The unfolding of non-finitist arithmetic

Solomon Feferman Thomas Strahm

Abstract

The *unfolding of schematic formal systems* is a novel concept which was initiated in Feferman [6]. This paper is mainly concerned with the proof-theoretic analysis of various unfolding systems for non-finitist arithmetic NFA. In particular, we examine two restricted unfoldings $\mathcal{U}_0(\mathsf{NFA})$ and $\mathcal{U}_1(\mathsf{NFA})$, as well as a full unfolding, $\mathcal{U}(\mathsf{NFA})$. The principal results then state: (i) $\mathcal{U}_0(\mathsf{NFA})$ is equivalent to PA; (ii) $\mathcal{U}_1(\mathsf{NFA})$ is equivalent to $\mathsf{RA}_{<\omega}$; (iii) $\mathcal{U}(\mathsf{NFA})$ is equivalent to $\mathsf{RA}_{<\Gamma_0}$. Thus $\mathcal{U}(\mathsf{NFA})$ is proof-theoretically equivalent to predicative analysis.

1 Introduction

The concept of unfolding of a schematic formal systems S was introduced in Feferman [6] in order to answer the following question:

> *Given a schematic system* S*, which operations and predicates, and which principles concerning them, ought to be accepted if one has accepted* S*?*

The basic system NFA of classical non-finitist arithmetic is paradigmatic for such S; it is given by the following axioms, where as usual we write x' for $Sc(x)$:

(1) $x' \neq 0$

(2) $Pd(x') = x$

(3) $P(0) \wedge (\forall x)[P(x) \rightarrow P(x')] \rightarrow (\forall x)P(x)$.

Here P is a *free predicate variable*, and the intention is to use the induction scheme (3) in a wider sense than is usual. Denote by $\mathcal{L}_0$ the language of NFA. The usual *narrow* sense in which (3) is to be applied is that we may substitute for P any formula B of $\mathcal{L}_0$ with distinguished free variable x. A *wider* sense is that we may substitute for P in (3) any formula B of any language $\mathcal{L}$ extending $\mathcal{L}_0$ whose basic notions one accepts as meaningful and for which concomitant axioms are provided. But this is still just a special case of the *general rule of substitution*

$$\text{(Subst)} \qquad A[P] \;\Rightarrow\; A[B/P]$$

for any formulas A, B of $\mathcal{L}$. In particular, as one unfolds NFA, its language and corresponding axioms expand and the class of formulas A, B to which one may apply (Subst) expands accordingly. This does not mean that the notions to which one may apply induction throughout mathematics are limited to those appearing in the unfolding of NFA. Rather, that only tells us which notions and principles are *implicit* in accepting NFA, i.e., which *ought* to be accepted if one accepts NFA at all. The acceptance of notions and principles beyond that, such as those coming from set theory or other conceptual arenas, must in each case be based on essentially new considerations.

Speaking of set theory, other formal systems which have natural schematic formalizations in the present sense are those of Zermelo and Zermelo-Fraenkel. For the former, the separation scheme is given by

$$\text{(Sep)} \qquad (\exists b)(\forall x)[x \in b \leftrightarrow x \in a \wedge P(x)].$$

Similarly, we may reformulate replacement in ZF in schematic terms using a free binary predicate variable R. Alternatively (and not necessarily equivalently) we may formulate it in a natural way using a *free partial function variable* f, as follows:

$$\text{(Rep)} \qquad (\forall x \in a)f(x)\downarrow \to (\exists b)(\forall y)[y \in b \leftrightarrow (\exists x \in a)f(x) = y],$$

which simply says that the range of f on a exists. For further discussion and examples of schematic systems using free predicate and/or partial function variables, in the wider sense indicated here, see [6].

The concept of unfolding applies to any schematic systems S. Its definition is given first in a restricted form $\mathcal{U}_0(\mathsf{S})$ and then in a full form $\mathcal{U}(\mathsf{S})$; consideration of the latter leads to a natural intermediate form $\mathcal{U}_1(\mathsf{S})$. The system $\mathcal{U}_0(\mathsf{S})$ is called the *operational unfolding* of S; it tells us *which operations from and to individuals, and which principles concerning them, ought to be accepted if one has accepted* S. It is obtained by adding free partial function variables and partial function and functional constants which are introduced successively from the basic operations of S by schemata of *explicit definition* (*ED*) and *least fixed point recursion* (*LFP*). The principles that are added concerning these are simply the equations they are intended to satisfy.

The full unfolding $\mathcal{U}(\mathsf{S})$ is also given in operational terms. It tells us, further, *which operations on and to predicates, and which principles concerning them, ought to be accepted if one has accepted* S. This will depend to begin with on which *logical operations*, viewed as operations on predicates, are accepted as basic in S. In the case of (classical) NFA these may be taken as the operations of negation, conjunction and universal quantification. If the initial system S is, for example, some form of finitist arithmetic or constructive non-finitist arithmetic, other choices of basic operations on predicates will be dictated. Once supplied with a choice of such basic logical operations, the further operations (and associated principles) on and to predicates in $\mathcal{U}(\mathsf{S})$ are generated again by (*ED*) and (*LFP*) schemata. In addition, we have an operation of *Join*, specific to the case of predicates, which allows us to pass in a canonical way from a sequence of n-ary predicates to an $(n+1)$-ary predicate. The intermediate unfolding $\mathcal{U}_1(\mathsf{S})$ is obtained in the same way as $\mathcal{U}(\mathsf{S})$, except that *Join* is not applied here.

The main results of this paper characterize proof-theoretically these three unfolding systems of NFA, as follows, where $\equiv$ denotes proof-theoretical equivalence (and where in each case we have conservation with respect to suitable classes of formulas of the system on the left over the system on the right).

I. $\mathcal{U}_0(\mathsf{NFA}) \equiv \mathsf{PA}$

II. $\mathcal{U}_1(\mathsf{NFA}) \equiv \mathsf{RA}_{<\omega}$

III. $\mathcal{U}(\mathsf{NFA}) \equiv \mathsf{RA}_{<\Gamma_0}$.

Here, as usual, $\mathsf{RA}_{<\alpha}$ denotes the system of ramified analysis in levels $< \alpha$, and Γ_0 is the so-called Feferman-Schütte ordinal, which has been identified as the limit of the predicatively provable ordinals. Thus $\mathcal{U}(\mathsf{NFA})$ is equivalent to predicative analysis.

For related earlier work and the historical background to the concept of unfolding, as well as for some directions of possible further work, see [6].

2 The operational unfolding of a schematically presented formal system S

It is the purpose of this section to define the operational unfolding $\mathcal{U}_0(\mathsf{S})$ of a schematically presented formal system S. The precise definition of the terms and axioms for explicit definition (*ED*) and least fixed point recursion (*LFP*) to be used in $\mathcal{U}_0(\mathsf{S})$ is given by formalization of the generalization of recursion theory (g.r.t.) to arbitrary structures $\mathcal{A}$ due to Feferman [4, 5]. An alternative but essentially equivalent form of g.r.t. is due to Moschovakis [11], and both have their roots in Platek's thesis [12]. For our purposes here, the structures to which this is to be applied are of the form

$$\mathcal{A} = (\mathbb{A}, F_0, \ldots, F_n),$$

where $\mathbb{A}$ is the domain of $\mathcal{A}$ and $F_0, \ldots, F_n$ are objects of type level ≤ 2 over $\mathbb{A}$, i.e. where each of these is either an individual of $\mathbb{A}$ or a partial function or partial monotonic functional of type level 2 of appropriate arity; see below for details concerning the type structure. For simplicity, we are only considering one-sorted structures $\mathcal{A}$ here, however, the treatment of many-sorted structures is straightforward. The basic structure to consider in the case of arithmetic is of course $(\mathbb{N}, Sc, Pd, 0)$, where $\mathbb{N}$ is the set of natural numbers and Sc and Pd denote the successor and predecessor operation, respectively.

The terms considered below are either individual terms, partial function terms or partial functional terms. Accordingly, their types can be divided into three classes.

Typ 1. ι denotes the type of individuals from $\mathbb{A}$, and we let $\bar{\iota}$ range over the types of finite (possibly empty) sequences of individuals.

Typ 2. τ_0, σ_0 range over the types of partial functions of the form $\bar{\iota} \tilde{\to} \iota$, and again we let $\bar{\tau}_0$ range over the types of finite sequences of such.

Typ 3. $(\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$ is used for partial functional types. These reduce to partial function types in case that $\bar{\tau}_0$ is empty.

The terms $(r, s, t, \ldots)$ are now inductively generated as follows, where we use the notation $r : \rho$ in order to indicate that the term r is of type ρ.

Tm 1. We have infinitely many variables $x, y, z, a, b, c, \ldots$ of type ι.

Tm 2. For each partial function type τ_0 we have infinitely many partial function variables $f, g, h, \ldots$ of type τ_0.

Tm 3. For each basic functional of the structure $\mathcal{A}$ we are given functional constants F_i of appropriate functional type.

Tm 4. $Cond(r, s) : (\bar{\tau}_0, \bar{\iota}, \iota, \iota \tilde{\to} \iota)$ for $r, s : (\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$.

Tm 5. $r(\bar{s}, \bar{t}) : \iota$ for $r : (\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$, $\bar{s} : \bar{\tau}_0$ and $\bar{t} : \bar{\iota}$.

Tm 6. $(\lambda \bar{f}, \bar{x}.t) : (\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$ for $\bar{f} : \bar{\tau}_0$, $\bar{x} : \bar{\iota}$ and $t : \iota$.

Tm 7. $LFP(\lambda f, \bar{x}.t) : (\bar{\iota} \tilde{\to} \iota)$ for $f : \bar{\iota} \tilde{\to} \iota$, $\bar{x} : \bar{\iota}$ and $t : \iota$.

The formulas $(A, B, C, \ldots)$ of $\mathcal{U}_0(\mathsf{S})$ are inductively given by:

Fm 1. The atomic formulas are $(r = s)$, $r{\downarrow}$ and $P(\bar{t})$ for $r, s : \iota$, $\bar{t} : \bar{\iota}$ and P a free relation symbol.

Fm 2. If A and B are formulas, then so also are $\neg A$, $(A \wedge B)$ and $(\forall x)A$.

Observe that we do not allow quantification over partial function variables. As we base all our systems on classical logic, we assume that the remaining logical connectives and quantifiers are defined as usual. However, let us mention that the results established in this article also hold for the corresponding systems based on intuitionistic logic.

In the following we write $t[\bar{f}, \bar{x}]$ to indicate a sequence $\bar{f}, \bar{x}$ of free variables possibly appearing in the term t; however, t may contain other free variables than those shown by using this bracket notation. The meaning of $A[\bar{f}, \bar{x}]$ is analogous.

The logic of $\mathcal{U}_0(\mathsf{S})$ is the classical *logic of partial terms LPT* of Beeson [1] for the individual sort ι, and usual (quantifier free) predicate logic for the other sorts. We recall that LPT embodies strictness axioms saying that all subterms of a defined compound term are defined as well. Moreover, if $(s = t)$ holds then both s and t

are defined, and s is defined provided $P(s)$ holds. As usual, one defines a partial equality relation between individual type terms by setting

$$s \simeq t \ := \ s\downarrow \vee t\downarrow \ \rightarrow \ s = t.$$

We are now ready to spell out the axioms of $\mathcal{U}_0(\mathsf{S})$, which essentially just bring out the obvious meaning of the terms specified above.

Ax 1. The axioms of S, including the defining axioms for the F_i's.

Ax 2. $\mathit{Cond}(s,t)(\bar{f},\bar{x},y,y) \simeq s(\bar{f},\bar{x}) \ \wedge \ y \neq z \rightarrow \mathit{Cond}(s,t)(\bar{f},\bar{x},y,z) \simeq t(\bar{f},\bar{x})$.

Ax 3. $(\lambda\bar{f},\bar{x}.t[\bar{f},\bar{x}])(\bar{f},\bar{x}) \simeq t[\bar{f},\bar{x}]$.

Ax 4. For $s \equiv \mathit{LFP}(\lambda f,\bar{x}.t[f,\bar{x}])$ we take:

(i) $s(\bar{x}) \simeq t[s,\bar{x}]$,

(ii) $(\forall\bar{x})(f(\bar{x}) \simeq t[f,\bar{x}]) \ \rightarrow \ (\forall\bar{x})(s(\bar{x})\downarrow \rightarrow f(\bar{x}) = s(\bar{x}))$.

Finally, crucial for the formulation of $\mathcal{U}_0(\mathsf{S})$ is the predicate substitution rule, which reads as follows:

(Subst) $$A[P] \ \Rightarrow \ A[B/P].$$

In the conclusion of this rule, B is an arbitrary formula and $A[B/P]$ denotes the formula $A[P]$ with each subformula $P(\bar{t})$ replaced by $(\exists\bar{x})(\bar{t} \simeq \bar{x} \wedge B(\bar{x}))$. This completes the description of the system $\mathcal{U}_0(\mathsf{S})$.

Before we turn to the definition of the intermediate and full unfolding of S, we very briefly address the proof-theoretic strength of $\mathcal{U}_0(\mathsf{NFA})$. Recall that NFA was presented by axioms (1)-(3) where (1) and (2) are the axioms for Sc and Pd, while (3) is the schematic axiom of induction; these are what take the place of Ax 1 above in the case of $\mathsf{S} = \mathsf{NFA}$.

3 The proof-theoretic strength of $\mathcal{U}_0(\mathsf{NFA})$

In this section we give a short sketch for the proof-theoretic equivalence of $\mathcal{U}_0(\mathsf{NFA})$ and Peano arithmetic PA.[1] Hence, in the sequel ι denotes the basic type of natural numbers.

As to the lower bound, we first have to bootstrap $\mathcal{U}_0(\mathsf{NFA})$ and show that the primitive recursive functions can be introduced there. This is of course straightforward by making use of LFP recursion. For notational convenience, let us write

$$\{\text{if } y = 0 \text{ then } s[\bar{f},\bar{x}] \text{ else } t[\bar{f},\bar{x}]\} \quad \text{for} \quad \mathit{Cond}(\lambda\bar{f},\bar{x}.s,\lambda\bar{f},\bar{x}.t)(\bar{f},\bar{x},y,0).$$

[1] To be precise, we use a conservative extension of PA in which the predicate symbol P may occur in formulas in the induction schema.

To show closure under primitive recursion in $\mathcal{U}_0(\mathsf{NFA})$, let r and s be terms of function type taking number type arguments $(\bar{x})$ and $(\bar{x}, y, z)$, respectively, and assume that r and s have been shown to be total in $\mathcal{U}_0(\mathsf{NFA})$. Let t be given by

$$t \;:=\; LFP(\lambda f, \bar{x}, y.\{\text{if } y = 0 \text{ then } r(\bar{x}) \text{ else } s(\bar{x}, Pd(y), f(\bar{x}, Pd(y)))\}).$$

Then one obtains by axiom Ax 4(i) of $\mathcal{U}_0(\mathsf{NFA})$ that

$$t(\bar{x}, 0) \simeq r(\bar{x}) \quad \text{and} \quad t(\bar{x}, y') \simeq s(\bar{x}, y, t(\bar{x}, y)).$$

One then establishes by induction on y that $t(\bar{x}, y)\downarrow$. Here one has to apply the substitution rule to the schematic form of the induction axiom of NFA. Hence, we can define all primitive recursive functions in $\mathcal{U}_0(\mathsf{NFA})$; of course, by (Subst), we have complete induction on the natural numbers available in $\mathcal{U}_0(\mathsf{NFA})$ for arbitrary formulas and, therefore, we see that Peano arithmetic PA in its usual non-schematic form is contained in $\mathcal{U}_0(\mathsf{NFA})$.

For the upper bound we make use of a direct embedding of $\mathcal{U}_0(\mathsf{NFA})$ into PA. Basically, we let partial function variables of type $\bar{\iota} \tilde{\to} \iota$ range over indices of partial recursive functions. It is then easy to find (indices of) partial recursive function(al)s serving as appropriate interpretations for the function(al) terms of $\mathcal{U}_0(\mathsf{NFA})$. In particular, closure under the LFP schema is guaranteed in the usual way as follows. Suppose $\lambda f, \bar{x}.t[f, \vec{x}]$ is already given to us as a partial recursive functional, possibly depending on additional parameters. Then we define (uniformly in n) a sequence g_n of partial recursive functions by

$$\begin{aligned} g_0 &:= \text{ the empty function}, \\ g_{n+1} &:= \lambda \bar{x}.t[g_n, \bar{x}]. \end{aligned}$$

Now we can take $g := \bigcup_{n \in \mathbb{N}} g_n$ as our interpretation of $LFP(\lambda f, \bar{x}.t[f, \bar{x}])$. We have that g is partial recursive, and the usual argument shows that it has the required properties, cf. Feferman [4, 5] for more details. Finally, the substitution rule (Subst) is easily seen to be validated under our embedding and, hence, we are in a position to state the following theorem:

Theorem 1 $\mathcal{U}_0(\mathsf{NFA})$ *is proof-theoretically equivalent to* PA *and conservatively extends* PA.

This concludes our brief sketch of the equivalence of $\mathcal{U}_0(\mathsf{NFA})$ and PA.

4 The full and intermediate unfolding of a schematically presented formal system S

It is the aim of the present section to define the full unfolding $\mathcal{U}(\mathsf{S})$ of a schematically presented formal system S and its natural restriction $\mathcal{U}_1(\mathsf{S})$. Whereas $\mathcal{U}_0(\mathsf{S})$ addresses the question of which operations on $\mathbb{A}$ ought to be accepted given a schematic system S for a structure $\mathcal{A} = (\mathbb{A}, F_0, \ldots, F_n)$, the central question concerning $\mathcal{U}_1(\mathsf{S})$ and $\mathcal{U}(\mathsf{S})$ can be stated as follows: which operations on and to predicates — and which principles concerning them — ought to be accepted if one has accepted S?

Let us now stepwise describe $\mathcal{U}(\mathsf{S})$ and its subsystem $\mathcal{U}_1(\mathsf{S})$. For that purpose, we introduce for each natural number n a new atomic type π_n of n-ary predicates on $\mathbb{A}$. Hence, our basic types now include ι and π_n for each $n \in \mathbb{N}$. The type structure to be considered for the full unfolding is spelled out below. Crucial is the presence of new function types of the form $\bar{\iota} \tilde{\rightarrow} \pi_n$, which will allow the definition of operations from individuals to predicates by means of least fixed point recursion. Definitions of this kind are very natural as we will see below; for example, in the unfolding of NFA one may define initial segments of the hyperarithmetic hierarchy simply by *LFP* recursion from numbers to predicates.

Typ 1. The basic types are ι and π_n for each $n \in \mathbb{N}$. We let κ, ν range over these basic types, and $\bar{\kappa}$ over types of finite sequences of objects of basic type.

Typ 2. We have partial function types of the two forms $\bar{\iota} \tilde{\rightarrow} \iota$ and $\bar{\iota} \tilde{\rightarrow} \pi_n$. As above, we let τ_0, σ_0 range over the former and, in addition, τ, σ range over partial function types of both forms. Sequence notation is used as before.

Typ 3. $(\bar{\tau}_0, \bar{\iota} \tilde{\rightarrow} \iota)$ and $(\bar{\tau}, \bar{\kappa} \tilde{\rightarrow} \pi_n)$ are our partial functional types.

Note that in Typ 2 and Typ 3 we do not allow the use of predicate arguments or arguments of type $\bar{\iota} \tilde{\rightarrow} \pi_n$ when defining function(al)s whose values are of type ι, i.e. are individuals. These *are* allowed in Typ 3 when defining functionals of type $(\bar{\tau}, \bar{\kappa} \tilde{\rightarrow} \pi_n)$; in particular, when $\bar{\tau}$ is empty, such functionals reduce to partial functions from individuals and predicates to predicates. The reason for the restriction of functional types with values of type ι is that in some sense individuals and operations on individuals are being treated as conceptually prior to operations on predicates; in particular, we want schemas to generate operations from individuals to individuals which do not depend on what predicates are available. However, we will see that we can prove more functions from individuals to individuals to be total in the presence of suitable predicates.[2]

The term building operations Tm 1-7 are now extended to the larger type structure in a straightforward manner as follows.

Tm 1. For each basic type κ we have infinitely many variables of type κ. In the following we usually reserve $x, y, z, a, b, c, \ldots$ as variables of type ι and $X^n, Y^n, Z^n, \ldots$ as variables of type π_n; we omit the superscript 'n' if it is given by the context.

[2]We have considered the alternative possibility in the definition of $\mathcal{U}(\mathsf{S})$ and $\mathcal{U}_1(\mathsf{S})$ of treating individuals and predicates on a par, without the restrictions taken here in Typ 1 – Typ 3. This would make sense from the computational point of view of [4], if not the logical point of view. It is an open question whether the upper bounds obtained in Sections 5 and 6 for $\mathcal{U}(\mathsf{NFA})$, resp. $\mathcal{U}_1(\mathsf{NFA})$, still hold in this unrestricted formulation.

Tm 2. For each partial function type τ we have infinitely many partial function variables $f, g, h, \ldots$ of type τ.

Tm 3. For each basic functional of the structure $\mathcal{A}$ we are given functional constants F_i of appropriate functional type.

Tm 4. $Cond(r,s) : (\bar{\tau}_0, \bar{\iota}, \iota, \iota \tilde{\to} \iota)$ for $r, s : (\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$;

$Cond(r,s) : (\bar{\tau}, \bar{\kappa}, \iota, \iota \tilde{\to} \pi_n)$ for $r, s : (\bar{\tau}, \bar{\kappa} \tilde{\to} \pi_n)$.

Tm 5. $r(\bar{s}, \bar{t}) : \iota$ for $r : (\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$, $\bar{s} : \bar{\tau}_0$ and $\bar{t} : \bar{\iota}$;

$r(\bar{s}, \bar{t}) : \pi_n$ for $r : (\bar{\tau}, \bar{\kappa} \tilde{\to} \pi_n)$, $\bar{s} : \bar{\tau}$ and $\bar{t} : \bar{\kappa}$.

Tm 6. $(\lambda \bar{f}, \bar{x}.t) : (\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$ for $\bar{f} : \bar{\tau}_0$, $\bar{x} : \bar{\iota}$ and $t : \iota$;

$(\lambda \bar{f}, \bar{x}.t) : (\bar{\tau}, \bar{\kappa} \tilde{\to} \pi_n)$ for $\bar{f} : \bar{\tau}$, $\bar{x} : \bar{\kappa}$ and $t : \pi_n$.

Tm 7. $LFP(\lambda f, \bar{x}.t) : (\bar{\iota} \tilde{\to} \nu)$ for $f : \bar{\iota} \tilde{\to} \nu$, $\bar{x} : \bar{\iota}$ and $t : \nu$.

The term forming operations Tm 1-7 are now extended in order to incorporate new operations on and to predicates, namely *Eq* (equality), Pr_P (free predicate symbol P), *Inv* (inverse image), *Neg* (negation), *Conj* (conjunction), *Un* (universal quantification) and *Join* (disjoint union). The operations *Neg*, *Conj* and *Un* are appropriate if one accepts classical predicate calculus, as, for example, one does in non-finitist arithmetic. Other systems (e.g. finitist or constructive arithmetic) may call for other choices of basic logical operations.

Tm 8. $Eq : \pi_2$.

Tm 9. $Pr_P : \pi_n$ for P n-ary predicate letter.

Tm 10. $Inv(s, t_1, \ldots, t_m) : \pi_n$ for $s : \pi_m$, $t_1, \ldots, t_m : \bar{\iota} \tilde{\to} \iota$ and $\bar{\iota}$ of length n.

Tm 11. $Neg(t) : \pi_n$ for $t : \pi_n$.

Tm 12. $Conj(s,t) : \pi_n$ for $s, t : \pi_n$.

Tm 13. $Un(t) : \pi_n$ for $t : \pi_{n+1}$.

Tm 14. $Join(t) : \pi_{n+1}$ for $t : \iota \tilde{\to} \pi_n$.

The formulas of $\mathcal{U}(\mathsf{S})$ are built in the same way as the formulas of $\mathcal{U}_0(\mathsf{S})$, but now taking into account new atomic formulas involving predicate type terms.

Fm 1. We have the atomic formulas:

(i) $(r = s)$, $r\!\downarrow$ and $P(\bar{t})$ for $r, s : \iota$, $\bar{t} : \bar{\iota}$ and P a free relation symbol;

(ii) $(r = s)$, $r\!\downarrow$ and $(t_1, \ldots, t_n) \in r$ for $r, s : \pi_n$ and $t_1, \ldots, t_n : \iota$.

Fm 2. If A and B are formulas, then so also are $\neg A$, $(A \wedge B)$ and $(\forall x)A$.

It is crucial to observe here that we do *not* allow quantification over predicate variables. Accordingly, the logic used for the sort π_n is a quantifier-free version of the logic of partial terms LPT.

The axioms of $\mathcal{U}(\mathsf{S})$ include Ax 1-4 of $\mathcal{U}_0(\mathsf{S})$, now extended to the new language. First Ax 5, we do not distinguish between predicates which are extensionally identical; i.e. it is only how predicates behave on their arguments that counts. In addition, the axioms for the new predicate forming operations are spelled out in the expected manner in Ax 6-12.

Ax 5. $(\forall \bar{x})[(\bar{x}) \in X \leftrightarrow (\bar{x}) \in Y] \rightarrow X = Y$.

Ax 6. $Eq\downarrow \wedge (\forall x, y)[(x, y) \in Eq \leftrightarrow x = y]$.

Ax 7. $Pr_P\downarrow \wedge (\forall \bar{x})[(\bar{x}) \in Pr_P \leftrightarrow P(\bar{x})]$.

Ax 8. $Inv(X, f_1, .., f_m)\downarrow \wedge (\forall \bar{x})[(\bar{x}) \in Inv(X, f_1, .., f_m) \leftrightarrow (f_1(\bar{x}), .., f_m(\bar{x})) \in X]$.

Ax 9. $Neg(X)\downarrow \wedge (\forall \bar{x})[(\bar{x}) \in Neg(X) \leftrightarrow (\bar{x}) \notin X]$.

Ax 10. $Conj(X, Y)\downarrow \wedge (\forall \bar{x})[(\bar{x}) \in Conj(X, Y) \leftrightarrow (\bar{x}) \in X \wedge (\bar{x}) \in Y]$.

Ax 11. $Un(X)\downarrow \wedge (\forall \bar{x})[(\bar{x}) \in Un(X) \leftrightarrow (\forall y)((\bar{x}, y) \in X)]$.

Ax 12. For $f : \iota \tilde{\rightarrow} \pi_n$ we take:

$$(\forall y)f(y)\downarrow \rightarrow Join(f)\downarrow \wedge (\forall \bar{x}, y)[(\bar{x}, y) \in Join(f) \leftrightarrow (\bar{x}) \in f(y)].$$

Further, $\mathcal{U}(\mathsf{S})$ contains the substitution rule (Subst), i.e. $A[P] \Rightarrow A[B/P]$, where now B denotes an arbitrary formula of $\mathcal{U}(\mathsf{S})$, but $A[P]$ is required to be a formula in the language of $\mathcal{U}_0(\mathsf{S})$. This last restriction is due to the fact that predicate terms in general depend on the predicate parameter P. Finally, we obtain an intermediate unfolding system $\mathcal{U}_1(\mathsf{S})$ by leaving out Ax 12, i.e. $\mathcal{U}_1(\mathsf{S})$ is just $\mathcal{U}(\mathsf{S})$ without $Join$.

The full unfolding $\mathcal{U}(\mathsf{S})$ described here is slightly different from the one given in Feferman [6]. There predicates are understood as (total) propositional functions, taking individuals as arguments and yielding propositions as values. For the latter, a type π of propositions is presupposed together with a predicate $T(x)$ expressing that the proposition x is true, thus providing an approach to predicates via Frege structures. As pointed out in [6], the Frege-type approach to predicates does not allow one to single out the role of join, and since we are interested in the intermediate unfolding $\mathcal{U}_1(\mathsf{S})$ here as well, we have assumed types for n-ary predicates as basic.

The rest of this paper is devoted to establishing the proof-theoretic equivalence of $\mathcal{U}(\mathsf{NFA})$ with $\mathsf{RA}_{<\Gamma_0}$ as well as $\mathcal{U}_1(\mathsf{NFA})$ with $\mathsf{RA}_{<\omega}$.

5 The proof-theoretic strength of $\mathcal{U}(\mathsf{NFA})$

In this section we compute the exact proof-theoretic strength of $\mathcal{U}(\mathsf{S})$ for the case that S is the basic schematic system of non-finitist arithmetic NFA. As before we let ι denote the basic type of natural numbers and, in addition, we write Pr instead of Pr_P, since P is the only free predicate letter needed.

In Section 5.1 we show that transfinite induction is derivable in $\mathcal{U}(\mathsf{NFA})$ below the Feferman-Schütte ordinal Γ_0 by carrying through a detailed wellordering proof; this is used to reduce $\mathsf{RA}_{<\Gamma_0}$ to $\mathcal{U}(\mathsf{NFA})$. Then in 5.2 we give the definition of a suitable version of Peano arithmetic with ordinals plus substitution rule, $\mathsf{PA}^+_\Omega + (\mathsf{Subst})$, which was introduced in Strahm [16]; it is shown op. cit. that $\mathsf{PA}^+_\Omega + (\mathsf{Subst})$ has proof-theoretic ordinal Γ_0. Finally, we show in Section 5.3 how to model $\mathcal{U}(\mathsf{NFA})$ in $\mathsf{PA}^+_\Omega + (\mathsf{Subst})$, thereby establishing $\mathsf{RA}_{<\Gamma_0}$ as an upper bound of $\mathcal{U}(\mathsf{NFA})$.

5.1 Lower bounds

The aim of this section is to establish the Feferman-Schütte ordinal Γ_0 as a lower bound of $\mathcal{U}(\mathsf{NFA})$. For that purpose we presuppose a primitive recursive standard wellordering $\prec$ of order type Γ_0 with least element 0 and field $\mathbb{N}$. We further assume familiarity with the Veblen functions $\varphi\alpha$ (or φ_α); cf. Schütte [15] or Pohlers [13] for details. When working in formal theories, we identify ordinal operations with their primitive recursive analogues acting on codes of our notation system; we also identify primitive recursive relations with the $\mathcal{U}_0(\mathsf{NFA})$ terms which represent their characteristic functions. If A is a formula with designated free variable x of type ι and $s : \iota$, then we define as usual:

$$\begin{aligned} \mathit{Prog}(A) &:= (\forall x)[(\forall y \prec x)A(y) \to A(x)], \\ \mathit{TI}(s, A) &:= \mathit{Prog}(A) \to (\forall x \prec s)A(x). \end{aligned}$$

If we want to stress the relevant induction variable of the formula A, we sometimes write $\mathit{Prog}(\lambda x.A(x))$ instead of $\mathit{Prog}(A)$; the expression $\mathit{TI}(s, \lambda x.A(x))$ reads similarly. Moreover, if $t : \pi_1$, then we often use the notation $\mathit{Prog}(t)$ instead of $\mathit{Prog}(\lambda x.x \in t)$, and similarly for $\mathit{TI}(s, t)$.

Let $(\gamma_n)_{n\in\mathbb{N}}$ denote a canonical fundamental sequence for Γ_0 given by $\gamma_0 = \omega$ and $\gamma_{n+1} = \varphi\gamma_n 0$. It is our aim to establish $\mathit{TI}(\gamma_n, P)$ for each $n \in \mathbb{N}$ within $\mathcal{U}(\mathsf{NFA})$. Crucial in the argument below is the following: as soon as we have shown $\mathit{TI}(\omega^\alpha, P)$, and if a suitable jump hierarchy above P exists below $\omega^\alpha + 1$, then we are able to deduce $\mathit{TI}(\varphi\alpha 0, P)$. Hence, our first task is to show how arithmetic jump hierarchies can be built uniformly within $\mathcal{U}(\mathsf{NFA})$.

Let $A(X, a, y)$ be an arithmetic formula with at most X, a, y free and $X : \pi_2$. Then the A *jump hierarchy up to* $\alpha < \Gamma_0$ *starting with* P is given set-theoretically by the following transfinite recursion:

$$Y_0 \;:=\; \{m : P(m)\},$$

$$Y_a \;:=\; \{m : A(Y^a, a, m)\} \qquad (0 \prec a \prec \alpha),$$

where Y^a denotes the set $\{(m, b) : b \prec a \wedge m \in Y_b\}$. We now show how to represent $(Y_a)_{a \prec \alpha}$ in $\mathcal{U}(\mathsf{NFA})$ *provided* that we have already established $TI(\alpha, P)$. First of all, by making use of the predicate axioms Ax 6–11 only, we can find a term $r_A : (\pi_2, \iota \tilde{\to} \iota)$ so that $\mathcal{U}(\mathsf{NFA})$ proves

$$r_A(X, a){\downarrow} \;\wedge\; (\forall y)(y \in r_A(X, a) \leftrightarrow A(X, a, y)).$$

Further, there exists a term $s : (\iota \tilde{\to} \pi_1, \iota, \iota \tilde{\to} \pi_1)$ which provably satisfies

$$s(f, a, x) \simeq \{\text{if } x \prec a \text{ then } f(x) \text{ else } \emptyset\},$$

where $\emptyset$ denotes a canonical term of type π_1 for the empty set. Finally, let $hier_A$ of type $(\iota \tilde{\to} \pi_1)$ be given by least fixed point recursion as follows:

$$hier_A \;:=\; LFP(\lambda f, a.\{\text{if } a = 0 \text{ then } Pr \text{ else } r_A(Join(\lambda x.s(f, a, x)), a)\}).$$

Since we have assumed $TI(\alpha, P)$, we can derive by the substitution rule (Subst) that $TI(\alpha, \lambda a.hier_A(a){\downarrow})$. Since it is easily shown that $Prog(\lambda a.hier_A(a){\downarrow})$, we can conclude

$$(\forall a \prec \alpha) hier_A(a){\downarrow}.$$

Hence, we have established the existence of the A jump hierarchy below α starting with P in $\mathcal{U}(\mathsf{NFA})$, and its defining properties are directly provable there.

In order to carry through our wellordering proof we make use of a very specific jump formula $\mathcal{A}(X, a, y)$, which is given in Schütte [15], pp. 184ff. Before we can give an explicit definition of $\mathcal{A}$, we need some preparations. First, let h and e be primitive recursive auxiliary functions on our ordinal notations, which satisfy

- $h(0) = e(0) = 0$; $h(\omega^a) = 0$ and $e(\omega^a) = a$;
- if $a = \omega^{a_1} + \cdots + \omega^{a_n}$ for more than one summand so that $a_n \preceq \cdots \preceq a_1$, then $h(a) = \omega^{a_1} + \cdots + \omega^{a_{n-1}}$ and $e(a) = a_n$.

In addition, let us define a kind of jump operator $\mathcal{J}$, which is given by the following arithmetic definition:

$$\mathcal{J}(X, a) \;:=\; (\forall y)((\forall x \prec y)(x \in X) \to (\forall x \prec y + a)(x \in X)).$$

Finally, the jump formula $\mathcal{A}(X, a, y)$ is given by

$$\mathcal{A}(X, a, y) \;:=\; (\forall z)\,(h(a) \preceq z \prec a \;\to\; \mathcal{J}((X)_z, \varphi(e(a), y))).$$

Here X is of type π_2 and a formula $t \in (X)_z$ must be read as $(t, z) \in X$. By the considerations above, we have a term $hier_{\mathcal{A}} : \iota \tilde{\to} \pi_1$ which provably in $\mathcal{U}(\mathsf{NFA})$ represents the $\mathcal{A}$ jump hierarchy starting from P below α, provided we have previously established $TI(\alpha, P)$ in $\mathcal{U}(\mathsf{NFA})$.

The following statement is crucial in the wellordering proof for $\mathcal{U}(\mathsf{NFA})$. It is Lemma 9 in Schütte [15] p. 186, re-written in the present notation, and its proof, which we omit, follows exactly the same steps loc. cit.

Lemma 2 *Assume that $TI(\alpha, P)$ is provable in $\mathcal{U}(\mathsf{NFA})$ for an ordinal α less than Γ_0. Then we have that $\mathcal{U}(\mathsf{NFA})$ proves*

$$(\forall a)[0 \prec a \prec \alpha \wedge (\forall b \prec a) Prog(hier_{\mathcal{A}}(b)) \rightarrow Prog(hier_{\mathcal{A}}(a))].$$

The ground is now prepared for the main theorem of this section:

Theorem 3 *$\mathcal{U}(\mathsf{NFA})$ proves $TI(\gamma_n, P)$ for each natural number n.*

Proof. The claim is established by metamathematical induction on n. The case $n = 0$ is trivial, since $TI(\omega, P)$ is certainly provable in $\mathcal{U}(\mathsf{NFA})$. For the step from n to $n+1$ we assume that $TI(\gamma_n, P)$ is derivable in $\mathcal{U}(\mathsf{NFA})$. By making use of the usual wellordering proof of PA, this readily implies that $\mathcal{U}(\mathsf{NFA})$ also proves

$$TI(\omega^{\gamma_n} + 1, P). \tag{1}$$

Hence, according to our discussion above, we have the $\mathcal{A}$ jump hierarchy below $\omega^{\gamma_n} + 1$ available in $\mathcal{U}(\mathsf{NFA})$, i.e. $\mathcal{U}(\mathsf{NFA})$ proves

$$(\forall a \prec \omega^{\gamma_n} + 1) hier_{\mathcal{A}}(a)\downarrow. \tag{2}$$

In the following we work informally in $\mathcal{U}(\mathsf{NFA})$ and we want to derive $TI(\varphi\gamma_n 0, P)$. For that purpose let us assume $Prog(P)$. Then the previous lemma readily entails

$$(\forall a \prec \omega^{\gamma_n} + 1)[(\forall b \prec a) Prog(hier_{\mathcal{A}}(b)) \rightarrow Prog(hier_{\mathcal{A}}(a))]. \tag{3}$$

From (3) and an application of (Subst) to (1) we can derive

$$Prog(hier_{\mathcal{A}}(\omega^{\gamma_n})). \tag{4}$$

In particular, (4) entails that $0 \in hier_{\mathcal{A}}(\omega^{\gamma_n})$, and since $h(\omega^{\gamma_n}) = 0$ and $e(\omega^{\gamma_n}) = \gamma_n$, we have that

$$(\forall z \prec \omega^{\gamma_n}) \mathcal{J}(hier_{\mathcal{A}}(z), \varphi\gamma_n 0). \tag{5}$$

Substituting $z = 0$ this means $\mathcal{J}(P, \varphi\gamma_n 0)$, which implies

$$(\forall x \prec \varphi\gamma_n 0) P(x). \tag{6}$$

All together we have established $TI(\gamma_{n+1}, P)$ as desired. $\square$

Corollary 4 *$\mathcal{U}(\mathsf{NFA})$ proves $TI(\alpha, P)$ for each ordinal α less than Γ_0.*

From this corollary we can conclude as usual:

Corollary 5 *$(\Pi^0_1\text{-}\mathsf{CA})_{<\Gamma_0}$ is contained in $\mathcal{U}(\mathsf{NFA})$.*

For an alternative way of climbing up the Γ_0 ladder we refer the reader to Feferman [3, 2]. There it is shown how to make use of the *ordinary* jump hierarchy in a wellordering proof below Γ_0.

We have established $(\Pi^0_1\text{-}\mathsf{CA})_{<\Gamma_0}$ or, equivalently, $\mathsf{RA}_{<\Gamma_0}$ as lower bound of the full unfolding system $\mathcal{U}(\mathsf{NFA})$ for non-finitist arithmetic NFA. The next two sections are devoted to proving the converse direction, namely that $\mathsf{RA}_{<\Gamma_0}$ is also an upper bound for $\mathcal{U}(\mathsf{NFA})$.

5.2 Peano arithmetic with ordinals and the bar rule

In this section we introduce the theory $\mathsf{PA}_\Omega^+ + (\mathsf{Subst})$ of Peano arithmetic with ordinals plus substitution (or bar) rule, which — among other things — is analyzed proof-theoretically in Strahm [16]. This theory will be used in the next section in order to provide an upper bound for the system $\mathcal{U}(\mathsf{NFA})$.

Fixed point theories over Peano arithmetic with ordinals were introduced in Jäger [10]. They have previously been applied in the proof-theoretic analysis of systems of explicit mathematics with the non-constructive μ operator in an essential way, cf. for example Feferman and Jäger [7, 8] and Glaß and Strahm [9].

In the following we let $\mathcal{L}$ denote the usual language of first order arithmetic with function and relation symbols for all primitive recursive functions and relations. We also assume that $\mathcal{L}$ includes the unary free relation symbol P. If X is an n-ary relation symbol distinct from P, then $\mathcal{L}(X)$ denotes the extension of $\mathcal{L}$ by X. An $\mathcal{L}(X)$ formula is called X *positive* if each occurrence of X in it is positive. We call those X positive formulas which contain at most $\bar{x} = x_1, \ldots, x_n$ free, *inductive operator forms*; $\mathcal{A}(X, \bar{x})$ ranges over such forms. Observe that the relation symbol P can have positive *and* negative occurrences in an inductive operator form $\mathcal{A}(X, \bar{x})$.

Now we extend $\mathcal{L}$ to a new first order language $\mathcal{L}_\Omega$ by adding a new sort of *ordinal variables* $(\alpha, \beta, \gamma, \ldots)$, new binary relation symbols $<$ and $=$ for the less-than and equality relations on ordinals[3] and an $(n+1)$-ary relation symbol $\mathcal{P}_\mathcal{A}$ for each inductive operator form $\mathcal{A}(X, \bar{x})$ for which X is n-ary.

The *number terms* of $\mathcal{L}_\Omega$ $(r, s, t, \ldots)$ are the number terms of $\mathcal{L}$; the *ordinal terms* of $\mathcal{L}_\Omega$ are the ordinal variables of $\mathcal{L}_\Omega$. The *formulas* of $\mathcal{L}_\Omega$ $(A, B, C, \ldots)$ are inductively defined as follows:

Fm 1. If R is an n-ary relation symbol of $\mathcal{L}$, then $R(\bar{s})$ is an atomic formula of $\mathcal{L}_\Omega$.

Fm 2. The formulas $(\alpha < \beta)$, $(\alpha = \beta)$ and $\mathcal{P}_\mathcal{A}(\alpha, \bar{s})$ are atomic formulas of $\mathcal{L}_\Omega$.

Fm 3. If A and B are $\mathcal{L}_\Omega$ formulas, then so also are $\neg A$, $(A \wedge B)$, $(\exists x)A$ and $(\forall x)A$.

Fm 4. If A is an $\mathcal{L}_\Omega$ formula, then so also are $(\exists \alpha < \beta)A$, $(\forall \alpha < \beta)A$, $(\exists \alpha)A$ and $(\forall \alpha)A$.

The remaining logical connectives are defined as usual. For every $\mathcal{L}_\Omega$ formula A we write A^α to denote the $\mathcal{L}_\Omega$ formula which is obtained by replacing all unbounded ordinal quantifiers $(\mathcal{Q}\beta)$ in A by $(\mathcal{Q}\beta < \alpha)$. Additional abbreviations are:

$$\mathcal{P}_\mathcal{A}^\alpha(\bar{s}) := \mathcal{P}_\mathcal{A}(\alpha, \bar{s}), \quad \mathcal{P}_\mathcal{A}^{<\alpha}(\bar{s}) := (\exists \beta < \alpha)\mathcal{P}_\mathcal{A}^\beta(\bar{s}), \quad \mathcal{P}_\mathcal{A}(\bar{s}) := (\exists \alpha)\mathcal{P}_\mathcal{A}^\alpha(\bar{s}).$$

We introduce several classes of $\mathcal{L}_\Omega$ formulas, which will be important for the ordinal part of our fixed point theories. The Δ_0^Ω formulas are the $\mathcal{L}_\Omega$ formulas which do

[3] In general it will be clear from the context whether $<$ and $=$ denote the less-than and equality relations on the nonnegative integers or on the ordinals.

not contain unbounded ordinal quantifiers; the Σ^Ω [Π^Ω] formulas are the $\mathcal{L}_\Omega$ formulas which do not contain positive universal [existential] and negative existential [universal] ordinal quantifiers. The union of Σ^Ω and Π^Ω is denoted by ∇^Ω.

We are now ready to give the exact formulation of the theory PA_Ω^+. It is based on the usual two-sorted predicate calculus with equality and classical logic. The non-logical axioms of PA_Ω^+ are divided into the following six groups:

Ax 1. The axioms of Peano arithmetic PA with the exception of complete induction on the natural numbers.

Ax 2. For all inductive operator forms $\mathcal{A}(X, \bar{x})$:

$$\mathcal{P}_\mathcal{A}^\alpha(\bar{s}) \leftrightarrow \mathcal{A}(\mathcal{P}_\mathcal{A}^{<\alpha}, \bar{s}).$$

Ax 3. Σ^Ω Reflection axioms. For all Σ^Ω formulas A:

$$A \to (\exists\alpha)A^\alpha.$$

Ax 4. Linearity axioms:

$$\alpha \not< \alpha \wedge (\alpha < \beta \wedge \beta < \gamma \to \alpha < \gamma) \wedge (\alpha < \beta \vee \alpha = \beta \vee \beta < \alpha).$$

Ax 5. Formula induction on the natural numbers. For all $\mathcal{L}_\Omega$ formulas $A(x)$:

$$A(0) \wedge (\forall x)(A(x) \to A(x')) \to (\forall x)A(x).$$

Ax 6. Σ^Ω induction on the ordinals. For all Σ^Ω formulas $A(\alpha)$:

$$(\forall\alpha)[(\forall\beta < \alpha)A(\beta) \to A(\alpha)] \to (\forall\alpha)A(\alpha).$$

Remark 6 From the inductive operator and Σ^Ω reflection axioms one can easily deduce that the Σ^Ω formulas $\mathcal{P}_\mathcal{A}$ describe fixed points of the inductive operator form $\mathcal{A}(X, \bar{x})$.

Crucial for the embedding of $\mathcal{U}(\mathsf{NFA})$ below is an adequate substitution or bar rule, which we formulate as follows.

(Subst) $$A[P] \Rightarrow A[B/P].$$

Here A is a formula in the language $\mathcal{L}$ and B is allowed to be an *arbitrary* $\mathcal{L}_\Omega$ formula. $A[B/P]$ just means $A[P]$ with each occurrence of $P(t)$ replaced by $B(t)$.

The following theorem about the proof-theoretic strength of $\mathsf{PA}_\Omega^+ + (\mathsf{Subst})$ is established in Strahm [16]. It will be used in the upper bound computation of $\mathcal{U}(\mathsf{NFA})$ in the next section.

Theorem 7 $\mathsf{PA}_\Omega^+ + (\mathsf{Subst})$ *is proof-theoretically equivalent to* $\mathsf{RA}_{<\Gamma_0}$ *and conservatively extends* $\mathsf{RA}_{<\Gamma_0}$ *with respect to arithmetic statements.*

We finish this section by mentioning that this theorem still holds if we allow induction on the ordinals for formulas in ∇^Ω, i.e. formulas which are either Σ^Ω or Π^Ω, cf. Strahm [16] for details.

5.3 Upper bounds

In this paragraph we establish an embedding of $\mathcal{U}(\mathsf{NFA})$ into $\mathsf{PA}^+_\Omega + (\mathsf{Subst})$, thereby showing the equivalence of $\mathcal{U}(\mathsf{NFA})$ and $\mathsf{RA}_{<\Gamma_0}$. Since types of the form $(\bar{\tau}_0, \bar{\iota} \tilde{\to} \iota)$ do not involve predicate types, the corresponding terms can be interpreted in the very same way as in Section 3 for $\mathcal{U}_0(\mathsf{NFA})$. The treatment of the predicate part of $\mathcal{U}(\mathsf{NFA})$ is much more involved. Basically, function variables of type $(\bar{\iota} \tilde{\to} \pi_n)$ are also interpreted as partial recursive functions, but the ranges and hence definedness conditions for such functions will be more complicated, as we will see below.

In the sequel we make use of the usual primitive recursive coding machinery: $\langle \ldots \rangle$ is a standard primitive recursive function for forming n-tuples $\langle t_1, \ldots, t_n \rangle$; Seq denotes the primitive recursive set of sequence numbers; $lh(t)$ denotes the length of (the sequence coded by) t; $Seq_n(t)$ abbreviates $Seq(t) \wedge lh(t) = n$; $(t)_i$ is the ith component of (the sequence coded by) t for $i < lh(t)$, in particular, $t = \langle (t)_0, \ldots, (t)_{lh(t) \dot{-} 1} \rangle$ for sequence numbers t; we write $(t)_{i,j}$ instead of $((t)_i)_j$; $\star$ denotes the usual primitive recursive operation of sequence concatenation; finally, if $\bar{x} = x_1, \ldots, x_n$ then we write $\langle \bar{x} \rangle$ for $\langle x_1, \ldots, x_n \rangle$.

We start off by inductively defining a collection Π of (non-unique) codes for the predicates of $\mathcal{U}(\mathsf{NFA})$ together with an $\in$ relation which determines the extension for each such code. Since a direct inductive definition of Π and $\in$ would involve negative occurrences of $\in$, we also have to define a complementary relation $\bar{\in}$ for $\in$ in order to fit into the framework for *positive inductive definitions* which is available in PA^+_Ω. We use the following codes for n-ary predicates:

- $\langle 0, 2 \rangle$ for the binary predicate Eq,
- $\langle 1, 1 \rangle$ for the unary predicate Pr,
- $\langle 2, n, a, f_1, \ldots, f_m \rangle$ for the n-ary predicate $Inv(a, f_1, \ldots, f_m)$,
- $\langle 3, n, a \rangle$ for the n-ary predicate $Neg(a)$,
- $\langle 4, n, a, b \rangle$ for the n-ary predicate $Conj(a, b)$,
- $\langle 5, n, a \rangle$ for the n-ary predicate $Un(a)$,
- $\langle 6, n, f \rangle$ for the n-ary predicate $Join(f)$.

Now is is possible to give a simultaneous inductive definition for Π, $\in$ and $\bar{\in}$, which is readily encoded into a single one in order to fit into $\mathcal{L}_\Omega$. The defining condition for $\Pi(a)$ is given by the disjunction of the following clauses:

(i) $a = \langle 0, 2 \rangle$,

(ii) $a = \langle 1, 1 \rangle$,

(iii) $Seq(a) \wedge (a)_0 = 2 \wedge lh(a) = (a)_{2,1} + 3 \wedge \Pi((a)_2)$,

(iv) $Seq_3(a) \wedge (a)_0 = 3 \wedge (a)_{2,1} = (a)_1 \wedge \Pi((a)_2)$,

(v) $Seq_4(a) \wedge (a)_0 = 4 \wedge (a)_{2,1} = (a)_1 \wedge (a)_{3,1} = (a)_1 \wedge \Pi((a)_2) \wedge \Pi((a)_3)$,

(vi) $Seq_3(a) \wedge (a)_0 = 5 \wedge (a)_{2,1} = (a)_1 + 1 \wedge \Pi((a)_2)$,

(vii) $Seq_3(a) \wedge (a)_0 = 6 \wedge (a)_1 > 0 \wedge$
$(\forall x)(\exists b)[\{(a)_2\}(x) = b \wedge (b)_1 = (a)_1 \dot{-} 1 \wedge \Pi(b)]$.

The clauses for $(x \in a)$ are straightforward but rather lengthy to spell out. Observe that in clause (iii) references to $\bar{\in}$ are needed. We encode multiple arguments of a predicate a into a single one by making use of sequence numbering as usual. Summing up, the defining clauses for the relation $(x \in a)$ read as follows:

(i) $a = \langle 0, 2\rangle \wedge Seq_2(x) \wedge (x)_0 = (x)_1$,

(ii) $a = \langle 1, 1\rangle \wedge Seq_1(x) \wedge P((x)_0)$,

(iii) $Seq(a) \wedge (a)_0 = 2 \wedge lh(a) = (a)_{2,1} + 3 \wedge \Pi((a)_2) \wedge Seq_{(a)_1}(x) \wedge$
$\langle \{(a)_3\}((x)_0, \dots, (x)_{lh(x) \dot{-} 1}), \dots, \{(a)_{lh(a) \dot{-} 1}\}((x)_0, \dots, (x)_{lh(x) \dot{-} 1})\rangle \in (a)_2$,

(iv) $Seq_3(a) \wedge (a)_0 = 3 \wedge (a)_{2,1} = (a)_1 \wedge \Pi((a)_2) \wedge$
$Seq_{(a)_1}(x) \wedge x \,\bar{\in}\, (a)_2$,

(v) $Seq_4(a) \wedge (a)_0 = 4 \wedge (a)_{2,1} = (a)_1 \wedge (a)_{3,1} = (a)_1 \wedge \Pi((a)_2) \wedge \Pi((a)_3) \wedge$
$Seq_{(a)_1}(x) \wedge x \in (a)_3 \wedge x \in (a)_4$,

(vi) $Seq_3(a) \wedge (a)_0 = 5 \wedge (a)_{2,1} = (a)_1 + 1 \wedge \Pi((a)_2) \wedge$
$Seq_{(a)_1}(x) \wedge (\forall y)[x \star \langle y\rangle \in (a)_2]$,

(vii) $Seq_3(a) \wedge (a)_0 = 6 \wedge (a)_1 > 0 \wedge$
$(\forall x)(\exists b)[\{(a)_2\}(x) = b \wedge (b)_1 = (a)_1 \dot{-} 1 \wedge \Pi(b)] \wedge$
$Seq_{(a)_1}(x) \wedge \langle (x)_0, \dots, (x)_{(a)_1 \dot{-} 2}\rangle \in \{(a)_2\}((x)_{(a)_1 \dot{-} 1})$.

The clauses for $x \,\bar{\in}\, a$ are similar and, therefore, we omit them. All together we have convinced ourselves that Π, $\in$ and $\bar{\in}$ can be generated by a (simultaneous) positive inductive definition. Hence, Π, $\in$ and $\bar{\in}$ exist in PA^+_Ω as slices of a suitable fixed point, and the above defining conditions are derivable there. Recall that $\Pi, \in$ and $\bar{\in}$ are Σ^Ω, whereas the corresponding stage predicates $\Pi^\alpha, \in^\alpha$ and $\bar{\in}^\alpha$ are Δ^Ω_0.

Observe that it does not follow from the fixed point property alone that $\in$ and $\bar{\in}$ are complementary on predicate codes. However, this property can be shown by Δ^Ω_0 induction on the ordinals. This is the content of the following lemma.

Lemma 8 *We have that* PA^+_Ω *proves:*

$$(\forall a)[\Pi(a) \rightarrow (\forall x)(Seq_{(a)_1}(x) \rightarrow (x \in a \leftrightarrow \neg(x \,\bar{\in}\, a)))].$$

Proof. We first notice that the extension of a predicate code is fully determined at the first stage it gets into Π. Formally, one easily establishes the following statement by Δ_0^Ω induction on α:

$$(\forall\beta<\alpha)(\forall a)[\Pi^\beta(a) \to \\ (\forall x)(\mathit{Seq}_{(a)_1}(x) \to ((x \in^\beta a \leftrightarrow x \in^\alpha a) \wedge (x \bar{\in}^\beta a \leftrightarrow x \bar{\in}^\alpha a)))].$$

Now the claim of the lemma is immediate from

$$(\forall a)[\Pi^\alpha(a) \to (\forall x)(\mathit{Seq}_{(a)_1}(x) \to (x \in^\alpha a \leftrightarrow \neg(x \bar{\in}^\alpha a)))],$$

which is readily shown by Δ_0^Ω induction on the ordinals as well. $\square$

In the following $\Pi_n(a)$ means that a is a code for a n-ary predicate, i.e. $\Pi_n(a)$ abbreviates $\Pi(a) \wedge (a)_1 = n$. Similarly, $\Pi_n^\alpha(a)$ expresses that a codes an n-ary predicate on stage α; observe that $\Pi_n^\alpha(a)$ is a Δ_0^Ω statement of $\mathcal{L}_\Omega$. Equality between predicates must be understood *extensionally* in the sequel; accordingly, we define

$$a =_{\pi_n} b \ := \ \Pi_n(a) \wedge \Pi_n(b) \wedge (\forall\bar{x})(\langle\bar{x}\rangle \in a \leftrightarrow \langle\bar{x}\rangle \in b).$$

It is our aim next to show how terms involving predicate types can be interpreted in order to satisfy their defining axioms, provably in PA_Ω^+. As we have already indicated earlier, we let variables f of type $(\bar{\iota} \tilde{\to} \pi_n)$ range over indices of partial recursive functions, but definedness for such functions must now be understood with respect to Π, namely $f(\bar{x})\downarrow$ means $(\exists y)[\{f\}(\bar{x}) \simeq y \wedge \Pi_n(y)]$, in short $\Pi_n(\{f\}(\bar{x}))$.

The interpretation of $\mathcal{U}(\mathsf{NFA})$ expressions in the partial recursive functions is obvious except for terms of the form $\mathit{LFP}(\lambda f, \bar{x}.t)$ with $f : \bar{\iota} \tilde{\to} \pi_n$, $\bar{x} : \bar{\iota}$ and $t : \pi_n$. Moreover, terms can always be interpreted in such a way that testing for definedness with respect to Π is delayed to the end of the evaluation procedure. The treatment of LFP terms is rather delicate and requires some specific monotonicity notions, which we want to introduce now. We set for all natural numbers a, b, f, g:

$$\begin{aligned} a \subseteq_{\pi_n}^{\alpha} b &:= \Pi_n^\alpha(a) \to a =_{\pi_n} b, \\ a \subseteq_{\pi_n}^{<\alpha} b &:= (\forall\beta<\alpha)(a \subseteq_{\pi_n}^{\beta} b), \\ f \subseteq_{\bar{\iota}\tilde{\to}\pi_n}^{\alpha} g &:= (\forall\bar{x})(\{f\}(\bar{x}) \subseteq_{\pi_n}^{\alpha} \{g\}(\bar{x})), \\ f \subseteq_{\bar{\iota}\tilde{\to}\pi_n}^{<\alpha} g &:= (\forall\beta<\alpha)(f \subseteq_{\bar{\iota}\tilde{\to}\pi_n}^{\beta} g). \end{aligned}$$

We often omit the subscripts 'π_n' and '$\bar{\iota} \tilde{\to} \pi_n$' if they are given by the context. In addition, we use the vector notation as in $\bar{a} \subseteq^\alpha \bar{b}$ and $\bar{f} \subseteq^\alpha \bar{g}$ with its usual componentwise meaning.

Let us now assume that we are given a term $t[\bar{f}, \bar{a}, \bar{u}] : \pi_n$ with all free variables shown, $\bar{f} : \bar{\iota} \tilde{\to} \pi$, $\bar{a} : \pi$, and $\bar{u}$ variables not involving predicate types.[4] We further

[4]If the arity of a predicate can vary for each element of a given sequence, then we just use the notation π.

assume that $t[\bar{f}, \bar{a}, \bar{u}]$ has already been given a suitable recursion-theoretic interpretation, and we identify t with its interpretation if we are working informally in PA^+_Ω. Then we call t *progressively monotonic* (provably in PA^+_Ω), if PA^+_Ω proves:

$$\bar{f} \subseteq^{<\alpha} \bar{g} \wedge \bar{a} \subseteq^{<\alpha} \bar{b} \ \rightarrow \ t[\bar{f}, \bar{a}, \bar{u}] \subseteq^{\alpha} t[\bar{g}, \bar{b}, \bar{u}].^5$$

In the sequel we abbreviate "progressively monotonic" to "p-monotonic". For example, it is easy to see that the term $Join(f)$ for $f : \bar{\iota} \tilde{\rightarrow} \pi_n$ is p-monotonic. Indeed, assume that $f \subseteq^{<\alpha} g$ and $Join(f)$ is a predicate on level α; then we have (essentially by Ax 2 of PA^+_Ω) for all $\bar{x}$ that $\{f\}(\bar{x})$ is a predicate on level less than α and, hence, $\{f\}(\bar{x}) =_\pi \{g\}(\bar{x})$ by our assumption; this entails $Join(f) =_\pi Join(g)$ as desired. On the other hand, a term such as $f(\bar{x})$ is obviously *not* p-monotonic.

Our aim now is to give a straightforward recursion-theoretic interpretation of terms $LFP(\lambda f, \bar{x}.t[f, \bar{x}]) : \bar{\iota} \tilde{\rightarrow} \pi_n$ for previously interpreted terms $t[f, \bar{x}]$ which are p-monotonic in the above defined sense. The key in the proof below is the fact that induction on the ordinals is available for Σ^Ω statements in PA^+_Ω.

Lemma 9 *Assume that $t[f, \bar{x}] : \pi_n$ with $f : \bar{\iota} \tilde{\rightarrow} \pi_n$ and $\bar{x} : \bar{\iota}$, possibly having other free variables, is provably p-monotonic in PA^+_Ω. Then there exists a recursion-theoretic interpretation e, primitive recursively depending on these other parameters, of $LFP(\lambda f, \bar{x}.t[f, \bar{x}])$ so that the corresponding LFP axioms are provable in PA^+_Ω.*

Proof. In the following we work informally in PA^+_Ω. By the second recursion theorem, choose an e (depending primitive recursively on the additional parameters of t) so that we have for all $\bar{x}$:

$$\{e\}(\bar{x}) \simeq t[e, \bar{x}]. \tag{1}$$

We claim that e is indeed the least fixed point of t. To see this, let f be another fixed point, i.e. assume that we have for all $\bar{x}$:

$$\{f\}(\bar{x}) \simeq_{\pi_n} t[f, \bar{x}], \tag{2}$$

where $\simeq_{\pi_n}$ denotes the natural partial equality relation which is associated to $=_{\pi_n}$. Then we establish by induction on the ordinals the following statement:

$$(\forall \bar{x})[\Pi^\alpha_n(\{e\}(\bar{x})) \ \rightarrow \ \{e\}(\bar{x}) =_{\pi_n} \{f\}(\bar{x})]. \tag{3}$$

Due to Lemma 8, it is easily seen that (3) is in fact a Σ^Ω statement. Let us assume that (3) holds for all $\beta < \alpha$. Then we have by the definition of $\subseteq^{<\alpha}_{\pi_n}$ that

$$e \subseteq^{<\alpha}_{\pi_n} f. \tag{4}$$

Since t is p-monotonic, this implies for all $\bar{x}$:

$$t[e, \bar{x}] \subseteq^{\alpha}_{\pi_n} t[f, \bar{x}]. \tag{5}$$

[5] We use $f, g, a, b, u \ldots$ as variables of $\mathcal{L}$ here.

If we now assume $\Pi_n^\alpha(\{e\}(\bar{x}))$ for some $\bar{x}$, then we obtain from (1), (2) and (5):

$$\{e\}(\bar{x}) = t[e,\bar{x}] =_{\pi_n} t[f,\bar{x}] =_{\pi_n} \{f\}(\bar{x}). \tag{6}$$

This is exactly our claim and, hence, we are done. $\Box$

It is important to mention here that p-monotonicity of $t[f,\bar{x}]$ is crucial for the fact that indeed *any* recursion-theoretic fixed point is already the least one. If we take, for example, the (non p-monotonic) term $t[f,\bar{x}] \equiv f(\bar{x})$ from above, then a fixed point can be anything and, in particular, there are solutions for t which are not least.

In a next step we now want to associate to each previously interpreted term t of type π_n a term $t^* : \pi_n$ having the same parameters as t in such a way that (i) t^* is p-monotonic, and (ii) $t^* \simeq_{\pi_n} t$, both provably in PA_Ω^+. This will immediately yield a suitable interpretation of LFP expressions: in order to obtain a recursion-theoretic interpretation of $LFP(\lambda f,\bar{x}.t)$, one just takes the least fixed point of t^*, which exists by the previous lemma; of course, this fixed point is also the least fixed point of t. For t^* we can choose, for example,

$$t^* \;:=\; Conj(t,t).$$

The following lemma tells us that t^* is indeed p-monotonic, provably in PA_Ω^+.

Lemma 10 *We have that t^* is provably p-monotonic in PA_Ω^+ for each term $t : \pi_n$.*

Proof. In the sequel we extend our $*$ translation to terms t of type $\bar{\iota} \tilde{\to} \pi_n$ by setting $t^* := t(\bar{x})^*$. In order to establish the claim of the lemma, one shows claims (i) and (ii) below simultaneously by induction on the complexity of t:

(i) If $t : \pi_n$, then t^* is provably p-monotonic in PA_Ω^+;

(ii) If $t : (\bar{\tau},\bar{\kappa} \tilde{\to} \pi_n)$, then we have for all $\bar{r} : \bar{\tau}$ and $\bar{s} : \bar{\kappa}$: if the $*$ translations of the terms involving predicate types in $\bar{r},\bar{s}$ are provably p-monotonic in PA_Ω^+, then so also $t(\bar{r},\bar{s})^*$ is provably p-monotonic in PA_Ω^+.

If t is a variable $a : \pi_n$ or $f : \bar{\iota} \tilde{\to} \pi_n$, then our claim is immediate from

$$a \subseteq^{<\alpha} b \;\to\; Conj(a,a) \subseteq^\alpha Conj(b,b), \tag{1}$$

$$f \subseteq^{<\alpha} g \;\to\; Conj(f(\bar{s}),f(\bar{s})) \subseteq^\alpha Conj(g(\bar{s}),g(\bar{s})), \tag{2}$$

which are both derivable in PA_Ω^+. In all other cases except possibly the LFP construction, the claim follows almost immediately from the induction hypotheses. It remains to discuss the case where t has the form $LFP(\lambda h,\bar{x}.s)$ for $h : \bar{\iota} \tilde{\to} \pi_n$, $\bar{x} : \bar{\iota}$ and $s : \pi_n$. This term may depend on additional parameters $\bar{f} : \bar{\iota} \tilde{\to} \pi$ and $\bar{a} : \pi$ involving predicate types (and other parameters involving number types only). According to our discussion above, we have a suitable interpretation of $LFP(\lambda h,\bar{x}.s)$, say $\ell[\bar{f},\bar{a}]$, which satisfies for all $\bar{x}$:

$$\{\ell[\bar{f},\bar{a}]\}(\bar{x}) \simeq s^*[\ell[\bar{f},\bar{a}],\bar{f},\bar{a},\bar{x}]. \tag{3}$$

It is our aim to show that ℓ is p-monotonic with respect to $\bar{f}, \bar{a}$, which will immediately imply that also its $*$ translation is p-monotonic. For that purpose we assume that

$$\bar{f} \subseteq^{<\alpha} \bar{g} \ \wedge \ \bar{a} \subseteq^{<\alpha} \bar{b}. \tag{4}$$

We want to derive $\ell[\bar{f}, \bar{a}] \subseteq^{\alpha} \ell[\bar{g}, \bar{b}]$. For this it enough to show for all ordinals $\beta \leq \alpha$ that

$$\ell[\bar{f}, \bar{a}] \subseteq^{\beta} \ell[\bar{g}, \bar{b}], \tag{5}$$

which we want to prove by Σ^{Ω} induction on the ordinals. Therefore, assume that (5) holds for all ordinals $\gamma < \beta \leq \alpha$, i.e.

$$\ell[\bar{f}, \bar{a}] \subseteq^{<\beta} \ell[\bar{g}, \bar{b}]. \tag{6}$$

By the induction hypothesis for (i) we know that s^* is p-monotonic so that we readily obtain from (4) and (6) for all $\bar{x}$:

$$s^*[\ell[\bar{f}, \bar{a}], \bar{f}, \bar{a}, \bar{x}] \subseteq^{\beta} s^*[\ell[\bar{g}, \bar{b}], \bar{g}, \bar{b}, \bar{x}]. \tag{7}$$

From (7) and the fixed point property (3) we can now immediately derive

$$\ell[\bar{f}, \bar{a}] \subseteq^{\beta} \ell[\bar{g}, \bar{b}] \tag{8}$$

as desired. All together we have shown $\ell[\bar{f}, \bar{a}] \subseteq^{\alpha} \ell[\bar{g}, \bar{b}]$ and, hence, we have established that our interpretation of *LFP* expressions preserves p-monotonicity in parameters. This concludes the proof of our lemma. $\square$

We have shown how to interpret the predicate type part of $\mathcal{U}(\mathsf{NFA})$ in the subsystem PA_{Ω}^{+} of Peano arithmetic with ordinals. Crucial use has been made of Σ^{Ω} induction on the ordinals. In order to complete the embedding of $\mathcal{U}(\mathsf{NFA})$ in $\mathsf{PA}_{\Omega}^{+} + (\mathsf{Subst})$ we readily observe that the substitution rule (Subst) of $\mathcal{U}(\mathsf{NFA})$ directly translates into (Subst) as it is formulated in $\mathcal{L}_{\Omega}$; to see this, notice that the premise of (Subst) in $\mathcal{U}(\mathsf{NFA})$ is a $\mathcal{U}_0(\mathsf{NFA})$ formula whose interpretation is just a formula of $\mathcal{L}$ and, therefore, matches the premise of the (Subst) rule of $\mathcal{L}_{\Omega}$. We have thus established the following theorem.

Theorem 11 $\mathcal{U}(\mathsf{NFA})$ *is contained in* $\mathsf{PA}_{\Omega}^{+} + (\mathsf{Subst})$.

Together with Corollary 5 and Theorem 7 we are now able to state the following corollary about the proof-theoretic strength of $\mathcal{U}(\mathsf{NFA})$.

Corollary 12 $\mathcal{U}(\mathsf{NFA})$ *is proof-theoretically equivalent to* $\mathsf{RA}_{<\Gamma_0}$ *and conservatively extends* $\mathsf{RA}_{<\Gamma_0}$ *with respect to arithmetic statements.*

6 The proof-theoretic strength of $\mathcal{U}_1(\mathsf{NFA})$

In this section we establish the proof-theoretic equivalence of the intermediate unfolding system $\mathcal{U}_1(\mathsf{NFA})$ and ramified analysis in all finite levels, $\mathsf{RA}_{<\omega}$. It readily follows with the techniques presented in Schütte [15] that the proof-theoretic ordinal of $\mathsf{RA}_{<\omega}$ is exactly the first fixed point of the ε function, namely $\varphi 20$. The ordinal $\varphi 20$ is also known as the least ordinal which is not autonomous in a semiformal system for pure number theory, cf. [15], p. 214. Other known systems which possess proof-theoretic ordinal $\varphi 20$ and are proof-theoretically equivalent to $\mathsf{RA}_{<\omega}$ include: (i) $(\Pi^0_1\text{-}\mathsf{CA})$ plus bar or substitution rule (Subst), cf. Rathjen [14]; (ii) the system $(\Pi^0_1\text{-}\mathsf{CA})_\omega\!\upharpoonright$ in which the existence of the Π^0_1 jump hierarchy is claimed up to and including ω, cf [14].

We divide our considerations into two subsections. First, we show that transfinite induction on each initial segment of $\varphi 20$ is available in $\mathcal{U}_1(\mathsf{NFA})$. In the second paragraph we sketch an adaptation of the arguments of Section 3.1 in order to yield an embedding of $\mathcal{U}_1(\mathsf{NFA})$ in $(\Pi^0_1\text{-}\mathsf{CA}) + (\mathsf{Subst})$.

6.1 Lower bounds

In the sequel let us briefly sketch a wellordering proof for the intermediate unfolding system $\mathcal{U}_1(\mathsf{NFA})$. In particular, we show that $\mathcal{U}_1(\mathsf{NFA})$ proves $TI(\alpha, P)$ for each ordinal α less than the first fixed point of the ε function, $\varphi 20$. This will yield $\mathsf{RA}_{<\omega}$ as a proof-theoretic lower bound of $\mathcal{U}_1(\mathsf{NFA})$.

In a first step we define a sequence of sets Y_i $(i \in \mathbb{N})$ of natural numbers in $\mathcal{U}_1(\mathsf{NFA})$ inductively as follows:

$$\begin{aligned} Y_0 &:= \{m : P(m)\}, \\ Y_{i+1} &:= \{m : \mathcal{J}(Y_i, \omega^m)\}. \end{aligned}$$

Observe that here $\mathcal{J}$ denotes the jump operation from Section 5.1. Notice further that the initial segments of this Y hierarchy are the main ingredients used in the wellordering proof of PA. In order to represent Y_i $(i \in \mathbb{N})$ uniformly in $\mathcal{U}_1(\mathsf{NFA})$, we first choose a term $s : \pi_1 \tilde{\to} \pi_1$ so that $\mathcal{U}_1(\mathsf{NFA})$ proves:

$$s(X)\!\downarrow \wedge\ (\forall y)(y \in s(X) \leftrightarrow \mathcal{J}(X, \omega^y)).$$

Such an s exists by the predicate axioms Ax 6-11 only, in particular *Join* is *not* used for the definition of s. Now, let $t : \iota \tilde{\to} \pi_1$ be given by least fixed point recursion as follows:

$$t := LFP(\lambda f, x.\{\text{if } x = 0 \text{ then } Pr \text{ else } s(Pd(x))\}).$$

A straightforward induction on x yields that $(\forall x)t(x)\!\downarrow$, and indeed t uniformly represents the Y hierarchy in $\mathcal{U}_1(\mathsf{NFA})$.

The following lemma is a straightforward adaptation to our context of Lemma 7, d) in Schütte [15], p. 183. Usually, this lemma is the main step in the wellordering proof for $(\Pi^0_1\text{-}\mathsf{CA})$. As the proof in our framework is literally the same, we omit it and refer the reader to [15] for details.

Lemma 13 $\mathcal{U}_1(\mathsf{NFA})$ *proves* $Prog(\lambda a.(\forall x)\, TI(\varepsilon_a, t(x)))$.

Let us now assume that $(\alpha_n)_{n\in\mathbb{N}}$ is a canonical fundamental sequence for $\varphi 20$ given by $\alpha_0 = \omega$ and $\alpha_{n+1} = \varepsilon_{\alpha_n}$. Then we have the following theorem.

Theorem 14 $\mathcal{U}_1(\mathsf{NFA})$ *proves* $TI(\alpha_n, P)$ *for each natural number* n.

Proof. We establish the claim of the theorem by metamathematical induction on n. The case $n = 0$ is trivial. For the step from n to $n+1$ let us assume $TI(\alpha_n, P)$. An invocation of (Subst) thus readily yields

$$TI(\alpha_n, \lambda a.(\forall x)\, TI(\varepsilon_a, t(x))). \tag{1}$$

On the other hand, we know by the previous lemma that

$$Prog(\lambda a.(\forall x)\, TI(\varepsilon_a, t(x))). \tag{2}$$

From (1) and (2) we can immediately derive

$$(\forall x)\, TI(\alpha_{n+1}, t(x)), \tag{3}$$

which readily entails $TI(\alpha_{n+1}, P)$ as claimed. This finishes our argument. □

Corollary 15 $\mathcal{U}_1(\mathsf{NFA})$ *proves* $TI(\alpha, P)$ *for each* α *less than* $\varphi 20$.

6.2 Upper bounds

In this section we briefly indicate how the upper bound argument for the full unfolding system $\mathcal{U}(\mathsf{NFA})$ given in Section 5.3 is to be modified in order to yield an upper bound of $\mathcal{U}_1(\mathsf{NFA})$ in $(\Pi^0_1\text{-}\mathsf{CA}) + (\mathsf{Subst})$.

One first inductively defines a collection Π^- of (non-unique) codes for the predicates of $\mathcal{U}_1(\mathsf{NFA})$; the generating clauses for Π^- are just the clauses (i)-(vi) of the definition of Π in Section 5.3. It is straightforward that the so-obtained inductive definition produces a primitive recursive set of codes Π^-. In the next step one associates to each code $c \in \Pi^-$ its natural extension; this can easily be done via a Δ^1_1 truth definition for arithmetic sentences, which is available in $(\Pi^0_1\text{-}\mathsf{CA})$ as usual.

Once we have at hand the collection Π^- of codes for predicates of $\mathcal{U}_1(\mathsf{NFA})$ together with the intended extension for each such code, we can basically proceed as in Section 5.3 in order to interpret the $\mathcal{U}_1(\mathsf{NFA})$ terms involving predicate types. The main difference is that ordinal levels of the sets in Π can now be replaced by natural numbers corresponding to the stage on which codes for $\mathcal{U}_1(\mathsf{NFA})$ predicates are produced according to the inductive definition of Π^-. Then the monotonicity notions as well as Lemma 9 and Lemma 10 carry over directly to this much simpler framework. Hence, the predicate part of $\mathcal{U}_1(\mathsf{NFA})$ can be modeled directly in $(\Pi^0_1\text{-}\mathsf{CA})$. Moreover, the substitution rule of $\mathcal{U}_1(\mathsf{NFA})$ immediately translates into the corresponding rule formulated in the language of $(\Pi^0_1\text{-}\mathsf{CA})$. Summing up, we have established an embedding of $\mathcal{U}_1(\mathsf{NFA})$ into $(\Pi^0_1\text{-}\mathsf{CA}) + (\mathsf{Subst})$.

Theorem 16 $\mathcal{U}_1(\mathsf{NFA})$ *is contained in* $(\Pi^0_1\text{-}\mathsf{CA}) + (\mathsf{Subst})$.

Corollary 17 $\mathcal{U}_1(\mathsf{NFA})$ *is proof-theoretically equivalent to* $\mathsf{RA}_{<\omega}$ *and conservatively extends* $\mathsf{RA}_{<\omega}$ *with respect to arithmetic statements.*

References

[1] Beeson, M. J. *Foundations of Constructive Mathematics: Metamathematical Studies.* Springer, Berlin, 1985.

[2] Feferman, S. A more perspicuous system for predicativity. In *Konstruktionen vs. Positionen I.* de Gruyter, Berlin, 1979, pp. 87–139.

[3] Feferman, S. Reflecting on incompleteness. *Journal of Symbolic Logic 56*, 1 (1991), 1–49.

[4] Feferman, S. A new approach to abstract data types, II. Computation on ADTs as ordinary computation. In *Computer Science Logic '91*, E. Börger et al., Ed., vol. 626 of *Lecture Notes in Computer Science*. Springer, Berlin, 1992, pp. 79–95.

[5] Feferman, S. Computation on abstract data types. The extensional approach, with an application to streams. *Annals of Pure and Applied Logic 81* (1996), 75–113.

[6] Feferman, S. Gödel's program for new axioms: Why, where, how and what? In *Gödel '96*, P. Hájek, Ed., vol. 6 of *Lecture Notes in Logic*. Springer, Berlin, 1996, pp. 3–22.

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

[8] Feferman, S., and Jäger, G. Systems of explicit mathematics with non-constructive μ-operator. Part II. *Annals of Pure and Applied Logic 79*, 1 (1996).

[9] Glass, T., and Strahm, T. Systems of explicit mathematics with non-constructive μ-operator and join. *Annals of Pure and Applied Logic 82* (1996), 193–219.

[10] Jäger, G. Fixed points in Peano arithmetic with ordinals. *Annals of Pure and Applied Logic 60*, 2 (1993), 119–132.

[11] Moschovakis, Y. The formal language of recursion. *Journal of Symbolic Logic 54* (1989), 1216–1252.

[12] Platek, R. *Foundations of Recursion Theory*. PhD thesis, Stanford University, 1966.

[13] Pohlers, W. *Proof Theory: An Introduction*, vol. 1407 of *Lecture Notes in Mathematics*. Springer, Berlin, 1988.

[14] Rathjen, M. The role of parameters in bar rule and bar induction. *Journal of Symbolic Logic 56*, 2 (1991), 715–730.

[15] Schütte, K. *Proof Theory*. Springer, Berlin, 1977.

[16] Strahm, T. The non-constructive μ operator, fixed point theories with ordinals, and the bar rule. *Annals of Pure and Applied Logic*. To appear.