X$, for every $n \in \mathbb{N}$. Therefore $\mathsf{Seq} \subseteq X$. Since additionally we are given $X \subseteq \mathsf{Seq}$, by $\mathbf{Ext}'$ we obtain $X = \mathsf{Seq}$. □

Lemma 2.5 (Recursion on Seq)
If X is a set, x is an element of X, and $f{:}X \times \mathsf{Seq} \times \mathrm{V} \mapsto X$, then there exists a unique function $g{:}\mathsf{Seq} \mapsto X$ s.t. $g(\mathsf{nil}) = x$ and $g(\langle 1, \langle y, u\rangle\rangle) = f(g(y), y, u)$ for each $y \in \mathsf{Seq}$, $u \in \mathrm{V}$.

Proof. Define a function $H{:}\mathfrak{X} \mapsto \mathfrak{X}$, where $\mathfrak{X} := \{Y \subseteq \mathsf{Seq} \times X \mid \exists z\, z \in Y\}$, so that

$$H(A) := \{\langle\langle 1, \langle y, u\rangle\rangle, f(v, y, u)\rangle \mid \langle y, v\rangle \in A\}.$$

By recursion on $\mathbb{N}$ ($X' := \mathfrak{X}$, $x' := \{\langle\langle 0,0\rangle, x\rangle\}$, $f'(Y,n) := H(Y)$) define

$$\begin{cases} F(0) := \{\langle\langle 0,0\rangle, x\rangle\}, \\ F(S(n)) := \{\langle\langle 1, \langle y,u\rangle\rangle, f(v,y,u)\rangle \mid \langle y,v\rangle \in F(n)\}. \end{cases} \tag{8}$$

<u>*Claim 1.*</u> $\forall y \in \mathsf{sq}(n) \exists! v\, \langle y,v\rangle \in F(n)$.

/- By induction on n. Obvious when $n = 0$. Assume $n > 0$ and $y \in \mathsf{sq}(n)$. Then by (6) and (5) $y = \langle 1, \langle \mathsf{p}_0(\mathsf{p}_1(y)), \mathsf{p}_1(\mathsf{p}_1(y))\rangle\rangle$ with $\mathsf{p}_0(\mathsf{p}_1(y)) \in \mathsf{sq}(n-1)$. By IH $\exists! v_0\, \langle \mathsf{p}_0(\mathsf{p}_1(y)), v_0\rangle \in F(n-1)$. Consequently, by (8), $\exists! v\, \langle y,v\rangle \in F(n)$. -/

<u>*Claim 2.*</u> $\langle y,v\rangle \in F(n) \to y \in \mathsf{sq}(n)$.

/- By induction on n, using (8). -/

<u>*Claim 3.*</u> $y \in \mathsf{sq}(n) \wedge m > n \to y \notin \mathsf{sq}(m)$.

/- By induction on n. Obvious when $n = 0$, since $0 \neq 1$. Assume $n > 0$. If $y \in \mathsf{sq}(n) \wedge y \in \mathsf{sq}(m)$, then $\mathsf{p}_0(\mathsf{p}_1(y)) \in \mathsf{sq}(n-1) \wedge \mathsf{p}_0(\mathsf{p}_1(y)) \in \mathsf{sq}(m-1)$, which contradicts the IH. -/

Claims 1–3 show

$$\forall y \in \mathsf{Seq} \exists! n \in \mathbb{N} \exists! v\, \langle y,v\rangle \in F(n),$$

or

$$\forall y \in \mathsf{Seq} \exists! v\, \langle y,v\rangle \in \bigcup_{n\in\mathbb{N}} F(n).$$

Therefore

$$g := \{\langle y,v\rangle \mid y \in \mathsf{Seq} \wedge \langle y,v\rangle \in \bigcup_{n\in\mathbb{N}} F(n)\}$$

is a function defined on Seq. $\mathrm{rng}(g) \subseteq X$, $g(\mathsf{nil}) = x$ and $g(\langle 1, \langle y,u\rangle\rangle) = f(g(y), y, u)$ follow from (8).

Uniqueness of such a g is proved by induction on Seq.

□

One defines the *length* function $\mathsf{ln}\colon \mathsf{Seq} \mapsto \mathbb{N}$ by recursion on Seq to satisfy the following equations:

$$\begin{aligned} \mathsf{ln}(\mathsf{nil}) &:= 0, \\ \mathsf{ln}(\langle 1, \langle a,b\rangle\rangle) &:= \mathsf{ln}(a) + 1 : \end{aligned}$$

take in Lemma 2.5 $X := \mathbb{N}$, $x := 0$, and $f(k,a,b) := k+1$.

From the definition of ln above we immediately have (by induction on Seq, Lemma 2.4), for $c \in \mathsf{Seq}$,

$$\mathsf{ln}(c) = 0 \leftrightarrow c = \mathsf{nil}. \tag{9}$$

By recursion on $\mathbb{N}$ (Lemma 2.3) one defines the result of erasing the last k members from a sequence c. For this, we set

$$\begin{cases} \mathsf{rem}_0 &:= \{\langle x,x\rangle \mid x \in \mathsf{Seq}\}; \\ \mathsf{rem}_{k+1} &:= \{\langle x, \mathsf{nil}\rangle \mid x \in \mathsf{Seq} \wedge \mathsf{ln}(x) \leq k+1\} \\ & \bigcup \{\langle\langle 1, \langle a,b\rangle\rangle, y\rangle \mid a \in \mathsf{Seq} \wedge \mathsf{ln}(a) \geq k+1 \wedge \langle a,y\rangle \in \mathsf{rem}_k\}. \end{cases} \tag{10}$$

By induction on k we prove

$$\forall k \in \mathbb{N} \forall c \in \mathsf{Seq} \exists! d \in \mathsf{Seq}\, \langle c,d\rangle \in \mathsf{rem}_k,$$

i.e. all rem_k are functions $\mathsf{Seq} \mapsto \mathsf{Seq}$. We denote $\mathsf{rem}(c,k) = d$ for $\langle c,d\rangle \in \mathsf{rem}_k$. By induction on k (10) also gives us

$$\mathsf{rem}(c,k) = \mathsf{nil} \quad \text{if} \quad \mathsf{ln}(c) \leq k.$$

The operation rem allows us to define the k-th last element $(c)_k$ of a sequence c, $1 \leq k \leq \mathsf{ln}(c)$:

$$(c)_k \quad := \quad \mathsf{p}_1(\mathsf{p}_1(\mathsf{rem}(c, k-1))).$$

We also define, for $c \neq \mathsf{nil}$,

$$\mathsf{head}(c) := \mathsf{rem}(c, \mathsf{ln}(c) - 1).$$

The operation head, from a non-zero sequence c, gives a one-element sequence $\mathsf{head}(c)$ consisting of the first (from the beginning) member of c. We will also need a complementary operation, $\mathsf{bodyt}(c)$, the remainder from c after the head is "cut off":

by recursion on Seq, taking in Lemma 2.5 $X := \mathsf{Seq}$, $x := \mathsf{nil}$ and

$$f(d,a,b) := \begin{cases} \mathsf{nil} & \text{if } a = \mathsf{nil}, \\ \langle 1, \langle d, b\rangle\rangle & \text{otherwise,} \end{cases}$$

one defines $\mathsf{bodyt}(c)$ for $c \in \mathsf{Seq}$ in the following way:

$$\begin{aligned} \mathsf{bodyt}(\mathsf{nil}) &:= \mathsf{nil}, \\ \mathsf{bodyt}(\langle 1, \langle a, b\rangle\rangle) &:= f(\mathsf{bodyt}(a), a, b). \end{aligned}$$

Note that this definition yields

$$\mathsf{bodyt}(\langle 1, \langle a, b\rangle\rangle) = \begin{cases} \mathsf{nil} & \text{if } a = \mathsf{nil}, \\ \langle 1, \langle \mathsf{bodyt}(a), b\rangle\rangle & \text{otherwise.} \end{cases}$$

Now we define the *concatenation* operation $x * y$ by recursion on $y \in \mathsf{Seq}$ ($X := \mathsf{Seq}$, $f(z, y, u) := \langle 1, \langle z, u\rangle\rangle$):

$$\begin{aligned} x * \mathsf{nil} &:= x, \\ x * \langle 1, \langle y, u\rangle\rangle &:= \langle 1, \langle x * y, u\rangle\rangle, \end{aligned}$$

for $x \in \mathsf{Seq}$.

Observe that $x * y$ is a homogeneous function: all three variables must have the same type in any stratification of "$x * y = z$".

It's also a routine check that for any $c \in \mathsf{Seq}$, $c \neq \mathsf{nil}$,

$$\mathsf{head}(c) * \mathsf{bodyt}(c) = c. \tag{11}$$

Lemma 2.6 *Concatenation is associative, i.e.*

$$\forall x \in \mathsf{Seq} \forall y \in \mathsf{Seq} \forall z \in \mathsf{Seq}\ x * (y * z) = (x * y) * z.$$

Proof. By induction on z. □

Definition 2.7 (cf. [29, Def.2.1] and [30, Def.2])
By **SCA** *sets* $\sqsupseteq$ *and* $\sqsupseteq^1$ *are defined as below:*

$$\sqsupseteq\ := \{\langle x, y\rangle \mid x \in \mathsf{Seq} \wedge y \in \mathsf{Seq} \wedge \exists z \in \mathsf{Seq}\, (y * z = x)\},$$

$$\sqsupseteq^1\ := \{\langle x, y\rangle \mid x \in \mathsf{Seq} \wedge y \in \mathsf{Seq} \wedge \exists z \in \mathsf{Seq}\, (\mathsf{ln}(z) = 1 \wedge y * z = x)\}.$$

We will use $x \sqsupseteq y$ and $x \sqsupseteq^1 y$ in place of $\langle x, y\rangle \in \sqsupseteq$ and $\langle x, y\rangle \in \sqsupseteq^1$, resp.

Lemma 2.8

$$\forall x \in \mathsf{Seq} \forall y \in \mathsf{Seq} \forall z \in \mathsf{Seq}\, (y \sqsupseteq z \to x * y \sqsupseteq x * z).$$

Proof. By associativity (Lemma 2.6). □

A *tree* is a non-empty set of sequences, downwards closed with respect to the $\sqsupseteq$-relation:

Definition 2.9 (cf. [29, Def.2.3] and [30, Def.3])
By **SCA** *we define*

$$\mathsf{Tree} := \{T \subseteq \mathsf{Seq} \mid \mathsf{nil} \in T \wedge \forall y{\in}T \forall z\,(y \sqsupseteq z \to z \in T)\}.$$

If $T \in \mathsf{Tree}$, $x \sqsupseteq_T y$ *and* $x \sqsupseteq^1_T y$ *will mean* $x \in T \wedge y \in T \wedge x \sqsupseteq y$ *and* $x \in T \wedge y \in T \wedge x \sqsupseteq^1 y$, *resp. With these notations we will make a familiar use of bounded quantifiers: e.g.* $\forall x' \sqsupseteq^1_T x\, \varphi[x']$ *will mean* $\forall x'\,(x' \sqsupseteq^1_T x \to \varphi[x'])$.

Definition 2.10 *If* $T, T' \in \mathsf{Tree}$ *we say that* R *is a* bisimulation *between* T *and* T', *written* $BS(R,T,T')$, *iff* $R \subseteq T \times T'$, $\langle \mathsf{nil}, \mathsf{nil}\rangle \in R$, *and the following holds:*

$$\begin{gathered}\forall x{\in}T \forall y{\in}T' \big(\langle x,y\rangle \in R \;\longrightarrow \\ \forall x' {\sqsupseteq}^1_T x \exists y' {\sqsupseteq}^1_{T'} y\, \langle x',y'\rangle \in R \;\bigwedge\; \forall y' {\sqsupseteq}^1_{T'} y \exists x' {\sqsupseteq}^1_T x\, \langle x',y'\rangle \in R\big)\,.\end{gathered} \tag{12}$$

Definition 2.11 *We define*

$$T \cong T' :\Leftrightarrow T \in \mathsf{Tree} \wedge T' \in \mathsf{Tree} \wedge \exists R\, BS(R,T,T').$$

Lemma 2.12 $\cong$ *is an equivalence relation on* Tree, *i.e. for every* $T, T', T'' \in \mathsf{Tree}$ *the following hold:*

$$T \cong T; \tag{13}$$

$$T \cong T' \to T' \cong T; \tag{14}$$

$$T \cong T' \wedge T' \cong T'' \to T \cong T''. \tag{15}$$

Proof. (13) is provided by the identity relation on T: $\{\langle x,x\rangle \mid x \in T\}$. (14) is provided by the inverse relation: when $BS(R,T,T')$, set $R^{-1} := \{\langle y,x\rangle \mid \langle x,y\rangle \in R\}$. (15) is provided by the composition: when $BS(R_1,T,T') \wedge BS(R_2,T',T'')$, set $R_2 \circ R_1 := \{\langle x,z\rangle \mid \exists y(\langle x,y\rangle \in R_1 \wedge \langle y,z\rangle \in R_2)\}$. □

Definition 2.13 *For* $T \in \mathsf{Tree}$ *and* $x \in T$ *we define*

$$T_x := \{y \in \mathsf{Seq} \mid x * y \in T\}.$$

Lemma 2.14 *If* $T \in \mathsf{Tree}$ *and* $x \in T$ *then* $T_x \in \mathsf{Tree}$.

Proof. By Definition 2.9 we need to prove

$$T_x \subseteq \mathsf{Seq} \wedge \mathsf{nil} \in T_x \wedge \forall y{\in}T_x \forall z (y \sqsupseteq z \to z \in T_x).$$

$T_x \subseteq \mathsf{Seq}$ is immediate from Definition 2.13. $\mathsf{nil} \in T_x$ follows from $x * \mathsf{nil} = x \in T$. Now assume $y \in T_x \wedge y \sqsupseteq z$. We then have

$$x * y \in T$$

and by Lemma 2.8

$$x * y \sqsupseteq x * z.$$

Since $T \in \mathsf{Tree}$, it must hold $x * z \in T$, i.e. $z \in T_x$.

□

Lemma 2.15 *If* $T, T' {\in} \mathsf{Tree}$, $BS(R,T,T')$ *and* $\langle x,y\rangle \in R$ *then* $T_x \cong T'_y$.

Proof. $T_x, T'_y \in \mathsf{Tree}$ by Lemma 2.14. Consider

$$R' := \{\langle x', y'\rangle \mid \langle x * x', y * y'\rangle \in R\}.$$

From $R \subseteq T \times T'$ we have $R' \subseteq T_x \times T'_y$. From $\langle x, y\rangle \in R$ we have $\langle \mathsf{nil}, \mathsf{nil}\rangle \in R'$. Finally,

$$\forall x' \in T_x \forall y' \in T'_y \left(\langle x', y'\rangle \in R' \longrightarrow \right.$$
$$\left. \forall x'' \sqsupseteq^1_{T_x} x' \exists y'' \sqsupseteq^1_{T'_y} y' \, \langle x'', y''\rangle \in R' \;\bigwedge\; \forall y'' \sqsupseteq^1_{T'_y} y' \exists x'' \sqsupseteq^1_{T_x} x' \, \langle x'', y''\rangle \in R' \right)$$

follows from the condition (12), so that we can conclude $BS(R', T_x, T'_y)$.

□

Lemma 2.16 *If* $T, T' \in \mathsf{Tree}$ *and* $T \cong T'$ *then*

$$\forall x \, (\langle \mathsf{nil}, x\rangle \in T \to \exists y \, (\langle \mathsf{nil}, y\rangle \in T' \wedge T_{\langle \mathsf{nil}, x\rangle} \cong T'_{\langle \mathsf{nil}, y\rangle}))$$
$$\bigwedge \quad \forall y \, (\langle \mathsf{nil}, y\rangle \in T' \to \exists x \, (\langle \mathsf{nil}, x\rangle \in T \wedge T_{\langle \mathsf{nil}, x\rangle} \cong T'_{\langle \mathsf{nil}, y\rangle})).$$

Proof. Let $T, T' \in \mathsf{Tree}$ and $BS(R, T, T')$. By the Definition 2.10 we have $\langle \mathsf{nil}, \mathsf{nil}\rangle \in R$ and

$$\forall x \sqsupseteq^1_T \mathsf{nil} \exists y \sqsupseteq^1_{T'} \mathsf{nil} \; \langle x, y\rangle \in R \;\bigwedge\; \forall y \sqsupseteq^1_{T'} \mathsf{nil} \exists x \sqsupseteq^1_T \mathsf{nil} \; \langle x, y\rangle \in R.$$

The claim now follows from Lemma 2.15. □

Lemma 2.17

$$\forall T \in \mathsf{Tree} \forall T' \in \mathsf{Tree} \quad \Big(\forall x \, (\langle \mathsf{nil}, x\rangle \in T \to \exists y \, (\langle \mathsf{nil}, y\rangle \in T' \wedge T_{\langle \mathsf{nil}, x\rangle} \cong T'_{\langle \mathsf{nil}, y\rangle}))$$
$$\bigwedge \forall y \, (\langle \mathsf{nil}, y\rangle \in T' \to \exists x \, (\langle \mathsf{nil}, x\rangle \in T \wedge T_{\langle \mathsf{nil}, x\rangle} \cong T'_{\langle \mathsf{nil}, y\rangle})) \quad \to \quad T \cong T' \Big).$$

Proof. Given

$$T \in \mathsf{Tree} \wedge T' \in \mathsf{Tree}$$

and

$$\forall x \, (\langle \mathsf{nil}, x\rangle \in T \to \exists y \, (\langle \mathsf{nil}, y\rangle \in T' \wedge T_{\langle \mathsf{nil}, x\rangle} \cong T'_{\langle \mathsf{nil}, y\rangle})) \tag{16}$$
$$\bigwedge \quad \forall y \, (\langle \mathsf{nil}, y\rangle \in T' \to \exists x \, (\langle \mathsf{nil}, x\rangle \in T \wedge T_{\langle \mathsf{nil}, x\rangle} \cong T'_{\langle \mathsf{nil}, y\rangle})), \tag{17}$$

set

$$R \quad := \quad \{\langle \mathsf{nil}, \mathsf{nil}\rangle\} \bigcup \{\langle x, y\rangle \mid x \in T - \{\mathsf{nil}\} \wedge y \in T' - \{\mathsf{nil}\} \wedge T_x \cong T'_y\}. \tag{18}$$

<u>*Claim.*</u> *R is a bisimulation between T and T'.*

/- From (18) we immediately have $R \subseteq T \times T'$ and $\langle \mathsf{nil}, \mathsf{nil}\rangle \in R$. We must now show

$$\forall x \in T \forall y \in T' \left(\langle x, y\rangle \in R \longrightarrow \forall x' \sqsupseteq^1_T x \exists y' \sqsupseteq^1_{T'} y \, \langle x', y'\rangle \in R \;\bigwedge\; \forall y' \sqsupseteq^1_{T'} y \exists x' \sqsupseteq^1_T x \, \langle x', y'\rangle \in R \right). \tag{19}$$

Fix $x \in T$, $y \in T'$. First consider the case $x = \mathsf{nil} = y$. Fix $x' \sqsupseteq^1_T \mathsf{nil}$. By (16) $\exists y' \sqsupseteq^1_{T'} \mathsf{nil} \; T_{x'} \cong T'_{y'}$. By (18) $\langle x', y'\rangle \in R$ for these x', y'. Similarly if we start with $y' \sqsupseteq^1_{T'} \mathsf{nil}$.

Observe that (18) implies $\langle x, y\rangle \in R \to (x \neq \mathsf{nil} \leftrightarrow y \neq \mathsf{nil})$. So it remains to consider the case $\langle x, y\rangle \in R \wedge x \neq \mathsf{nil} \neq y$. Assuming $x \neq \mathsf{nil} \neq y$, $\langle x, y\rangle \in R$ yields $T_x \cong T'_y$. By Lemma 2.16

$$\forall x' \, (\langle \mathsf{nil}, x'\rangle \in T_x \to \exists y' \, (\langle \mathsf{nil}, y'\rangle \in T'_y \wedge T_{\langle x, x'\rangle} \cong T'_{\langle y, y'\rangle}))$$
$$\bigwedge \quad \forall y' \, (\langle \mathsf{nil}, y'\rangle \in T'_y \to \exists x' \, (\langle \mathsf{nil}, x'\rangle \in T_x \wedge T_{\langle x, x'\rangle} \cong T'_{\langle y, y'\rangle})),$$

i.e.

$$\forall x' \sqsupseteq^1_T x \exists y' \sqsupseteq^1_{T'} y \, T_{x'} \cong T'_{y'} \quad \bigwedge \quad \forall y' \sqsupseteq^1_{T'} y \exists x' \sqsupseteq^1_T x \, T_{x'} \cong T'_{y'},$$

which yields the conclusion of (19).

-/

□

Now, if

$$T = \{\mathsf{nil}\} \bigcup \{\langle \mathsf{nil}, y_1, \ldots, y_n\rangle \in T\} \in \mathsf{Tree},$$

by $\breve{T}$ we want to denote a tree

$$\{\mathsf{nil}\} \bigcup \{\langle \mathsf{nil}, \{y_1\}, \ldots, \{y_n\}\rangle \mid \langle \mathsf{nil}, y_1, \ldots, y_n\rangle \in T\}.$$

For establishing properties of $\breve{T}$, we will use the **NFU**-fact

$$\forall x \forall y \ (x = y \leftrightarrow \{x\} = \{y\}). \tag{20}$$

The exact definitions are below.

Definition 2.18 *Set*

$$\begin{array}{rcl} =_0 & := & \{\langle \mathsf{nil}, \mathsf{nil}\rangle\}, \\ =_1 & := & \{\langle \{p\}, q\rangle \mid p \in \mathsf{Seq} \wedge q \in \mathsf{Seq} \wedge \mathsf{ln}(p) = 1 \wedge \mathsf{ln}(q) = 1 \wedge \{(p)_1\} = (q)_1\}, \\ =_{k+2} & := & \{\langle \{p\}, q\rangle \mid p \in \mathsf{Seq} \wedge q \in \mathsf{Seq} \wedge \mathsf{ln}(p) > 1 \wedge \mathsf{ln}(q) > 1 \wedge \{(p)_1\} = (q)_1 \wedge \langle \{\mathsf{rem}(p,1)\}, \mathsf{rem}(q,1)\rangle \in =_{k+1}\}. \end{array}$$

By recursion on $\mathbb{N}$ *(Lemma 2.3) there exists a function* g *s.t.*

$$\forall k \in \mathbb{N} \, g(k) = =_k \, .$$

Finally we set

$$p^+ = q \quad :\Leftrightarrow \quad p = \mathsf{nil} \wedge q = \mathsf{nil} \ \vee \ \exists k \in \mathbb{N} - \{0\} \, \langle \{p\}, q\rangle \in g(k).$$

Definition 2.19 *For* $T \in \mathsf{Tree}$ *we define*

$$\breve{T} := \{q \in \mathsf{Seq} \mid \exists p \in T \, p^+ = q\}.$$

Lemma 2.20

$$\forall p \in \mathsf{Seq} \exists! q \in \mathsf{Seq} \ p^+ = q.$$

Proof. By induction on Seq, using the facts (9), (20) and the axiom **Pair**. □

Lemma 2.21

$$\forall T \in \mathsf{Tree} \exists! U \in \mathsf{Tree} \ U = \breve{T}.$$

Proof. Use the Definition 2.19, Lemma 2.20, Definition 2.9 and the *Equality* axioms of **NFUP**. □

Lemma 2.22 *For* $T_1, T_2 \in \mathsf{Tree}$ *it holds:*

$$T_1 \cong T_2 \leftrightarrow \breve{T}_1 \cong \breve{T}_2.$$

Proof. It suffices to use the equivalence

$$BS(R, T_1, T_2) \leftrightarrow BS(\breve{R}, \breve{T}_1, \breve{T}_2),$$

where

$$\breve{R} := \{\langle q_1, q_2\rangle \mid \langle p_1, p_2\rangle \in R \wedge p_1^+ = q_1 \wedge p_2^+ = q_2\},$$

and note that both R and $\breve{R}$ are definable from each other in a stratified way. □

3 Modelling NF

Definition 3.1 *We define*

$$S \mathrel{\breve{\in}} T \quad :\Leftrightarrow \quad S \in \mathsf{Tree} \wedge T \in \mathsf{Tree} \wedge \exists x \left(\langle \mathsf{nil}, x \rangle \in T \wedge \breve{S} \cong T_{\langle \mathsf{nil}, x \rangle} \right).$$

Lemma 3.2 *For $S, S', T, T' \in \mathsf{Tree}$ the following hold:*
(1) $S \cong S' \wedge S \mathrel{\breve{\in}} T \to S' \mathrel{\breve{\in}} T$;
(2) $T \cong T' \wedge S \mathrel{\breve{\in}} T \to S \mathrel{\breve{\in}} T'$.

Proof. (1) follows from Lemmata 2.22 and 2.12. (2) follows from the Definition 3.1, Lemma 2.16 and Lemma 2.12. □

Definition 3.3 *Let φ be an $\mathcal{L}_\in$-formula and $\mathfrak{Z}$ be a set. By $\varphi^{\mathfrak{Z}}$ we denote the formula obtained from φ by replacing $=$ by $\cong$, $\in$ by $\breve{\in}$, and all quantifiers Qz by $QZ \in \mathfrak{Z}$.*
When φ is a statement, we say that $\mathfrak{Z}$ satisfies *φ, $\mathfrak{Z} \models \varphi$, iff $\varphi^{\mathfrak{Z}}$ holds.*

Lemma 3.4 *Let $\varphi(x, y_1, \dots, y_k)$ be a formula of $\mathcal{L}_\in$ with all free variables shown and $\mathfrak{Z}$ be a set. Let $Y_i \in \mathsf{Tree}$ for all $1 \le i \le k$. Let $X_1, X_2 \in \mathsf{Tree}$ and $X_1 \cong X_2$. Then*

$$\varphi^{\mathfrak{Z}}[X_1] \leftrightarrow \varphi^{\mathfrak{Z}}[X_2].$$

In other words, any set $\mathfrak{Z}$ satisfies the Equality axioms of **NF**.

Proof. By induction on φ. The atomic case follows from Lemmata 2.12 and 3.2. □

Lemma 3.5 *The defining formulae in the Definitions 2.11 and 3.1 are stratified. In any stratification of $T \cong T'$, T and T' must have the same type, and in any stratification of $S \mathrel{\breve{\in}} T$, the type of T must be 1 higher than the type of S.*

Proof. By inspection. □

Lemma 3.6 *$\varphi^{\mathfrak{Z}}$ satisfies Separation for any stratified φ, i.e. if $\varphi[x]$ is a stratified formula of $\mathcal{L}_\in$ and $\mathfrak{Z}$ is a set, then*

$$\exists Y \forall X \; \left(X \in Y \leftrightarrow X \in \mathfrak{Z} \wedge \varphi^{\mathfrak{Z}}[X] \right). \tag{21}$$

Proof. In view of Lemma 3.5, the only obstacle why the formula $X \in \mathfrak{Z} \wedge \varphi^{\mathfrak{Z}}[X]$ could be unstratified is that it might contain several occurrences of the variable $\mathfrak{Z}$. Let $\psi^{\mathfrak{Z}_1 \cdots \mathfrak{Z}_n}[X]$ be a new formula, obtained from $X \in \mathfrak{Z} \wedge \varphi^{\mathfrak{Z}}[X]$ by replacing each occurrence of $\mathfrak{Z}$ by occurrence of a new variable $\mathfrak{Z}_i$. Then the formula $\psi^{\mathfrak{Z}_1 \cdots \mathfrak{Z}_n}[X]$ is stratified. By **SCA**, we have

$$\forall \mathfrak{Z}_1 \dots \forall \mathfrak{Z}_n \, \exists Y \forall X \; \left(X \in Y \leftrightarrow \psi^{\mathfrak{Z}_1 \cdots \mathfrak{Z}_n}[X] \right). \tag{22}$$

Substituting now $\mathfrak{Z}$ for $\mathfrak{Z}_1, \dots, \mathfrak{Z}_n$, we obtain (21). □

Now we introduce the following construction. If

$$T = \{\mathsf{nil}\} \bigcup \{\langle \mathsf{nil}, y_1, \dots, y_n \rangle \in T\} \in \mathsf{Tree},$$

by $\overline{T}$ we want to denote a tree

$$\{\mathsf{nil}\} \bigcup \{\langle \mathsf{nil}, T, \{y_1\}, \dots, \{y_n\} \rangle\rangle \mid \langle \mathsf{nil}, y_1, \dots, y_n \rangle \in T\}.$$

The exact definition is below.

Definition 3.7 *For $T \in \mathsf{Tree}$ we define:*

$$\overline{T} \quad := \quad \{\mathsf{nil}\} \bigcup \{\langle \mathsf{nil}, T\rangle * q \mid q \in \breve{T}\}.$$

Lemma 3.8 *For any $T \in \mathsf{Tree}$ it holds*

$$\overline{T} \in \mathsf{Tree} \ \wedge \ \overline{T}_{\langle \mathsf{nil},T\rangle} = \breve{T}.$$

Proof is straightforward, using Definitions 2.19, 3.7 and the axiom $\mathbf{Ext}'$. □

Definition 3.9 *For any $\mathfrak{Y} \subseteq \mathsf{Tree}$ we define*

$$\mathfrak{Y}^* := \{\mathsf{nil}\} \cup \bigcup \{\overline{T} \mid T \in \mathfrak{Y}\}.$$

Lemma 3.10 *For any $\mathfrak{Y} \subseteq \mathsf{Tree}$ we have $\mathfrak{Y}^* \in \mathsf{Tree}$ and*

$$\forall T \left(\langle \mathsf{nil}, T\rangle \in \mathfrak{Y}^* \to \mathfrak{Y}^*_{\langle \mathsf{nil},T\rangle} = \breve{T} \wedge T \in \mathfrak{Y}\right). \tag{23}$$

Proof. $\mathfrak{Y}^* \in \mathsf{Tree}$ is obvious from the definition of $\mathfrak{Y}^*$. For (23) we additionally employ Lemma 3.8. □

Lemma 3.11

$$\forall \mathfrak{Y} \subseteq \mathsf{Tree} \exists ! T \in \mathsf{Tree} \ T = \mathfrak{Y}^*.$$

Proof. Existence follows from Lemma 3.10. Uniqueness follows from the *Equality* axioms of **NFUP**. □

Definition 3.12 *For any $\mathfrak{Z} \subseteq \mathsf{Tree}$ we define*

$$\mathbf{pw}(\mathfrak{Z}) := \{\mathfrak{Y}^* \mid \mathfrak{Y} \subseteq \mathfrak{Z}\}.$$

Lemma 3.13

$$\forall \mathfrak{Z} \subseteq \mathsf{Tree} \exists ! \mathfrak{W} \subseteq \mathsf{Tree} \ \mathfrak{W} = \mathbf{pw}(\mathfrak{Z}).$$

Proof. Existence follows from **SCA** and Lemma 3.11. Uniqueness follows from the *Equality* axioms of **NFUP**. □

Lemma 3.14 *The operation* **pw** *is monotone on* Tree, *i.e.*

$$\forall \mathfrak{Z}_1 \subseteq \mathsf{Tree} \forall \mathfrak{Z}_2 \subseteq \mathsf{Tree} \left(\mathfrak{Z}_1 \subseteq \mathfrak{Z}_2 \to \mathbf{pw}(\mathfrak{Z}_1) \subseteq \mathbf{pw}(\mathfrak{Z}_2)\right).$$

Proof. To show $\{\mathfrak{Z}^* \mid \mathfrak{Z} \subseteq \mathfrak{Z}_1\} \subseteq \{\mathfrak{Z}^* \mid \mathfrak{Z} \subseteq \mathfrak{Z}_2\}$, we observe that if $\mathfrak{Z} \subseteq \mathfrak{Z}_1$ then $\mathfrak{Z} \subseteq \mathfrak{Z}_2$, so $\mathfrak{Z}^* \in \mathbf{pw}(\mathfrak{Z}_2)$. □

Definition 3.15
1. *A set $\mathfrak{Z} \subseteq \mathsf{Tree}$ is called a* (**pw**-) fixpoint *iff* $\mathbf{pw}(\mathfrak{Z}) \subseteq \mathfrak{Z}$.
2. *A set $\mathfrak{Z} \subseteq \mathsf{Tree}$ is called a* (**pw**-) least fixpoint *iff it is a fixpoint and* $\forall \mathfrak{Y} \subseteq \mathsf{Tree} \left(\mathbf{pw}(\mathfrak{Y}) \subseteq \mathfrak{Y} \to \mathfrak{Z} \subseteq \mathfrak{Y}\right)$.

Lemma 3.16 *If $\mathfrak{Z}$ is a least fixpoint then $\mathfrak{Z} = \mathbf{pw}(\mathfrak{Z})$.*

Proof. Since $\Lambda^* \in \mathbf{pw}(\mathfrak{Z})$, by $\mathbf{Ext}'$ it's sufficient to show

$$\mathbf{pw}(\mathfrak{Z}) \subseteq \mathfrak{Z} \tag{24}$$

and

$$\mathfrak{Z} \subseteq \mathbf{pw}(\mathfrak{Z}). \tag{25}$$

(24) follows from the fact that $\mathfrak{Z}$ is a fixpoint. Since the operation **pw** is monotone (Lemma 3.14), we obtain

$$\mathbf{pw}(\mathbf{pw}(\mathfrak{Z})) \subseteq \mathbf{pw}(\mathfrak{Z}),$$

i.e. $\mathbf{pw}(\mathfrak{Z})$ is also a fixpoint. But since $\mathfrak{Z}$ is a *least* fixpoint, we obtain (25).

□

Lemma 3.17 *A least fixpoint, if exists, is unique.*

Proof. Let $\mathfrak{Z}_1$ and $\mathfrak{Z}_2$ be two least fixpoints. By Lemma 3.16 $\mathfrak{Z}_1 = \mathbf{pw}(\mathfrak{Z}_1)$ and $\mathfrak{Z}_2 = \mathbf{pw}(\mathfrak{Z}_2)$. Then we also have $\Lambda^* \in \mathfrak{Z}_1$ and $\Lambda^* \in \mathfrak{Z}_2$. Since $\mathfrak{Z}_1$ and $\mathfrak{Z}_2$ are both least fixpoints, $\mathfrak{Z}_1 \subseteq \mathfrak{Z}_2$ and $\mathfrak{Z}_2 \subseteq \mathfrak{Z}_1$ both hold. It remains to apply the $\mathbf{Ext}'$ axiom of **NFUP**. □

Lemma 3.18 *If $\mathfrak{Z}$ is a least fixpoint then the following holds:*

$$\forall T \in \mathfrak{Z} \forall T' \in \mathfrak{Z} \left(\forall S \in \mathfrak{Z} \, (S \,\breve{\in}\, T \leftrightarrow S \,\breve{\in}\, T') \to T \cong T'\right).$$

In other words, any least fixpoint satisfies the Extensionality axiom of **NF**.

Proof. Given

$$T \in \mathfrak{Z} \wedge T' \in \mathfrak{Z} \wedge \forall S \in \mathfrak{Z} \, (S \,\breve{\in}\, T \leftrightarrow S \,\breve{\in}\, T'), \tag{26}$$

first we observe, since $\mathfrak{Z} \subseteq \mathsf{Tree}$, that

$$T \in \mathsf{Tree} \wedge T' \in \mathsf{Tree}. \tag{27}$$

Now we aim to show

$$\forall x \, (\langle \mathsf{nil}, x \rangle \in T \to \exists y \, (\langle \mathsf{nil}, y \rangle \in T' \wedge T_{\langle \mathsf{nil}, x \rangle} \cong T'_{\langle \mathsf{nil}, y \rangle})) \tag{28}$$

$$\bigwedge \quad \forall y \, (\langle \mathsf{nil}, y \rangle \in T' \to \exists x \, (\langle \mathsf{nil}, x \rangle \in T \wedge T_{\langle \mathsf{nil}, x \rangle} \cong T'_{\langle \mathsf{nil}, y \rangle})). \tag{29}$$

From (26) we have

$$\forall S \in \mathfrak{Z} \, (S \,\breve{\in}\, T \leftrightarrow S \,\breve{\in}\, T'). \tag{30}$$

In order to prove (28), assume $\langle \mathsf{nil}, x \rangle \in T$. Since $T \in \mathfrak{Z}$ and $\mathfrak{Z} = \mathbf{pw}(\mathfrak{Z})$ (Lemma 3.16), we have $T \in \mathbf{pw}(\mathfrak{Z})$, i.e.

$$\exists \mathfrak{Y} \subseteq \mathfrak{Z} \; \mathfrak{Y}^* = T. \tag{31}$$

By Lemma 3.10

$$x \in \mathfrak{Y} \wedge \mathfrak{Y}^*_{\langle \mathsf{nil}, x \rangle} = \breve{x}, \tag{32}$$

which implies

$$x \in \mathfrak{Z} \wedge T_{\langle \mathsf{nil}, x \rangle} = \breve{x}. \tag{33}$$

Then we must have $x \,\breve{\in}\, T$, and then by (30) $x \,\breve{\in}\, T'$, i.e.

$$\exists y \left(\langle \mathsf{nil}, y \rangle \in T' \wedge \breve{x} \cong T'_{\langle \mathsf{nil}, y \rangle}\right). \tag{34}$$

From (33) and (34) we obtain

$$T_{\langle \mathsf{nil}, x \rangle} \cong T'_{\langle \mathsf{nil}, y \rangle}$$

for the abovementioned x, y.

For (29), we proceed in the similar manner, now employing the direction $\leftarrow$ of (30).

This establishes (28) and (29), and hence, by Lemma 2.17,

$$T \cong T'.$$

□

Comment. Does the operation **pw** have fixpoints? Yes, – for example the sets Tree, $\mathbf{pw}(\mathsf{Tree})$, $\mathbf{pw}(\mathbf{pw}(\mathsf{Tree}))$, ... But we don't know whether it's consistent to assume that it has a *least* fixpoint.

Lemma 3.19 *Any fixpoint satisfies* **SCA** *of* **NF**.

Proof. Let $\mathfrak{Z}$ be a fixpoint. Let $\varphi(x, y_1, \ldots, y_k)$ be a stratified formula of $\mathcal{L}_\in$ with all free variables shown. Let $Y_i \in \mathfrak{Z}$ for all $1 \le i \le k$. We need to prove

$$\exists \mathfrak{Y}^* \in \mathfrak{Z} \forall X \in \mathfrak{Z} \left(X \breve{\in} \mathfrak{Y}^* \leftrightarrow \varphi^{\mathfrak{Z}}(X, Y_1, \ldots, Y_k) \right). \tag{35}$$

By Lemma 3.6 set

$$\mathfrak{Y} := \{X \in \mathfrak{Z} \mid \varphi^{\mathfrak{Z}}[X]\}. \tag{36}$$

Defining $\mathfrak{Y}^*$ as in Definition 3.9 and using that $\mathfrak{Z}$ is a fixpoint, we conclude $\mathfrak{Y}^* \in \mathfrak{Z}$.
Now, assuming $T \in \mathfrak{Z}$, it remains to prove

$$T \breve{\in} \mathfrak{Y}^* \leftrightarrow \varphi^{\mathfrak{Z}}[T].$$

In $\to$ direction, assume $T \breve{\in} \mathfrak{Y}^*$. By Definition 3.1 this means

$$\exists T' \left(\langle \mathsf{nil}, T' \rangle \in \mathfrak{Y}^* \wedge \breve{T} \cong \mathfrak{Y}^*_{\langle \mathsf{nil}, T' \rangle} \right), \tag{37}$$

which by Lemma 3.10 implies

$$\exists T' \in \mathfrak{Y} \left(\breve{T} \cong \mathfrak{Y}^*_{\langle \mathsf{nil}, T' \rangle} = \breve{T}' \right). \tag{38}$$

By Lemma 2.22 we have now

$$T \cong T'. \tag{39}$$

Now from (36) and Lemma 3.4 we conclude $\varphi^{\mathfrak{Z}}[T]$.
In the converse direction, assume $\varphi^{\mathfrak{Z}}[T]$. Then by (36)

$$T \in \mathfrak{Y}, \tag{40}$$

and by Definition 3.9 and Lemma 3.8

$$T \breve{\in} \mathfrak{Y}^*. \tag{41}$$

□

Definition 3.20 *Let* **MID(pw)** *be the axiom saying*

There exists a least fixpoint of the **pw** *operation.*

Theorem 1 **NF** *is consistent relative to* **NFUP** + **MID(pw)**.

Proof. Follows from Lemmata 3.4, 3.19 and 3.18. □

References

[1] P. Aczel. *Non-Well-Founded Sets.* CSLI Lecture Notes No. 14, Stanford University, 1988

[2] J. Barwise, L. Moss. *Vicious Circles.* CSLI Lecture Notes No. 60, Stanford University, 1996

[3] M. Boffa. The consistency problem for NF. *Journal of Symbolic Logic* 42, pp. 215–220, 1977

[4] W. Buchholz, S. Feferman, W. Pohlers, W. Sieg. *Iterated Inductive Definitions and Subsystems of Analysis.* Lecture Notes in Mathematics 897, Springer, 1981

[5] A. Cantini. Relating Quine's NF to Feferman's EM. *Studia Logica* 62, pp. 141-163, 1999

[6] S. Feferman. Some formal systems for the unlimited theory of structures and categories. *Unpublished manuscript,* 52 pp., available at `http://math.stanford.edu/~feferman/papers/Unlimited.pdf`. *Abstract in the Journal of Symbolic Logic* 39, pp. 374–375, 1974

[7] S. Feferman. A language and axioms for explicit mathematics. In: Algebra and Logic, *Lecture Notes in Mathematics* 450: 87–139, 1975

[8] S. Feferman. Constructive theories of functions and classes. In: *Logic Colloquium '78,* J.N. Crossley (ed.), 159–224, 1979

[9] S. Feferman. Monotone inductive definitions. In: *The L.E.J. Brower Centenary Symposium,* A.S. Troelstra, D. van Dallen (eds.), North-Holland, pp. 77–89, 1982

[10] S. Feferman. Typical ambiguity: trying to have your cake and eat it too. In: G. Link (ed.), *One Hundred Years of Russell's Paradox,* de Gruyter, Berlin, 135–151, 2004

[11] H. Friedman. One Hundred and Two Problems in Mathematical Logic. *Journal of Symbolic Logic* 40(2): 113–129, 1975

[12] T. E. Forster. *Set Theory with a Universal Set,* second edition. Clarendon Press, Oxford, 1995

[13] T. E. Forster. Quine's NF, 60 years on. *American Mathematical Monthly,* vol. 104, no. 9, pp. 838–845, 1997

[14] T. Glass, M. Rathjen, A. Schlüter. On the proof-theoretic strength of monotone induction in explicit mathematics. *Annals of Pure and Applied Logic,* 85: 1-46, 1997

[15] M. R. Holmes. The Axiom of Anti-Foundation in Jensen's 'New Foundations with Ur-Elements.' *Bulletin de la Societe Mathematique de Belgique (serie B)* 43, pp. 167–179, 1991

[16] M. R. Holmes. *Elementary Set Theory with a Universal Set,* vol. 10 of the Cahiers du Centre de logique, Academia, Louvain-la-Neuve (Belgium), 242 pp., 1998

[17] M. R. Holmes. Strong axioms of infinity in NFU. *Journal of Symbolic Logic,* vol. 66, no. 1, pp. 87–116, 2001

[18] M. R. Holmes. *New Foundations home page.* `http://math.boisestate.edu/~holmes/holmes/nf.html`

[19] R. B. Jensen. On the consistency of a slight(?) modification of Quine's NF. *Synthese* 19, pp. 250-263, 1969

[20] W. V. Quine. New foundations for mathematical logic. *American Mathematical Monthly* 44, pp. 70–80, 1937

[21] W. V. Quine. On ordered pairs. *Journal of Symbolic Logic* 10, pp. 95–96, 1945

[22] M. Rathjen. Explicit Mathematics with monotone inductive definitions: a survey. In: W. Sieg et al. (eds.), *Reflections on the Foundations of Mathematics: Essays in Honour of Solomon Feferman,* Lecture Notes in Logic 15, pp. 329–346, 2002

[23] J. B. Rosser. On the consistency of Quine's new foundations for mathematical logic. *Journal of Symbolic Logic* 4, pp. 15–24, 1939

[24] J. B. Rosser. The axiom of infinity in Quine's New Foundations. *Journal of Symbolic Logic* 17, pp. 238–242, 1952

[25] R. Solovay. The consistency strength of NFUB. *Unpublished manuscript,* 39 pp., available at `http://front.math.ucdavis.edu/author/Solovay-R*&c=LO`, 1997

[26] E. P. Specker. The axiom of choice in Quine's new foundations for mathematical logic. *Proceedings of the National Academy of Sciences of the USA,* 39, pp. 972–975, 1953

[27] E. P. Specker. Typical ambiguity. In: E. Nagel (ed.), *Logic, methodology and philosophy of science,* Stanford University Press, pp. 116–123, 1962

[28] S. Tupailo. Realization of analysis into Explicit Mathematics. *Journal of Symbolic Logic* 66(4), 1848–1864, 2001

[29] S. Tupailo. Realization of Constructive Set Theory into Explicit Mathematics: a lower bound for impredicative Mahlo universe. *Annals of Pure and Applied Logic,* vol. 120/1–3, pp. 165–196, 2003

[30] S. Tupailo. On Non-wellfounded Constructive Set Theory: Construction of Non-wellfounded Sets in Explicit Mathematics. In: G. Mints, R. Muskens (eds.), *Games, Logic, and Constructive Sets,* 109–125, CSLI Publications, 2003

[31] S. Tupailo. On the intuitionistic strength of monotone inductive definitions. *Journal of Symbolic Logic* 69, no. 3, 790–798, 2004