A Finite Axiomatization of Inductive-Recursive Definitions

Peter Dybjer*and Anton Setzer†

May 16, 2005

Abstract

Induction-recursion is a schema which formalizes the principles for introducing new sets in Martin-Löf's type theory. It states that we may inductively define a set while simultaneously defining a function from this set into an arbitrary type by structural recursion. This extends the notion of an inductively defined set substantially and allows us to introduce universes and higher order universes (but not a Mahlo universe). In this article we give a finite axiomatization of inductive-recursive definitions. We prove consistency by constructing a set-theoretic model which makes use of one Mahlo cardinal.

1 Introduction

In this article we present an elegant, uniform method for introducing large sets in type theory. We draw on experience from proof theory, category theory, and set theory to formulate a compact, completely formal theory of inductive-recursive definitions, and to prove its consistency.

Induction-recursion is a schema for introducing new sets in type theory developed by Dybjer [7]. All the usual sets in Martin-Löf's type theory and practically all sets (data types), which are defined in analogy with it, are instances of this schema. Applications of induction-recursion include not only a variety of type-theoretic analogues of large cardinals (inaccessible cardinals, hyper-inaccessible cardinals, etc) but also various powerful notions needed for the type-theoretic formalization of metamathematics (such as reducibility predicates and logical relations for dependent types). Induction-recursion can also provide novel ways to formalize simple concepts such as the set of lists with distinct elements [7].

The original presentation of induction-recursion was as an external schema [7]. In this article we internalize this concept. The new theory has a special type of codes for inductive-recursive definitions. New sets defined by induction-recursion are introduced by deriving codes in this type. Therefore we achieve full precision of the concept of an inductive-recursive definition. The meta-theory becomes easier, as will be demonstrated by building a full function space model.

Ordinary dependent type theory with generalized inductive definitions (that is, Martin-Löf's type theory without universes) has a natural full function space interpretation in classical set theory [1, 8]. As shown by our construction of a set-theoretic model the step from inductive to inductive-recursive definitions in type theory is roughly analogous to moving from ordinary ZF set theory to ZF set theory with a Mahlo cardinal. The proof-theoretic strength of type theory increases accordingly when inductive-recursive definitions are added. The consistency of the

*Department of Mathematics and Computing Science, Chalmers University of Technology. Email: peterd@cs.chalmers.se

†Department of Mathematics, Uppsala University. Email: setzer@math.uu.se

theory is shown without assuming the positivity restriction on parameters needed for Dybjer's original realizability model of inductive-recursive definitions [7].

The new theory explains that induction-recursion can be viewed as a very general reflection principle: given finitely many (possibly infinitary) operations on a type D, we can construct by simultaneous induction-recursion a universe U with decoding function $\mathrm{T} : \mathrm{U} \to D$, which reflects each of the D-operations. This reflection principle can be expressed formally by a diagram which extends the initial algebra diagram used for categorical semantics of inductively defined sets. The resulting theory has been implemented in the *Half* system, a proof assistant for Martin-Löf's type theory developed by Coquand and Synek, see Cederquist [4].

Plan of the paper. In Section 2 we present Martin-Löf's Logical Framework. In Section 3 we recall how to use initial algebras for giving categorical semantics of inductive types in the simply typed lambda calculus. In Section 4 we discuss the step from induction to induction-recursion and how we need to modify the notion of an endofunctor Φ and of an initial Φ-algebra in order to capture the formal rules for induction-recursion. We then show how to give a finite axiomatization of inductive-recursive definitions by introducing a type of codes for such modified endofunctors. In Section 5 we show how to recover some well-known set constructors by giving appropriate codes. In Section 6 we build a set-theoretic model. In Section 7 we mention some related work.

2 An Extension of the Logical Framework

The Logical Framework (see [15]) has the following forms of judgements: Γ context, and A : type, $A = B$: type, $a : A$, $a = b : A$, depending on contexts Γ (written as $\Gamma \Rightarrow A$: type, etc.). We have set : type and if A : set, then A : type. The collection of types is closed under the formation of dependent function types written as $(x : A) \to B$, with elements formed by abstraction $(x : A)a$, application written in the form $a(b)$ and which has the η-rule. Types are also closed under the formation of dependent products written as $(x : A) \times B$, with elements $\langle a, b\rangle$, projections π_0 and π_1 and again the η-rule (surjective pairing). There is also the type 1, with unique element $\langle\rangle : 1$ and η-rule expressing, that if $a : 1$, then $a = \langle\rangle : 1$.

We will add a level between set and type, which we call stype for small types: stype : type. (The reason for the need for stype is discussed in [7].) If a : set then a : stype. Moreover, stype is also closed under dependent function types, dependent products and includes the one-element type. However, set itself will not be in stype.

Finally, in order to make it possible to code all constructors into one (see the remark on page 4), we add the set $\mathbb{B}$ of booleans with elements tt for true and ff for false and as elimination rule case distinction if a then b else $c : D$ for $a : \mathbb{B}$, D : type and $b, c : D$.

We also use some abbreviations, such as omitting the type in an abstraction, that is, writing $(x)a$ instead of $(x : A)a$, and writing repeated application as $a(b_1, \ldots, b_n)$ instead of $a(b_1) \cdots (b_n)$ and repeated abstraction as $(x_1 : A_1, \ldots, x_n : A_n)a$ instead of $(x_1 : A_1) \cdots (x_n : A_n)a$.

3 Inductive Types as Initial Algebras

Let us first consider the question of how to formalize inductive types in the setting of the simply typed λ-calculus. We shall consider *generalized inductive definitions* of types given by a finite number of constructors

$$\mathrm{intro}_i : \Phi_i(\mathrm{U}) \to \mathrm{U} \ ,$$

where Φ_i are *strictly positive* in the following restricted sense:

- The constant functor $\Phi(D) = 1$ is strictly positive. This is the base case corresponding to an introduction rule with no premises.
- If Ψ is strictly positive and A is an stype, then $\Phi(D) = A \times \Psi(D)$ is strictly positive. This corresponds to the addition of a *non-inductive*[1] premise.
- If Ψ is strictly positive and A is an stype, then $\Phi(D) = (A \to D) \times \Psi(D)$ is strictly positive. This corresponds to the addition of an *inductive* premise, where A corresponds to the hypotheses of this premise in a generalized inductive definition (and when $A = 1$ we have the special case of an *ordinary inductive definition*).

Note that all occurrences of U in $\Phi(\mathrm{U})$ are strictly positive in the standard sense that U does not occur to the left of an arrow in $\Phi(\mathrm{U})$.

Assume $\Phi_1, \ldots, \Phi_n$ are strictly positive functors, and let $\vec{\Phi} := (\Phi_1, \ldots, \Phi_n)$. Then the inductive type generated by $\vec{\Phi}$ can be captured categorically as an initial $\vec{\Phi}$-algebra, that is, a sequence of arrows ($i = 1, \ldots, n$)

$$\Phi_i(\mathrm{U}) \xrightarrow{\mathrm{intro}_i} \mathrm{U}$$

such that for any other $\vec{\Phi}$-algebra

$$\Phi_i(D) \xrightarrow{d_i} D$$

there is a unique arrow $\mathrm{T} : \mathrm{U} \longrightarrow D$, such that the following diagrams commute

$$\begin{tikzcd}
\Phi_i(\mathrm{U}) \arrow[r, "\mathrm{intro}_i"] \arrow[d, "\Phi_i(\mathrm{T})"'] & \mathrm{U} \arrow[d, "\mathrm{T}"] \\
\Phi_i(D) \arrow[r, "d_i"] & D
\end{tikzcd}$$

4 Inductive-Recursive Definitions

4.1 From Inductive to Inductive-Recursive Definitions

In the presence of dependent types more inductive definitions become possible. Let us look at some examples:

The set $\Sigma(A, B)$ has one constructor $\mathrm{p} : (x : A) \to (y : B(x)) \to \Sigma(A, B)$. It has two non-inductive arguments, where the type $B(x)$ of the second argument depends on the first premise $x : A$.

The well-ordering set $\mathrm{W}(A, B)$ has one constructor $\sup : (x : A) \to (y : B(x) \to \mathrm{W}(A, B)) \to \mathrm{W}(A, B)$. It has a first non-inductive argument x and a second $B(x)$-indexed inductive argument y. So the second argument depends on the first non-inductive argument.

Both are examples of inductive definitions (no simultaneously defined function participates in the definition yet). For this case later premises can only depend on earlier non-inductive premises, but not on earlier inductive premises. We cannot make use of inductive premises, because they only give information about the set we are currently defining.

To capture inductive definitions of sets in the presence of dependent types [8, 9], we thus only need to change the notion of a strictly positive functor Φ above by replacing the non-inductive case by:

[1] In [7] the terminology "non-recursive premise" was used, but "non-inductive premise" seems better in connection with induction-recursion, since it primarily has to do with the inductively defined set and not with the recursively defined function. Similarly we will use "inductive premise" instead of "recursive premise".

- If A is an stype, and Ψ_x is a strictly positive functor depending on $x : A$, then $\Phi(D) = (x : A) \times \Psi_x(D)$ is strictly positive.

We shall now replace the sequence of functors $(\Phi_1, \ldots, \Phi_n)$ by a single functor by defining $\Phi(D) := (x : \mathrm{N}_n) \times \Phi_x(D)$. In order to make this possible we need the existence of finite sets with n elements N_n. An easy observation shows that $\mathbb{B}$ and the empty set N_0 suffice. (It will however be possible to define N_0, see section 5).

In the case of inductive-recursive definitions however, a later premise may also depend on an earlier inductive premise. We consider the key example, the ordinary first universe U à la Tarski [12], which is defined inductively, while simultaneously defining the decoding function $\mathrm{T} : \mathrm{U} \to \mathrm{set}$ recursively. Consider one of its constructors, $\widehat{\Sigma} : (x : \mathrm{U}) \to (y : \mathrm{T}(x) \to \mathrm{U}) \to \mathrm{U}$ with the defining equality $\mathrm{T}(\widehat{\Sigma}(a, b)) = \Sigma(\mathrm{T}(a), \mathrm{T} \circ b) : \mathrm{set}$. Here we have two inductive premises: $x : \mathrm{U}$ (implicitly indexed by the one-element type 1) and $y : \mathrm{U}$ indexed by $\mathrm{T}(x)$. The second argument depends on the first inductive argument via T.

Is U inductively generated by a strictly positive functor Φ as was the case for inductively defined sets? If this is the case, Φ must depend on the recursively defined function T as well: we need something like $\Phi : (U : \mathrm{set}) \to (T : U \to \mathrm{set}) \to \mathrm{set}$ defined by $\Phi(U, T) = (x : U) \times (T(x) \to U)$!

In general, induction-recursion allows that a simultaneously defined function $\mathrm{T} : \mathrm{U} \to D$ for an arbitrary fixed type D may participate in the inductive generation of the set U.

- The modified non-inductive case thus becomes: if A is an stype, and Ψ_x is a strictly positive functor depending on $x : A$, then $\Phi(U, T) = (x : A) \times \Psi_x(U, T)$ is strictly positive.

- The modified inductive case becomes: if A is an stype, and Ψ_g is a strictly positive functor depending on $g : A \to D$, then $\Phi(U, T) = (f : A \to U) \times \Psi_{T \circ f}(U, T)$ is strictly positive.

We see that the Φ which generates U (as defined above), is isomorphic to the following strictly positive functor: $\Phi(U, T) = (f : 1 \to U) \times (T(f(\langle\rangle)) \to U)$ [2].

Furthermore, T is defined by $\mathrm{T}(\widehat{\Sigma}(a, b)) = \Sigma(\mathrm{T}(a), \mathrm{T} \circ b)$, i.e. $\mathrm{T}(\widehat{\Sigma}(a, b)) = d(\mathrm{T}(a), \mathrm{T} \circ b)$ with $d : (A : \mathrm{set}) \to (A \to \mathrm{set}) \to \mathrm{set}$ and $d(A, B) := \Sigma(A, B)$.

In general, we need an additional component Φ^{Arg} which specifies the domain of d. (Note that this domain only depends on D and not on U and T!). Finally, we need a third component $\Phi^{\mathrm{map}}(U, T) : \Phi^{\mathrm{arg}}(U, T) \to \Phi^{\mathrm{Arg}}$ and then we can draw a diagram

$$\begin{tikzcd}
\Phi^{\mathrm{arg}}(\mathrm{U}, \mathrm{T}) \arrow[r, "\mathrm{intro}"] \arrow[d, "\Phi^{\mathrm{map}}(\mathrm{U}, \mathrm{T})"'] & \mathrm{U} \arrow[d, "\mathrm{T}"] \\
\Phi^{\mathrm{Arg}} \arrow[r, "d"] & D
\end{tikzcd}$$

which summarizes the inductive definition of U and recursive definition of T. Think of D as a type of "semantic" objects and of $d : \Phi^{\mathrm{Arg}} \to D$ as a (possibly infinitary) "semantic" operation with Φ^{Arg} as the domain (or generalized arity) of d. U is a universe of codes for objects in D and $\mathrm{T} : \mathrm{U} \to D$ is the decoding function. The constructor intro is the syntactic reflection of $d : \Phi^{\mathrm{Arg}} \to D$.[3]

[2] As this example shows the term "strictly positive" may no longer be wholly appropriate, since the T-argument now can appear negatively. Allen [2] used the alternative term "half-positive" for this reason. U always appears strictly positively however.

[3] Recall that the term "universe à la Tarski" was chosen by Martin-Löf [12] because of the similarity between the definition of T (for the ordinary first universe) and Tarski's definition of truth.

Note the similarity between the above diagram for induction-recursion and the ordinary diagram for an initial algebra of an endofunctor which was displayed in Section 2! The key difference is that here Φ is no longer a functor in the ordinary sense, but consists of three components: Φ^{Arg}, Φ^{arg}, and Φ^{map}. These will be axiomatized below.

4.2 A Finite Axiomatization

We shall now give the formal rules for the inductive-recursive definition of a set U and a function $\mathrm{T} : \mathrm{U} \to D$. Such a definition is always parameterized with respect to the target type D of T, since a particular inductive-recursive definition generates a universe for a finite number of D-operations.

The main step is to introduce a new type SP_D, the objects of which are representatives of strictly positive "functors" Φ as above:

$$\frac{D : \mathrm{type}}{\mathrm{SP}_D : \mathrm{type}}$$

(There is also a rule, which lets us infer that $\mathrm{SP}_D = \mathrm{SP}_{D'}$ if $D = D'$, but we will omit all such equality preservation rules.)

SP_D has three associated operations corresponding to Φ^{Arg}, Φ^{arg}, and Φ^{map} in the informal exposition above:

$$\frac{D : \mathrm{type} \qquad \phi : \mathrm{SP}_D}{\mathrm{Arg}_{D,\phi} : \mathrm{type}}$$

$$\frac{\phi : \mathrm{SP}_D \qquad U : \mathrm{set} \qquad T : U \to D}{\mathrm{arg}_\phi(U,T) : \mathrm{stype}}$$

$$\frac{\phi : \mathrm{SP}_D \qquad U : \mathrm{set} \qquad T : U \to D}{\mathrm{map}_\phi(U,T) : (\mathrm{arg}_\phi(U,T)) \to \mathrm{Arg}_{D,\phi}}$$

To simplify notation we have suppressed the parameter (the "global" premise) D : type and the argument D for the second and third operation[4]. It should be emphasized that arg_ϕ and map_ϕ are only abbreviations of the proper formal expressions $\mathrm{arg}_{D,\phi}$ and $\mathrm{map}_{D,\phi}$. Similarly, we will suppress the D in some later operations as well.

With this new notation the diagram for the inductive-recursive definition of U and T becomes:

$$\begin{tikzcd}
\mathrm{arg}_\phi(\mathrm{U}_{\phi,d}, \mathrm{T}_{\phi,d}) \arrow[r, "\mathrm{intro}_{\phi,d}"] \arrow[d, "\mathrm{map}_\phi(\mathrm{U}_{\phi,d}, \mathrm{T}_{\phi,d})"'] & \mathrm{U}_{\phi,d} \arrow[d, "\mathrm{T}_{\phi,d}"] \\
\mathrm{Arg}_{D,\phi} \arrow[r, "d"] & D
\end{tikzcd}$$

We have the following introduction rules for SP_D (again with D suppressed):

$$\mathrm{nil} : \mathrm{SP}_D$$

$$\frac{A \text{ stype} \qquad \phi : A \to \mathrm{SP}_D}{\mathrm{nonind}(A,\phi) : \mathrm{SP}_D}$$

$$\frac{A \text{ stype} \qquad \phi : (A \to D) \to \mathrm{SP}_D}{\mathrm{ind}(A,\phi) : \mathrm{SP}_D}$$

[4]In Arg we have not suppressed it, since the equality rules for it will make use of D.

$$
\begin{aligned}
\mathrm{Arg}_{D,\mathrm{nil}} &= 1\\
\mathrm{Arg}_{D,\mathrm{nonind}(A,\phi)} &= (x : A) \times \mathrm{Arg}_{D,\phi(x)}\\
\mathrm{Arg}_{D,\mathrm{ind}(A,\phi)} &= (f : A \to D) \times \mathrm{Arg}_{D,\phi(f)}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{arg}_{\mathrm{nil}}(U,T) &= 1\\
\mathrm{arg}_{\mathrm{nonind}(A,\phi)}(U,T) &= (x : A) \times (\mathrm{arg}_{\phi(x)}(U,T))\\
\mathrm{arg}_{\mathrm{ind}(A,\phi)}(U,T) &= (f : A \to U) \times (\mathrm{arg}_{\phi(T\circ f)}(U,T))
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{map}_{\mathrm{nil}}(U,T,\langle\rangle) &= \langle\rangle\\
\mathrm{map}_{\mathrm{nonind}(A,\phi)}(U,T,\langle a,\gamma\rangle) &= \langle a, \mathrm{map}_{\phi(a)}(U,T,\gamma)\rangle\\
\mathrm{map}_{\mathrm{ind}(A,\phi)}(U,T,\langle f,\gamma\rangle) &= \langle T\circ f, \mathrm{map}_{\phi(T\circ f)}(U,T,\gamma)\rangle
\end{aligned}
$$

We are now ready to give the formal rules for U and T. These rules have the common premises D : type, $\phi : \mathrm{SP}_D$ and $d : \mathrm{Arg}_{D,\phi} \to D$ which will be omitted.

Formation rules:

$$\mathrm{U}_{\phi,d} : \mathrm{set}$$

$$\mathrm{T}_{\phi,d} : \mathrm{U}_{\phi,d} \to D$$

Introduction rule:

$$\frac{a : \mathrm{arg}_{\phi}(\mathrm{U}_{\phi,d}, \mathrm{T}_{\phi,d})}{\mathrm{intro}_{\phi,d}(a) : \mathrm{U}_{\phi,d}}$$

Equality rule:

$$\frac{a : \mathrm{arg}_{\phi}(\mathrm{U}_{\phi,d}, \mathrm{T}_{\phi,d})}{\mathrm{T}_{\phi,d}(\mathrm{intro}_{\phi,d}(a)) = d(\mathrm{map}_{\phi}(\mathrm{U}_{\phi,d}, \mathrm{T}_{\phi,d}, a))}$$

Moreover, structural recursion on U into a type D', that is, the analogue of universe elimination, is expressed by the following diagram (we omit the indices ϕ, d of U, T, intro, R, write $D'[t]$ for the substitution of some fixed variable in D' by t and, when used as an argument, D' instead of $(x)D'[x]$; assume in the following $x : \mathrm{U}_{\phi,d} \Rightarrow D'[x]$: type as a global premise)

$$
\begin{tikzcd}
\mathrm{arg}_{\phi}(\mathrm{U},\mathrm{T}) \arrow[r, "\mathrm{intro}"] \arrow[d, "{\langle \mathrm{id}, \mathrm{mapIH}_{\phi,\mathrm{U},\mathrm{T},D'}(\mathrm{R}_{D'}(e))\rangle}"] & \mathrm{U} \arrow[d, "{\langle \mathrm{id}, \mathrm{R}_{D'}(e)\rangle}"] \\
(\gamma : \mathrm{arg}_{\phi}(\mathrm{U},\mathrm{T})) \times \mathrm{IH}_{\phi,\mathrm{U},\mathrm{T},D'}(\gamma) \arrow[r, "{\langle \mathrm{intro}\circ\pi_0, e\rangle}"] & (x : \mathrm{U}) \times D'[x]
\end{tikzcd}
$$

where we have used the operation IH which generates the induction hypothesis and the operation mapIH which generates the recursive call:

$$\frac{U : \mathrm{set} \qquad T : (x : U) \to D \qquad x : U \Rightarrow D'[x] : \mathrm{type} \qquad \gamma : \mathrm{arg}_{\phi}(U,T)}{\mathrm{IH}_{\phi,U,T,D'}(\gamma) : \mathrm{type}}$$

$$\frac{U : \mathrm{set} \qquad T : (x : U) \to D \qquad x : U \Rightarrow D'[x] : \mathrm{type} \qquad R : (x : U) \to D'[x]}{\mathrm{mapIH}_{\phi,U,T,D'}(R) : (x : \mathrm{arg}_{\phi}(U,T)) \to \mathrm{IH}_{\phi,U,T,D'}(x)}$$

$$
\begin{aligned}
\mathrm{IH}_{\mathrm{nil},U,T,D'}(\langle\rangle) &= 1\\
\mathrm{IH}_{\mathrm{nonind}(A,\phi),U,T,D'}(\langle a,\gamma\rangle) &= \mathrm{IH}_{\phi(a),U,T,D'}(\gamma)\\
\mathrm{IH}_{\mathrm{ind}(A,\phi),U,T,D'}(\langle f,\gamma\rangle) &= ((y : A) \to D'[f(y)]) \times (\mathrm{IH}_{\phi(T\circ f),U,T,D'}(\gamma))
\end{aligned}
$$

$$\mathrm{mapIH}_{\mathrm{nil},U,T,D'}(R, \langle\rangle) = \langle\rangle$$

$$\begin{array}{rcl}\mathrm{mapIH}_{\mathrm{nonind}(A,\phi),U,T,D'}(R,\langle a,\gamma\rangle) & = & \mathrm{mapIH}_{\phi(a),U,T,D'}(R,\gamma)\\ \mathrm{mapIH}_{\mathrm{ind}(A,\phi),U,T,D'}(R,\langle f,\gamma\rangle) & = & \langle R\circ f,\mathrm{mapIH}_{\phi(T\circ f),U,T,D'}(R,\gamma)\rangle\end{array}$$

Elimination rule (universe elimination):

$$\frac{e:(\gamma:\mathrm{arg}_\phi(\mathrm{U}_{\phi,d},\mathrm{T}_{\phi,d}))\to(\mathrm{IH}_{\phi,\mathrm{U}_{\phi,d},\mathrm{T}_{\phi,d},D'}(\gamma))\to(D'[\mathrm{intro}_{\phi,d}(\gamma)])}{\mathrm{R}_{\phi,d,D'}(e):(a:\mathrm{U}_{\phi,d})\to D'[a]}$$

Equality rule (universe elimination, premises omitted):

$$\mathrm{R}_{\phi,d,D'}(e,\mathrm{intro}_{\phi,d}(\gamma))=e(\gamma,\mathrm{mapIH}_{\phi,\mathrm{U}_{\phi,d},\mathrm{T}_{\phi,d},D'}(\mathrm{R}_{\phi,d,D'}(e),\gamma))$$

5 Examples

We shall show how to find $\phi:\mathrm{SP}_D$ for some well-known set constructors. (Compare the informal discussion in Section 4.1.) We will write intro instead of $\mathrm{intro}_{\phi,d}$. Let in the first examples $D:=1$ and $d:=(x:C)\langle\rangle$ for some suitable type C, since this is how we obtain inductive definitions as degenerate cases of inductive-recursive definitions.

Σ-sets. Let

$$\phi_{A,B}:=\mathrm{nonind}(A,(x)\mathrm{nonind}(B(x),(y)\mathrm{nil}))$$

in the context $A:\mathrm{set},B:A\to\mathrm{set}$. It follows that $\Sigma(A,B):=\mathrm{U}_{\phi_{A,B},d}:\mathrm{set}$. This set has the constructor $\mathrm{intro}:((x:A)\times(B(x)\times 1))\to\Sigma(A,B)$. If we define $\mathrm{p}:=(A,B,x,y)\mathrm{intro}(\langle x,\langle y,\langle\rangle\rangle\rangle)$, then $\mathrm{p}:(A:\mathrm{set},B:A\to\mathrm{set},x:A,y:B(x))\to\Sigma(A,B)$ and one can easily derive the ordinary elimination rules as if p were the constructor of Σ. Note that this illustrates that we get dependencies on parameters (in the sense of Dybjer [9, 7]) like A, B for free.

Natural numbers. Let

$$\begin{array}{rcl}\phi & := & \mathrm{nonind}(\mathbb{B},(x)\text{ if }x\text{ then nil else ind}(1,(y)\mathrm{nil}))\ ,\\ \mathrm{N} & := & \mathrm{U}_{\phi,d}\ ,\\ 0 & := & \mathrm{intro}(\langle\mathrm{tt},\langle\rangle\rangle):\mathrm{N}\ ,\\ \mathrm{S} & := & (n)\mathrm{intro}(\langle\mathrm{ff},\langle(y)n,\langle\rangle\rangle\rangle):\mathrm{N}\to\mathrm{N}\ .\end{array}$$

Although this definition is like the definition of N by the equation $\mathrm{N}=1+(1\to\mathrm{N})\times 1$, because of the η-rule on 1 this is equivalent to the ordinary definition of N. The usual elimination rules for N can be derived.

The empty set. Let

$$\phi:=\mathrm{ind}(1,(x)\mathrm{nil})\ ,$$

and define $\mathrm{N}_0:=\mathrm{U}_{\phi,d}$. Then we can show the elimination rule for the empty set N_0. Note that this corresponds to the definition of N_0 by having one constructor $\mathrm{intro}:\mathrm{N}_0\to\mathrm{N}_0$. We can define now $\mathrm{N}_0':=\mathrm{U}_{\mathrm{nonind}(\mathrm{N}_0,(x)\mathrm{nil}),d}$, which can be regarded as the empty set with no constructors. However, one might prefer to add the set N_0 like the set $\mathbb{B}$ as a basic set.

Well-orderings. Let

$$\phi_{A,B}:=\mathrm{nonind}(A,(x)\mathrm{ind}(B(x),(y)\mathrm{nil}))\ ,$$

in the context $A:\mathrm{set},B:(x:A)\to\mathrm{set}$, and define $\mathrm{W}(A,B):=\mathrm{U}_{\phi_{A,B},d}:\mathrm{set}$ with the constructor $\mathrm{intro}:((x:A)\times((B(x)\to\mathrm{W}(A,B))\times 1))\to\mathrm{W}(A,B)$. As before we can define the ordinary constructor sup with its elimination rules.

A universe closed under N **and** Σ**.** Let $D :=$ set,

$$\phi := \mathrm{nonind}(\mathbb{B}, (x) \text{ if } x \text{ then nil else } \mathrm{ind}(1, (f)\mathrm{ind}(f(\langle\rangle), (y)\mathrm{nil}))) \ .$$

Hence $\mathrm{Arg}_{D,\phi} = (x : \mathbb{B}) \times \mathrm{E}(x)$, with

$$\begin{aligned} \mathrm{E}(\mathrm{tt}) &= 1 \ , \\ \mathrm{E}(\mathrm{ff}) &= (x : 1 \to \mathrm{set}) \times (f : x(\langle\rangle) \to \mathrm{set}) \times 1 \ . \end{aligned}$$

Moreover, let $\mathrm{d} : \mathrm{Arg}_{D,\phi} \to \mathrm{set}$ be defined such that

$$\begin{aligned} \mathrm{d}(\langle \mathrm{tt}, \langle\rangle\rangle) &= \mathrm{N} \ , \\ \mathrm{d}(\langle \mathrm{ff}, \langle A, \langle B, \langle\rangle\rangle\rangle\rangle) &= \Sigma(A(\langle\rangle), (y)B(y)) \ , \end{aligned}$$

using the elimination rules for $\mathbb{B}$ and product. Define $\mathrm{U}' := \mathrm{U}_{\phi,d}$, $\mathrm{T}' := \mathrm{T}_{\phi,d}$, and

$$\begin{aligned} \widehat{\mathrm{N}} &:= \mathrm{intro}(\langle \mathrm{tt}, \langle\rangle\rangle) : \mathrm{U}' \ , \\ \widehat{\Sigma} &:= (a, b)\mathrm{intro}(\langle \mathrm{ff}, \langle (x)a, \langle b, \langle\rangle\rangle\rangle\rangle) : (a : \mathrm{U}', b : \mathrm{T}'(a) \to \mathrm{U}') \to \mathrm{U}' \ . \end{aligned}$$

$\widehat{\mathrm{N}}$ and $\widehat{\Sigma}$ are essentially the two constructors of the universe U', T' and we have $\mathrm{T}'(\widehat{\mathrm{N}}) = \mathrm{N}$, $\mathrm{T}'(\widehat{\Sigma}(a, b)) = \Sigma(\mathrm{T}'(a), \mathrm{T}' \circ b)$.

Lists with distinct elements. Assume A : set and $\# : (A \times A) \to A$, where $\#$ is an (infix) apartness relation on A. In [7] the set Dlist of lists with elements which are distinct with respect to the relation $\#$ is defined inductively together with the recursively defined relation (family of sets) Fresh : Dlist $\to A \to$ set, where $\mathrm{Fresh}(l, a)$ expresses that a is distinct from all elements in l. (If we wish to make the dependence on the parameters A and $\#$ explicit, we may write $\mathrm{Dlist}(A, \#)$ and $\mathrm{Fresh}(A, \#)$.) Dlist has the constructors[5]

$$\begin{aligned} \mathrm{empty} &: \mathrm{Dlist} \ , \\ \mathrm{cons} &: (a : A, u : \mathrm{Dlist}, \mathrm{Fresh}(u, a)) \to \mathrm{Dlist} \ , \end{aligned}$$

and $\mathrm{Fresh}(l, a)$ is defined such that

$$\begin{aligned} \mathrm{Fresh}(\mathrm{empty}) &= (b)1 \ , \\ \mathrm{Fresh}(\mathrm{cons}(a, u, p)) &= (b)((b\#a) \wedge \mathrm{Fresh}(u, a)) \ . \end{aligned}$$

Then $\mathrm{Dlist} = \mathrm{U}_{\phi_{A,\#}, \mathrm{d}_{A,\#}}$, $\mathrm{Fresh} = \mathrm{T}_{\phi_{A,\#}, \mathrm{d}_{A,\#}}$, where $D := A \to$ set,

$$\begin{aligned} \phi_{A,\#} := \mathrm{nonind}(\mathbb{B}, (x) &\text{ if } x \text{ then nil} \\ &\text{else } \mathrm{nonind}(A, (a)\mathrm{ind}(1, (u)\mathrm{nonind}(u(\langle\rangle, a), \mathrm{nil})))) \ , \\ \mathrm{d}_{A,\#}(\langle \mathrm{tt}, \langle\rangle\rangle) &= (b)1 \ , \\ \mathrm{d}_{A,\#}(\langle \mathrm{ff}, \langle a, \langle u, \langle p, \langle\rangle\rangle\rangle\rangle\rangle) &= (b)((b\#a) \wedge u(\langle\rangle, a)) \ . \end{aligned}$$

The above examples show that we can derive all inductive-recursive sets in a form, which is close to the way we would ordinarily like to write them down. We must for example write the arguments in list notation and, if we have a non-indexed inductive argument, write it as an argument depending on the type 1. In an implementation of the calculus one could of course easily avoid this administrative overhead.

6 Set-Theoretic Model

6.1 Interpretation of Expressions

The idea behind the model is simple: interpret all constructions in set theory in the obvious way! In particular, each type is interpreted as a set, equal types are

[5] We have here renamed the constructor nil in [7] to empty.

interpreted as equal sets, $a : A$ is interpreted as $a \in A$, and $a = b : A$ is interpreted as a and b are equal elements of A. Moreover, $A \to B$ is interpreted as the set of all functions from A to B in the set-theoretic sense, and $(x : A) \to B$ as the set-theoretic cartesian product $\Pi_{x \in A} B$, etc.

The inductively defined type SP_D of codes for strictly positive operators is interpreted as an inductively defined set in the set-theoretic sense, that is, as a set generated by iterating a monotone operator up to a fixed point. Similarly, the inductive-recursively defined set U and function $\mathrm{T} : \mathrm{U} \to D$, are also interpreted by iterating a monotone operator up to a fixed point.

In order to ensure that a fixed point indeed can be reached we postulate the existence of one Mahlo cardinal in addition to the ordinary axioms of ZF set theory.[6] We also need the the axiom of choice to deal with cardinals, and for simplicity we assume the generalized continuum hypothesis.[7]

Note, that a cardinal κ is inaccessible, iff it is regular and $\aleph_\kappa = \kappa$, where $\aleph_\alpha$ enumerates the infinite cardinals. An inaccessible cardinal κ is a Mahlo cardinal, iff every normal function $f : \kappa \to \kappa$ has a regular fixed point. (A normal function f is a (strictly) monotone function, which is continuous at limit ordinals λ, i.e. $f(\lambda) = \sup_{\alpha<\lambda} f(\alpha)$.) The standard model of our extension of ZF is $\mathrm{V}_{\mathrm{M}^+}$, where M^+ is the first inaccessible above M, however all types will be interpreted as elements of V_Λ, where Λ is the first (non-regular) fixed point of $\lambda\alpha.\aleph_\alpha$ above M.

We will develop the semantics following the approach in [8]. Let $\lambda_0 := \aleph_{\mathrm{M}+1}$, $\lambda_{n+1} := \aleph_{\lambda_n}$, and $\Lambda := \sup_{n\in\omega} \lambda_n$.

If $a, a_1, \ldots, a_n, c$ are sets, and b is a function with domain a, let

$$\begin{aligned} \Pi_{x\in a} b(x) &:= \{f \mid f \text{ function } \wedge \mathrm{dom}(f) = a \wedge \forall x \in a. f(x) \in b(x)\} \ , \\ \lambda x \in a.b(x) &:= \{\langle x, b(x)\rangle \mid x \in a\} \ , \\ \Sigma_{x\in a} b(x) &:= \{\langle c, d\rangle \mid c \in a \wedge d \in b(c)\} \ , \\ a_0 + \cdots + a_n &:= \Sigma_{i\in\{0,\ldots,n\}} a_i \text{ (if } n \geq 1) \ , \\ (a \to c) &:= \Pi_{x\in a} c \ . \end{aligned}$$

Moreover, $(a)_i := a_i$, if $a = \langle a_0, \ldots, a_i\rangle$ and undefined otherwise.

Whenever we introduce sets A^α indexed by ordinals α, let in the following

$$A^{<\alpha} := \bigcup_{\beta<\alpha} A^\beta \ .$$

We shall use the set V_Λ as the set-theoretic universe for our interpretation. All types and objects of types will thus be interpreted as elements of V_Λ [8]. Terms which depend on free variables will be interpreted relative to an assignment ρ, that is, a function, which maps a finite set of variables to elements of V_Λ. In the following ρ (possibly with indices or accents) will always be an assignment. If $a \in \mathrm{V}_\Lambda$, then ρ_x^a is the assignment with $\mathrm{dom}(\rho_x^a) := \mathrm{dom}(\rho) \cup \{x\}$, such that

$$\rho_x^a(y) := \begin{cases} a & \text{if } x = y, \\ \rho(y) & \text{otherwise.} \end{cases}$$

Let terms be the set of expressions which possibly occur as elements of a type or as types: So variables are terms and if $a, b, a_1, \ldots, a_n$ are terms, x is a variable, and C is an n-ary constructor (including set, stype and constructors like $\langle \cdot, \cdot \rangle$, π_0

[6] In Sect. 7 ("Constructive versions of the model") we will discuss how to replace these strong set theoretic requirements by far weaker ones.

[7] Without the generalized continuum hypothesis one has to replace Mahlo and inaccessible by strongly Mahlo and strongly inaccessible, respectively, and $\aleph_\alpha$ by $\beth_\alpha$.

[8] We here use a notion of model which only requires all *derivable* types to be interpreted as elements of V_Λ. Note however that V_Λ is not closed under the formation of dependent function types. If we wish to satisfy this requirement we can either reinterpret type as the class of all sets or as V_I for some inaccessible cardinal $\mathrm{I} > \mathrm{M}$.

but excluding type) of the system, then $(x : a) \to b$, $(x : a)b$, $(x : a) \times b$ and $C(a_1, \dots, a_n)$ are terms.

For terms t and assignments ρ we will determine, whether its interpretation t^*_ρ is defined, and if it is defined, the value of t^*_ρ. This will be done in such a way that for every term t and every $n \in \omega$ there exists an $m \in \omega$ such that, if $\mathrm{rng}(\rho) \subseteq \mathrm{V}_{\lambda_n}$, $t^*_\rho \in \mathrm{V}_{\lambda_m}$. For closed terms, t^*_ρ will not depend on ρ and we therefore omit the subscript ρ. We will use $\simeq$ for partial equality in the usual sense, and also let $t^*_\rho :\simeq s$ mean that the interpretation of t under assignment ρ is defined to be s, provided s is defined, and is undefined otherwise. We extend this definition further by defining

$$\mathrm{type}^* := V_\Lambda \ .$$

The interpretation of terms is given by

$$\begin{array}{ll}
x^*_\rho :\simeq \rho(x) \ , & \mathrm{set}^* :\simeq \mathrm{stype}^* :\simeq \mathrm{V_M} \ , \\
((x : A) \to B)^*_\rho :\simeq \Pi_{y \in A^*_\rho} B^*_{\rho^y_x} \ , & ((x : A)a)^*_\rho :\simeq \lambda y \in A^*_\rho . a^*_{\rho^y_x} \ , \\
(a(b))^*_\rho :\simeq a^*_\rho(b^*_\rho) \ , & ((x : A) \times B)^*_\rho :\simeq \Sigma_{y \in A^*_\rho} B^*_{\rho^y_x} \ , \\
\langle a, b \rangle^*_\rho :\simeq \langle a^*_\rho, b^*_\rho \rangle \ , & (\pi_0(a))^*_\rho :\simeq (a)_0 \ , \\
(\pi_1(a))^*_\rho :\simeq (a)_1 \ , & 1^* :\simeq 1 \ , \\
\langle\rangle^* :\simeq 0 \ , & \mathbb{B}^* :\simeq \{0, 1\} \ , \\
\mathrm{tt}^* :\simeq 0 \ , & \mathrm{ff}^* :\simeq 1 \ ,
\end{array}$$

$$(\text{if } a \text{ then } b \text{ else } c)^*_\rho :\simeq \begin{cases} b^*_\rho & \text{if } a^*_\rho = 0 \ , \\ c^*_\rho & \text{if } a^*_\rho = 1 \ , \\ \text{undefined} & \text{otherwise} \ . \end{cases}$$

To interpret terms with constructors SP, nonind, ind, Arg, ..., we first define SP^*, nonind^*, ind^*, Arg^*, arg^*, map^*, IH^*, mapIH^*, U^*, T^* and interpret

$$\begin{array}{rcl}
(\mathrm{SP}_D)^*_\rho & :\simeq & \mathrm{SP}^*(D^*_\rho) \ , \\
(\mathrm{nonind}(a, b))^*_\rho & :\simeq & \mathrm{nonind}^*(a^*_\rho, b^*_\rho) \ , \\
(\mathrm{Arg}_{D,\phi})^*_\rho & :\simeq & \mathrm{Arg}^*(D^*_\rho, \phi^*_\rho) \ , \\
(\mathrm{map}_{D,\phi}(U, T))^*_\rho & :\simeq & \lambda x \in \mathrm{arg}^*(D^*_\rho, \phi^*_\rho, U^*_\rho, T^*_\rho).\mathrm{map}^*(D^*_\rho, \phi^*_\rho, U^*_\rho, T^*_\rho, x) \ , \\
& & \text{etc.}
\end{array}$$

$\mathrm{SP}^*(D)$ is defined for $D \in \mathrm{type}^*$ as the least set such that

$$\mathrm{SP}^*(D) = 1 + \Sigma_{a \in \mathrm{set}^*}(a \to \mathrm{SP}^*(D)) + \Sigma_{a \in \mathrm{set}^*}((a \to D) \to \mathrm{SP}^*(D)) \ ,$$

which we get by iterating the appropriate operator κ times, if for all $a \in \mathrm{set}^*$ the cardinality of a and of $a \to D$ is less than κ. If $D \in \mathrm{V}_{\lambda_n}$, therefore $\mathrm{SP}^*(D) \in \mathrm{V}_{\lambda_{n+1}}$.

$$\mathrm{nil}^* :\simeq \langle 0, 0 \rangle, \ \mathrm{nonind}^*(a, b) :\simeq \langle 1, \langle a, b \rangle \rangle, \ \mathrm{ind}^*(a, b) :\simeq \langle 2, \langle a, b \rangle \rangle \ .$$

$\mathrm{Arg}^*(D, \phi)$ is defined, if $\phi \in \mathrm{SP}^*(D)$, and then defined in accordance with the equations for Arg, that is,

$$\begin{array}{rcl}
\mathrm{Arg}^*(D, \mathrm{nil}^*) & :\simeq & 1 \ , \\
\mathrm{Arg}^*(D, \mathrm{nonind}^*(A, \phi)) & :\simeq & \Sigma_{x \in A} \mathrm{Arg}^*(D, \phi(x)) \ , \\
\mathrm{Arg}^*(D, \mathrm{ind}^*(A, \phi)) & :\simeq & \Sigma_{f \in (A \to D)} \mathrm{Arg}^*(D, \phi(f)) \ .
\end{array}$$

Similarly, we define $\mathrm{arg}^*(D, \phi, U, T)$, $\mathrm{map}^*(D, \phi, U, T, a)$, and for $D' \in (U \to \mathrm{type}^*)$, $\mathrm{IH}^*(D, \phi, U, T, D', \gamma)$, $\mathrm{mapIH}^*(D, \phi, U, T, D', R, a)$.

$$\begin{array}{rcl}
\mathrm{U}^*(D, \phi, d) & :\simeq & \mathrm{U^M}(D, \phi, d) \ , \\
\mathrm{T}^*(D, \phi, d) & :\simeq & \lambda x \in \mathrm{U^M}(D, \phi, d).\mathrm{T^M}(D, \phi, d, x) \ ,
\end{array}$$

where $\mathrm{U}^\alpha(D, \phi, d)$ and $\mathrm{T}^\alpha(D, \phi, d)$ or shorter U^α and T^α are simultaneously defined by recursion on α as

$$\mathrm{U}^\alpha \ :\simeq \ \mathrm{arg}^*(D, \phi, \mathrm{U}^{<\alpha}, \mathrm{T}^{<\alpha}) \ ,$$

$$
\begin{aligned}
\mathrm{T}^{\alpha}(a) &:\simeq d(\mathrm{map}^*(D,\phi,\mathrm{U}^{<\alpha},\mathrm{T}^{<\alpha},a)) \ ,\\
\\
\mathrm{intro}(a)^*_{\rho} &:\simeq a^*_{\rho} \ ,\\
\mathrm{R}^*(D,\phi,d,D',e,a) &:\simeq \mathrm{R}^{\mathrm{M}}(D,\phi,d,D',e,a), \text{ where}\\
\mathrm{R}^{\alpha}(D,\phi,d,D',e,a) &:\simeq e(a,\mathrm{mapIH}^*(D,\phi,\mathrm{U}^*(D,\phi,d),\mathrm{T}^*(D,\phi,d) \ ,\\
&\qquad\qquad D',\mathrm{R}^{<\alpha}(D,\phi,d,D',e),a)) \ .
\end{aligned}
$$

Contexts will be interpreted as sets of assignments:

$$
\emptyset^* :\simeq \emptyset \ , \quad (\Gamma, x : A)^* :\simeq \{\rho^a_x \mid \rho \in \Gamma^*_{\rho} \wedge a \in A^*_{\rho}\} \ .
$$

6.2 Soundness of the Rules

Theorem 1 (Soundness theorem)

(a) If $\vdash \Gamma$ context*, then* Γ^* *is defined.*

(b) If $\vdash \Gamma \Rightarrow A : E$*, where* $E \equiv \mathrm{type}$ *or* E *is a term, then* Γ^* *is defined,* $\forall\rho \in \Gamma^*.A^*_{\rho} \in E^*_{\rho}$*, and if* $E \not\equiv \mathrm{type}$*,* $\forall\rho \in \Gamma^*.E^*_{\rho} \in \mathrm{type}^*$*.*

(c) If $\vdash \Gamma \Rightarrow A = B : E$*, where* $E \equiv \mathrm{type}$ *or* E *is a term, then* Γ^* *is defined,* $\forall\rho \in \Gamma^*(A^*_{\rho} \in E^*_{\rho} \wedge B^*_{\rho} = A^*_{\rho})$*, and if* $E \not\equiv \mathrm{type}$*,* $\forall\rho \in \Gamma^*.E^*_{\rho} \in \mathrm{type}^*$*.*

(d) $\not\vdash a : \mathrm{N}_0$*, where* N_0 *is the empty set, for any of the possibilities mentioned in Section 5.*

The proof of the Soundness theorem is more or less routine, except for the verification that $\mathrm{U} : \mathrm{set}$. In order to prove this we will need some lemmata.

First we need to verify that U^{α} is increasing with α and that for $\alpha < \beta$ T^{α} and T^{β} coincide on U^{α}. In order to prove this we need to verify that $\mathrm{arg}^*(D,\phi,U,T)$ and $\mathrm{map}^*(D,\phi,U,T)$ are monotone in U, T, as expressed by the following lemma:

Lemma 1 *Assume* $D \in \mathrm{type}^*$*,* $\phi \in \mathrm{SP}^*(D)$*,* $U \subseteq U' \in \mathrm{set}^*$*,* $T' : U' \to D$*,* $T = T' \restriction U$*. Then*

(a) $\mathrm{arg}^*(D,\phi,U,T) \subseteq \mathrm{arg}^*(D,\phi,U',T')$ *and*

(b) $\mathrm{map}^*(D,\phi,U',T') \restriction \mathrm{arg}^*(D,\phi,U,T) = \mathrm{map}^*(D,\phi,U,T)$*.*

We want to show that there is a $\kappa < \mathrm{M}$ such that $\mathrm{U}^{<\kappa} = \mathrm{U}^{\kappa}$. This is the case if κ is a limit ordinal such that $\mathrm{arg}^*(D,\phi,U,T)$ is κ-continuous in U and T, that is,

$$
\mathrm{arg}^*(D,\phi,\mathrm{U}^{<\kappa},\mathrm{T}^{<\kappa}) = \bigcup_{\alpha<\kappa} \mathrm{arg}^*(D,\phi,\mathrm{U}^{\alpha},\mathrm{T}^{\alpha}) \ . \tag{1}
$$

To obtain this we need that all index sets, which start an inductive argument, have cardinality less than κ. The set $\mathrm{Aux}(D,\phi,U,T) \in \mathrm{set}^*$, where $D \in \mathrm{type}^*$, $\phi \in \mathrm{SP}^*(D)$, $U \in \mathrm{set}^*$, $T \in U \to D$, collects all possible such index sets. It is defined by induction on ϕ:

$$
\begin{aligned}
\mathrm{Aux}(D,\mathrm{nil}^*,U,T) &:= 1 \ ,\\
\mathrm{Aux}(D,\mathrm{nonind}^*(A,\phi),U,T) &:= \Pi_{x\in A}\mathrm{Aux}(D,\phi(x),U,T) \ ,\\
\mathrm{Aux}(D,\mathrm{ind}^*(A,\phi),U,T) &:= A + \Pi_{f\in(A\to U)}\mathrm{Aux}(D,\phi(T\circ f),U,T) \ .
\end{aligned}
$$

Lemma 2 *Assume* $D \in \mathrm{type}^*$*,* $\phi \in \mathrm{SP}^*(D)$*. Let* κ *be inaccessible and let for* $\alpha < \kappa$ $U^{\alpha} \in \mathrm{set}^*$*,* $T^{\alpha} : U^{\alpha} \to D$ *such that for* $\alpha < \beta$*,* $U^{\alpha} \subseteq U^{\beta}$*,* $T^{\alpha} = T^{\beta} \restriction U^{\alpha}$*. Assume also for some* $\alpha_0 < \kappa$ *and for all* $\alpha_0 \leq \alpha < \kappa$

$$
\mathrm{Aux}(D,\phi,U^{\alpha},T^{\alpha}) \in V_{\kappa} \ . \tag{2}
$$

Then $\mathrm{arg}^*(D,\phi,U,T)$ *is* κ*-continuous in* U *and* T*, that is, (1) holds.*

Proof: "$\supseteq$" follows by Lemma 1b.

"$\subseteq$" follows by induction on ϕ. We treat only the main case $\phi = \mathrm{ind}^*(A,\gamma)$. Assume $a \in \arg^*(D,\phi,U^{<\kappa},T^{<\kappa})$, and show $a \in \arg^*(D,\phi,U^\alpha,T^\alpha)$ for some $\alpha < \kappa$. We know $a = \langle f, c\rangle$ for some $f : A \to U^{<\kappa}$, $c \in \arg^*(D,\gamma(T^{<\kappa} \circ f),U^{<\kappa},T^{<\kappa})$. By (2) it follows $A \in V_\kappa$, and by the inaccessibility of κ there exists a $\beta < \kappa$ such that $f : A \to U^{<\beta}$, especially $f : A \to U^\beta$. W.l.o.g. $\alpha_0 \leq \beta$. For $\beta \leq \alpha < \kappa$ it follows $\mathrm{Aux}(D,\gamma(T^\alpha \circ f),U^\alpha,T^\alpha) \in V_\kappa$ and therefore by induction hypothesis there exists a β' such that $c \in \arg^*(D,\gamma(T^{\beta'} \circ f),U^{\beta'},T^{\beta'})$. With $\alpha := \max\{\beta,\beta'\}$ follows the assertion. $\Box$

Lemma 3 *Assume* $\phi \in \mathrm{SP}^*(D)$, $s \in \mathrm{Arg}^*(D,\phi) \to D$. *Abbreviate* $\mathrm{U}^\alpha := \mathrm{U}^\alpha(D,\phi,d)$, $\mathrm{T}^\alpha := \mathrm{T}^\alpha(D,\phi,d)$ *and note that* $\mathrm{U}^*(D,\phi,d) = \mathrm{U}^\mathrm{M}$, $\mathrm{T}^*(D,\phi,d) = \mathrm{T}^\mathrm{M}$.

(a) $\mathrm{T}^\alpha : \mathrm{U}^\alpha \to D$, *and if* $\alpha < \mathrm{M}$, $\mathrm{U}^\alpha \in \mathrm{V}_\mathrm{M}$.

(b) If $\alpha < \beta$ *then* $\mathrm{U}^\alpha \subseteq \mathrm{U}^\beta$ *and* $\mathrm{T}^\beta \restriction \mathrm{U}^\alpha = \mathrm{T}^\alpha$.

(c) There exists $\kappa < \mathrm{M}$ *such that* $\mathrm{U}^\alpha = \mathrm{U}^\kappa$ *(and therefore* $\mathrm{T}^\alpha = \mathrm{T}^\kappa$*) for all* $\alpha > \kappa$.

(d) $\mathrm{U}^\mathrm{M} \in \mathrm{V}_\mathrm{M}$, $\arg^*(D,\phi,\mathrm{U}^\mathrm{M},\mathrm{T}^\mathrm{M}) \subseteq \mathrm{U}^\mathrm{M}$.

Proof:

(a) Easy induction on α.

(b) Induction on α, β, by using Lemma 1(b).

(c) Define $f : \mathrm{Ord} \to \mathrm{Ord}$ by transfinite recursion:

$$\begin{aligned} f(\beta) = \min\{\alpha \quad | \quad & \forall\beta' < \beta(f(\beta') < \alpha) \wedge \\ & \forall\beta' < \mathrm{M}(\mathrm{U}^{\beta'} \subseteq \mathrm{V}_\beta \to \mathrm{U}^{\beta'+1} \cup \mathrm{Aux}(D,\phi,\mathrm{U}^{\beta'},\mathrm{T}^{\beta'}) \subseteq \mathrm{V}_\alpha\} \end{aligned}$$

$f : \mathrm{M} \to \mathrm{M}$ follows immediately by M being inaccessible, since

$$\{\mathrm{U}^{\beta'} \mid \beta' < \mathrm{M} \wedge \mathrm{U}^{\beta'} \subseteq \mathrm{V}_\beta\} \in \mathrm{V}_{\beta+1} \subseteq \mathrm{V}_M \ .$$

Let for $\alpha < \mathrm{M}$ $\theta(\alpha) := f^\alpha(0)$. By the regularity of M we have $\theta : \mathrm{M} \to \mathrm{M}$. Since f is increasing, θ is normal. Hence, since M is Mahlo, θ has an inaccessible fixed point $\kappa < \mathrm{M}$.

Therefore $f : \kappa \to \kappa$: Assume $\alpha < \kappa$. κ is a limit ordinal, therefore $\alpha < \theta(\beta)$ for some $\beta < \kappa$, $f(\alpha) < f(\theta(\beta)) = \theta(\beta+1) < \theta(\kappa) = \kappa$. By induction on α, using the regularity of κ, for $\alpha < \kappa$ $\mathrm{U}^\alpha \in \mathrm{V}_\kappa$, $\mathrm{Aux}(D,\phi,\mathrm{U}^\alpha,\mathrm{T}^\alpha) \in V_\kappa$, and therefore by Lemma 2

$$\begin{aligned} \mathrm{U}^\kappa &= \arg^*(D,\phi,\mathrm{U}^{<\kappa},\mathrm{T}^{<\kappa}) \\ &= \bigcup_{\alpha<\kappa} \arg^*(D,\phi,\mathrm{U}^\alpha,\mathrm{T}^\alpha) \\ &= \bigcup_{\alpha<\kappa} \mathrm{U}^{\alpha+1} = \mathrm{U}^{<\kappa}. \end{aligned}$$

By induction on α for all $\alpha \geq \kappa$ $\mathrm{U}^\alpha = \mathrm{U}^{<\kappa} = \mathrm{U}^\kappa$.

(d) $\mathrm{U}^\mathrm{M} = \mathrm{U}^\kappa \in \mathrm{V}_\mathrm{M}$, $\arg^*(D,\phi,\mathrm{U}^\mathrm{M},\mathrm{T}^\mathrm{M}) = \arg^*(D,\phi,\mathrm{U}^\kappa,\mathrm{T}^\kappa) \subseteq \mathrm{U}^{\kappa+1} \subseteq \mathrm{U}^\mathrm{M}$. $\Box$

7 Related and Future Work

Universes in type theory. The first example of an inductive-recursive definition in type theory was Martin-Löf's universe à la Tarski [12]. [9] Then Palmgren [17] defined external and internal universe hierarchies and also a super universe. Rathjen, Griffor, and Palmgren [19] defined quantifier universes and Palmgren [16] defined higher order universe hierarchies. All these constructions use induction-recursion, whereas Setzer [20] defined a Mahlo universe, which goes beyond it.

[9] There are earlier examples of informal inductive-recursive definitions, for example, Martin-Löf's simultaneous definition of the notions of computable type and term [13] from 1972. However, the explicitly inductive-recursive nature of type-theoretic universes was only brought out when they were formulated à la Tarski rather than à la Russell.

Inductive definitions in type theory. Previous work on formalization of inductive definitions in Martin-Löf's type theory has mainly used external schemata in the style of Martin-Löf's intuitionistic theory of iterated inductive definitions in predicate logic [11]. See for example Backhouse [3], Dybjer [9], and Paulin [18]. A schema for inductive-recursive definitions was introduced by Dybjer [7].

Categorical semantics of inductive types and of universes. The categorical semantics of inductively defined dependent types has been discussed for example by Coquand and Paulin [5] and Mendler [14]. The latter article also discusses categorical semantics of universes in type theory. In a future article we plan to extend Mendler's work, by giving categorical semantics of inductive-recursive definitions in terms of initial algebras on endofunctors in slice categories. We will also show how such semantics suggest an alternative finite axiomatization of inductive-recursive definitions.

Set-theoretic semantics of type theory. It is well-known that Martin-Löf's type theory has a "naive" full function-space model, see for example the introduction in Troelstra [26]. Dybjer [8] gives a full function space model of Martin-Löf's type theory with an external schema for inductive definitions. Aczel's recent article [1] contains further information about set-theoretic interpretations of type theory.

Large cardinals in set theory. Induction-recursion gives quite a general approach to type-theoretic analogues of large cardinals in set theory. See for example Drake [6] for an introduction to large cardinals. Induction-recursion gives rise to analogues of for example inaccessible, hyper-inaccessible cardinals, and more generally Mahlo's π-numbers [19], but does not justify the definition of a *set*, which is an analogue of a Mahlo cardinal. However, the *type* of sets has closure properties similar to those of a Mahlo cardinal.

Constructive versions of the model. The current model requires much more proof theoretic power than is actually needed: the strength of the type theory considered is very weak relative to ZF, even without any addition of large cardinals. Aczel [1] shows that the set theoretic models interpret as well the principle of excluded middle of type theory, an enormous strengthening of the type theory. In order to get a model in a theory which has the same strength, Aczel modifies the model and replaces ZF by constructive set theory CZF. One can as well define a model in a theory of the same strength by giving a realizability interpretation in Kripke-Platek set theory extended by a recursive Mahlo ordinal and ω admissibles above, extending [21, 24, 23]. Both models require some extra work, which exceeds the space available in this article.[10]

Proof-theoretic strength of type theory. It should be easy to develop a term model of the theory in KPM^+ used in [23] for the interpretation of Mahlo type theory. Such a model, which will make use of a (countable) *recursive Mahlo* ordinal and ω admissibles above it only, would show that the strength of the current type theory is at most as big as the Mahlo universe. On the other hand, set can be seen as being almost a Mahlo-universe, since we have induction over arbitrary types. What is missing to get the full strength is the possibility of having the W-type on top of the universe. In [10] together with [22], [24], [25] it was shown that in case of one universe such a restriction reduces the strength from $|\mathrm{KPI}^+|$ to $|\mathrm{KPI}|$ and with a similar argument for the lower bound as in [10] it is very likely that using

[10] The interpretation in the extension of Kripke-Platek set theory will be presented in an extended version of this article.

the Mahlo-feature of set we have a lower bound |KPM|. Therefore it seems that the strength of our theory lies in the interval $[|\mathrm{KPM}|, |\mathrm{KPM}^+|]$.

Inductive-recursive definitions seem to cover what is by many (but not all) researchers considered at the moment as predicative type theory. Even if some extensions are not covered by our calculus, it seems unlikely that such extensions will get beyond the strength of the Mahlo universe. This indicates that Mahloness is a natural boundary in the world of predicativity, which can only be crossed by adding principles such as the existence of the Mahlo universe as a set. The second author regards such principles as predicatively justifiable.

Inductive-recursive definition of indexed families. The external schema by Dybjer [7] considers the more general case of the simultaneous inductive-recursive definition of a set-indexed family of sets and functions. The present finite axiomatization can be extended to this case too, but we postpone the presentation of this to a future article.

References

[1] P. Aczel. On relating type theories and set theories. Submitted to *TYPES' 98*, LNCS, Springer-Verlag.

[2] S. Allen. *A Non-Type-Theoretic Semantics for Type-Theoretic Language*. PhD thesis, Department of Computer Science, Cornell University, 1987.

[3] R. Backhouse. On the meaning and construction of the rules in Martin-Löf's theory of types. In A. Avron, B. Harper, F. Honsell, I. Mason, and G. Plotkin, editors, *Proceedings of the Workshop on General Logic, Edinburgh, February 1987*. Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, 1988. ECS-LFCS-88-52.

[4] J. Cederquist. *Pointfree Approach to Constructive Analysis in Type Theory*. PhD thesis, Department of Computing Science, Chalmers University of Technology and University of Göteborg, 1997.

[5] T. Coquand and C. Paulin. Inductively defined types, preliminary version. In *LNCS 417, COLOG '88, International Conference on Computer Logic*. Springer-Verlag, 1990.

[6] F. R. Drake. *Set Theory - an Introduction to Large Cardinals*. North Holland, 1974.

[7] P. Dybjer. A general formulation of simultaneous inductive-recursive definitions in type theory. To appear in *Journal of Symbolic Logic*.

[8] P. Dybjer. Inductive sets and families in Martin-Löf's type theory and their set-theoretic semantics. In G. Huet and G. Plotkin, editors, *Logical Frameworks*, pages 280–306. Cambridge University Press, 1991.

[9] P. Dybjer. Inductive families. *Formal Aspects of Computing*, 6:440–465, 1994.

[10] E. Griffor and M. Rathjen. The strength of some Martin-Löf type theories. *Archive for Mathematical Logic*, 33:347 – 385, 1994.

[11] P. Martin-Löf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In J. E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 179–216. North-Holland, 1971.

[12] P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

[13] P. Martin-Löf. An intuitionistic theory of types. In G. Sambin and J. Smith, editors, *Twenty-Five Years of Constructive Type Theory*. Oxford University Press, 1998. To appear. Reprinted version of an unpublished report from 1972.

[14] P. F. Mendler. Predicative type universes and primitive recursion. In *Proceedings Sixth Annual Synposium on Logic in Computer Science*. IEEE Computer Society Press, 1991.

[15] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: an Introduction*. Oxford University Press, 1990.

[16] E. Palmgren. On universes in type theory. To appear in: G. Sambin, and J. Smith, editors: *Twenty-Five Years of Constructive Type Theory*.

[17] E. Palmgren. *On Fixed Point Operators, Inductive Definitions and Universes in Martin-Löf's Type Theory*. PhD thesis, Uppsala University, 1991.

[18] C. Paulin-Mohring. Inductive definitions in the system Coq - rules and properties. In *Proceedings Typed λ-Calculus and Applications*, pages 328–245. Springer-Verlag, LNCS, March 1993.

[19] M. Rathjen, E. R. Griffor, and E. Palmgren. Inaccessibility in constructive set theory and type theory. *Annals of Pure and Applied Logic*, 94:181 – 200, 1998.

[20] A. Setzer. Extending Martin-Löf Type Theory by one Mahlo-universe. To appear in *Archive for Mathematical Logic*.

[21] A. Setzer. *Proof theoretical strength of Martin-Löf Type Theory with W-type and one universe*. PhD thesis, Fakultät für Mathematik der Ludwig-Maximilians-Universität München, 1993.

[22] A. Setzer. *Proof theoretical strength of Martin-Löf Type Theory with W-type and one universe*. PhD thesis, Universität München, 1993.

[23] A. Setzer. A model for a type theory with Mahlo universe. Draft, 1996.

[24] A. Setzer. An upper bound for the proof theoretical strength of Martin-Löf Type Theory with W-type and one Universe. Draft, 1996.

[25] A. Setzer. Well-ordering proofs for Martin-Löf type theory. *Annals of Pure and Applied Logic*, 92:113 – 159, 1998.

[26] A. S. Troelstra. On the syntax of Martin-Löf's type theories. *Theoretical Computer Science*, 51:1–26, 1987.