A Theory of Explicit Mathematics Equivalent to ID_1

Reinhard Kahle[1] and Thomas Studer[2]

[1] WSI, Universität Tübingen,
Sand 13, D-72076 Tübingen, Germany
Tel. +49-7071-29 74036, Fax: +49-7071-29 5060
kahle@informatik.uni-tuebingen.de
[2] IAM, Universität Bern,
Neubrückstr. 10, CH-3012 Bern, Switzerland
Tel. +41-31-631 4976, Fax: +41-31-631 3965
tstuder@iam.unibe.ch

Abstract. We show that the addition of *name induction* to the theory $\mathsf{EETJ} + (\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I_N})$ of explicit elementary types with join yields a theory proof-theoretically equivalent to ID_1.

Keywords: Proof theory, explicit mathematics, inductive definitions.

1 Introduction

In this paper, we introduce a theory of explicit mathematics which is proof-theoretically equivalent to the well-known theory ID_1 of non-iterated positive arithmetical inductive definitions.

Explicit mathematics was introduced by Feferman to formalize Bishop-style constructive mathematics [Fef75,Fef79]. In the following, it turned out that this framework is important for proof-theoretic studies of subsystems of analysis and Kripke-Platek set theory. Moreover, it provides a very useful account to theoretical computer science, particularly, it is well-suited for the study of functional and object-oriented programming, cf. [Fef90,Fef91,Fef92,Stä97,Stä98,Stu0x].

Theories of explicit mathematics are formulated in a two sorted language. The first-order part, consisting of so-called *applicative theories*, is based on partial combinatory logic which can be extended axiomatically by additional constants, cf. [JKS99]. *Types* build the second sort of objects in explicit mathematics. They are extensional in the usual set-theoretic sense, but a special naming relation due to Jäger [Jäg88] allows us to deal with *names* of the types on the first-order level. These names show an intensional behaviour.

There exist a wide variety of theories of explicit mathematics. The proof-theoretic strength of the different theories cover a broad part of the landscape of mathematical theories. Nevertheless, the theory presented here is the first theory of explicit mathematics equivalent to ID_1.

The well-known theory ID_1 of non-iterated inductive definitions is one of the most prominent theories in proof theory. Formalizing least fixed points of positive arithmetical operator forms, it can be regarded as the most elementary *impredicative* theory. Going back to Kreisel [Kre63], its proof-theoretic study (and the study of its iterations) can be found in [Fef70,BFPS81,Poh89].

In order to get a theory with the proof-theoretic strength of ID_1, we will add the concept of *name induction* to the theory EETJ of *explicit elementary types with join*. That means that names of types can be built by use of *generators* only, i.e. that the naming relation $\Re$ is, so to say, *least*.

In the context of Martin-Löf's type theory, this leastness condition corresponds to certain elimination rules which have first been considered by Palmgren and later by Rathjen, also in connection with *universes*, [Pal98,GR94]. For applicative theories, the concept of name induction in the presence of universes is studied in detail in a joint work with Jäger, [JKS0x]. The theories studied in that paper exceed the strength of ID_1 substantially by having proof-theoretic strength of Feferman's theory T_0. For the notion of proof-theoretic strength, we refer to Feferman [Fef88,Fef0x].

In type systems dealing with record or object types the concept of structural rule is important. Simplifying, we can say that these rules rely on the assumption that the universe of types consists of record or object types only, cf. e.g. [AC96]. Name induction can be seen as a generalization of this idea since it allows us to prove that the only types that exists are those which are created by the generators.

The structure of the paper is as follows. In the next section, we introduce the theory NEM of explicit mathematics with name induction and state some basic results. As the core of the paper, we prove in Section 3 that NEM allows for the definition of *accessible parts*. This result is used in the fourth section to give an interpretation of $\mathsf{ID}_1^{\mathsf{acc}}$, a theory equivalent to ID_1, in NEM. In the final section, we describe a model of NEM which can be formalized in ID_1.

A substantial part of the work of the first author was elaborated while visiting Sol Feferman at Stanford University under support of the Deutsche Forschungsgemeinschaft. The work of the second author is supported by the Schweizerische Nationalfonds. This article benefits from fruitful discussion with Gerhard Jäger.

2 The Theory NEM of Explicit Mathematics with Name Induction

2.1 Explicit Mathematics

In this section, we present the theory EETJ of explicit elementary types with join.

The underlying language $\mathcal{L}_{\mathsf{EM}}$ is comprised of

- individual variables $a, b, c, f, u, v, w, x, y, z, \ldots$,
- type variables $A, B, S, T, U, V, X, Y, Z, \ldots$,
- individual constants k, s (combinators), $\mathsf{p}, \mathsf{p}_0, \mathsf{p}_1$ (pairing and projections), 0 (zero), s_N (successor), p_N (predecessor) and d_N (definition by numerical cases),

- *generators* which are special individual constants, namely nat (natural numbers), id (identity), co (complement), int (intersection), dom (domain), inv (inverse image) and j (join),
- one binary function symbol $\cdot$ for (partial) application of individuals to individuals,
- unary relation symbols $\downarrow$ (defined) and N (natural numbers) and
- binary relation symbols $\in$ (membership), $=$ (equality) and $\Re$ (naming or representation).

Individual terms $(r, s, t, r_1, s_1, t_1, \ldots)$ of $\mathcal{L}_{\mathsf{EM}}$ are built up from individual variables and individual constants by means of the function symbol $\cdot$. We use (st) or st as an abbreviation for $(s \cdot t)$ and adopt the convention of association to the left, i.e. $s_1 s_2 \ldots s_n$ stands for $(\ldots(s_1 \cdot s_2) \ldots s_n)$.

Atomic formulae of $\mathcal{L}_{\mathsf{EM}}$ are $\mathsf{N}(s)$, $s\downarrow$, $s = t$, $U = V$, $s \in U$ and $\Re(s, U)$. $\mathsf{N}(s)$ means that s is a natural number. $s\downarrow$ means that s *is defined* or s *has a value.* $\Re(s, U)$ is the naming relation, expressing that the individual s *represents* the type U or is a *name* of U.

The *formulae* of $\mathcal{L}_{\mathsf{EM}}$ $(\varphi, \psi, \ldots)$ are built up from the atomic formulae by use of the usual propositional connectives and quantification in both sorts, over individuals as well as over types.

A formula which contains neither quantifiers over types nor the naming relation $\Re$ is called *elementary.*

As abbreviations, we use:

$$
\begin{aligned}
t' &:= \mathsf{s_N} t, \\
(s, t) &:= \mathsf{p} st, \\
s \simeq t &:= s\downarrow \vee t\downarrow \rightarrow s = t, \\
s \neq t &:= s\downarrow \wedge t\downarrow \wedge \neg(s = t), \\
s \in \mathsf{N} &:= \mathsf{N}(s), \\
\exists x \in \mathsf{N}.\varphi(x) &:= \exists x.x \in \mathsf{N} \wedge \varphi(x), \\
\forall x \in \mathsf{N}.\varphi(x) &:= \forall x.x \in \mathsf{N} \rightarrow \varphi(x), \\
s \mathrel{\dot\in} t &:= \exists X.\Re(t, X) \wedge s \in X, \\
\exists x \mathrel{\dot\in} s.\varphi(x) &:= \exists x.x \mathrel{\dot\in} s \wedge \varphi(x), \\
\forall x \mathrel{\dot\in} s.\varphi(x) &:= \forall x.x \mathrel{\dot\in} s \rightarrow \varphi(x), \\
\Re(s) &:= \exists X.\Re(s, X).
\end{aligned}
$$

The logic for the first-order part of theories of explicit mathematics is Beeson's classical *logic of partial terms*, cf. [Bee85,TvD88]. The second order part is based on classical logic with equality.

The nonlogical axioms of EETJ can be divided into the following groups.

I. Applicative axioms.

(1) $\mathsf{k}ab = a$,

(2) $\mathsf{s}ab\downarrow \wedge \mathsf{s}abc \simeq ac(bc)$,
(3) $\mathsf{p_0}(a,b) = a \wedge \mathsf{p_1}(a,b) = b$,
(4) $0 \in \mathsf{N} \wedge \forall x \in \mathsf{N}.x' \in \mathsf{N}$,
(5) $\forall x \in \mathsf{N}.x' \neq 0 \wedge \mathsf{p_N}(x') = x$,
(6) $\forall x \in \mathsf{N}.x \neq 0 \rightarrow \mathsf{p_N}x \in \mathsf{N} \wedge (\mathsf{p_N}x)' = x$,
(7) $a \in \mathsf{N} \wedge b \in \mathsf{N} \wedge a = b \rightarrow \mathsf{d_N}xyab = x$,
(8) $a \in \mathsf{N} \wedge b \in \mathsf{N} \wedge a \neq b \rightarrow \mathsf{d_N}xyab = y$.

II. Explicit representation and extensionality.

(1) $\exists x.\Re(x,U)$,
(2) $\Re(a,U) \wedge \Re(a,V) \rightarrow U = V$,
(3) $(\forall x.x \in U \leftrightarrow x \in U) \rightarrow U = V$.

III. Basic type existence axioms.

Natural numbers

$\Re(\mathsf{nat}) \wedge \forall x.x \dot{\in} \mathsf{nat} \leftrightarrow \mathsf{N}(x)$.

Identity

$\Re(\mathsf{id}) \wedge \forall x.x \dot{\in} \mathsf{id} \leftrightarrow \exists y.x = (y,y)$.

Complements

$\Re(a) \rightarrow \Re(\mathsf{co}(a)) \wedge \forall x.x \dot{\in} \mathsf{co}(a) \leftrightarrow x \dot{\notin} a$.

Intersections

$\Re(a) \wedge \Re(b) \rightarrow \Re(\mathsf{int}(a,b)) \wedge \forall x.x \dot{\in} \mathsf{int}(a,b) \leftrightarrow x \dot{\in} a \wedge x \dot{\in} b$.

Domains

$\Re(a) \rightarrow \Re(\mathsf{dom}(a)) \wedge \forall x.x \dot{\in} \mathsf{dom}(a) \leftrightarrow \exists y.(x,y) \dot{\in} a$.

Inverse images

$\Re(a) \rightarrow \Re(\mathsf{inv}(a,f)) \wedge \forall x.x \dot{\in} \mathsf{inv}(a,f) \leftrightarrow fx \dot{\in} a$.

Joins

$\Re(a) \wedge (\forall x \dot{\in} a.\Re(fx)) \rightarrow \Re(\mathsf{j}(a,f)) \wedge \Sigma(a,f,\mathsf{j}(a,f))$,

where $\Sigma(a,f,b)$ means that b names the disjoint union of f over a, defined as

$$\Sigma(a,f,b) := \forall x.x \dot{\in} b \leftrightarrow \exists y,z.x = (y,z) \wedge y \dot{\in} a \wedge z \dot{\in} fy.$$

IV. Uniqueness of generators. With respect to $\mathcal{L}_{\mathsf{EM}}$, it is given by the collection ($\mathcal{L}_{\mathsf{EM}}$-UG) of the following axioms for all syntactically different generators r_0 and r_1 and arbitrary generators s and t of $\mathcal{L}_{\mathsf{EM}}$:

(1) $r_0 \neq r_1$,
(2) $\forall x.sx \neq \mathsf{nat} \wedge sx \neq \mathsf{id}$,

(3) $\forall x, y. sx = ty \rightarrow s = t \wedge x = y$.

EETJ is the theory consisting of all axioms of the groups I. – IV.

As addition to the axioms of EETJ, we will consider the induction principle ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{I}_{\mathsf{N}}$), the schema of complete induction on N for arbitrary formulae $\varphi(u)$:

$$(\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I}_{\mathsf{N}}) \qquad \varphi(0) \wedge (\forall x \in \mathsf{N}. \varphi(x) \rightarrow \varphi(x')) \rightarrow \forall x \in \mathsf{N}. \varphi(x)$$

It is a well-known result that we can introduce λ abstraction and recursion using the combinator axioms (1) and (2), cf. [Fef75,Bee85].

Proposition 1.

1. *For every variable x and every term t of $\mathcal{L}_{\mathsf{EM}}$, there exists a term $\lambda x.t$ of $\mathcal{L}_{\mathsf{EM}}$ whose free variables are those of t, excluding x, such that*

$$\mathsf{EETJ} \vdash \lambda x.t \downarrow \wedge (\lambda x.t)\, x \simeq t.$$

2. *There exists a term* rec *of* $\mathcal{L}_{\mathsf{EM}}$ *such that*

$$\mathsf{EETJ} \vdash \mathsf{rec}\, f \downarrow \wedge \forall x. \mathsf{rec}\, f\, x \simeq f\, (\mathsf{rec}\, f)\, x.$$

Our definition EETJ is based on a finite axiomatization of elementary comprehension. This approach is essential for the formulation of name induction below. In contrast, the original definition of EETJ employed an infinite axiom schema. A theorem of Feferman and Jäger [FJ96] shows that this schema is derivable from the finite axiomatization.

Lemma 1 (Elementary comprehension). *Let φ be an elementary $\mathcal{L}_{\mathsf{EM}}$ formula with no (distinct) individual variables other than $z_1, \ldots, z_{m+1}$ and no (distinct) type variables other than $Z_1, \ldots, Z_n$. Then there exists a closed individual term t of $\mathcal{L}_{\mathsf{EM}}$, depending on φ, such that* EETJ *proves for all individual terms $\boldsymbol{a} = a_1, \ldots, a_m$, $\boldsymbol{b} = b_1, \ldots, b_n$ and type terms $\boldsymbol{S} = S_1, \ldots, S_n$ that:*

1. $\Re(\boldsymbol{b}, \boldsymbol{S}) \rightarrow \Re(t(\boldsymbol{a}, \boldsymbol{b}))$,
2. $\Re(\boldsymbol{b}, \boldsymbol{S}) \rightarrow \forall x (x \,\dot{\in}\, t(\boldsymbol{a}, \boldsymbol{b}) \leftrightarrow \varphi[x, \boldsymbol{a}, \boldsymbol{S}])$.

Informally, we will write $\{x : \varphi(x)\}$ for the collection of all individuals c satisfying $\varphi(c)$. Using this notation, the lemma expresses that, for elementary formulae $\varphi[u, \boldsymbol{y}, \boldsymbol{Y}]$, the following hold:

1. $\{x : \varphi[x, \boldsymbol{a}, \boldsymbol{S}]\}$ is a type,
2. there is a name $t(\boldsymbol{a}, \boldsymbol{b})$ for this type which is given uniformly in the individual parameters and the names of the type parameters.

2.2 Name Induction

In this section, we define the schema of *name induction*. This induction principle states that names can be defined by means of generators only. Because, in a certain sense, names can be seen as intensional representations of sets, we get an intensional version of $\in$ induction.

In order to state the formal definition of name induction, we introduce as auxiliary notation the closure condition $\mathcal{C}(\varphi, a)$ as the disjunction of the following formulae:

(1) $a = \mathsf{nat} \vee a = \mathsf{id}$,
(2) $\exists x.a = \mathsf{co}(x) \wedge \varphi(x)$,
(3) $\exists x, y.a = \mathsf{int}(x, y) \wedge \varphi(x) \wedge \varphi(y)$,
(4) $\exists x.a = \mathsf{dom}(x) \wedge \varphi(x)$,
(5) $\exists f, x.a = \mathsf{inv}(f, x) \wedge \varphi(x)$,
(6) $\exists f, x.a = \mathsf{j}(x, f) \wedge \varphi(x) \wedge \forall y \,\dot{\in}\, x.\varphi(fy)$.

The schema of name induction is now given by

$$(\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I}_{\Re}) \qquad (\forall x.\mathcal{C}(\varphi, x) \to \varphi(x)) \to \forall x.\Re(x) \to \varphi(x),$$

for arbitrary formulae $\varphi(x)$ of $\mathcal{L}_{\mathsf{EM}}$.

The theory NEM of *explicit mathematics with name induction* consists of the axioms of EETJ plus ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{I}_{\mathsf{N}}$) and ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{I}_{\Re}$).

As a first consequence of ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{I}_{\Re}$), we prove *name strictness* which, more explicitly, says the (appropriate) arguments of generators of names are names, too. This is represented by the conjunction $\mathsf{Str}(\Re)$ of the following clauses:

(1) $\forall x.\Re(\mathsf{co}(x)) \to \Re(x)$,
(2) $\forall x, y.\Re(\mathsf{int}(x, y)) \to \Re(x) \wedge \Re(y)$,
(3) $\forall x.\Re(\mathsf{dom}(x)) \to \Re(x)$,
(4) $\forall f, x.\Re(\mathsf{inv}(f, x)) \to \Re(x)$,
(5) $\forall f, x.\Re(\mathsf{j}(x, f)) \to \Re(x) \wedge \forall y \,\dot{\in}\, x.\Re(fy)$.

To show $\mathsf{Str}(\Re)$ in NEM, we first note that the closure of the names under condition $\mathcal{C}$ is guaranteed by the type existence axioms of EETJ:

$$\mathsf{EETJ} \vdash \mathcal{C}(\Re, x) \to \Re(x).$$

Lemma 2. $\mathsf{NEM} \vdash \mathsf{Str}(\Re)$.

Proof. The proof is straightforward using ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{I}_{\Re}$) on the formula $\mathcal{C}(\Re, x)$, i.e. we have

$$(\forall x.\mathcal{C}(\mathcal{C}(\Re, x), x) \to \mathcal{C}(\Re, x)) \to \forall x.\Re(x) \to \mathcal{C}(\Re, x).$$

The premise follows immediately from the preceding remark and the fact that φ occurs only positively in $\mathcal{C}(\varphi, x)$. From the consequence $\forall x.\Re(x) \to \mathcal{C}(\Re, x)$ we get the required conclusion $\mathsf{Str}(\Re)$ by substituting the different names. For example, for clause (5) we have

$$\begin{aligned}\Re(\mathsf{j}(x, f)) &\to \mathcal{C}(\Re, \mathsf{j}(x, f)) \\ &\to \exists g, z.\mathsf{j}(x, f) = \mathsf{j}(z, g) \wedge \Re(z) \wedge \forall y \,\dot{\in}\, z.\Re(gy) \\ &\to \exists g, z.x = z \wedge f = g \wedge \Re(z) \wedge \forall y \,\dot{\in}\, z.\Re(gy) \\ &\to \Re(x) \wedge \forall y \,\dot{\in}\, x.\Re(fy)\end{aligned}$$

For this argument, the uniqueness of generators ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{UG}$) is essential.

3 Accessible Parts in NEM

For the proof-theoretic analysis of NEM, the crucial property is the possibility of defining *accessible parts*. This will be used in the next section to embed the theory $\mathsf{ID}_1^{\mathsf{acc}}$ in NEM.

Let us introduce the following abbreviation:

$$\mathsf{Closed}(a, b, \varphi) := \forall x \,\dot{\in}\, a.(\forall y \,\dot{\in}\, a.(y, x) \,\dot{\in}\, b \to \varphi(y)) \to \varphi(x).$$

If b is a name for a binary relation, then $\mathsf{Closed}(a, b, \varphi)$ expresses that φ holds for all elements $c \,\dot{\in}\, a$ if it holds for all predecessors of c in a with respect to the relation named by b.

Using this abbreviation we can state the following proposition which is the essential step of the embedding of $\mathsf{ID}_1^{\mathsf{acc}}$.

Theorem 1. *There exists a formula* $\mathsf{Acc}(a, b, x)$ *such that* NEM *proves for arbitrary formulae* $\varphi(x)$*:*

(Acc.1) $\Re(a) \wedge \Re(b) \to \mathsf{Closed}(a, b, \mathsf{Acc}(a, b, \cdot))$,
(Acc.2) $\Re(a) \wedge \Re(b) \wedge \mathsf{Closed}(a, b, \varphi) \to \forall x.\mathsf{Acc}(a, b, x) \to \varphi(x)$.

Proof. Let us assume $\Re(a, A)$ and $\Re(b, B)$. We set $A_x = \{y \in A | (y, x) \in B\}$, i.e. the subset of A consisting of all B-predecessors of x. By elementary comprehension, there exists a closed term pd so that $\Re(\mathsf{pd}\,(a, b, x), A_x)$.

By use of the recursion theorem, we can define a term f satisfying the equation:

$$\mathsf{f}\,(a, b, c) \simeq \mathsf{j}\,(\mathsf{pd}\,(a, b, c), \lambda y.\mathsf{f}(a, b, y)). \qquad (\star)$$

Hence, f maps an element $c \in A$ to the disjoint union of all f-images of B-predecessors of c. Using f, we define the formula Acc in the following way:

$$\mathsf{Acc}(a, b, c) := c \,\dot{\in}\, a \wedge \Re(\mathsf{f}\,(a, b, c)).$$

If $\mathsf{Acc}(a, b, c)$ holds we say that "c is accessible". The idea of its definition is the following. $\mathsf{pd}\,(a, b, c)$ is the name of the set A_c which contains of all B-predecessors of c in A. Using join, we associate this set with a set of elements which can be proven to be names if $\mathsf{f}\,(a, b, c)$ is a name. This trick allows us to encode *arbitrary* objects of our language by *names*, and then name induction can be used to prove the required properties.

(Acc.1) To show $\mathsf{Closed}(a, b, \mathsf{Acc}(a, b, \cdot))$, we choose an element c of A such that

$$\forall y \,\dot{\in}\, a.(y, c) \,\dot{\in}\, b \to \mathsf{Acc}(a, b, y).$$

The definition of pd yields

$$\forall y.y \,\dot{\in}\, \mathsf{pd}\,(a, b, c) \to \mathsf{Acc}(a, b, y).$$

This implies by the definition of Acc that

$$\forall y.y \,\dot{\in}\, \mathsf{pd}\,(a, b, c) \to \Re(\mathsf{f}\,(a, b, y)).$$

From the axioms about join, we obtain

$$\Re(\mathsf{j}\,(\mathsf{pd}\,(a,b,c),\mathsf{f})).$$

By the equation ($\star$), this means $\Re(\mathsf{f}\,(a,b,c))$. Together with the assumption $c \dot\in a$ we have $\mathsf{Acc}(a,b,c)$. Since c was chosen arbitrarily, the proof of $\mathsf{Closed}(a,b,\mathsf{Acc})$ is completed.

(Acc.2) To prove the second assertion we first show two auxiliary statements (A) and (B).

(A) says that if c is accessible, then all its b predecessors are accessible, too.

$$\mathsf{Acc}(a,b,c) \rightarrow (\forall x \dot\in \mathsf{pd}\,(a,b,c).\mathsf{Acc}(a,b,x)). \tag{A}$$

Assuming $\mathsf{Acc}(a,b,c)$, we get by ($\star$) that $\Re(\mathsf{j}\,(\mathsf{pd}\,(a,b,c),\lambda y.\mathsf{f}(a,b,y)))$ holds. Then $\forall x \dot\in \mathsf{pd}\,(a,b,c).\Re(\mathsf{f}(a,b,x))$ is a consequence of Lemma 2 about name strictness. To complete the proof of (A), we have to check that $\forall x \dot\in \mathsf{pd}\,(a,b,c).x \dot\in a$, which immediately follows from the definition of pd.

In order to formulate the assertion (B), we define an additional formula $\psi_\varphi(u,v,w)$ depending on a formula $\varphi(x)$ which will be used as induction formula in the schema of name induction. Using the definition of f, here we "replace" an arbitrary objects by their associated names.

$$\psi_\varphi(a,b,u) := \forall y.\mathsf{Acc}(a,b,y) \wedge \mathsf{f}\,(a,b,y) = u \rightarrow \varphi(y).$$

Now, the statement (B) reads as

$$\mathsf{Closed}(a,b,\varphi) \wedge \mathcal{C}(\psi_\varphi(a,b,\cdot),u) \rightarrow \psi_\varphi(a,b,u). \tag{B}$$

For the proof of (B), we assume $\mathsf{Closed}(a,b,\varphi) \wedge \mathcal{C}(\psi_\varphi(a,b,\cdot),u)$ and $\mathsf{Acc}(a,b,c) \wedge \mathsf{f}\,(a,b,c) = u$, from which we have to show $\varphi(c)$. From the last assumption, we get by ($\star$):

$$u = \mathsf{j}\,(\mathsf{pd}\,(a,b,c),\lambda y.\mathsf{f}(a,b,y)).$$

Uniqueness of generators and clause (5) of $\mathcal{C}(\psi_\varphi(a,b,\cdot),u)$ yield

$$\forall x \dot\in \mathsf{pd}\,(a,b,c).\psi_\varphi(a,b,\mathsf{f}(a,b,x))).$$

By the definition of ψ_φ, this reads

$$\forall x \dot\in \mathsf{pd}\,(a,b,c).\forall y.\mathsf{Acc}(a,b,y) \wedge \mathsf{f}\,(a,b,y) = \mathsf{f}(a,b,x) \rightarrow \varphi(y).$$

Choosing x for y, we get

$$\forall x \dot\in \mathsf{pd}\,(a,b,c).\mathsf{Acc}(a,b,x) \rightarrow \varphi(x).$$

Assuming $\mathsf{Acc}(a,b,c)$, we obtain by (A) that $\forall x \dot\in \mathsf{pd}\,(a,b,c).\mathsf{Acc}(a,b,x)$ holds. So we have

$$\forall x \dot\in \mathsf{pd}\,(a,b,c).\varphi(x).$$

But this is the premise of the assumption $\mathsf{Closed}(a,b,\varphi)$ and we get $A(c)$. Thus, (B) is proven.

To prove the second assertion (Acc.2), we now take an arbitrary formula $\varphi(x)$ and assume $\mathsf{Closed}(a,b,\varphi)$ and $\mathsf{Acc}(a,b,x)$. For the first assumption (B) yields

$$\forall y.\mathcal{C}(\psi_\varphi(a,b,\cdot),y) \to \psi_\varphi(a,b,y).$$

This is just the premise of name induction for $\psi_\varphi(a,b,y)$ and we get from ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{I}_{\Re}$)

$$\forall y.\Re(y) \to \psi_\varphi(a,b,y).$$

By the definition of $\psi_\varphi(a,b,y)$, this is

$$\forall y.\Re(y) \to \forall x.\mathsf{Acc}(a,b,x) \wedge \mathsf{f}\,(a,b,x) = y \to \varphi(x).$$

Since the assumption $\mathsf{Acc}(a,b,x)$ implies $\Re(\mathsf{f}\,(a,b,x))$, we can choose y as $\mathsf{f}\,(a,b,x)$ and all premises are satisfied. Therefore we finally obtain the required result $\varphi(x)$.

In this proof we followed the presentation of the corresponding proof in [JKS0x], where the principle of *inductive generation* is verified in the presence of *universes*.

4 Modelling $\mathsf{ID}_1^{\mathsf{acc}}$ in NEM

To show the lower bound of NEM, we will embed the theory $\mathsf{ID}_1^{\mathsf{acc}}$ of *accessibility elementary inductive definitions*, cf. [BFPS81,Can96]. Let $\mathcal{L}_1$ be the language of Peano arithmetic. In order to obtain $\mathcal{L}_{\mathsf{ID}}$, we extend this language by adding new unary predicate symbols $\mathcal{P}_\varphi$ for every formula $\varphi(x,y)$ of $\mathcal{L}_1$ containing two distinct free variables. For the definition of $\mathsf{ID}_1^{\mathsf{acc}}$, we extend the axioms of PA to the new language, including formulae induction for arbitrary $\mathcal{L}_{\mathsf{ID}}$ formulae, and add for each new predicate symbol $\mathcal{P}_\varphi$ and each $\mathcal{L}_{\mathsf{ID}}$ formula ψ the following two axioms:

($\mathsf{ID}_1^{\mathsf{acc}}$.1) $\quad \forall x.(\forall y.\varphi(x,y) \to \mathcal{P}_\varphi(y)) \to \mathcal{P}_\varphi(x)$

($\mathsf{ID}_1^{\mathsf{acc}}$.2) $\quad (\forall x.(\forall y.\varphi(x,y) \to \psi(y)) \to \psi(x)) \to \forall x.\mathcal{P}_\varphi(x) \to \psi(x)$

It is well-known that Peano arithmetic can be embedded in $\mathsf{EETJ} + (\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I}_{\mathsf{N}})$, indeed in its applicative fragment $\mathsf{BON} + (\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I}_{\mathsf{N}})$, using an interpretation $\cdot^N$, cf. [FJ93]. This interpretation translates formulae of $\mathcal{L}_1$ into elementary formulae of $\mathcal{L}_{\mathsf{EM}}$. Thus, by elementary comprehension we get for every binary formulae $\varphi(x,y)$ of $\mathcal{L}_1$ a name t_{φ^N} for the corresponding type, i.e. EETJ proves that t_{φ^N} is a name for $\{(x,y)|x \in \mathsf{N} \wedge y \in \mathsf{N} \wedge \varphi^N(x,y)\}$. These names will be employed in the proof of the following theorem to represent the binary relations which are used in the definition of $\mathsf{ID}_1^{\mathsf{acc}}$.

Theorem 2. *There exists a translation $\cdot^N$ from $\mathcal{L}_{\mathsf{ID}}$ to $\mathcal{L}_{\mathsf{EM}}$ such that*

$$\mathsf{ID}_1^{\mathsf{acc}} \vdash \varphi \quad \Rightarrow \quad \mathsf{NEM} \vdash \varphi^N$$

for all $\mathcal{L}_{\mathsf{ID}}$ formulae φ.

Proof. To interpret $\mathsf{ID}_1^{\mathsf{acc}}$ in NEM we extend the translation $\cdot^N$ by setting

$$[\mathcal{P}_\varphi(x)]^N := \mathsf{Acc}(\mathsf{nat}, t_{\varphi^N}, x),$$

where $\mathsf{Acc}(x, y, z)$ is defined as in Theorem 1. Then the proof runs by induction on the length of the derivation of $\mathsf{ID}_1^{\mathsf{acc}} \vdash \varphi$. In addition to the embedding of PA in EETJ, we need only to check the axioms for the new predicate symbols. The translation of $(\mathsf{ID}_1^{\mathsf{acc}}.1)$ reads as

$$\begin{aligned}
&[\forall x.(\forall y.\varphi(x, y) \to \mathcal{P}_\varphi(y)) \to \mathcal{P}_\varphi(x)]^N \\
\leftrightarrow\ & \forall x \mathrel{\dot\in} \mathsf{nat}.(\forall y \mathrel{\dot\in} \mathsf{nat}.\varphi^N(x, y) \to \mathsf{Acc}(\mathsf{nat}, t_{\varphi^N}, y)) \to \mathsf{Acc}(\mathsf{nat}, t_{\varphi^N}, x) \\
\leftrightarrow\ & \mathsf{Closed}(\mathsf{nat}, t_{\varphi^N}, \mathsf{Acc}(\mathsf{nat}, t_{\varphi^N}, \cdot)).
\end{aligned}$$

Since the last line is an instance of (Acc.1) of Theorem 1, this axiom is verified. In the same way, $(\mathsf{ID}_1^{\mathsf{acc}}.2)^N$ follows from (Acc.2):

$$\begin{aligned}
&[(\forall x.(\forall y.\varphi(x, y) \to \psi(y)) \to \psi(x)) \to \forall x.\mathcal{P}_\varphi(x) \to \psi(x)]^N \\
\leftrightarrow\ & (\forall x \mathrel{\dot\in} \mathsf{nat}.(\forall y \mathrel{\dot\in} \mathsf{nat}.\varphi^N(x, y) \to \psi^N(y)) \to \psi^N(x)) \\
& \qquad\qquad \to \forall x \mathrel{\dot\in} \mathsf{nat}.\mathsf{Acc}(\mathsf{nat}, t_{\varphi^N}, x) \to \psi^N(x) \\
\leftrightarrow\ & \mathsf{Closed}(\mathsf{nat}, t_{\varphi^N}, \psi^N) \to \forall x.\mathsf{Acc}(\mathsf{nat}, t_{\varphi^N}, x) \to \psi^N(x).
\end{aligned}$$

The last line is an instance of (Acc.2), and we have finished the embedding of $\mathsf{ID}_1^{\mathsf{acc}}$.

5 Modelling NEM in ID_1

In this section, we embed NEM in the theory ID_1 of non-iterated inductive definitions. This extension of Peano arithmetic postulates the existence of least fixed points for positive arithmetical operator forms. These are formulae $\varphi(R, x)$ in the language $\mathcal{L}_1$ with one additional relation symbol R that has only positive occurrences in φ. The language of ID_1 is $\mathcal{L}_1$ extended by new predicate symbols $\mathcal{P}_\varphi$ for each positive operator form $\varphi(R, x)$. As axioms, we choose those of PA, including formulae induction extended to the new language and the following two principles for each new predicate symbol $\mathcal{P}_\varphi$ and arbitrary formulae ψ:

($\mathsf{ID}_1.1$) $\quad \forall x.\varphi(\mathcal{P}_\varphi, x) \to \mathcal{P}_\varphi(x)$

($\mathsf{ID}_1.2$) $\quad (\forall x.\varphi(\psi/R, x) \to \psi(x)) \to \forall x.\mathcal{P}_\varphi(x) \to \psi(x)$

Here $\varphi(\psi/R, x)$ denotes the result of substituting any occurrence of $R(t)$ in φ by $\psi(t/x)$.

In [Fef75], Feferman presented an inductive model construction for explicit mathematics. Beeson showed in [Bee85] that for the system $\mathsf{EETJ} + (\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I_N})$ this construction can be carried out in the theory $\widehat{\mathsf{ID}}_1$, cf. also [Mar94,MS98]. This theory stating only the existence of (not necessarily least) fixed points of positive arithmetical operator forms can be obtained from ID_1 by replacing the axioms (ID_1.1) and (ID_1.2) by

$$(\widehat{\mathsf{ID}}_1) \qquad \forall x.\varphi(\mathcal{P}_\varphi, x) \leftrightarrow \mathcal{P}_\varphi(x).$$

In fact, we can use Beeson's formalization for the analysis of NEM using, in addition, the induction principle of ID_1 to verify name induction ($\mathcal{L}_{\mathsf{EM}}$-$\mathsf{I}_\Re$). The only differences are the adaption to the finite axiomatization of elementary comprehension and the (trivial) verification of uniqueness of generators ($\mathcal{L}_{\mathsf{EM}}$-UG) which was not part of the original formulation of EETJ.

We start with a standard interpretation $\cdot^\star$ of the applicative structure using the relation $App(x, y, z) := \{x\}(y) \simeq y$ in the sense of ordinary recursion, cf. [FJ93]. Here, the constants of $\mathcal{L}_{\mathsf{EM}}$ are interpreted by numerals of $\mathcal{L}_1$ coding appropriate number-theoretic functions satisfying the axioms of EETJ. With respect to the generators we have to choose numerals according to the following codes which will be used for the interpretation of the type structure:

- $\langle 1 \rangle$ codes the type of numerals,
- $\langle 2 \rangle$ codes the type of pairs with identical elements,
- $\langle 3, a \rangle$ codes the complement of the type coded by a,
- $\langle 4, a, b \rangle$ codes the intersection of the two types coded by a and b,
- $\langle 5, a \rangle$ codes the domain of a function given as a type of ordered pairs coded by a
- $\langle 6, f, a \rangle$ codes the inverse images of f, i.e. the type of all individuals x with fx is an element of the type coded by a,
- $\langle 7, a, f \rangle$ codes the join of f over the type coded by a.

By choosing the codes for the generators according to these conditions, the axioms about uniqueness of generators are obviously satisfied.

To interpret the second order part of NEM we define three relations Typ, In and $\overline{\mathsf{In}}$, using appropriate operator forms. The meaning of these predicates and their relation to $\mathcal{L}_{\mathsf{EM}}$ is as follows. Let s, t be terms of ID_1 interpreting types S, T of $\mathcal{L}_{\mathsf{EM}}$, respectively, and let r be the interpretation of an arbitrary $\mathcal{L}_{\mathsf{EM}}$ term, then we have:

- $\mathsf{Typ}(t)$ represents that t is a code of a type.
- $\mathsf{In}(r, t)$ interprets the formula $r \in T$.
- $\overline{\mathsf{In}}(r, t)$ holds for $\neg r \in T$.
- We have to introduce the relation $\overline{\mathsf{In}}$ in order to guarantee that the defining operator forms are positive. As a consequence, we have to prove that $\mathsf{In}(r, t)$ is equivalent to $\neg\overline{\mathsf{In}}(r, t)$.
- $T = S$ is interpreted by $\mathsf{Typ}(t) \wedge \mathsf{Typ}(s) \wedge \forall x.\mathsf{In}(x, t) \leftrightarrow \mathsf{In}(x, s)$, i.e. as extensional equality.
- $\Re(t, S)$ is also modelled by $\mathsf{Typ}(t) \wedge \mathsf{Typ}(s) \wedge \forall x.\mathsf{In}(x, t) \leftrightarrow \mathsf{In}(x, s)$.

In order to define $\mathsf{Typ}(x)$, $\mathsf{In}(x, y)$ and $\overline{\mathsf{In}}(x, y)$ we need some coding. Let us use $\varphi^0(x)$, $\varphi^1(x, y)$ and $\varphi^2(x, y)$ as abbreviations for $\varphi(\langle 0, x \rangle)$, $\varphi(\langle 1, \langle x, y \rangle \rangle)$ and $\varphi(\langle 2, \langle x, y \rangle \rangle)$, respectively. With this notation we can define $\mathsf{Typ}(x)$, $\mathsf{In}(x, y)$ and

$\overline{\mathsf{In}}(x,y)$ as the "projections" $\mathcal{P}^0_{\boldsymbol{\varphi}}(x)$, $\mathcal{P}^1_{\boldsymbol{\varphi}}(x,y)$ and $\mathcal{P}^2_{\boldsymbol{\varphi}}(x,y)$ of the fixed point $\mathcal{P}_{\boldsymbol{\varphi}}$ of the positive operator form:

$$\begin{aligned}\boldsymbol{\varphi}(\psi,z) := {} & (\exists y.z = \langle 0,y\rangle \wedge \mathcal{C}_{\mathsf{Typ}}(\psi,y)) \vee \\ & (\exists x,y.z = \langle 1,\langle x,y\rangle\rangle \wedge \mathcal{C}_{\mathsf{In}}(\psi,x,y)) \vee \\ & (\exists x,y.z = \langle 2,\langle x,y\rangle\rangle \wedge \mathcal{C}_{\overline{\mathsf{In}}}(\psi,x,y))\end{aligned}$$

with the following closure conditions (where it is helpful to keep in mind the intended meanings of ψ^0, ψ^1 and ψ^2, namely Typ, In and $\overline{\mathsf{In}}$, respectively). $\mathcal{C}_{\mathsf{Typ}}(\psi,z)$ is the disjunction of the following clauses:

- $z = \langle 1\rangle$,
- $z = \langle 2\rangle$,
- $\exists x.z = \langle 3,x\rangle \wedge \psi^0(x)$,
- $\exists x,y.z = \langle 4,x,y\rangle \wedge \psi^0(x) \wedge \psi^0(x)$,
- $\exists x.z = \langle 5,x\rangle \wedge \psi^0(x)$,
- $\exists f,x.z = \langle 6,f,x\rangle \wedge \psi^0(x)$,
- $\exists f,x.z = \langle 7,x,f\rangle \wedge \psi^0(x) \wedge \forall y.\neg\psi^2(y,x) \to \psi^0(\{f\}(y))$.

$\mathcal{C}_{\mathsf{In}}(\psi,u,z)$ is the disjunction of the following clauses:

- $z = \langle 0\rangle$,
- $z = \langle 1\rangle \wedge \exists y.u = \langle y,y\rangle$,
- $\exists x.z = \langle 2,x\rangle \wedge \psi^0(x) \wedge \psi^2(u,x)$,
- $\exists x,y.z = \langle 4,x,y\rangle \wedge \psi^0(x) \wedge \psi^0(x) \wedge \psi^1(u,x) \wedge \psi^1(u,y)$,
- $\exists x.z = \langle 5,x\rangle \wedge \psi^0(x) \wedge \exists v.\psi^1(\langle u,v\rangle,x)$,
- $\exists f,x.z = \langle 6,f,x\rangle \wedge \psi^0(x) \wedge \psi^1(\{f\}(u),x)$,
- $\exists f,x.z = \langle 7,x,f\rangle \wedge \psi^0(x) \wedge (\forall y.\neg\psi^2(y,x) \to \psi^0(\{f\}(y))) \wedge$
  $\exists v,w.u = \langle v,w\rangle \wedge \psi^1(v,x) \wedge \psi^1(w,\{f\}(v))$.

The defining clauses for $\mathcal{C}_{\overline{\mathsf{In}}}$ are analogous, also containing positive occurrences of ψ only.

Without the leastness property for the fixed point defined by $\boldsymbol{\varphi}$ we cannot prove that In and $\overline{\mathsf{In}}$ are complementary. Hence, for embedding $\mathsf{EETJ} + (\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I_N})$ in $\widehat{\mathsf{ID}}_1$ one has to make use of Aczel's trick of sorting out all codes a for types where $\mathsf{In}(\cdot,a)$ is not the complement of $\overline{\mathsf{In}}(\cdot,a)$. However, in ID_1 the leastness condition allows for a direct proof that In and $\overline{\mathsf{In}}$ are complements, cf. [Bee85].

Lemma 3. $\mathsf{ID}_1 \vdash \mathsf{Typ}(y) \to \forall x.\mathsf{In}(x,y) \leftrightarrow \neg\overline{\mathsf{In}}(x,y)$.

Theorem 3. NEM *can be embedded in* ID_1.

Proof. The interpretation $\cdot^\star$ is chosen according to the remarks above. The verification of the axioms of EETJ and the induction schema $(\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I_N})$ is straightforward, cf. [Bee85] and [Mar94]. It only remains to check the principle of name induction,

$(\mathcal{L}_{\mathsf{EM}}\text{-}\mathsf{I}_{\Re})$ $\qquad (\forall x.\mathcal{C}(\chi,x) \to \chi(x)) \to \forall x.\Re(x) \to \chi(x).$

This can be derived from the leastness principle for $\mathcal{P}_{\boldsymbol{\varphi}}$

$$(\forall z.\boldsymbol{\varphi}(\psi, z) \to \psi(z)) \to \forall z.\mathcal{P}_{\boldsymbol{\varphi}}(z) \to \psi(z)$$

by choosing a formula $\psi(z)$ so that

$$\begin{aligned} \psi(\langle 0, x\rangle) &\leftrightarrow \chi^\star(x), \\ \psi(\langle 1, \langle x, y\rangle\rangle) &\leftrightarrow \mathsf{In}(x, y), \\ \psi(\langle 2, \langle x, y\rangle\rangle) &\leftrightarrow \overline{\mathsf{In}}(x, y), \\ \psi(z) &\leftrightarrow 0 = 0 \qquad \text{for every other argument } z. \end{aligned}$$

Starting from the premise $[\forall x.\mathcal{C}(\chi, x) \to \chi(x)]^\star$ we obtain $(\forall z.\boldsymbol{\varphi}(\psi, z) \to \psi(z))$: assume $\boldsymbol{\varphi}(\psi, z)$ holds with $z = \langle 0, x\rangle$ for some x. Then we get $\mathcal{C}_{\mathsf{Typ}}(\psi, x)$ which implies $[\mathcal{C}(\chi, x)]^\star$. So $\chi^\star(x)$ follows by our premise and $\psi(\langle 0, x\rangle)$ holds by the definition of ψ. If $\boldsymbol{\varphi}(\psi, z)$ holds and there is no x with $z = \langle 0, x\rangle$, then $\psi(z)$ is trivially fulfilled. Hence we conclude by the leastness condition for $\mathcal{P}_{\boldsymbol{\varphi}}$ that $\forall z.\mathcal{P}_{\boldsymbol{\varphi}}(z) \to \psi(z)$ holds. Let z be $\langle 0, x\rangle$, then we have $\mathcal{P}_{\boldsymbol{\varphi}}(\langle 0, x\rangle) \to \psi(\langle 0, x\rangle)$ which reads as $\mathsf{Typ}(x) \to \chi^\star(x)$. Because $\Re(x)$ is interpreted as $\mathsf{Typ}(x)$ we are finished.

This theorem, together with Theorem 2 and the well-known proof-theoretic equivalence of $\mathsf{ID}_1^{\mathsf{acc}}$ and ID_1, yields the final result:

Theorem 4. *The theory* NEM *of explicit mathematics with name induction is proof-theoretically equivalent to* ID_1*, and its proof-theoretic ordinal is the Bachmann-Howard ordinal.*

References

[AC96] Martín Abadi and Luca Cardelli. *A Theory of Objects*. Springer, 1996.

[Bee85] Michael Beeson. *Foundations of Constructive Mathematics*. Ergebnisse der Mathematik und ihrer Grenzgebiete; 3.Folge, Bd. 6. Springer, Berlin, 1985.

[BFPS81] Wilfried Buchholz, Solomon Feferman, Wolfram Pohlers, and Wilfried Sieg. *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical studies*, volume 897 of *Lecture Notes in Mathematics*. Springer-Verlag, 1981.

[Can96] Andrea Cantini. *Logical Frameworks for Truth and Abstraction*, volume 135 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1996.

[Fef70] Solomon Feferman. Formal theories for transfinite iterations of generalized inductive definitions and some subsystems of analysis. In A. Kino, J. Myhill, and R. Vesley, editors, *Intuitionismus and Proof Theory*, pages 303–326. North Holland, Amsterdam, 1970.

[Fef75] Solomon Feferman. A language and axioms for explicit mathematics. In J. Crossley, editor, *Algebra and Logic*, volume 450 of *Lecture Notes in Mathematics*, pages 87–139. Springer, 1975.

[Fef79] Solomon Feferman. Constructive theories of functions and classes. In M. Boffa, D. van Dalen, and K. McAloon, editors, *Logic Colloquium 78*, pages 159–224. North–Holland, Amsterdam, 1979.

[Fef88] Solomon Feferman. Hilbert's program relativized: Proof-theoretical and foundational reductions. *Journal of Symbolic Logic*, 53(2):364–384, 1988.

[Fef90] Solomon Feferman. Polymorphic typed lambda-calculus in a type-free axiomatic framework. In W. Sieg, editor, *Logic and Computation*, volume 106 of *Contemporary Mathematics*, pages 101–136. American Mathematical Society, 1990.

[Fef91] Solomon Feferman. Logics for termination and correctness of functional programs. In Y. Moschovakis, editor, *Logic from Computer Sciences*, pages 95–127. Springer, 1991.

[Fef92] Solomon Feferman. Logics for termination and correctness of functional programs II: Logics of strength PRA. In P. Aczel, H. Simmons, and S. S. Wainer, editors, *Proof Theory*, pages 195–225. Cambridge University Press, 1992.

[Fef0x] Solomon Feferman. Does reductive proof theory have a viable rationale? *Erkenntnis*, 200x. To appear.

[FJ93] Solomon Feferman and Gerhard Jäger. Systems of explicit mathematics with non-constructive μ-operator. Part I. *Annals of Pure and Applied Logic*, 65(3):243–263, 1993.

[FJ96] Solomon Feferman and Gerhard Jäger. Systems of explicit mathematics with non-constructive μ-operator. Part II. *Annals of Pure and Applied Logic*, 79:37–52, 1996.

[GR94] Ed Griffor and Michael Rathjen. The strength of some Martin-Löf type theories. *Archive for Mathematical Logic*, 33:347–385, 1994.

[Jäg88] Gerhard Jäger. Induction in the elementary theory of types and names. In E. Börger, H. Kleine Büning, and M.M. Richter, editors, *Computer Science Logic '87*, volume 329 of *Lecture Notes in Computer Science*, pages 118–128. Springer, 1988.

[JKS99] Gerhard Jäger, Reinhard Kahle, and Thomas Strahm. On applicative theories. In A. Cantini, E. Casari, and P. Minari, editors, *Logic and Foundation of Mathematics*, pages 83–92. Kluwer, 1999.

[JKS0x] Gerhard Jäger, Reinhard Kahle, and Thomas Studer. Universes in explicit mathematics. 200x. Submitted.

[Kre63] Georg Kreisel. Generalized inductive definitions. Technical report, Stanford Report, 1963.

[Mar94] Markus Marzetta. *Predicative Theories of Types and Names*. Dissertation, Universität Bern, Institut für Informatik und angewandte Mathematik, 1994.

[MS98] Markus Marzetta and Thomas Strahm. The μ quantification operator in explicit mathematics with universes and iterated fixed point theories with ordinals. *Archive for Mathematical Logic*, 37:391–413, 1998.

[Pal98] Erik Palmgren. On universes in type theory. In G. Sambin and J. Smith, editors, *Twenty Five Years of Constructive Type Theory*, pages 191–204. Oxford University Press, 1998.

[Poh89] Wolfram Pohlers. *Proof Theory*, volume 1407 of *Lecture Notes in Mathematics*. Springer, 1989.

[Stä97] Robert Stärk. Call-by-value, call-by-name and the logic of values. In D. van Dalen and M. Bezem, editors, *Computer Science Logic CSL '96: Selected Papers*, volume 1258 of *Lecture Notes in Computer Science*, pages 431–445. Springer, 1997.

[Stä98] Robert Stärk. Why the constant 'undefined'? Logics of partial terms for strict and non-strict functional programming languages. *Journal of Functional Programming*, 8(2):97–129, 1998.

[Stu0x] Thomas Studer. A semantics for $\lambda^{\{\}}_{str}$: a calculus with overloading and late-binding. *Journal of Logic and Computation*, 200x. To appear.

[TvD88] Anne Troelstra and Dirk van Dalen. *Constructivism in Mathematics*, volume II. North Holland, Amsterdam, 1988.