Proof-theoretic strength of PRON with various extensions

Diplomarbeit
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

David Steiner

2001

Leiter der Arbeit:

Prof. Dr. Gerhard Jäger
Institut für Informatik und angewandte Mathematik

Contents

0 Introduction

In this thesis we will introduce the theory PRON of *primitive recursive operations* and *numbers*. It is more or less the same theory like $\mathsf{PEA}^+ + (\mathsf{r})$ introduced by Schlüter in [12]. But it will be very helpful to formulate the axioms of PRON similar to the axioms of the well known applicative theory BON. There are the following differences between PRON and BON:

1. In PRON we have the three constants i, a_2, and b_2 satisfying $\mathsf{i}a = a$, $\mathsf{a}_2{<}a, b{>}c \simeq {<}ac, bc{>}$, and $\mathsf{b}_2{<}a, b{>}c \simeq a(bc)$. But we do not have a constant $\hat{\mathsf{s}}$ with $\hat{\mathsf{s}}abc \simeq ac(bc)$ like in BON.

2. We define a function symbol $<>$ instead of a constant $\hat{\mathsf{p}}$ for pairing. The reason for this difference is that we will define a restricted λ abstraction $(\lambda^* x.t)$ by induction on the build-up of the term t. In this definition we will have to distinct between a pair and an application.
   The addition of the function symbol $<>$ allows us to define one more point differently: For $n > 0$ we will define a term t to be in $(\mathsf{N}^n \to \mathsf{N})$, if $x_0, \ldots, x_{n-1} \in \mathsf{N}$ implies $t{<}x_0, \ldots, x_{n-1}{>} \in \mathsf{N}$.

3. In the axioms for primitive recursion we do not claim totality. This is only necessary in section 5, when we will replace formula induction $(\mathcal{L}\text{-}\mathsf{I_N})$ by set induction $(\mathsf{S}\text{-}\mathsf{I_N})$. In this case we will write $\mathsf{PRON^t}$ instead of PRON.

There are certain technical difficulties because of these differences: For instance, the case $n = 0$ must always be handled separately, because an n-ary function is not defined by iterated term application. It is also true that we cannot define full λ abstraction, because we do not have the constant $\hat{\mathsf{s}}$ of BON.

In section 2 we will give a total recursion theoretic model of PRON. We know from Kahle [9] that this is not possible for BON, because there exists a term $\mathsf{not_N}$ so that BON proves $\mathsf{not_N} \notin \mathsf{N}$. The constants of PRON are going to be interpreted as unary indices of primitive recursive functions. In section 4 we will enhance this model by the interpretations of the additional constants $\boldsymbol{\mu}$ and E_1. Because of the choice of this kind of model, the axioms for the non-constructive μ operator and the Suslin operator E_1 will be defined in the following way:

$(\boldsymbol{\mu}.1)\qquad (\forall x \in \mathsf{N})(f{<}a, x{>} \in \mathsf{N}) \leftrightarrow \boldsymbol{\mu} f a \in \mathsf{N}$

$(\boldsymbol{\mu}.2)\qquad (\forall x \in \mathsf{N})(f{<}a, x{>} \in \mathsf{N}) \to$
$\qquad\qquad [(\exists x \in \mathsf{N})(f{<}a, x{>} = \mathsf{0}) \to f{<}a, \boldsymbol{\mu} f a{>} = \mathsf{0}]$

$(\mathsf{E_1}.1)\quad (\forall x, y \in \mathsf{N})(f{<}a, x, y{>} \in \mathsf{N}) \leftrightarrow \mathsf{E_1} fa \in \mathsf{N}$

$(\mathsf{E_1}.2)\quad (\forall x, y \in \mathsf{N})(f{<}a, x, y{>} \in \mathsf{N}) \rightarrow [(\exists g)[g \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge (\forall x \in \mathsf{N})(f{<}a, g(\mathsf{s_N} x), gx{>} = \mathsf{0})] \leftrightarrow \mathsf{E_1} fa = \mathsf{0}]$

Let $n > 0$. If a is an n-tuple of terms, then ${<}a, x{>}$ is an $(n+1)$-tuple, so $\boldsymbol{\mu}$ can be applied to any term in $(\mathsf{N}^{n+1} \rightarrow \mathsf{N})$. The constant $\mathsf{E_1}$ can be applied to every term in $(\mathsf{N}^{n+2} \rightarrow \mathsf{N})$ for the same reason. In section 2 we will also give the exact axiomatization of BON. We will give several reasons why BON must be the stronger theory than PRON. But PRON is strong enough to have a term which represents an arbitrary primitive recursive function.

We will write $\mathsf{PRON}(\boldsymbol{\mu})$ and $\mathsf{PRON^t}(\boldsymbol{\mu})$ for PRON and $\mathsf{PRON^t}$ plus the axioms about $\boldsymbol{\mu}$. Further, PRON(SUS) and $\mathsf{PRON^t}$(SUS) will denote the theories $\mathsf{PRON}(\boldsymbol{\mu})$ and $\mathsf{PRON^t}(\boldsymbol{\mu})$ extended by the axioms about $\mathsf{E_1}$. In this thesis we will be able to show the following proof-theoretic results:

1. $\mathsf{PRA} \subseteq \mathsf{PRON^t} + (\mathsf{S}\text{-}\mathsf{I_N}) \subseteq \mathsf{PRA^-} + (\Sigma_1\text{-}\mathsf{I_N})$
2. $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N}) \equiv \mathsf{PA}$
3. $\mathsf{PRON^t}(\boldsymbol{\mu}) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{ACA_0}$
4. $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N}) \equiv \mathsf{ACA}$
5. $\mathsf{PRON^t}(\mathsf{SUS}) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \Pi^1_1\text{-}\mathsf{CA_0}$
6. $\mathsf{PRON}(\mathsf{SUS}) + (\mathcal{L}\text{-}\mathsf{I_N}) \equiv \Pi^1_1\text{-}\mathsf{CA}$

In section 3 we start with the embeddings of PRON, $\mathsf{PRON}(\boldsymbol{\mu})$, and PRON(SUS), always with formula induction. The lower proof-theoretic bounds can be proved with a translation from arithmetic to the language $\mathcal{L}$ of applicative theories. For this purpose, we profit by the possibility to define a so-called *characteristic term* in $\mathcal{L}$ for every Π^1_0 (Π^1_1) formula. This term represents a subset of N, so it is helpful to derive the translation of the comprehension scheme. We can define a similar characteristic term in section 5 where we embed the same theories but with set induction instead of formula induction. Hereby, this characteristic term is also helpful to derive set induction and induction for quantifier free formulas.

In section 4 we are going to determine the upper bounds of the mentioned theories with formula induction. We will embed $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N})$ in $\mathsf{BON} + (\hat{\mathcal{L}}\text{-}\mathsf{I_N})$ and construct a model of $\mathsf{PRON}(\boldsymbol{\mu})$ in second order arithmetic. Then we will formalize a model of PRON(SUS) in Π^1_1-CA. For this purpose the crucial point is to prove that, given an e coding an $(n+2)$-ary Relation on N, the following equivalence holds: there exists a set-theoretic function which is a descending chain for e, if and only if there exists a natural number coding a total function which is also a descending chain for e. The upper bounds with set induction in section 5 are easier, because we can embed $\mathsf{PRON}^{\mathsf{t}} + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$ with the additional axioms in $\mathsf{BON}+(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$, $\mathsf{BON}(\hat{\boldsymbol{\mu}})+(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$, and $\mathsf{SUS}+(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$.

I am grateful to Prof. Dr. Gerhard Jäger for introducing me to applicative theories and PD Dr. Thomas Strahm for guiding me during my work. I also thank Dieter Probst and Marc Heissenbüttel for their competent advise. Further, I acknowledge the support I received from the entire research group. Last but not least I would like to thank my girlfriend and my parents who never ceased to motivate me.

David Steiner
Bern, October 1, 2001

1 The formal frameworks

1.1 Functions and indices

In this section we are going to introduce the class *PRIM* of primitive recursive functions as well as the extensions $PRIM(\mu)$ and $PRIM(SUS)$. We will define the most relevant primitive recursive functions like the coding of sequence numbers. Further, we will introduce a set of indices for every class of functions and we are going to define the level of an index.

Definition 1.1.1

1. *Let* n, m, k, x_0, $\ldots$, x_{n-1} *be arbitrary natural numbers. We define the following basic functions:*

   (a) *Successor.* $\mathcal{S}(x_0) := x_0 + 1$

   (b) *Constant functions.* $Cs^n_m(\vec{x}) := m$

   (c) *Projections. If* $k < n$ *then* $Pr^n_k(\vec{x}) := x_k$

2. *Let* n, m, x_0, $\ldots$, x_{n-1}, y *be arbitrary natural numbers and* $\mathcal{K}$ *be a class of number theoretic functions. We define the following closure characteristics of functions of* $\mathcal{K}$*:*

   (a) *Composition. If* $m > 0$ *and* f *is an* m*-ary function of* $\mathcal{K}$ *and* g_0, $\ldots$, g_{m-1} *are* n*-ary functions of* $\mathcal{K}$*, then the* n*-ary function*

   $$Comp^n(f, g_0, \ldots, g_{m-1})(\vec{x}) \quad := \quad f(g_0(\vec{x}), \ldots, g_{m-1}(\vec{x}))$$

   is an element of $\mathcal{K}$*.*

   (b) *Primitive recursion. If* f *is an* n*-ary function of* $\mathcal{K}$ *and* g *is an* $(n+2)$*-ary function of* $\mathcal{K}$*, then the* $(n+1)$*-ary function*

   $$Rec^{n+1}(f,g)(\vec{x}, y) \;:=\; \begin{cases} f(\vec{x}) & \textit{if } y = 0 \\ g(\vec{x}, y, Rec^{n+1}(f,g)(\vec{x}, y-1)) & \textit{if } y > 0 \end{cases}$$

   is an element of $\mathcal{K}$*.*

   (c) μ *operator. If* f *is an* $(n+1)$*-ary function of* $\mathcal{K}$*, then the* n*-ary function*

   $$Zero^n(f)(\vec{x}) \quad := \quad \begin{cases} \min\{y \mid f(\vec{x}, y) = 0\} & \textit{if there is an } y \textit{ so} \\ & \quad\textit{that } f(\vec{x}, y) = 0 \\ 0 & \textit{otherwise} \end{cases}$$

   is an element of $\mathcal{K}$*.*

(d) *Suslin operator. If f is an $(n+2)$-ary function of $\mathcal{K}$, then the n-ary function*

$$Sus^n(f)(\vec{x}) \;:=\; \begin{cases} 0 & \textit{if there is a unary function } g\textit{, so} \\ & \quad \textit{that } f(\vec{x}, g(\mathcal{S}(z)), g(z)) = 0 \\ & \quad \textit{for every natural number } z \\ 1 & \textit{otherwise} \end{cases}$$

is an element of $\mathcal{K}$.

3. *Now we can introduce the following classes of functions:*

   (a) *The class PRIM of primitive recursive functions consists of the basic functions and is closed under composition and primitive recursion.*

   (b) *The class PRIM(μ) consists of the basic functions and is closed under composition, primitive recursion, and the μ operator.*

   (c) *The class PRIM(SUS) consists of the basic functions and is closed under composition, primitive recursion, the μ operator, and the Suslin operator.*

Before we are going to introduce indices for the mentioned classes of functions, we will define certain primitive recursive functions such as the coding of sequence numbers and characteristic functions.
The reader should be familiar with primitive recursive functions and relations. We will never prove that a certain function or relation is primitive recursive. A good book about recursion theory is Hinman [7], for example.

Definition 1.1.2 *The following primitive recursive functions will be relevant in the sequel:*

$$\begin{aligned} x \dot{-} y &:= \begin{cases} x - y & \textit{if } y < x \\ 0 & \textit{otherwise} \end{cases} \\ min(x,y) &:= x \dot{-} (x \dot{-} y) \;=\; \min\{x,y\} \\ p(x) &:= x\textit{-th prime number starting with } p(0) = 2 \end{aligned}$$

Now we are ready to introduce a primitive recursive coding of sequence numbers. We chose this kind of coding, because it is very easy to understand and the projections are also primitive recursive.

Definition 1.1.3 *For all natural numbers $n > 0$, $x_0, \ldots, x_{n-1}$ we define an n-ary primitive recursive function ν^n from $\mathbb{N}^n$ to $\mathbb{N}$ by*

$$\nu^n(x_0, \ldots, x_{n-1}) \;:=\; \prod_{k<n} p(k)^{x_k+1}$$

and we define the coding of sequence numbers as follows:

$$\begin{array}{rcl} \langle\rangle & := & 1 \\ \langle x_0, \ldots, x_{n-1}\rangle & := & \nu^n(x_0, \ldots, x_{n-1}) \end{array}$$

In addition, we define the primitive recursive functions lh (length) and π *(projections), as well as the primitive recursive relation Seq (sequence numbers) for an easy calculating with sequence numbers.*

$$\begin{array}{rcl} lh(s) & := & \min\{x \leq s \mid p(x) \not| \, s\} \\ \pi(s,k) & := & \min\{x \leq s \mid p(k)^{x+2} \not| \, s\} \\ s \in Seq & :\Leftrightarrow & s = \prod_{k<lh(s)} p(k)^{\pi(s,k)+1} \end{array}$$

In the sequel we will abbreviate $\pi(s,k)$ *by* $(s)_k$ *and* $((\ldots(s)_{k_0})_{k_1}\ldots)_{k_{m-1}}$ *by* $(s)_{k_0,\ldots,k_{m-1}}$ *for every natural number* $m > 0$*.*

Primitive recursive functions and relations can easily be translated into each other.

Definition 1.1.4 *For each n-ary primitive recursive relation* $\mathcal{R}$ *we define its n-ary primitive recursive characteristic function* $ch_{\mathcal{R}}$ *by:*

$$ch_{\mathcal{R}}(\vec{x}) \; := \; \begin{cases} 0 & \text{if } \mathcal{R}(\vec{x}) \\ 1 & \text{otherwise} \end{cases}$$

for all natural numbers $n > 0$, x_0, $\ldots$, x_{n-1}.
Conversely, given an n-ary primitive recursive function $\mathcal{F}$*, its graph* $Gr_{\mathcal{F}}$ *is an* $(n+1)$*-ary primitive recursive relation,*

$$Gr_{\mathcal{F}} \; := \; \{(\vec{x}, y) \mid \vec{x}, y \in \mathbb{N} \text{ and } \mathcal{F}(\vec{x}) = y\}$$

for every natural number n.

Now we are going to define indices for functions of the classes $PRIM$, $PRIM(\mu)$, and $PRIM(SUS)$. We start with the indices for the primitive recursive functions.

Definition 1.1.5 *Prim is the set of indices for the primitive recursive functions and is inductively defined as follows:*

$$
\begin{aligned}
& s \in Prim \;\Leftrightarrow \\
& s \in Seq \wedge [[(s)_0 = 0 \wedge lh(s) = 2 \wedge (s)_1 = 1] \vee \\
& [(s)_0 = 1 \wedge lh(s) = 3] \vee [(s)_0 = 2 \wedge lh(s) = 3 \wedge (s)_1 > (s)_2] \vee \\
& [(s)_0 = 3 \wedge lh(s) = (s)_{2,1} + 3 \wedge (s)_2 \in Prim \wedge (s)_{2,1} > 0 \wedge \\
& \quad (\forall k < (s)_{2,1})((s)_{k+3} \in Prim \wedge (s)_{k+3,1} = (s)_1)] \vee \\
& [(s)_0 = 4 \wedge lh(s) = 4 \wedge (s)_2 \in Prim \wedge (s)_3 \in Prim \wedge \\
& \quad (s)_1 = (s)_{2,1} + 1 \wedge (s)_{3,1} = (s)_1 + 1]]
\end{aligned}
$$

Note that *Prim* is a primitive recursive set. We have to define which functions are coded with the various indices.

Definition 1.1.6 *Let $s \in Prim$ be an index of a function of PRIM. The function Φ_s is defined by induction on the build-up of s as follows:*

$$
\begin{aligned}
\Phi_{\langle 0,1\rangle} &:= \mathcal{S} \\
\Phi_{\langle 1,n,m\rangle} &:= Cs^n_m \\
\Phi_{\langle 2,n,k\rangle} &:= Pr^n_k \\
\Phi_{\langle 3,n,f,g_0,\ldots,g_{m-1}\rangle} &:= Comp^n(\Phi_f, \Phi_{g_0}, \ldots, \Phi_{g_{m-1}}) \\
\Phi_{\langle 4,n+1,f,g\rangle} &:= Rec^{n+1}(\Phi_f, \Phi_g)
\end{aligned}
$$

We often write $[s]$ instead of Φ_s.

The primitive recursion evaluation function *PrimEv* is calculating the function Φ_s, if s is a unary index of *Prim*.

Definition 1.1.7 *PrimEv is the following recursive function on $\mathbb{N}^2$:*

$$
PrimEv(x,y) \;:=\; \begin{cases} \Phi_x(y) & \textit{if } x \in Prim \wedge (x)_1 = 1 \\ 0 & \textit{otherwise} \end{cases}
$$

Now we introduce the indices for every function of $PRIM(\mu)$.

Definition 1.1.8 *$\mu Prim$ is the set of indices for the primitive recursive functions plus the non-constructive μ operator and is inductively defined as follows:*

$$
\begin{aligned}
& s \in \mu Prim \;\; \Leftrightarrow \\
& s \in Seq \wedge [[(s)_0 = 0 \wedge lh(s) = 2 \wedge (s)_1 = 1] \vee \\
& [(s)_0 = 1 \wedge lh(s) = 3] \vee [(s)_0 = 2 \wedge lh(s) = 3 \wedge (s)_1 > (s)_2] \vee \\
& [(s)_0 = 3 \wedge lh(s) = (s)_{2,1} + 3 \wedge (s)_2 \in \mu Prim \wedge (s)_{2,1} > 0 \wedge \\
& \quad (\forall k < (s)_{2,1})((s)_{k+3} \in \mu Prim \wedge (s)_{k+3,1} = (s)_1)] \vee \\
& [(s)_0 = 4 \wedge lh(s) = 4 \wedge (s)_2 \in \mu Prim \wedge (s)_3 \in \mu Prim \wedge \\
& \quad (s)_1 = (s)_{2,1} + 1 \wedge (s)_{3,1} = (s)_1 + 1] \vee \\
& [(s)_0 = 5 \wedge lh(s) = 3 \wedge (s)_2 \in \mu Prim \wedge (s)_{2,1} = (s)_1 + 1]]
\end{aligned}
$$

Note that $\mu Prim$ is a primitive recursive set and $Prim \subseteq \mu Prim$. We have to define which function of $PRIM(\mu)$ is coded with an arbitrary index s of $\mu Prim$.

Definition 1.1.9 *Let $s \in \mu Prim$ be an index of a function of $PRIM(\mu)$. The function Ψ_s is defined by induction on the build-up of s as follows:*

$$
\begin{aligned}
\Psi_{\langle 0,1 \rangle} &:= \mathcal{S} \\
\Psi_{\langle 1,n,m \rangle} &:= Cs^n_m \\
\Psi_{\langle 2,n,k \rangle} &:= Pr^n_k \\
\Psi_{\langle 3,n,f,g_0,\ldots,g_{m-1} \rangle} &:= Comp^n(\Psi_f, \Psi_{g_0}, \ldots, \Psi_{g_{m-1}}) \\
\Psi_{\langle 4,n+1,f,g \rangle} &:= Rec^{n+1}(\Psi_f, \Psi_g) \\
\Psi_{\langle 5,n,f \rangle} &:= Zero^n(\Psi_f)
\end{aligned}
$$

We also write $[s]$ for Ψ_s if it is clear that we talk about $PRIM(\mu)$.

The evaluation function $\mu PrimEv$ is calculating the function Ψ_s, if s is a unary index of $\mu Prim$.

Definition 1.1.10 *$\mu PrimEv$ is the following function on $\mathbb{N}^2$:*

$$
\mu PrimEv(x, y) \;\; := \;\; \begin{cases} \Psi_x(y) & \textit{if } x \in \mu Prim \wedge (x)_1 = 1 \\ 0 & \textit{otherwise} \end{cases}
$$

Now we introduce the indices for every function of $PRIM(SUS)$.

Definition 1.1.11 *$SusPrim$ is the set of indices for the primitive recursive functions with the non-constructive μ operator plus the indices for the Suslin operator E_1 and is inductively defined as follows:*

$$\begin{aligned}
&s \in \mathit{SusPrim} \;\;\Leftrightarrow \\
&s \in \mathit{Seq} \wedge [[(s)_0 = 0 \wedge \mathit{lh}(s) = 2 \wedge (s)_1 = 1] \vee \\
&[(s)_0 = 1 \wedge \mathit{lh}(s) = 3] \vee [(s)_0 = 2 \wedge \mathit{lh}(s) = 3 \wedge (s)_1 > (s)_2] \vee \\
&[(s)_0 = 3 \wedge \mathit{lh}(s) = (s)_{2,1} + 3 \wedge (s)_2 \in \mathit{SusPrim} \wedge (s)_{2,1} > 0 \wedge \\
&\quad (\forall k < (s)_{2,1})((s)_{k+3} \in \mathit{SusPrim} \wedge (s)_{k+3,1} = (s)_1)] \vee \\
&[(s)_0 = 4 \wedge \mathit{lh}(s) = 4 \wedge (s)_2 \in \mathit{SusPrim} \wedge (s)_3 \in \mathit{SusPrim} \wedge \\
&\quad (s)_1 = (s)_{2,1} + 1 \wedge (s)_{3,1} = (s)_1 + 1] \vee \\
&[(s)_0 = 5 \wedge \mathit{lh}(s) = 3 \wedge (s)_2 \in \mathit{SusPrim} \wedge (s)_{2,1} = (s)_1 + 1] \vee \\
&[(s)_0 = 6 \wedge \mathit{lh}(s) = 3 \wedge (s)_2 \in \mathit{SusPrim} \wedge (s)_{2,1} = (s)_1 + 2]]
\end{aligned}$$

Note that $\mathit{SusPrim}$ is a primitive recursive set and $\mu\mathit{Prim} \subseteq \mathit{SusPrim}$. We have to define which function of $\mathit{PRIM}(\mathit{SUS})$ is coded with an arbitrary index s of $\mathit{SusPrim}$.

Definition 1.1.12 *Let $s \in \mathit{SusPrim}$ be an index of an arbitrary function of $\mathit{PRIM}(\mathit{SUS})$. The function Θ_s is defined by induction on the build-up of s as follows:*

$$\begin{aligned}
\Theta_{\langle 0,1\rangle} &:= \mathcal{S} \\
\Theta_{\langle 1,n,m\rangle} &:= Cs^n_m \\
\Theta_{\langle 2,n,k\rangle} &:= Pr^n_k \\
\Theta_{\langle 3,n,f,g_0,\ldots,g_{m-1}\rangle} &:= \mathit{Comp}^n(\Theta_f, \Theta_{g_0}, \ldots, \Theta_{g_{m-1}}) \\
\Theta_{\langle 4,n+1,f,g\rangle} &:= \mathit{Rec}^{n+1}(\Theta_f, \Theta_g) \\
\Theta_{\langle 5,n,f\rangle} &:= \mathit{Zero}^n(\Theta_f) \\
\Theta_{\langle 6,n,f\rangle} &:= \mathit{Sus}^n(\Theta_f)
\end{aligned}$$

We again use the notation $[s]$ for Θ_s if it is clear that we talk about $\mathit{PRIM}(\mathit{SUS})$.

The evaluation function $\mathit{SusPrimEv}$ is calculating the function Θ_s, if s is a unary index of $\mathit{SusPrim}$.

Definition 1.1.13 *$\mathit{SusPrimEv}$ is the following function on $\mathbb{N}^2$:*

$$\mathit{SusPrimEv}(x,y) \;:=\; \begin{cases} \Theta_x(y) & \text{if } x \in \mathit{SusPrim} \wedge (x)_1 = 1 \\ 0 & \text{otherwise} \end{cases}$$

For every index s of $\mathit{SusPrim}$ we can define the Level $\mathit{Lev}(s)$. The Level of an index s is the number how many times s is nested.

Definition 1.1.14 *Let $s \in SusPrim$ be an index of an arbitrary function of $PRIM(SUS)$. The primitive recursive function $Lev(s)$ is defined as follows:*

$$Lev(s) \quad := \quad \begin{cases} 0 & \text{if } (s)_0 \leq 2 \\ \max\{Lev((s)_2), \ldots, Lev((s)_{lh(s) \dot{-} 1})\} + 1 & \text{if } (s)_0 \geq 3 \end{cases}$$

Note that if s is an element of Prim or μPrim, then $Lev(s)$ is also defined and has the same meaning.

We will define models of PRON, $\mathsf{PRON}(\boldsymbol{\mu})$, and $\mathsf{PRON}(\mathsf{SUS})$ where we only have indices of unary functions. Therefore, we need to prove the existence of some auxiliary functions which change the arity of an index in the desired way.

Lemma 1.1.15 *Let $n > 1$, $x_0, \ldots, x_n$ be arbitrary natural numbers. Further, let $e \in SusPrim$ be an n-ary index and $f \in SusPrim$ be a unary index. Then there exist primitive recursive functions $\cdot'$, $\widetilde{\cdot}$, and $\overline{\overline{\cdot}}$ with the following properties:*

$$\begin{aligned} [\,e'\,](\langle \ldots \langle\langle x_0, x_1\rangle, x_2\rangle, \ldots, x_{n-1}\rangle) &= [e](x_0, \ldots, x_{n-1}) \\ [\,\widetilde{f}\,](x_0, x_1) &= [f](\langle x_0, x_1\rangle) \\ [\,\overline{\overline{f}}\,](x_0, x_1, x_2) &= [f](\langle\langle x_0, x_1\rangle, x_2\rangle) \end{aligned}$$

Proof For all $k < n$ let d^n_k be the index of the primitive recursive function $\langle \ldots \langle\langle x_0, x_1\rangle, x_2\rangle, \ldots, x_{n-1}\rangle \to x_k$. In addition, let c_2 be the index of the primitive recursive function $(x, y) \to \langle x, y\rangle$. Then there are primitive recursive functions $\cdot'$, $\widetilde{\cdot}$, and $\overline{\overline{\cdot}}$ so that:

$$\begin{aligned} e' &= \langle 3, 1, e, d^n_0, \ldots, d^n_{n \dot{-} 1}\rangle \\ \widetilde{f} &= \langle 3, 2, f, \langle 3, 2, c_2, \langle 2, 2, 0\rangle, \langle 2, 2, 1\rangle\rangle\rangle \\ \overline{\overline{f}} &= \langle 3, 3, f, \langle 3, 3, c_2, \langle 3, 3, c_2, \langle 2, 3, 0\rangle, \langle 2, 3, 1\rangle\rangle, \langle 2, 3, 2\rangle\rangle\rangle \end{aligned}$$

It is an easy recursion theoretic exercise to prove that the three functions have the mentioned properties. $\square$

Note that we can apply the three functions $\cdot'$, $\widetilde{\cdot}$, and $\overline{\overline{\cdot}}$ to elements e, f of *Prim* or μ*Prim* because $Prim \subseteq \mu Prim \subseteq SusPrim$. In this case, the abbreviation $[\cdot]$ stands for *PrimEv* or μ*PrimEv*, respectively.

1.2 Arithmetics

In this section we will define the first order language $\mathcal{L}_1$ and the second order language $\mathcal{L}_2$. We are working with a form of second order arithmetic with

set and function variables. The reason for this choice is that theorem 3.3.1 (Π^1_1 normal forms) is most suitable for our purposes.
Furthermore, we will introduce the two outstanding systems of first order arithmetic, namely primitive recursive arithmetic PRA and Peano arithmetic PA. The subsystems of second order arithmetic we are going to introduce are ACA_0, ACA, Π^1_1-CA_0, and Π^1_1-CA.

Definition 1.2.1 *The language $\mathcal{L}_1$ is a first order language with:*

1. *countably many number variables u, v, w, x, y, z, ... (possibly with subscripts).*

2. *an n-ary function symbol for every n-ary primitive recursive function and every natural number n. The function symbols are denoted by $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, ... (possibly with subscripts). We sometimes write c, c_0, c_1, c_2, ... for the 0-ary function symbols we call constants.*

3. *an n-ary relation symbol for every n-ary primitive recursive relation and every natural number $n > 0$. We mostly write $\mathcal{R}$, $\mathcal{R}_0$, $\mathcal{R}_1$, $\mathcal{R}_2$, ... for the relation symbols.*

4. *the logical symbols $\neg$ (negation), $\vee$ (disjunction), and $\exists$ (existential quantifier), as well as useful symbols like (, [,),], and the comma for a better readability of expressions.*

The second order language $\mathcal{L}_2$ contains the language $\mathcal{L}_1$ plus:

5. *countably many set variables U, V, W, X, Y, Z, ... (possibly with subscripts).*

6. *countably many function variables F, G, H, ... (possibly with subscripts).*

7. *the element relation $\in$ between natural numbers and sets of natural numbers.*

In particular, we use the constant 0, the unary function symbol $\mathcal{S}$ for the successor function, the binary (infix) relation symbol $=$ for the equality relation, and the binary (infix) relation symbol $<$ for the less relation. Very often we also write the same expression for a primitive recursive function (relation) and the associated function (relation) symbol.
There is only a basic symbol for equality between numbers. Therefore, we have to define two additional kinds of equality.

Definition 1.2.2 *Equality between sets of numbers and functions is defined as follows:*

$$\begin{array}{rcl} U = V & := & (\forall x)(x \in U \leftrightarrow x \in V) \\ F = G & := & (\forall x)(F(x) = G(x)) \end{array}$$

The number terms and formulas of $\mathcal{L}_1$ and $\mathcal{L}_2$ are defined in the usual way.

Definition 1.2.3 *The number terms r, s, t, ... (possibly with subscripts) of $\mathcal{L}_1$ ($\mathcal{L}_2$) are inductively generated as follows:*

1. *The number variables and constants are number terms of $\mathcal{L}_1$ ($\mathcal{L}_2$).*

2. *If t_0, ..., t_{n-1} are number terms of $\mathcal{L}_1$ ($\mathcal{L}_2$), then so also is $\mathcal{F}(t_0, \ldots, t_{n-1})$ for every n-ary function symbol $\mathcal{F}$ and all natural numbers $n > 0$.*

3. *If t is a number term of $\mathcal{L}_2$, then so also is $F(t)$ for every function variable F.*

The $\mathcal{L}_1$ ($\mathcal{L}_2$) formulas A, B, C, ... (possibly with subscripts) are inductively generated as follows:

1. *If t_0, ..., t_{n-1} are number terms of $\mathcal{L}_1$ ($\mathcal{L}_2$), then $\mathcal{R}(t_0, \ldots, t_{n-1})$ is an atomic $\mathcal{L}_1$ ($\mathcal{L}_2$) formula for every n-ary relation symbol $\mathcal{R}$ and all natural numbers $n > 0$.*

2. *If t is a number term of $\mathcal{L}_2$, then $(t \in U)$ is an atomic $\mathcal{L}_2$ formula for every set variable U.*

3. *If A and B are $\mathcal{L}_1$ ($\mathcal{L}_2$) formulas, then so also are $\neg A$, $(A \vee B)$, and $(\exists x)A$.*

4. *If A and B are $\mathcal{L}_2$ formulas, then so also are $(\exists X)A$, and $(\exists F)A$.*

Free and bound variables are defined as usual. We use the notation $FV(A)$ for the set of all free number variables which have no bound appearance in the formula A. A sentence is a formula with no free variables. In addition, we will use the notation $A[\vec{U}, \vec{F}, \vec{v}]$ to express that a formula A contains the free variables U_0, ..., U_{k-1}, F_0, ..., F_{l-1}, v_0, ..., v_{m-1} for all natural numbers k, l, and m.

Definition 1.2.4 (Substitution) *Let $n > 0$ be a natural number, t, s_0, ..., s_{n-1} be arbitrary terms and A be an arbitrary formula. Further, let u_0, ..., u_{n-1} be variables in $FV(A)$. We write $t[s_0, \ldots, s_{n-1}/u_0, \ldots, u_{n-1}]$*

and $A[s_0, \ldots, s_{n-1}/u_0, \ldots, u_{n-1}]$ to express that we simultaneously replace in t and A every appearance of u_k by s_k for all $k < n$. Sometimes we will abbreviate this notation by $t[\vec{s}/\vec{u}]$, $A[\vec{s}/\vec{u}]$, or $A(\vec{s})$.

Definition 1.2.5 *Let A and B be arbitrary $\mathcal{L}_1$ ($\mathcal{L}_2$) formulas, and let t be an arbitrary number term of $\mathcal{L}_1$ ($\mathcal{L}_2$). Then we have the following abbreviations:*

$$\begin{array}{rcl}
(A \land B) & := & \neg(\neg A \lor \neg B) \\
(A \to B) & := & (\neg A \lor B) \\
(A \leftrightarrow B) & := & (A \to B \land B \to A) \\
(\forall x)A & := & \neg(\exists x)(\neg A) \\
(\exists! x)A & := & (\exists x)[A \land (\forall y)(A[y/x] \to y = x)] \\
(\exists x < t)A & := & (\exists x)(x < t \land A) \\
(\forall x < t)A & := & (\forall x)(x < t \to A)
\end{array}$$

Now, let A be an arbitrary $\mathcal{L}_2$ formula. We additionally define the following abbreviations:

$$\begin{array}{rcl}
(\exists x \in U)A & := & (\exists x)(x \in U \land A) \\
(\forall x \in U)A & := & (\forall x)(x \in U \to A) \\
(\forall X)A & := & \neg(\exists X)(\neg A) \\
(\exists! X)A & := & (\exists X)[A \land (\forall Y)(A[Y/X] \to Y = X)] \\
(\forall F)A & := & \neg(\exists F)(\neg A) \\
(\exists! F)A & := & (\exists F)[A \land (\forall G)(A[G/F] \to G = F)]
\end{array}$$

We will be interested in systems of first and second order arithmetic, which are based on induction and comprehension principles for classes of formulas capturing levels of the arithmetic and the analytic hierarchy. We therefore have to build two hierarchies for formulas, whose level depends on the number of alternating quantifiers.

Definition 1.2.6 *First, we define QF to be the collection of all quantifier-free $\mathcal{L}_1$ formulas. Further, for all natural numbers n we generate the collections of Σ_n and Π_n formulas of $\mathcal{L}_1$ according to the arithmetic hierarchy by induction on n as follows:*

1. *The Σ_0 and Π_0 formulas of $\mathcal{L}_1$ are those $\mathcal{L}_1$ formulas which contain only bounded number quantifiers.*

2. *The Σ_{n+1} formulas of $\mathcal{L}_1$ comprise all Σ_n and Π_n formulas of $\mathcal{L}_1$ as well as all formulas of the form $(\exists x)A$ so that A is a Π_n formula of $\mathcal{L}_1$.*

3. *The* Π_{n+1} *formulas of* $\mathcal{L}_1$ *comprise all* Σ_n *and* Π_n *formulas of* $\mathcal{L}_1$ *as well as all formulas of the form* $(\forall x)A$ *so that* A *is a* Σ_n *formula of* $\mathcal{L}_1$.

Π_∞ *denotes the collection of all* $\mathcal{L}_1$ *formulas. Now we generate for all natural numbers* n *the collections of* Σ^1_n *and* Π^1_n *formulas of* $\mathcal{L}_2$ *according to the analytic hierarchy by induction on* n *as follows:*

1. *The* Σ^1_0 *and* Π^1_0 *formulas of* $\mathcal{L}_2$ *are those* $\mathcal{L}_2$ *formulas which contain only number quantifiers, i.e., which contain no set quantifier and no function quantifier. They are called the arithmetic* $\mathcal{L}_2$ *formulas.*

2. *The* Σ^1_{n+1} *formulas of* $\mathcal{L}_2$ *comprise all* Σ^1_n *and* Π^1_n *formulas of* $\mathcal{L}_2$ *as well as all formulas of the form* $(\exists X)A$ *so that* A *is a* Π^1_n *formula of* $\mathcal{L}_2$.

3. *The* Π^1_{n+1} *formulas of* $\mathcal{L}_2$ *comprise all* Σ^1_n *and* Π^1_n *formulas of* $\mathcal{L}_2$ *as well as all formulas of the form* $(\forall X)A$ *so that* A *is a* Σ^1_n *formula of* $\mathcal{L}_2$.

Π^1_∞ *denotes the collection of all* $\mathcal{L}_2$ *formulas.*

Our rudimentary theory in arithmetic is PRA^-, that is PRA without the induction axiom. In the definitions of our arithmetic theories PRA^- is always the initial theory.

Definition 1.2.7 (PRA^-) *The axioms and rules of inference of* PRA^- *are divided into the following four groups:*

i. Logical axioms

(1) *every instance of an axiom of classical propositional logic*

(2) $x = x$

(3) $x = y \land x = z \to y = z$

(4) $x_0 = y_0 \land \ldots \land x_{n-1} = y_{n-1} \to \mathcal{F}(x_0, \ldots, x_{n-1}) = \mathcal{F}(y_0, \ldots, y_{n-1})$
for every n*-ary function symbol* $\mathcal{F}$ *and all natural numbers* $n > 0$

(5) $x_0 = y_0 \land \ldots \land x_{n-1} = y_{n-1} \to (\mathcal{R}(x_0, \ldots, x_{n-1}) \leftrightarrow \mathcal{R}(y_0, \ldots, y_{n-1}))$
for every n*-ary relation symbol* $\mathcal{R}$ *and all natural numbers* $n > 0$

(6) $A[t/x] \to (\exists x)A$ *for every formula* A *and every term* t

ii. Number theoretic axioms

(7) $\neg(\mathcal{S}(x) = 0)$

(8) $\mathcal{S}(x) = \mathcal{S}(y) \rightarrow x = y$

(9) $\neg(x < 0)$

(10) $x < \mathcal{S}(y) \rightarrow (x < y \vee x = y)$

iii. Defining equations

(11) $Cs^n_m(x_0, \ldots, x_{n-1}) = m$
for all natural numbers n, m

(12) $Pr^n_k(x_0, \ldots, x_{n-1}) = x_k$
for all natural numbers $n > 0$ and $k < n$

(13) $Comp^n(\mathcal{F}, \mathcal{G}_0, \ldots, \mathcal{G}_{m-1})(x_0, \ldots, x_{n-1}) =$
$\mathcal{F}(\mathcal{G}_0(x_0, \ldots, x_{n-1}), \ldots, \mathcal{G}_{m-1}(x_0, \ldots, x_{n-1}))$
for all m-ary function symbols $\mathcal{F}$, all n-ary function symbols $\mathcal{G}_0$, ..., $\mathcal{G}_{m-1}$, and all natural numbers n and $m > 0$

(14) $Rec^{n+1}(\mathcal{F}, \mathcal{G})(x_0, \ldots, x_{n-1}, 0) = \mathcal{F}(x_0, \ldots, x_{n-1})$
for all n-ary function symbols $\mathcal{F}$, all $(n+2)$-ary function symbols $\mathcal{G}$, and all natural numbers n

(15) $Rec^{n+1}(\mathcal{F}, \mathcal{G})(x_0, \ldots, x_{n-1}, \mathcal{S}(y)) =$
$\mathcal{G}(x_0, \ldots, x_{n-1}, y, Rec^{n+1}(\mathcal{F}, \mathcal{G})(x_0, \ldots, x_{n-1}, y))$
for all n-ary function symbols $\mathcal{F}$, all $(n+2)$-ary function symbols $\mathcal{G}$, and all natural numbers n

iv. Rules of inference

(MP) $$\frac{A \qquad A \rightarrow B}{B}$$
for every formula A and B

($\exists$) $$\frac{A[u/x] \rightarrow B}{(\exists x)A \rightarrow B}$$
for every formula A and B so that u does not occur in B

We have to define several schemes and axioms before we can introduce the mentioned theories.

Definition 1.2.8 *Let $\mathcal{K}$ be a class of formulas. We define the following schemes and axioms:*

v. Induction scheme

($\mathcal{K}$-$\mathsf{I_N}$) $A[0/u] \wedge (\forall x)(A[x/u] \rightarrow A[S(x)/u]) \rightarrow (\forall x)A[x/u]$
for all formulas A in $\mathcal{K}$

vi. Set induction for $\mathcal{L}_2$:

$(\mathsf{IA_N})$ $\quad (\forall X)[0 \in X \wedge (\forall y)(y \in X \rightarrow S(y) \in X) \rightarrow (\forall y)(y \in X)]$

vii. Graph principle:

(GP) $\quad (\forall X)[(\forall y)(\exists! z)(\langle y, z\rangle \in X) \rightarrow (\exists F)(\forall y)(\langle y, F(y)\rangle \in X)]$

viii. Comprehension scheme

$(\mathcal{K}\text{-}\mathsf{CA})$ $\quad (\exists X)(\forall y)(y \in X \leftrightarrow A[y/u])$
for all formulas A in $\mathcal{K}$

Definition 1.2.9 *We will work with the following $\mathcal{L}_1$ theories in the sequel:*

$$\begin{aligned} \mathsf{PRA} &:= \mathsf{PRA}^- + (\mathsf{QF\text{-}I_N}) \\ \mathsf{PA} &:= \mathsf{PRA}^- + (\Pi_\infty\text{-}\mathsf{I_N}) \end{aligned}$$

Further, we define the following $\mathcal{L}_2$ theories:

$$\begin{aligned} \mathsf{ACA_0} &:= \mathsf{PRA}^- + (\mathsf{GP}) + (\mathsf{IA_N}) + (\Pi^1_0\text{-}\mathsf{CA}) \\ \mathsf{ACA} &:= \mathsf{PRA}^- + (\mathsf{GP}) + (\Pi^1_\infty\text{-}\mathsf{I_N}) + (\Pi^1_0\text{-}\mathsf{CA}) \\ \Pi^1_1\text{-}\mathsf{CA_0} &:= \mathsf{PRA}^- + (\mathsf{GP}) + (\mathsf{IA_N}) + (\Pi^1_1\text{-}\mathsf{CA}) \\ \Pi^1_1\text{-}\mathsf{CA} &:= \mathsf{PRA}^- + (\mathsf{GP}) + (\Pi^1_\infty\text{-}\mathsf{I_N}) + (\Pi^1_1\text{-}\mathsf{CA}) \end{aligned}$$

The relation $\equiv$ denotes the usual notion of proof-theoretic equivalence as it is defined in Feferman [4]. In this thesis, two theories are defined to have the same proof-theoretic strength, if they prove (at least) the same Π_2 sentences.

1.3 Applicative theories

It is the purpose of this section to introduce the basic applicative framework as well as the precise axiomatizations of set and formula induction, the non-constructive μ operator, and the Suslin operator E_1. Further, we will define additional axioms like the axioms of totality and extensionality.

Definition 1.3.1 *The language $\mathcal{L}$ of our applicative theories is a first order language of partial terms with:*

1. *individual variables a, b, c, d, f, g, h, u, v, w, x, y, z, ... (possibly with subscripts).*

2. *individual constants i, k, a_2, b_2 (combinators), p_0, p_1 (unpairing), $\mathsf{0}$ (zero), $\mathsf{s_N}$ (numerical successor), $\mathsf{p_N}$ (numerical predecessor), $\mathsf{d_N}$ (definition by numerical cases), and r (primitive recursion).*

3. *two unary relation symbols* N *(natural numbers) and* $\downarrow$ *(defined), and one binary relation symbol* $=$ *(equality).*

4. *two binary function symbols* $\odot$ *(partial term application) and* $<>$ *(pairing).*

5. *the logical symbols* $\neg$ *(negation),* $\vee$ *(disjunction), and* $\exists$ *(existential quantifier), as well as useful symbols like* (, [,),], *and the comma for a better readability of expressions.*

We define individual terms of $\mathcal{L}$ as we define terms of every other first order language.

Definition 1.3.2 *The individual* $\mathcal{L}$ *terms* r, s, t, $\ldots$ *(possibly with subscripts) are inductively generated as follows:*

1. *The individual variables and individual constants are individual terms.*

2. *If* s *and* t *are individual terms, then so also are* $\odot(s,t)$ *and* $<>(s,t)$.

In the following we will always abbreviate $\odot(s,t)$ simply by (st), st or sometimes also $s(t)$; the context will always ensure that no confusion arises. We further adopt the convention of association to the left so that $t_0 t_1 t_2 \ldots t_{n-1}$ stands for $(\ldots((t_0 t_1) t_2) \ldots t_{n-1})$.

Definition 1.3.3 *We define general* n*-tupling of individual terms by induction on* $n > 1$ *as follows:*

$$\begin{array}{rcl} <t_0> & := & t_0 \\ <t_0, t_1> & := & <>(t_0, t_1) \\ <t_0, \ldots, t_n> & := & <<t_0, \ldots, t_{n-1}>, t_n> \end{array}$$

Note that $<t_0, t_1> = <<t_0>, t_1>$.

We define $\mathcal{L}$ formulas in our applicative framework the same way as in every other first order language.

Definition 1.3.4 *The* $\mathcal{L}$ *formulas* A, B, C, D, $\ldots$ *(possibly with subscripts) are inductively generated as follows:*

1. *If* s *and* t *are individual terms of* $\mathcal{L}$*, then the atomic formulas* $\mathsf{N}(t)$, $t\downarrow$, *and* $(s = t)$ *are* $\mathcal{L}$ *formulas.*

2. *If* A *and* B *are* $\mathcal{L}$ *formulas, then so also are* $\neg A$, $(A \vee B)$, *and* $(\exists x)A$.

Free variables and substitution like $t[\vec{s}/\vec{u}]$ and $A[\vec{s}/\vec{u}]$ are defined the same way as in section 1.2. We will also use the abbreviations $A[\vec{u}]$ and $A(\vec{s})$. We do not have any abbreviation for the substitution in terms like $t[s/u]$, because the expression $t(s)$ is already reserved for term application.

Definition 1.3.5 *For every natural number n we define additional formulas as follows:*

$$
\begin{array}{rcl}
(A \wedge B) & := & \neg(\neg A \vee \neg B) \\
(A \rightarrow B) & := & (\neg A \vee B) \\
(A \leftrightarrow B) & := & (A \rightarrow B \wedge B \rightarrow A) \\
(\exists! x)A & := & (\exists x)[A \wedge (\forall y)(A[y/x] \rightarrow y = x)] \\
(\exists \vec{x})A & := & (\exists x_0) \ldots (\exists x_{n-1})A \\
(\forall \vec{x})A & := & \neg(\exists \vec{x})(\neg A)
\end{array}
$$

Our applicative theories are based on partial term application. Hence, it is not guaranteed that terms have a value, and $t\downarrow$ is read as "t is defined" or "t has a value".

Definition 1.3.6 *We define the partial equality relation $\simeq$ and the negation of equality $\neq$ as follows:*

$$
\begin{array}{rcl}
(s \simeq t) & := & s\downarrow \vee\, t\downarrow \rightarrow (s = t) \\
(s \neq t) & := & s\downarrow \wedge\, t\downarrow \wedge \neg(s = t)
\end{array}
$$

If we want to describe n-ary total functions of natural numbers, we have to write very long formulas. For this reason, it is more comfortable to define additional abbreviations which help us writing shorter formulas.

Definition 1.3.7 *Let $n > 0$ and m be natural numbers and $t_0, \ldots, t_{m-1}$ be arbitrary $\mathcal{L}$ terms. Then we use the following abbreviations concerning the predicate N:*

$$
\begin{array}{rcl}
\vec{t} \in \mathsf{N} & := & \mathsf{N}(t_0) \wedge \ldots \wedge \mathsf{N}(t_{m-1}) \\
(\exists \vec{x} \in \mathsf{N})A & := & (\exists \vec{x})(\vec{x} \in \mathsf{N} \wedge A) \\
(\forall \vec{x} \in \mathsf{N})A & := & (\forall \vec{x})(\vec{x} \in \mathsf{N} \rightarrow A) \\
(\exists! x \in \mathsf{N})A & := & (\exists x \in \mathsf{N})[A \wedge (\forall y \in \mathsf{N})(A[y/x] \rightarrow y = x)] \\
\vec{t} \in (\mathsf{N}^n \rightarrow \mathsf{N}) & := & (\forall \vec{x} \in \mathsf{N})(t_0 {<} x_0, \ldots, x_{n-1} {>} \in \mathsf{N} \wedge \ldots \wedge \\
 & & t_{m-1} {<} x_0, \ldots, x_{n-1} {>} \in \mathsf{N})
\end{array}
$$

We will always write $\vec{t} \in (\mathsf{N} \rightarrow \mathsf{N})$ instead of $\vec{t} \in (\mathsf{N}^1 \rightarrow \mathsf{N})$.

Sometimes we need the existence of $\mathcal{L}$ terms which represent natural numbers. It is a usual habit to define numerals for this reason.

Definition 1.3.8 *For every natural number m we define the numeral $\overline{m}$ by induction on m as follows:*

$$\begin{array}{rcl} \overline{0} & := & \mathsf{0} \\ \overline{(m+1)} & := & \mathsf{s_N}\overline{m} \end{array}$$

We mostly write $\mathsf{0}$, $\mathsf{1}$ *for the numerals* $\overline{0}$, $\overline{1}$, *respectively. Of course, every numeral is a closed $\mathcal{L}$ term.*

Let us define the notion of a subset of N. Sets of natural numbers are represented via their characteristic functions which are total on N.

Definition 1.3.9 *We define a subset t of N with the intention that a natural number x belongs to the set $t \in \mathcal{P}(\mathsf{N})$ if and only if $tx = \mathsf{0}$.*

$$t \in \mathcal{P}(\mathsf{N}) \quad := \quad (\forall x \in \mathsf{N})(tx = \mathsf{0} \vee tx = \mathsf{1})$$

If n is an arbitrary natural number and $t_0, \ldots, t_{n-1}$ are arbitrary $\mathcal{L}$ terms, then we sometimes write $\vec{t} \in \mathcal{P}(\mathsf{N})$ for $t_0 \in \mathcal{P}(\mathsf{N}) \wedge \ldots \wedge t_{n-1} \in \mathcal{P}(\mathsf{N})$.

Now we are going to introduce the theory PRON of primitive recursive operations and numbers which has been treated in Schlüter [12] as $\mathsf{PEA}^+ + (\mathsf{r})$. Its underlying logic is the *classical logic of partial terms* due to Beeson [1], which is also described in Feferman, Jäger, and Strahm [6] and corresponds to E^+ logic with strictness and equality of Troelstra and Van Dalen [16].

Definition 1.3.10 (PRON **I)** *The logical axioms and the rules of inference of* PRON *are divided into the following four groups:*

A. Propositional and quantifier axioms

(a) *every instance of an axiom of classical propositional logic*

(b) $A[t/x] \wedge t\downarrow \rightarrow (\exists x)A$ *for every term t*

B. Definedness axioms

(c) $r\downarrow$, *provided that r is a variable or a constant*

(d) $(st)\downarrow \rightarrow s\downarrow \wedge t\downarrow$

(e) $<s,t>\downarrow \rightarrow s\downarrow \wedge t\downarrow$

(f) $s = t \to s\downarrow \wedge t\downarrow$

(g) $\mathsf{N}(t) \to t\downarrow$

C. Equality axioms.

(h) $r = r$, *provided that* r *is a variable or a constant*

(i) $s_0 = t_0 \wedge \ldots \wedge s_{n-1} = t_{n-1} \wedge A(s_0, \ldots, s_{n-1}) \to A(t_0 \ldots, t_{n-1})$
for all atomic formulas $A[\vec{v}]$ *and every natural number* n

D. Rules of inference

(MP) $$\frac{A \qquad A \to B}{B}$$
for every formula A *and* B

($\exists$) $$\frac{A[u/x] \to B}{(\exists x)A \to B}$$
for every formula A *and* B *so that* u *does not occur in* B

Definition 1.3.11 (PRON II) *The non-logical axioms of* PRON *are divided into the following five groups:*

I. Partial enumerative algebra

(1) $\mathsf{i}a = a$

(2) $\mathsf{k}ab = a$

(3) $\mathsf{a}_2{<}a, b{>}\downarrow \wedge\ \mathsf{a}_2{<}a, b{>}c \simeq {<}ac, bc{>}$

(4) $\mathsf{b}_2{<}a, b{>}\downarrow \wedge\ \mathsf{b}_2{<}a, b{>}c \simeq a(bc)$

II. Pairing and projection

(5) $\mathsf{p}_0{<}a, b{>} = a \wedge \mathsf{p}_1{<}a, b{>} = b$

III. Natural numbers

(6) $0 \in \mathsf{N} \wedge (\forall x \in \mathsf{N})(\mathsf{s_N} x \in \mathsf{N})$

(7) $(\forall x \in \mathsf{N})(\mathsf{s_N} x \neq 0 \wedge \mathsf{p_N}(\mathsf{s_N} x) = x)$

(8) $(\forall x \in \mathsf{N})(x \neq 0 \to \mathsf{p_N} x \in \mathsf{N} \wedge \mathsf{s_N}(\mathsf{p_N} x) = x)$

IV. Definition by numerical cases

(9) $c \in \mathsf{N} \wedge d \in \mathsf{N} \wedge c = d \to \mathsf{d_N}{<}a, b, c, d{>} = a$

(10) $c \in \mathsf{N} \wedge d \in \mathsf{N} \wedge c \neq d \rightarrow \mathsf{d_N}{<}a, b, c, d{>} = b$

V. Primitive recursion

(11) $\mathsf{r}{<}f, g{>}\downarrow \wedge\, \mathsf{r}{<}f, g{>}{<}a, \mathsf{0}{>} \simeq fa$

(12) $b \in \mathsf{N} \rightarrow \mathsf{r}{<}f, g{>}{<}a, \mathsf{s_N}b{>} \simeq g{<}a, b, \mathsf{r}{<}f, g{>}{<}a, b{>}{>}$

In the sequel we will write PRON^- *for the theory* PRON *without the axioms* (11) *and* (12) *about primitive recursion.*

Let us now turn to the two type 2 functionals which will be relevant in the sequel. For this purpose we add the two new constants $\boldsymbol{\mu}$ and $\mathsf{E_1}$ to our applicative framework. If $n > 0$ and $f \in (\mathsf{N}^{n+1} \rightarrow \mathsf{N})$, the non-constructive μ operator is checking the existence of a zero in f. If $n > 0$ and $f \in (\mathsf{N}^{n+2} \rightarrow \mathsf{N})$ represents an $(n+2)$-ary relation on N, the Suslin operator E_1 is testing for the wellfoundedness of f.

Definition 1.3.12 *We define the following axioms about the additional constants* $\boldsymbol{\mu}$ *and* $\mathsf{E_1}$*:*

VI. The non-constructive $\boldsymbol{\mu}$ operator

$(\boldsymbol{\mu}.1)$ $\quad (\forall x \in \mathsf{N})(f{<}a, x{>} \in \mathsf{N}) \leftrightarrow \boldsymbol{\mu} fa \in \mathsf{N}$

$(\boldsymbol{\mu}.2)$ $\quad (\forall x \in \mathsf{N})(f{<}a, x{>} \in \mathsf{N}) \rightarrow$
$\quad [(\exists x \in \mathsf{N})(f{<}a, x{>} = \mathsf{0}) \rightarrow f{<}a, \boldsymbol{\mu} fa{>} = \mathsf{0}]$

VII. The Suslin operator $\mathsf{E_1}$

$(\mathsf{E_1}.1)$ $\quad (\forall x, y \in \mathsf{N})(f{<}a, x, y{>} \in \mathsf{N}) \leftrightarrow \mathsf{E_1} fa \in \mathsf{N}$

$(\mathsf{E_1}.2)$ $\quad (\forall x, y \in \mathsf{N})(f{<}a, x, y{>} \in \mathsf{N}) \rightarrow [(\exists g)[g \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge$
$\quad (\forall x \in \mathsf{N})(f{<}a, g(\mathsf{s_N}x), gx{>} = \mathsf{0})] \leftrightarrow \mathsf{E_1} fa = \mathsf{0}]$

We are ready to define the two extensions $\mathsf{PRON}(\boldsymbol{\mu})$ and $\mathsf{PRON(SUS)}$ of PRON. We include the non-constructive μ operator in our basic axiomatic framework for the Suslin operator, because we do not know if the axioms for $\boldsymbol{\mu}$ are derivable from the axioms of $\mathsf{E_1}$.

Definition 1.3.13 *We define* $\mathsf{PRON}(\boldsymbol{\mu})$ *to be the* $\mathcal{L}$ *theory* PRON *extended by the axioms about* $\boldsymbol{\mu}$*, and we let* $\mathsf{PRON(SUS)}$ *denote the* $\mathcal{L}$ *theory which extends* PRON *by the axioms about* $\boldsymbol{\mu}$ *and* $\mathsf{E_1}$.

$$\begin{aligned} \mathsf{PRON}(\boldsymbol{\mu}) &:= \mathsf{PRON} + (\boldsymbol{\mu}.1) + (\boldsymbol{\mu}.2) \\ \mathsf{PRON(SUS)} &:= \mathsf{PRON}(\boldsymbol{\mu}) + (\mathsf{E_1}.1) + (\mathsf{E_1}.2) \end{aligned}$$

Note that the axioms of $\boldsymbol{\mu}$ and E_1 differ from the axiomatizations in BON as we can see in Feferman and Jäger [5] and in Jäger and Strahm [8]. The reason is that there came up some problems by generating a model with indices.

Definition 1.3.14 *We define two different induction schemes as follows:*

VIII. Set induction on N

$(\mathsf{S\text{-}I_N})$ $\quad t \in \mathcal{P}(\mathsf{N}) \wedge t\mathsf{0} = \mathsf{0} \wedge (\forall x \in \mathsf{N})(tx = \mathsf{0} \rightarrow t(\mathsf{s_N}x) = \mathsf{0}) \rightarrow$
$\quad (\forall x \in \mathsf{N})(tx = \mathsf{0})$
for every $\mathcal{L}$ term t

IX. Formula induction on N

$(\mathcal{L}\text{-}\mathsf{I_N})$ $\quad A[\mathsf{0}/u] \wedge (\forall x \in \mathsf{N})(A[x/u] \rightarrow A[\mathsf{s_N}x/u]) \rightarrow (\forall x \in \mathsf{N})A[x/u]$
for all $\mathcal{L}$ formulas A

We can define additional axioms which sometimes will extend our applicative theories containing PRON. In this cases we have to add the new constants $\mathsf{d_V}$, $\mathsf{c_=}$, $\mathsf{c_N}$, and $\mathsf{i_N}$ to $\mathcal{L}$.

Definition 1.3.15 *We define the following additional axioms:*

X. Various additional axioms

(Tot) $\quad (\forall x, y)(xy\downarrow)$

(Ext) $\quad (\forall f, g)[(\forall x)(fx \simeq gx) \rightarrow f = g]$

(Nat) $\quad (\forall x)\mathsf{N}(x)$

$(\mathsf{D_V})$ $\quad (c = d \rightarrow \mathsf{d_V}{<}a, b, c, d{>} = a) \wedge (c \neq d \rightarrow \mathsf{d_V}{<}a, b, c, d{>} = b)$

$(\mathsf{Ch_=})$ $\quad (x = y \rightarrow \mathsf{c_=}{<}x, y{>} = \mathsf{0}) \wedge (x \neq y \rightarrow \mathsf{c_=}{<}x, y{>} = \mathsf{1})$

$(\mathsf{Ch_N})$ $\quad (x \in \mathsf{N} \rightarrow \mathsf{c_N}x = \mathsf{0}) \wedge (x \notin \mathsf{N} \rightarrow \mathsf{c_N}x = \mathsf{1})$

$(\mathsf{Inj_N})$ $\quad (\forall x)(\mathsf{i_N}x \in \mathsf{N}) \wedge (\forall x, y)(x \neq y \rightarrow \mathsf{i_N}x \neq \mathsf{i_N}y)$

Finally, it will be useful to have abbreviations for the projections on tuples of terms.

Definition 1.3.16 *For all natural numbers $n > 0$ and $k < n$ we define the projections of an arbitrary $\mathcal{L}$ term t as follows:*

$$(t)^n_k \quad := \quad \begin{cases} \mathsf{p_0}(\mathsf{p_0}(\ldots \mathsf{p_0}(\mathsf{p_0}t)\ldots)) & \textit{if } k = 0 \\ \mathsf{p_1}(\mathsf{p_0}(\mathsf{p_0}(\ldots \mathsf{p_0}(\mathsf{p_0}t)\ldots))) & \textit{otherwise} \end{cases}$$

where the constant $\mathsf{p_0}$ appears $n - (k + 1)$ times.

We have to check if the abbreviations $(\cdot)^n_k$ for projections of terms are correctly defined.

Lemma 1.3.17 *For all natural numbers $n > 0$ and $k < n$ we have*

$$\mathsf{PRON} \;\vdash\; (<x_0, \ldots, x_{n-1}>)^n_k = x_k$$

PROOF We only need the axiom (5) of PRON and definition 1.3.3. □

2 Basic consequences and models

2.1 Restricted lambda abstraction

In BON we can express full λ abstraction because of the constant $\hat{\mathsf{s}}$. This is not possible in PRON, but there is a way to define a restricted form of lambda abstraction. Therefore, we have to introduce a property of variables.

Definition 2.1.1 *A variable x occurs in argument position in a term t, if one of the following conditions is true:*

1. *t is a variable or a constant*
2. *t is the term $<r, s>$ and x occurs in argument position in r and s*
3. *t is the term (rs) and x occurs in argument position in s, but does not occur in r, i.e. $x \notin FV(r)$*

Now we are able to define restricted lambda abstraction provided by the definition of PRON.

Definition 2.1.2 *Let x occur in argument position in t. The term $(\lambda^* x.t)$ is defined by induction on the definition of t by:*

$$(\lambda^* x.t) \quad := \quad \begin{cases} \mathsf{i} & \text{if } t \text{ is the variable } x \\ \mathsf{k}\, t & \text{if } t \text{ is a variable } y \neq x \\ & \quad \text{or a constant} \\ \mathsf{a}_2 <(\lambda^* x.r), (\lambda^* x.s)> & \text{if } t \text{ is the term } <r, s> \\ \mathsf{b}_2 <r, (\lambda^* x.s)> & \text{if } t \text{ is the term } (rs) \end{cases}$$

There are two fundamental properties of λ^* abstraction we have to mention.

Lemma 2.1.3 *Let x occur in argument position in t.*

1. $\mathsf{PRON} \vdash (\lambda^* x.t)\downarrow$
2. $FV(\lambda^* x.t) = FV(t) \setminus \{x\}$

PROOF These assertions are both easily proved by induction on the definition of t. □

It is not always easy to see, if the variable y is in argument position in $(\lambda^* x.t)$. For that reason we have to define another property of variables.

Definition 2.1.4 *By induction on the build-up of t we define that $Arg_t^y(x)$ holds if:*

1. *t is a variable or a constant.*

2. *t is the term $<r, s>$ and $Arg_r^y(x)$ and $Arg_s^y(x)$ hold.*

3. *t is the term (rs), $Arg_s^y(x)$ holds, and y is in argument position in r.*

Now we have the desired tool to decide, if y is in argument position in $(\lambda^* x.t)$.

Lemma 2.1.5 *$Arg_t^y(x)$ holds $\Leftrightarrow$ y is in argument position in $(\lambda^* x.t)$.*

Proof We can prove this lemma by induction on the definition of t:

1. t is the variable x: $Arg_t^y(x)$ holds and y is in argument position in $(\lambda^* x.t) = \mathsf{i}$.

2. t is a variable $z \neq x$ or a constant: $Arg_t^y(x)$ holds and y is in argument position in $(\lambda^* x.t) = \mathsf{k}t$ (no problem if $t = y$).

3. t is the term $< r, s >$: $Arg_t^y(x)$ holds $\Leftrightarrow$ $Arg_r^y(x)$ and $Arg_s^y(x)$ hold. By induction hypothesis this is the fact $\Leftrightarrow$ y is in argument position in $(\lambda^* x.r)$ and in $(\lambda^* x.s)$ $\Leftrightarrow$ y is in argument position in $(\lambda^* x.t) = \mathsf{a}_2 <(\lambda^* x.r), (\lambda^* x.s)>$.

4. t is the term (rs): $Arg_t^y(x)$ holds $\Leftrightarrow$ $Arg_s^y(x)$ holds and y is in argument position in r. By induction hypothesis this is the fact $\Leftrightarrow$ y is in argument position in r and in $(\lambda^* x.s)$ $\Leftrightarrow$ y is in argument position in $\lambda^* x.t = \mathsf{b}_2 <r, (\lambda^* x.s)>$. □

The following lemma shows that $Arg_t^y(x)$ does not imply that y is in argument position in t.

Lemma 2.1.6 *Let x occur in argument position in t.*

1. *If $y \neq x$ occurs in argument position in t, then y occurs in argument position in $(\lambda^* x.t)$.*

2. *If $y \neq x$ occurs in argument position in $(\lambda^* x.t)$, then y needs not occur in argument position in t.*

Proof The first assertion is easily proved by induction on the definition of t. For the second assertion let t be the term $\mathsf{p}_0 y <x, \mathsf{p}_1 yx>$. We know that x is in argument position in t, y is not in argument position in t, but y is in argument position in $(\lambda^* x.t)$, because $Arg_t^y(x)$ holds. □

The next theorem is the reason why we have defined λ^* abstraction this way. It has almost the same application as λ abstraction in BON.

Theorem 2.1.7 *Let x occur in argument position in t. Then we can prove the following assertions:*

1. $\mathsf{PRON} \vdash (\lambda^* x.t)x \simeq t$

2. $\mathsf{PRON} \vdash s\downarrow \rightarrow (\lambda^* x.t)s \simeq t[s/x]$

3. *If $Arg_t^y(x)$ holds and $x \notin FV(r)$ then*
   $\mathsf{PRON} \vdash r\downarrow \wedge s\downarrow \rightarrow (\lambda^* y.(\lambda^* x.t))rs \simeq t[r/y][s/x]$

PROOF These assertions are proved by induction on the definition of t. We will prove the third one, the first two are a straightforward exercise. Suppose that $r\downarrow$ and $s\downarrow$.

1. If t is the variable x then $(\lambda^* x.t) = \mathsf{i}$ and $(\lambda^* y.(\lambda^* x.t))rs \simeq (\lambda^* y.\mathsf{i})rs \simeq \mathsf{i}[r/y]s \simeq \mathsf{i}s \simeq s \simeq x[r/y][s/x] \simeq t[r/y][s/x]$

2. If t is the variable y then $(\lambda^* x.t) = \mathsf{k}y$ and $(\lambda^* y.(\lambda^* x.t))rs \simeq (\lambda^* y.\mathsf{k}y)rs \simeq (\mathsf{k}y)[r/y]s \simeq \mathsf{k}rs \simeq r \simeq y[r/y][s/x] \simeq t[r/y][s/x]$

3. If t is a variable z so that $x \neq z \neq y$ or t is a constant then $(\lambda^* x.t) = \mathsf{k}t$ and $(\lambda^* y.(\lambda^* x.t))rs \simeq (\lambda^* y.\mathsf{k}t)rs \simeq (\mathsf{k}t)[r/y]s \simeq \mathsf{k}ts \simeq t \simeq t[r/y][s/x]$

4. If t is the term $<t_0, t_1>$ then $(\lambda^* x.t) = \mathsf{a}_2<(\lambda^* x.t_0), (\lambda^* x.t_1)>$ and
   $(\lambda^* y.(\lambda^* x.t))rs \simeq (\lambda^* y.\mathsf{a}_2<(\lambda^* x.t_0), (\lambda^* x.t_1)>)rs \simeq$
   $(\mathsf{a}_2<(\lambda^* x.t_0), (\lambda^* x.t_1)>)[r/y]s \simeq \mathsf{a}_2<(\lambda^* x.t_0)[r/y], (\lambda^* x.t_1)[r/y]>s \simeq$
   $\mathsf{a}_2<(\lambda^* y.(\lambda^* x.t_0))r, (\lambda^* y.(\lambda^* x.t_1))r>s \simeq$
   $<(\lambda^* y.(\lambda^* x.t_0))rs, (\lambda^* y.(\lambda^* x.t_1))rs> \simeq <t_0[r/y][s/x], t_1[r/y][s/x]> \simeq$
   $<t_0, t_1>[r/y][s/x] \simeq t[r/y][s/x]$

5. If t is the term $(t_0 t_1)$ then $(\lambda^* x.t) = \mathsf{b}_2<t_0, (\lambda^* x.t_1)>$ and
   $(\lambda^* y.(\lambda^* x.t))rs \simeq (\lambda^* y.\mathsf{b}_2<t_0, (\lambda^* x.t_1)>)rs \simeq$
   $(\mathsf{b}_2<t_0, (\lambda^* x.t_1)>)[r/y]s \simeq \mathsf{b}_2<t_0[r/y], (\lambda^* x.t_1)[r/y]>s \simeq$
   $\mathsf{b}_2<t_0[r/y], (\lambda^* y.(\lambda^* x.t_1))r>s \simeq$
   $t_0[r/y]((\lambda^* y.(\lambda^* x.t_1))rs) \simeq t_0[r/y](t_1[r/y][s/x]) \simeq$
   $(t_0 t_1)[r/y][s/x]$ because $x \notin FV(t_0)$, so we have $t[r/y][s/x]$ □

Examples 2.1.8 *We give two examples of forbidden λ^* abstraction and an example of an $\mathcal{L}$ term where λ^* abstraction is possible.*

1. *We are not allowed to define a term like $(\lambda^* x.(\lambda^* y.(\lambda^* z.xz(yz))))$. Otherwise we would be able to construct the constant* $\hat{\mathsf{s}}$ *of* BON, *which must not be possible.*

2. *We are not allowed to define a term like* $(\lambda^* x.(\mathsf{p}_0 x)(\mathsf{p}_1 x))$. *Otherwise we would be able to construct the primitive recursion evaluation function PrimEv, which must not be possible, because PrimEv is recursive and not primitive recursive.*

3. *Let us remember the proof of lemma 2.1.6. We are allowed to define the term* $r := (\lambda^* y.(\lambda^* x.\mathsf{p}_0 y{<}x, \mathsf{p}_1 yx{>}))$. *It is a nice result to have an* $\mathcal{L}$ *term* r *with the property* $r{<}a,b{>}c = a{<}c, bc{>}$, *which follows from theorem 2.1.7.*

We are now able to prove a restricted version of the recursion theorem.

Theorem 2.1.9 *There exists an* $\mathcal{L}$ *term* rec *so that:*

$$\mathsf{PRON} \;\vdash\; \mathsf{rec} f{\downarrow} \wedge \mathsf{rec} f x \simeq f{<}\mathsf{rec} f, x{>}$$

Proof Let s be the term $s := (\lambda^* y.(\lambda^* z.\mathsf{p}_0 y{<}\mathsf{p}_1 y, z{>}))$ with the intention that $s{<}a,b{>}c \simeq a{<}b,c{>}$. Further, we define t to be the term $(\lambda^* f.(\lambda^* u.f{<}s{<}\mathsf{p}_0 u, \mathsf{p}_0 u{>}, \mathsf{p}_1 u{>}))$. Let now rec be the term $(\lambda^* f.s{<}tf, tf{>})$. Note that $\mathsf{rec} f \simeq s{<}tf,tf{>}$ and $s{<}tf,tf{>}{\downarrow}$ by lemma 2.1.3, so we know that $recf{\downarrow}$. We can easily check that:

$$\begin{aligned}
\mathsf{rec} f x &\simeq s{<}tf,tf{>}x \simeq tf{<}tf,x{>} \\
&\simeq (\lambda^* u.f{<}s{<}\mathsf{p}_0 u, \mathsf{p}_0 u{>}, \mathsf{p}_1 u{>}){<}tf,x{>} \simeq f{<}s{<}tf,tf{>},x{>} \\
&\simeq f{<}\mathsf{rec} f, x{>}
\end{aligned}$$

Surely, we have to check if it is allowed to define the terms s, t and rec because the variables for the abstraction have to be in argument position: z is in argument position in $\mathsf{p}_0 y{<}\mathsf{p}_1 y, z{>}$ and $Arg^{y}_{\mathsf{p}_0 y<\mathsf{p}_1 y,z>}(z)$ holds. The variable u is in argument position in $f{<}s{<}\mathsf{p}_0 u, \mathsf{p}_0 u{>}, \mathsf{p}_1 u{>}$ and $Arg^{f}_{f<s<\mathsf{p}_0 u,\mathsf{p}_0 u>,\mathsf{p}_1 u>}(u)$ holds. Finally, f is in argument position in $s{<}tf,tf{>}$. □

The existence of the constant r with its axioms (11) and (12) allows us to introduce $\mathcal{L}$ terms for all (names of) primitive recursive functions, so that the defining equations and totality of these terms are derivable in PRON. In the following inductive definition the corresponding terms are specified in detail.

Definition 2.1.10 *For each (description of a) primitive recursive function* $\mathcal{F}$ *we define a closed* $\mathcal{L}$ *term* $\mathsf{pr}_{\mathcal{F}}$ *by induction on the build-up of* $\mathcal{F}$ *as follows:*

1. *If* $\mathcal{F} = \mathcal{S}$ *then* $\mathsf{pr}_{\mathcal{F}} := (\lambda^* z.\mathsf{s}_{\mathsf{N}} z)$

2. *If* $\mathcal{F} = Cs^n_m$ *then* $\mathsf{pr}_{\mathcal{F}} := \begin{cases} \overline{m} & \text{if } n = 0 \\ (\lambda^* z.\overline{m}) & \text{if } n > 0 \end{cases}$

3. *If* $\mathcal{F} = Pr^n_k$ *then* $\mathsf{pr}_{\mathcal{F}} := (\lambda^* z.(z)^n_k)$

4. *If* $\mathcal{F} = Comp^n(\mathcal{G}, \mathcal{H}_0, \ldots, \mathcal{H}_{m-1})$ *then*

$$\mathsf{pr}_{\mathcal{F}} := \begin{cases} \mathsf{pr}_{\mathcal{G}}{<}\mathsf{pr}_{\mathcal{H}_0}, \ldots, \mathsf{pr}_{\mathcal{H}_{m-1}}{>} & \textit{if } n = 0 \\ (\lambda^* z.\mathsf{pr}_{\mathcal{G}}{<}\mathsf{pr}_{\mathcal{H}_0} z, \ldots, \mathsf{pr}_{\mathcal{H}_{m-1}} z{>}) & \textit{if } n > 0 \end{cases}$$

5. *If* $\mathcal{F} = Rec^{n+1}(\mathcal{G}, \mathcal{H})$ *then*

$$\mathsf{pr}_{\mathcal{F}} := \begin{cases} (\lambda^* z.\mathsf{r}{<}(\lambda^* y.\mathsf{pr}_{\mathcal{G}}), (\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>}){>}{<}0, z{>}) & \textit{if } n = 0 \\ (\lambda^* z.\mathsf{r}{<}\mathsf{pr}_{\mathcal{G}}, \mathsf{pr}_{\mathcal{H}}{>} z) & \textit{if } n > 0 \end{cases}$$

For the proof of the following theorem we need theorem 2.1.7, the axioms (11) and (12) of PRON concerning r, and the induction scheme ($\mathcal{L}$-$\mathsf{I_N}$) for induction on N.

Theorem 2.1.11 (Primitive recursive functions) *Let* $\mathcal{F}$ *be an arbitrary primitive recursive function. The defining equations of* $\mathcal{F}$ *are derivable in* $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N})$ *for* $\mathsf{pr}_{\mathcal{F}}$*. Moreover, we can prove the following assertions:*

1. *If* $\mathcal{F}$ *is a constant, then* $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N}) \vdash \mathsf{pr}_{\mathcal{F}} \in \mathsf{N}$.

2. *If the arity of* $\mathcal{F}$ *is* $m > 0$*, then* $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N}) \vdash \mathsf{pr}_{\mathcal{F}} \in (\mathsf{N}^m \to \mathsf{N})$

Proof We can prove this theorem by induction on the build-up of $\mathcal{F}$. The most interesting point is the case $\mathcal{F} = Rec^{n+1}(\mathcal{G}, \mathcal{H})$. Hereby, we have to distinct the two cases $n = 0$ and $n > 0$.
Let $n = 0$. By induction hypothesis there exist the terms $\mathsf{pr}_{\mathcal{G}} \in \mathsf{N}$ and $\mathsf{pr}_{\mathcal{H}} \in (\mathsf{N}^2 \to \mathsf{N})$. We show that $\mathsf{pr}_{\mathcal{F}} \in (\mathsf{N} \to \mathsf{N})$ by induction on x_0.

$x_0 = 0$**:** $\mathsf{pr}_{\mathcal{F}} x_0 \simeq (\lambda^* z.\mathsf{r}{<}(\lambda^* y.\mathsf{pr}_{\mathcal{G}}), (\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>}){>}{<}0, z{>})0 \simeq$
$\mathsf{r}{<}(\lambda^* y.\mathsf{pr}_{\mathcal{G}}), (\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>}){>}{<}0, 0{>} \simeq$
$(\lambda^* y.\mathsf{pr}_{\mathcal{G}})0 \simeq \mathsf{pr}_{\mathcal{G}} \in \mathsf{N}$

$x_0 \to \mathsf{s_N} x_0$**:** $\mathsf{pr}_{\mathcal{F}}(\mathsf{s_N} x_0) \simeq$
$(\lambda^* z.\mathsf{r}{<}(\lambda^* y.\mathsf{pr}_{\mathcal{G}}), (\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>}){>}{<}0, z{>})(\mathsf{s_N} x_0) \simeq$
$\mathsf{r}{<}(\lambda^* y.\mathsf{pr}_{\mathcal{G}}), (\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>}){>}{<}0, \mathsf{s_N} x_0{>} \simeq$

$(\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>})$
${<}0, x_0, \mathsf{r}{<}(\lambda^* y.\mathsf{pr}_{\mathcal{G}}), (\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>}){>}{<}0, x_0{>}{>} \simeq$

$\mathsf{pr}_{\mathcal{H}}{<}x_0, \mathsf{r}{<}(\lambda^* y.\mathsf{pr}_{\mathcal{G}}), (\lambda^* y.\mathsf{pr}_{\mathcal{H}}{<}(y)^3_1, (y)^3_2{>}){>}{<}0, x_0{>}{>} \simeq$
$\mathsf{pr}_{\mathcal{H}}{<}x_0, \mathsf{pr}_{\mathcal{F}} x_0{>} \in \mathsf{N}$ because $\mathsf{pr}_{\mathcal{F}} x_0 \in \mathsf{N}$ by induction hypothesis.

Now let $n > 0$. By induction hypothesis there exist the terms $\mathsf{pr}_{\mathcal{G}} \in (\mathsf{N}^n \to \mathsf{N})$ and $\mathsf{pr}_{\mathcal{H}} \in (\mathsf{N}^{n+2} \to \mathsf{N})$. We show that $\mathsf{pr}_{\mathcal{F}} \in (\mathsf{N}^{n+1} \to \mathsf{N})$ by induction on x_n. In the following, let $x_0, \ldots, x_{n-1}$ be in N.

$x_n = 0$: $\mathsf{pr}_{\mathcal{F}}{<}x_0, \ldots, x_n{>} \simeq (\lambda^* z.\mathsf{r}{<}\mathsf{pr}_{\mathcal{G}}, \mathsf{pr}_{\mathcal{H}}{>}z){<}x_0, \ldots, x_{n-1}, 0{>} \simeq$
$\mathsf{r}{<}\mathsf{pr}_{\mathcal{G}}, \mathsf{pr}_{\mathcal{H}}{>}{<\!\!<}x_0, \ldots, x_{n-1}{>}, 0{>} \simeq \mathsf{pr}_{\mathcal{G}}{<}x_0, \ldots, x_{n-1}{>} \in \mathsf{N}$

$x_n \to \mathsf{s_N} x_n$: $\mathsf{pr}_{\mathcal{F}}{<}x_0, \ldots, x_{n-1}, \mathsf{s_N} x_n{>} \simeq$
$(\lambda^* z.\mathsf{r}{<}\mathsf{pr}_{\mathcal{G}}, \mathsf{pr}_{\mathcal{H}}{>}z){<}x_0, \ldots, x_{n-1}, \mathsf{s_N} x_n{>} \simeq$
$\mathsf{r}{<}\mathsf{pr}_{\mathcal{G}}, \mathsf{pr}_{\mathcal{H}}{>}{<\!\!<}x_0, \ldots, x_{n-1}{>}, \mathsf{s_N} x_n{>} \simeq$
$\mathsf{pr}_{\mathcal{H}}{<\!\!<}x_0, \ldots, x_{n-1}{>}, x_n, \mathsf{r}{<}\mathsf{pr}_{\mathcal{G}}, \mathsf{pr}_{\mathcal{H}}{>}{<\!\!<}x_0, \ldots, x_{n-1}{>}, x_n{>\!\!>} \simeq$
$\mathsf{pr}_{\mathcal{H}}{<}x_0, \ldots, x_n, \mathsf{r}{<}\mathsf{pr}_{\mathcal{G}}, \mathsf{pr}_{\mathcal{H}}{>}{<}x_0, \ldots, x_n{>\!\!>} \simeq$
$\mathsf{pr}_{\mathcal{H}}{<}x_0, \ldots, x_n, \mathsf{pr}_{\mathcal{F}}{<}x_0, \ldots, x_n{>\!\!>} \in \mathsf{N}$ because
$\mathsf{pr}_{\mathcal{F}}{<}x_0, \ldots, x_n{>} \in \mathsf{N}$ by induction hypothesis. □

Let us make an example to see how the term $\mathsf{pr}_{\mathcal{F}}$ behaves.

Example 2.1.12 *The index for the addition + is*
$\langle 4, 2, \langle 2, 1, 0\rangle, \langle 3, 3, \langle 0, 1\rangle, \langle 2, 3, 2\rangle\rangle\rangle$.
The addition is the primitive recursive function
$Rec^2(Pr^1_0, Comp^3(\mathcal{S}, Pr^3_2))$.
The $\mathcal{L}$ *term* pr_+ *is defined as*
$(\lambda^* x.r{<}(\lambda^* y.(y)^1_0), (\lambda^* y.(\lambda^* z.\mathsf{s_N} z)((\lambda^* z.(z)^3_2)y)){>}x)$.
By the definitions 1.3.16 and 2.1.2 we have
$\mathsf{pr}_+ = \mathsf{b}_2{<}\mathsf{r}{<}\mathsf{i}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}, \mathsf{i}{>\!\!>\!\!>}, \mathsf{i}{>}$.
Now we can easily verify that $\mathsf{pr}_+{<}\overline{4}, 1{>} = \overline{5}$:

$$\mathsf{b}_2{<}\mathsf{r}{<}\mathsf{i}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}, \mathsf{i}{>\!\!>\!\!>}, \mathsf{i}{>}{<}\overline{4}, 1{>} \simeq$$
$$\mathsf{r}{<}\mathsf{i}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}, \mathsf{i}{>\!\!>\!\!>}(\mathsf{i}{<}\overline{4}, 1{>}) \simeq$$

$$\mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}, \mathsf{i}{>\!\!>}$$
$${<}\overline{4}, 0, \mathsf{r}{<}\mathsf{i}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}, \mathsf{i}{>\!\!>\!\!>}{<}\overline{4}, 0{>\!\!>} \simeq$$

$$\mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}, \mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}, \mathsf{i}{>\!\!>}{<}\overline{4}, 0, \mathsf{i}\overline{4}{>}$$
$$\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}(\mathsf{b}_2{<}\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}, \mathsf{i}{>}{<}\overline{4}, 0, \overline{4}{>}) \simeq$$
$$\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}(\mathsf{b}_2{<}\mathsf{p}_1, \mathsf{i}{>}(\mathsf{i}{<}\overline{4}, 0, \overline{4}{>})) \simeq$$
$$\mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}(\mathsf{p}_1(\mathsf{i}{<}\overline{4}, 0, \overline{4}{>})) \simeq \mathsf{b}_2{<}\mathsf{s_N}, \mathsf{i}{>}\overline{4} \simeq \mathsf{s_N}(\mathsf{i}\overline{4}) \simeq \overline{5}$$

2.2 A recursion theoretic model

In this section we will define the intended recursion theoretic model $\mathcal{M}$ of PRON. The universe of this model is the set $\mathbb{N}$ of all natural numbers and the function symbol $\odot$ is interpreted as the primitive recursion evaluation function *PrimEv*. The constants are interpreted as indices of unary primitive recursive functions. For a better understanding how the model works, we define the interpretations of the various symbols in the order of their difficulty.

Definition 2.2.1 $\mathcal{M} := (|\mathcal{M}|,\ \mathsf{N}^{\mathcal{M}},\ \downarrow^{\mathcal{M}},\ =^{\mathcal{M}},\ \odot^{\mathcal{M}},\ <>^{\mathcal{M}},\ \mathsf{0}^{\mathcal{M}},\ \mathsf{s_N}^{\mathcal{M}},\ \mathsf{i}^{\mathcal{M}},\ \mathsf{k}^{\mathcal{M}},\ \mathsf{p_N}^{\mathcal{M}},\ \mathsf{p_0}^{\mathcal{M}},\ \mathsf{p_1}^{\mathcal{M}},\ \mathsf{d_N}^{\mathcal{M}},\ \mathsf{b_2}^{\mathcal{M}},\ \mathsf{a_2}^{\mathcal{M}},\ \mathsf{r}^{\mathcal{M}})$ *is defined as follows:*

$$
\begin{array}{rcl}
|\mathcal{M}| & := & \mathbb{N} \\
\mathsf{N}^{\mathcal{M}} & := & \mathbb{N} \\
\downarrow^{\mathcal{M}} & := & \mathbb{N} \\
=^{\mathcal{M}} & := & \{(x,x) \in \mathbb{N}^2 \mid x \in \mathbb{N}\} \\
\odot^{\mathcal{M}} & := & \mathit{PrimEv} \\
<>^{\mathcal{M}} & := & \textit{the primitive recursive function } (x,y) \to \langle x,y\rangle \\
\mathsf{0}^{\mathcal{M}} & := & 0 \\
\mathsf{s_N}^{\mathcal{M}} & := & \langle 0,1\rangle \\
\mathsf{i}^{\mathcal{M}} & := & \langle 2,1,0\rangle \\
\mathsf{k}^{\mathcal{M}} & := & \textit{index of the primitive recursive function } x \to \langle 1,1,x\rangle \\
\mathsf{p_N}^{\mathcal{M}} & := & \textit{index of the primitive recursive function } x \to x \dot{-} 1 \\
\mathsf{p_0}^{\mathcal{M}} & := & \textit{index of the primitive recursive function } x \to (x)_0 \\
\mathsf{p_1}^{\mathcal{M}} & := & \textit{index of the primitive recursive function } x \to (x)_1 \\
\mathsf{d_N}^{\mathcal{M}} & := & \textit{index of the primitive recursive function} \\
 & & x \to \begin{cases} (x)_{0,0,0} & \textit{if } (x)_{0,1} = (x)_1 \\ (x)_{0,0,1} & \textit{otherwise} \end{cases} \\
\mathsf{b_2}^{\mathcal{M}} & := & \langle 3,1,s_1,\mathsf{p_0}^{\mathcal{M}},\mathsf{p_1}^{\mathcal{M}}\rangle \textit{ while } s_1 \textit{ is the} \\
 & & \textit{index of the primitive recursive function} \\
 & & (x,y) \to \\
 & & \quad \begin{cases} \langle 3,1,x,y\rangle & \textit{if } x \in \mathit{Prim} \wedge (x)_1 = 1 \wedge \\ & \quad y \in \mathit{Prim} \wedge (y)_1 = 1 \\ \langle 3,1,x,\langle 1,1,0\rangle\rangle & \textit{if } x \in \mathit{Prim} \wedge (x)_1 = 1 \wedge \\ & \quad (y \notin \mathit{Prim} \vee (y)_1 \neq 1) \\ \langle 1,1,0\rangle & \textit{otherwise} \end{cases}
\end{array}
$$

$$
\begin{aligned}
\mathsf{a_2}^{\mathcal{M}} \quad := \quad & \langle 3, 1, s_2, \langle 1, 1, <>^{\mathcal{M}}\rangle, \mathsf{p_0}^{\mathcal{M}}, \mathsf{p_1}^{\mathcal{M}}\rangle \textit{ while } s_2 \textit{ is the} \\
& \textit{index of the primitive recursive function} \\
& (x, y, z) \to \\
& \quad \begin{cases}
\langle 3, 1, x, y, z\rangle & \textit{if } y \in \textit{Prim} \wedge (y)_1 = 1 \wedge \\
& \quad z \in \textit{Prim} \wedge (z)_1 = 1 \\
\langle 3, 1, x, y, \langle 1, 1, 0\rangle\rangle & \textit{if } y \in \textit{Prim} \wedge (y)_1 = 1 \wedge \\
& \quad (z \notin \textit{Prim} \vee (z)_1 \neq 1) \\
\langle 3, 1, x, \langle 1, 1, 0\rangle, z\rangle & \textit{if } (y \notin \textit{Prim} \vee (y)_1 \neq 1) \wedge \\
& \quad z \in \textit{Prim} \wedge (z)_1 = 1 \\
\langle 3, 1, x, \langle 1, 1, 0\rangle, \langle 1, 1, 0\rangle\rangle & \textit{otherwise}
\end{cases}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{r}^{\mathcal{M}} \quad := \quad & \textit{index of the primitive recursive function} \\
& x \to \\
& \quad \begin{cases}
\langle 4, 2, (x)_0, \overline{\overline{\overline{(x)_1}}}\rangle' & \textit{if } (x)_0 \in \textit{Prim} \wedge (x)_{0,1} = 1 \wedge \\
& \quad (x)_1 \in \textit{Prim} \wedge (x)_{1,1} = 1 \\
\langle 4, 2, (x)_0, \langle 1, 3, 0\rangle\rangle' & \textit{if } (x)_0 \in \textit{Prim} \wedge (x)_{0,1} = 1 \wedge \\
& \quad ((x)_1 \notin \textit{Prim} \vee (x)_{1,1} \neq 1) \\
\langle 4, 2, \langle 1, 1, 0\rangle, \overline{\overline{\overline{(x)_1}}}\rangle' & \textit{if } ((x)_0 \notin \textit{Prim} \vee (x)_{0,1} \neq 1) \wedge \\
& \quad (x)_1 \in \textit{Prim} \wedge (x)_{1,1} = 1 \\
\langle 1, 1, 0\rangle & \textit{otherwise}
\end{cases}
\end{aligned}
$$

Theorem 2.2.2 *$\mathcal{M}$ is a model of* $\mathsf{PRON} + (\mathsf{Tot}) + (\mathsf{Nat})$.

Proof The axiom (Tot) is satisfied because *PrimEv* is a total function and the axiom (Nat) is satisfied because the universe of $\mathcal{M}$ is $\mathbb{N}$. The most axioms of PRON are easy to verify. We only check the axioms about the constants $\mathsf{b_2}$, $\mathsf{a_2}$, and r.

The axiom (4) about the combinator $\mathsf{b_2}$:
Suppose that $a^{\mathcal{M}}$ and $b^{\mathcal{M}}$ are indices of unary primitive recursive functions.
$(\mathsf{b_2}<a, b>c)^{\mathcal{M}} = [[\mathsf{b_2}^{\mathcal{M}}](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle)](c^{\mathcal{M}}) =$
$[[\langle 3, 1, s_1, \mathsf{p_0}^{\mathcal{M}}, \mathsf{p_1}^{\mathcal{M}}\rangle](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle)](c^{\mathcal{M}}) =$
$[[s_1]([\mathsf{p_0}^{\mathcal{M}}](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle), [\mathsf{p_1}^{\mathcal{M}}](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle))](c^{\mathcal{M}}) = [[s_1](a^{\mathcal{M}}, b^{\mathcal{M}})](c^{\mathcal{M}}) =$
$[\langle 3, 1, a^{\mathcal{M}}, b^{\mathcal{M}}\rangle](c^{\mathcal{M}}) = [a^{\mathcal{M}}]([b^{\mathcal{M}}](c^{\mathcal{M}})) = (a(bc))^{\mathcal{M}}$

The axiom (3) about the combinator $\mathsf{a_2}$:
Suppose that $a^{\mathcal{M}}$ and $b^{\mathcal{M}}$ are indices of unary primitive recursive functions.
$(\mathsf{a_2}<a, b>c)^{\mathcal{M}} = [[\mathsf{a_2}^{\mathcal{M}}](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle)](c^{\mathcal{M}}) =$
$[[\langle 3, 1, s_2, \langle 1, 1, <>^{\mathcal{M}}\rangle, \mathsf{p_0}^{\mathcal{M}}, \mathsf{p_1}^{\mathcal{M}}\rangle](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle)](c^{\mathcal{M}}) =$
$[[s_2]([\langle 1, 1, <>^{\mathcal{M}}\rangle](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle), [\mathsf{p_0}^{\mathcal{M}}](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle), [\mathsf{p_1}^{\mathcal{M}}](\langle a^{\mathcal{M}}, b^{\mathcal{M}}\rangle))](c^{\mathcal{M}}) =$
$[[s_2](<>^{\mathcal{M}}, a^{\mathcal{M}}, b^{\mathcal{M}})](c^{\mathcal{M}}) = [\langle 3, 1, <>^{\mathcal{M}}, a^{\mathcal{M}}, b^{\mathcal{M}}\rangle](c^{\mathcal{M}}) =$
$[<>^{\mathcal{M}}]([a^{\mathcal{M}}](c^{\mathcal{M}}), [b^{\mathcal{M}}](c^{\mathcal{M}})) = \langle [a^{\mathcal{M}}](c^{\mathcal{M}}), [b^{\mathcal{M}}](c^{\mathcal{M}})\rangle = <ac, bc>^{\mathcal{M}}$

The axioms (11) and (12) about the constant r:
Suppose that $f^{\mathcal{M}}$ and $g^{\mathcal{M}}$ are indices of unary primitive recursive functions.

1. $(\mathsf{r}{<}f,g{>})^{\mathcal{M}} = \langle 4,2,f^{\mathcal{M}},\overline{\overline{g^{\mathcal{M}}}}\rangle' \in \mathbb{N}$, so $\mathsf{r}{<}f,g{>}\downarrow$ is true in $\mathcal{M}$.

2. $(\mathsf{r}{<}f,g{>}{<}a,\mathsf{0}{>})^{\mathcal{M}} =$
$[[\mathsf{r}^{\mathcal{M}}](\langle f^{\mathcal{M}},g^{\mathcal{M}}\rangle)](\langle a^{\mathcal{M}},\mathsf{0}^{\mathcal{M}}\rangle) = [\langle 4,2,f^{\mathcal{M}},\overline{\overline{g^{\mathcal{M}}}}\rangle'](\langle a^{\mathcal{M}},0\rangle) =$
$[\langle 4,2,f^{\mathcal{M}},\overline{\overline{g^{\mathcal{M}}}}\rangle](a^{\mathcal{M}},0) = [f^{\mathcal{M}}](a^{\mathcal{M}}) = (fa)^{\mathcal{M}}$

3. $(\mathsf{r}{<}f,g{>}{<}a,\mathsf{s_N}b{>})^{\mathcal{M}} = [[\mathsf{r}^{\mathcal{M}}](\langle f^{\mathcal{M}},g^{\mathcal{M}}\rangle)](\langle a^{\mathcal{M}},[\mathsf{s_N}^{\mathcal{M}}](b^{\mathcal{M}})\rangle) =$
$[\langle 4,2,f^{\mathcal{M}},\overline{\overline{g^{\mathcal{M}}}}\rangle'](\langle a^{\mathcal{M}},[\langle 0,1\rangle](b^{\mathcal{M}})\rangle) = [\langle 4,2,f^{\mathcal{M}},\overline{\overline{g^{\mathcal{M}}}}\rangle](a^{\mathcal{M}},\mathcal{S}(b^{\mathcal{M}})) =$
$[\overline{\overline{g^{\mathcal{M}}}}](a^{\mathcal{M}},b^{\mathcal{M}},[\langle 4,2,f^{\mathcal{M}},\overline{\overline{g^{\mathcal{M}}}}\rangle](a^{\mathcal{M}},b^{\mathcal{M}})) =$
$[g^{\mathcal{M}}](\langle\langle a^{\mathcal{M}},b^{\mathcal{M}}\rangle,[\langle 4,2,f^{\mathcal{M}},\overline{\overline{g^{\mathcal{M}}}}\rangle'](\langle a^{\mathcal{M}},b^{\mathcal{M}}\rangle)\rangle) =$
$[g^{\mathcal{M}}](\langle\langle a^{\mathcal{M}},b^{\mathcal{M}}\rangle,[[\mathsf{r}^{\mathcal{M}}](\langle f^{\mathcal{M}},g^{\mathcal{M}}\rangle)](\langle a^{\mathcal{M}},b^{\mathcal{M}}\rangle)\rangle) =$
$(g{<}{<}a,b{>},\mathsf{r}{<}f,g{>}{<}a,b{>}{>})^{\mathcal{M}} = (g{<}a,b,\mathsf{r}{<}f,g{>}{<}a,b{>}{>})^{\mathcal{M}}$

If at least one of the numbers $f^{\mathcal{M}}$, $g^{\mathcal{M}}$, $a^{\mathcal{M}}$, $b^{\mathcal{M}}$ is not the index of a unary primitive recursive function, then the proofs are similar and even easier. □

We have chosen the interpretations of the constants a_2 and b_2 like that, because we wanted to be able to prove that every interpretation of λ^* abstraction is an index of a unary primitive recursive function.

Lemma 2.2.3 *Let t be an individual $\mathcal{L}$ term and the variable x be in argument position in t. Then $(\lambda^* x.t)^{\mathcal{M}}$ is a unary index of Prim.*

Proof Induction on the build-up of t:

1. If t is the variable x then $(\lambda^* x.t) = \mathsf{i}$ and $\mathsf{i}^{\mathcal{M}} = \langle 2,1,0\rangle$.

2. If t is a variable $y \neq x$ or a constant then $(\lambda^* x.t) = \mathsf{k}t$ and
$(\mathsf{k}t)^{\mathcal{M}} = [\mathsf{k}^{\mathcal{M}}](t^{\mathcal{M}}) = \langle 1,1,t^{\mathcal{M}}\rangle$.

3. If t is the term ${<}r,s{>}$ then $(\lambda^* x.t) = \mathsf{a}_2{<}(\lambda^* x.r),(\lambda^* x.s){>}$ and
$(\mathsf{a}_2{<}(\lambda^* x.r),(\lambda^* x.s){>})^{\mathcal{M}} = [s_2]({<}{>}^{\mathcal{M}},(\lambda^* x.r)^{\mathcal{M}},(\lambda^* x.s)^{\mathcal{M}}) =$
$\langle 3,1,{<}{>}^{\mathcal{M}},(\lambda^* x.r)^{\mathcal{M}},(\lambda^* x.s)^{\mathcal{M}}\rangle$ because $(\lambda^* x.r)^{\mathcal{M}}$ and $(\lambda^* x.s)^{\mathcal{M}}$ are unary indices of *Prim* by induction hypothesis.

4. If t is the term rs then $(\lambda^* x.t) = \mathsf{b}_2{<}r,(\lambda^* x.s){>}$ and
$(\mathsf{b}_2{<}r,(\lambda^* x.s){>})^{\mathcal{M}} = [s_1](r^{\mathcal{M}},(\lambda^* x.s)^{\mathcal{M}})$
$$= \begin{cases} \langle 3,1,r^{\mathcal{M}},(\lambda^* x.s)^{\mathcal{M}}\rangle & \text{if } r^{\mathcal{M}} \in \mathit{Prim} \wedge (r^{\mathcal{M}})_1 = 1 \\ \langle 1,1,0\rangle & \text{otherwise} \end{cases}$$
because $(\lambda^* x.s)^{\mathcal{M}}$ is an index of a unary primitive recursive function by induction hypothesis. □

The interpretation of numerals is very natural in $\mathcal{M}$, but the interpretation of tuples of terms in $\mathcal{M}$ is iterated pairing.

Lemma 2.2.4 *Let m and $n > 1$ be natural numbers and $t_0, \ldots, t_{n-1}$ be arbitrary $\mathcal{L}$ terms. Then we can prove the following assertions:*

1. $(\overline{m})^{\mathcal{M}} = m$
2. $<t_0, \ldots, t_{n-1}>^{\mathcal{M}} = \langle \ldots \langle \langle (t_0)^{\mathcal{M}}, (t_1)^{\mathcal{M}} \rangle, (t_2)^{\mathcal{M}} \rangle, \ldots, (t_{n-1})^{\mathcal{M}} \rangle$

PROOF The reader should be able to prove this lemma as an exercise. □

Note that $(<t_0, \ldots, t_{n-1}>^{\mathcal{M}})_k = (t_k)^{\mathcal{M}}$ if and only if $n = 2$ because of the different coding of sequence numbers.

Theorem 2.2.5 *Let n be an arbitrary natural number, $\mathcal{F}$ be an n-ary primitive recursive function, and $t_0, \ldots, t_{n-1}$ be arbitrary $\mathcal{L}$ terms. Then the interpretation of the $\mathcal{L}$ term $\mathsf{pr}_{\mathcal{F}}$ behaves the same way as the function $\mathcal{F}$:*

$$
\begin{array}{ll}
n = 0: & (\mathsf{pr}_{\mathcal{F}})^{\mathcal{M}} = \mathcal{F} \\
n > 0: & (\mathsf{pr}_{\mathcal{F}}<t_0, \ldots, t_{n-1}>)^{\mathcal{M}} = \mathcal{F}((t_0)^{\mathcal{M}}, \ldots, (t_{n-1})^{\mathcal{M}})
\end{array}
$$

PROOF This theorem can be proved by induction on the build-up of $\mathcal{F}$. Due to theorem 2.1.11 the proof is straightforward. □

Corollary 2.2.6 *Let n, $x_0, \ldots, x_{n-1}$ be arbitrary natural numbers and $\mathcal{F}$ be an n-ary primitive recursive function. Further, let $f \in Prim$ denote the index of $\mathcal{F}$. Then we can prove the following equations:*

$$
\begin{array}{ll}
n = 0: & (\mathsf{pr}_{\mathcal{F}})^{\mathcal{M}} = [f] \\
n = 1: & [(\mathsf{pr}_{\mathcal{F}})^{\mathcal{M}}](x_0) = [f](x_0) \\
n > 1: & [(\mathsf{pr}_{\mathcal{F}})^{\mathcal{M}}](\langle \ldots \langle \langle x_0, x_1 \rangle, x_2 \rangle, \ldots, x_{n-1} \rangle) = \\
 & \quad [f'](\langle \ldots \langle \langle x_0, x_1 \rangle, x_2 \rangle, \ldots, x_{n-1} \rangle)
\end{array}
$$

This can easily be proved with lemma 1.1.15 □

2.3 Comparison to BON

In this section we will define the language $\hat{\mathcal{L}}$ and the axioms of the basic theory BON of operations and numbers. Then we will examine the consistency of PRON with some of the additional axioms from definition 1.3.15 and will compare the results to BON.

Definition 2.3.1 *There are only little differences between $\mathcal{L}$ and $\hat{\mathcal{L}}$. We define the first order language $\hat{\mathcal{L}}$ of partial terms by making the following changes in $\mathcal{L}$:*

1. *The individual constants of $\hat{\mathcal{L}}$ are $\hat{\mathsf{k}}$, $\hat{\mathsf{s}}$ (combinators), $\hat{\mathsf{p}}$, $\hat{\mathsf{p}}_0$, $\hat{\mathsf{p}}_1$ (pairing and unpairing), $\mathsf{0}$ (zero), $\hat{\mathsf{s}}_{\mathsf{N}}$ (numerical successor), $\hat{\mathsf{p}}_{\mathsf{N}}$ (numerical predecessor), $\hat{\mathsf{d}}_{\mathsf{N}}$ (definition by numerical cases), and $\hat{\mathsf{r}}_{\mathsf{N}}$ (primitive recursion on N).*

2. *The symbol $\odot$ is the only function symbol of $\hat{\mathcal{L}}$.*

Definition 2.3.2 (BON) *The logical axioms and rules of inference of* BON *are the same as of* PRON. *The theory* BON *consists of the following non-logical axioms:*

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

(2) $\hat{\mathsf{s}}ab\downarrow \wedge \hat{\mathsf{s}}abc \simeq ac(bc)$

(3) $\hat{\mathsf{p}}_0(\hat{\mathsf{p}}ab) = a \wedge \hat{\mathsf{p}}_1(\hat{\mathsf{p}}ab) = b$

(4) $\mathsf{0} \in \mathsf{N} \wedge (\forall x \in \mathsf{N})(\hat{\mathsf{s}}_{\mathsf{N}}x \in \mathsf{N})$

(5) $(\forall x \in \mathsf{N})(\hat{\mathsf{s}}_{\mathsf{N}}x \neq \mathsf{0} \wedge \hat{\mathsf{p}}_{\mathsf{N}}(\hat{\mathsf{s}}_{\mathsf{N}}x) = x)$

(6) $(\forall x \in \mathsf{N})(x \neq \mathsf{0} \rightarrow \hat{\mathsf{p}}_{\mathsf{N}}x \in \mathsf{N} \wedge \hat{\mathsf{s}}_{\mathsf{N}}(\hat{\mathsf{p}}_{\mathsf{N}}x) = x)$

(7) $c \in \mathsf{N} \wedge d \in \mathsf{N} \wedge c = d \rightarrow \hat{\mathsf{d}}_{\mathsf{N}}abcd = a$

(8) $c \in \mathsf{N} \wedge d \in \mathsf{N} \wedge c \neq d \rightarrow \hat{\mathsf{d}}_{\mathsf{N}}abcd = b$

(9) $(\forall x, y \in \mathsf{N})(fxy \in \mathsf{N}) \wedge a \in \mathsf{N} \rightarrow (\forall x \in \mathsf{N})(\hat{\mathsf{r}}_{\mathsf{N}}fax \in \mathsf{N}) \wedge \hat{\mathsf{r}}_{\mathsf{N}}fa\mathsf{0} = a$

(10) $(\forall x, y \in \mathsf{N})(fxy \in \mathsf{N}) \wedge a \in \mathsf{N} \wedge b \in \mathsf{N} \rightarrow \hat{\mathsf{r}}_{\mathsf{N}}fa(\mathsf{s}_{\mathsf{N}}b) = fb(\hat{\mathsf{r}}_{\mathsf{N}}fab)$

We have to do some changes on several additional axioms because in $\hat{\mathcal{L}}$ the constants have different names and we have no function symbol for pairing. The abbreviations $\vec{t} \in \mathsf{N}$, $\vec{t} \in (\mathsf{N} \rightarrow \mathsf{N})$, and $\vec{f} \in \mathcal{P}(\mathsf{N})$ are the same as in PRON, but for $n > 1$ the abbreviation $\vec{t} \in (\mathsf{N}^n \rightarrow \mathsf{N})$ is not allowed in BON.

Definition 2.3.3 *The axioms* (Tot), (Ext), *and* (Nat) *and the induction scheme* (S-I_{N}) *of set induction are defined the same way as in* PRON. *The following additional axioms are different in $\hat{\mathcal{L}}$:*

($\hat{\mathsf{D}}_{\mathsf{V}}$) $(c = d \rightarrow \hat{\mathsf{d}}_{\mathsf{V}}abcd = a) \wedge (c \neq d \rightarrow \hat{\mathsf{d}}_{\mathsf{V}}abcd = b)$

($\hat{\mathsf{Ch}}_=$) $(x = y \rightarrow \hat{\mathsf{c}}_=xy = \mathsf{0}) \wedge (x \neq y \rightarrow \hat{\mathsf{c}}_=xy = \mathsf{1})$

($\hat{\mathsf{Ch}}_{\mathsf{N}}$) $(x \in \mathsf{N} \rightarrow \hat{\mathsf{c}}_{\mathsf{N}}x = \mathsf{0}) \wedge (x \notin \mathsf{N} \rightarrow \hat{\mathsf{c}}_{\mathsf{N}}x = \mathsf{1})$

($\hat{\mathsf{I}}\mathsf{nj}_{\mathsf{N}}$) $(\forall x)(\hat{\mathsf{i}}_{\mathsf{N}}x \in \mathsf{N}) \wedge (\forall x, y)(x \neq y \rightarrow \hat{\mathsf{i}}_{\mathsf{N}}x \neq \hat{\mathsf{i}}_{\mathsf{N}}y)$

($\hat{\boldsymbol{\mu}}$.1) $\quad f \in (\mathsf{N} \to \mathsf{N}) \leftrightarrow \hat{\boldsymbol{\mu}} f \in \mathsf{N}$

($\hat{\boldsymbol{\mu}}$.2) $\quad f \in (\mathsf{N} \to \mathsf{N}) \to [(\exists x \in \mathsf{N})(fx = \mathsf{0}) \to f(\hat{\boldsymbol{\mu}} f) = \mathsf{0}]$

($\hat{\mathsf{E}}_1$.1) $\quad (\forall x, y \in \mathsf{N})(fxy \in \mathsf{N}) \leftrightarrow \hat{\mathsf{E}}_1 f \in \mathsf{N}$

($\hat{\mathsf{E}}_1$.2) $\quad (\forall x, y \in \mathsf{N})(fxy \in \mathsf{N}) \to [(\exists g)(g \in (\mathsf{N} \to \mathsf{N}) \wedge$
$\quad (\forall x \in \mathsf{N})(f(g(\hat{\mathsf{s}}_{\mathsf{N}} x))(gx) = \mathsf{0})) \leftrightarrow \hat{\mathsf{E}}_1 f = \mathsf{0}]$

($\hat{\mathcal{L}}$-I_{N}) $\quad A[\mathsf{0}/u] \wedge (\forall x \in \mathsf{N})(A[x/u] \to A[\hat{\mathsf{s}}_{\mathsf{N}} x/u]) \to (\forall x \in \mathsf{N}) A[x/u]$
for all $\hat{\mathcal{L}}$ formulas A

Definition 2.3.4 *Let* BON^- *be the theory* BON *without the axioms* (9) *and* (10). *In addition, we define the following theories:*

$$\begin{aligned} \mathsf{BON}(\hat{\boldsymbol{\mu}}) &:= \mathsf{BON} + (\hat{\boldsymbol{\mu}}.1) + (\hat{\boldsymbol{\mu}}.2) \\ \mathsf{SUS} &:= \mathsf{BON}(\hat{\boldsymbol{\mu}}) + (\hat{\mathsf{E}}_1.1) + (\hat{\mathsf{E}}_1.2) \end{aligned}$$

There are two important theorems of BON we will need in the sequel. The existence of full λ abstraction and the fixed point theorem.

Theorem 2.3.5 (λ abstraction) *For each $\hat{\mathcal{L}}$ term t and all variables x there exists an $\hat{\mathcal{L}}$ term $(\lambda x.t)$ so that:*

$$\begin{aligned} \mathsf{BON} &\vdash (\lambda x.t)\downarrow \wedge (\lambda x.t)x \simeq t \\ \mathsf{BON} &\vdash s\downarrow \to (\lambda x.t)s \simeq t[s/x] \end{aligned}$$

Proof We define the $\hat{\mathcal{L}}$ term $(\lambda x.t)$ by induction on the build-up of t as follows:

1. If t is the variable x, then $(\lambda x.t)$ is defined to be the term $\hat{\mathsf{s}}\hat{\mathsf{k}}\hat{\mathsf{k}}$.

2. If t is a variable different from x or a constant, then $(\lambda x.t)$ is defined to be the term $\hat{\mathsf{k}}t$.

3. If t is an application (rs), then $(\lambda x.t)$ is defined to be the term $\hat{\mathsf{s}}(\lambda x.r)(\lambda x.s)$.

It is an easy exercise to verify the two assertions above. $\square$

We can generalize λ abstraction to several arguments by simply iterating abstraction for one argument. For all natural numbers $n > 0$, all $\hat{\mathcal{L}}$ terms t, and all variables $x_0, \ldots, x_{n-1}$, we write $(\lambda x_0 \ldots x_{n-1}.t)$ for $(\lambda x_0.(\ldots(\lambda x_{n-1}.t)\ldots))$.

Theorem 2.3.6 (Fixed point) *There exists a closed $\hat{\mathcal{L}}$ term* fix *so that:*

$$\mathsf{BON} \;\vdash\; \mathsf{fix}f{\downarrow} \wedge \mathsf{fix}fx \simeq f(\mathsf{fix}f)x$$

Proof We define fix to be the term

$$\lambda x.(\lambda yz.x(yy)z)(\lambda yz.x(yy)z)$$

It is an easy exercise to verify the properties of fix. □

The theory BON is much more powerful than PRON. It seems that $\mathsf{PRON}(\boldsymbol{\mu})$ is to weak to derive the axioms about $\hat{\boldsymbol{\mu}}$ and that $\mathsf{PRON}(\mathsf{SUS})$ is to weak to prove the axioms about $\hat{\mathsf{E}}_1$. There are plenty of sentences that are derivable in BON but not in PRON.

Remarks 2.3.7 *In* BON *we can show the following facts:*

1. *There exists an $\hat{\mathcal{L}}$ term* $\mathsf{not_N}$ *so that* BON *proves* $\neg\mathsf{N}(\mathsf{not_N})$*.*
   This is not true in PRON *because the model $\mathcal{M}$ of section 2.2 is a model of* $\mathsf{PRON} + (\mathsf{Tot}) + (\mathsf{Nat})$*.*

2. *There exists an $\hat{\mathcal{L}}$ term* prim *so that* $\mathsf{BON}^- + (\hat{\mathcal{L}}\text{-}\mathsf{I_N})$ *proves*

   (a) $(\forall x, y \in \mathsf{N})(fxy \in \mathsf{N}) \wedge a \in \mathsf{N} \rightarrow$
   $(\forall x \in \mathsf{N})(\mathsf{prim}fax \in \mathsf{N}) \wedge \mathsf{prim}fa0 = a$

   (b) $(\forall x, y \in \mathsf{N})(fxy \in \mathsf{N}) \wedge a \in \mathsf{N} \wedge b \in \mathsf{N} \rightarrow$
   $\mathsf{prim}fa(\hat{\mathsf{s}}_\mathsf{N}b) = fb(\mathsf{prim}fab)$

   It seems that there is no such term in $\mathsf{PRON}^- + (\mathcal{L}\text{-}\mathsf{I_N})$*.*

3. *There exists an $\hat{\mathcal{L}}$ term* $\mathsf{fix_t}$ *so that* $\mathsf{BON}+(\mathsf{Tot})$ *proves* $\mathsf{fix_t}f = f(\mathsf{fix_t}f)$*.*
   It seems that there is no such term in $\mathsf{PRON} + (\mathsf{Tot})$*.*

4. *We can represent the ω-jump for an arbitrary set in* $\mathsf{BON}(\hat{\boldsymbol{\mu}}) + (\hat{\mathcal{L}}\text{-}\mathsf{I_N})$*.*
   We will show in section 4.2 that $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N})$ *is to weak to give a representation of the ω-jump for an arbitrary set.*

5. *There exist $\hat{\mathcal{L}}$ terms which do not have a normal form, for example the term $(\lambda x.xx)(\lambda x.xx)$.*
   Note that $(\lambda^* x.xx)$ is not a valid term in PRON*.*

It would be far beyond the scope of this thesis to prove all these assertions. They can be looked up in Feferman, Jäger, and Strahm [6].

Now we will study additional principles about our applicative universe and ask the question whether these new principles lead to consistent theories or not.

Theorem 2.3.8 *The following theories are inconsistent:*

1. $\mathsf{BON} + (\mathsf{Tot}) + (\mathsf{Nat})$
2. $\mathsf{BON} + (\mathsf{Tot}) + (\hat{\mathsf{D}}_{\mathsf{V}})$
3. $\mathsf{BON} + (\mathsf{Tot}) + (\hat{\mathsf{Ch}}_{=})$
4. $\mathsf{BON} + (\mathsf{Tot}) + (\hat{\mathsf{Ch}}_{\mathsf{N}})$
5. $\mathsf{BON} + (\mathsf{Tot}) + (\hat{\mathsf{I}}\mathsf{nj}_{\mathsf{N}})$
6. $\mathsf{BON} + (\mathsf{Ext}) + (\mathsf{Nat})$
7. $\mathsf{BON} + (\mathsf{Ext}) + (\hat{\mathsf{D}}_{\mathsf{V}})$
8. $\mathsf{BON} + (\mathsf{Ext}) + (\hat{\mathsf{Ch}}_{=})$
9. $\mathsf{BON} + (\mathsf{Ext}) + (\hat{\mathsf{Ch}}_{\mathsf{N}})$
10. $\mathsf{BON} + (\mathsf{Ext}) + (\hat{\mathsf{I}}\mathsf{nj}_{\mathsf{N}})$

PROOF The ninth assertion is a result of Minari [10]. All the other assertions can be found in Strahm [15].

The theory $\mathsf{BON}+(\mathsf{Tot})+(\mathsf{Ext})$ is consistent. There are total term models of BON where extensionality is satisfied. This implies that $\mathsf{PRON}+(\mathsf{Tot})+(\mathsf{Ext})$ is consistent, too.

Theorem 2.3.9 *Let* $\mathsf{PRON}^{\mathsf{c}}$ *denote the following theory:*

$$\mathsf{PRON}^{\mathsf{c}} \;:=\; \mathsf{PRON} + (\mathsf{Tot}) + (\mathsf{Nat}) + (\mathsf{D}_{\mathsf{V}}) + (\mathsf{Ch}_{=}) + (\mathsf{Ch}_{\mathsf{N}}) + (\mathsf{Inj}_{\mathsf{N}})$$

Then the theory $\mathsf{PRON}^{\mathsf{c}}$ *is consistent.*

PROOF The recursion theoretic model $\mathcal{M}$ is total and the universe of $\mathcal{M}$ is the set $\mathbb{N}$ of natural numbers. We can extend it to a model of $\mathsf{PRON}^{\mathsf{c}}$ by adding the interpretations of the constants d_{V}, $\mathsf{c}_{=}$, c_{N}, and i_{N}.

$$\begin{aligned}
{\mathsf{d}_{\mathsf{V}}}^{\mathcal{M}} &:= {\mathsf{d}_{\mathsf{N}}}^{\mathcal{M}} \\
{\mathsf{c}_{=}}^{\mathcal{M}} &:= \text{index of the primitive recursive function } x \to ch_{=}((x)_0,(x)_1) \\
{\mathsf{c}_{\mathsf{N}}}^{\mathcal{M}} &:= \langle 1,1,0\rangle \\
{\mathsf{i}_{\mathsf{N}}}^{\mathcal{M}} &:= {\mathsf{s}_{\mathsf{N}}}^{\mathcal{M}}
\end{aligned}$$

It is now easy to verify that $\mathcal{M}$ is a model of $\mathsf{PRON}^{\mathsf{c}}$ $\square$

In the recursion theoretic model $\mathcal{M}$ we do not have extensionality because every function has infinitely many indices. The next theorem tells us that it is not possible to have a model in $\mathbb{N}$ with extensionality.

Theorem 2.3.10 *The following theories are inconsistent:*

1. $\mathsf{PRON} + (\mathsf{Ext}) + (\mathsf{D_V})$
2. $\mathsf{PRON} + (\mathsf{Ext}) + (\mathsf{Nat})$
3. $\mathsf{PRON} + (\mathsf{Ext}) + (\mathsf{Ch_=})$
4. $\mathsf{PRON} + (\mathsf{Ext}) + (\mathsf{Inj_N})$

PROOF For the first assertion we define s to be the $\mathcal{L}$ term $\mathsf{rec}(\lambda^* y.\mathsf{d_V}{<}\mathsf{1}, \mathsf{0}, \mathsf{p_0} y, (\lambda^* z.\mathsf{0}){>})$, which immediately leads to a contradiction: From theorem 2.1.9 we know that $s{\downarrow}$ and we have

$$\begin{aligned} sx &\simeq (\lambda^* y.\mathsf{d_V}{<}\mathsf{1}, \mathsf{0}, \mathsf{p_0} y, (\lambda^* z.\mathsf{0}){>}){<}s, x{>} \\ &\simeq \mathsf{d_V}{<}\mathsf{1}, \mathsf{0}, s, (\lambda^* z.\mathsf{0}){>} \simeq \begin{cases} \mathsf{1} & \text{if } s = (\lambda^* z.\mathsf{0}) \\ \mathsf{0} & \text{otherwise} \end{cases} \end{aligned}$$

The other three assertions immediately follow from the first one. □

We do not know if the theory $\mathsf{PRON} + (\mathsf{Ext}) + (\mathsf{Ch_N})$ is consistent or not, because we cannot apply the arguments of P. Minari to PRON.

3 Lower proof-theoretic bounds

3.1 $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N})$ contains PA

In this section we interpret the system PA of first order arithmetic into applicative theories. A crucial step for this embedding has already been done in Definition 2.1.10 and Theorem 2.1.11 by assigning to each primitive recursive function $\mathcal{F}$ a closed $\mathcal{L}$ term $\mathsf{pr}_{\mathcal{F}}$ which represents $\mathcal{F}$ in PRON.
First we have to define a translation $\cdot^{\diamond}$ from $\mathcal{L}_1$ terms and formulas to terms and formulas of $\mathcal{L}$. Every variable u of $\mathcal{L}_1$ has its corresponding variable in $\mathcal{L}$ and we mostly choose the individual variable u for the translation $u^{\diamond}$ of the number variable u.

Definition 3.1.1 *The individual $\mathcal{L}$ term $t^{\diamond}$ is defined by induction on the build-up of the term t.*

1. *If t is a number variable of $\mathcal{L}_1$, then $t^{\diamond}$ is the corresponding individual variable of $\mathcal{L}$.*

2. *If t is a constant of $\mathcal{L}_1$, then $t^{\diamond}$ is the corresponding numeral of $\mathcal{L}$.*

3. *If $n > 0$ and t is the $\mathcal{L}_1$ term $\mathcal{F}(s_0, \dots, s_{n-1})$ for the n-ary function symbol $\mathcal{F}$, then $t^{\diamond}$ is the $\mathcal{L}$ term*
$$\begin{cases} <(s_0)^{\diamond}, (s_1)^{\diamond}> & \text{if } t = \langle s_0, s_1 \rangle \\ \mathsf{pr}_{\mathcal{F}}<(s_0)^{\diamond}, \dots, (s_{n-1})^{\diamond}> & \text{otherwise} \end{cases}$$

Definition 3.1.2 *The $\mathcal{L}$ formula $A^{\diamond}$ is defined by induction on the build-up of the $\mathcal{L}_1$ formula A.*

1. *If $n > 0$ and $A = \mathcal{R}(t_0, \dots, t_{n-1})$ for the n-ary relation symbol $\mathcal{R}$, then*
$$A^{\diamond} := \begin{cases} (t_0)^{\diamond} = (t_1)^{\diamond} & \text{if } A = (t_0 = t_1) \\ \mathsf{pr}_{ch_{\mathcal{R}}}<(t_0)^{\diamond}, \dots, (t_{n-1})^{\diamond}> = \mathsf{0} & \text{otherwise} \end{cases}$$

2. *If $A = \neg B$, then $A^{\diamond} := \neg B^{\diamond}$.*

3. *If $A = B \vee C$, then $A^{\diamond} := B^{\diamond} \vee C^{\diamond}$.*

4. *If $A = (\exists x)B$, then $A^{\diamond} := (\exists x \in \mathsf{N})B^{\diamond}$.*

Finally, we have to ensure that the individual variables of the translated formulas $A^{\diamond}$ range over N, because the number variables of $\mathcal{L}_1$ are interpreted as natural numbers.

Definition 3.1.3 ($\mathcal{L}$ translation of $\mathcal{L}_1$ formulas) *Let $A[\vec{v}]$ be an $\mathcal{L}_1$ formula. Then the $\mathcal{L}$ translation $A^\diamond_{\mathsf{N}}$ of A is defined as follows:*

$$A^\diamond_{\mathsf{N}}(\vec{v}) \quad := \quad \vec{v} \in \mathsf{N} \to A^\diamond(\vec{v})$$

Due to the complete induction scheme ($\mathcal{L}$-I_N) we can embed the theory PA in $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I}_\mathsf{N})$ directly without any problems.

Theorem 3.1.4 (Embedding of PA) *Let $B[\vec{v}]$ be an $\mathcal{L}_1$ formula so that PA proves $B(\vec{v})$. Then we have*

$$\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I}_\mathsf{N}) \quad \vdash \quad B^\diamond_{\mathsf{N}}(\vec{v})$$

Proof We can prove this theorem by induction on the length of derivation. We only need to check the non-logical axioms because the logical axioms and the rules of inference are easy to verify.

1. The number theoretic axioms can be verified with the term $\mathsf{pr}_{ch_<}$ and the axioms (6) and (7) of PRON about the constant $\mathsf{s_N}$.

2. With $\mathsf{pr}_{\mathcal{F}}$ we can derive the defining equation of every primitive recursive function $\mathcal{F}$ due to theorem 2.1.11.

3. The $\mathcal{L}$ translation of the induction scheme (Π_∞-I_N) is
$\vec{v} \in \mathsf{N} \to$
$[A^\diamond(\vec{v}, \mathsf{0}) \land (\forall x \in \mathsf{N})(A^\diamond(\vec{v}, x) \to A^\diamond(\vec{v}, \mathsf{pr}_{\mathcal{S}} x)) \to (\forall x \in \mathsf{N}) A^\diamond(\vec{v}, x)]$.
Suppose, that $\vec{v} \in \mathsf{N}$. Then we have exactly the induction scheme ($\mathcal{L}$-I_N) because PRON proves $\mathsf{pr}_{\mathcal{S}} x = \mathsf{s_N} x$ by definition 2.1.10. □

3.2 $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I}_\mathsf{N})$ contains ACA

For a precise formulation of the embedding result of this section we have to extend the translation $\cdot^\diamond$ of section 3.1 to the language $\mathcal{L}_2$. We write $\cdot^\circ$ for this extended translation. In the following we assume that we have a translation of the number, set, and function variables of $\mathcal{L}_2$ into the variables of $\mathcal{L}$ so that no conflicts arise. For convenience we often simply write, for example, u, x, f for the translations of the number, set, and function variables u, X, F, respectively.

Definition 3.2.1 *The individual $\mathcal{L}$ term t° is defined by induction on the build-up of the term t.*

1. *If t is a number variable of $\mathcal{L}_2$, then t° is the corresponding individual variable of $\mathcal{L}$.*

2. *If t is a constant of $\mathcal{L}_2$, then t° is the corresponding numeral of $\mathcal{L}$.*

3. *If $n > 0$ and t is the $\mathcal{L}_2$ term $\mathcal{F}(s_0, \ldots, s_{n-1})$ for the n-ary function symbol $\mathcal{F}$, then t° is the $\mathcal{L}$ term*

$$\begin{cases} <(s_0)^\circ, (s_1)^\circ> & \text{if } t = \langle s_0, s_1 \rangle \\ \mathsf{pr}_{\mathcal{F}}<(s_0)^\circ, \ldots, (s_{n-1})^\circ> & \text{otherwise} \end{cases}$$

4. *If t is the $\mathcal{L}_2$ term $F(s)$ for the function variable F, then t° is the $\mathcal{L}$ term $f(s^\circ)$.*

Definition 3.2.2 *The $\mathcal{L}$ formula A° is defined by induction on the build-up of the $\mathcal{L}_2$ formula A.*

1. *If $n > 0$ and $A = \mathcal{R}(t_0, \ldots, t_{n-1})$ for the n-ary relation symbol $\mathcal{R}$, then*

$$A^\circ := \begin{cases} (t_0)^\circ = (t_1)^\circ & \text{if } A = (t_0 = t_1) \\ \mathsf{pr}_{ch_{\mathcal{R}}}<(t_0)^\circ, \ldots, (t_{n-1})^\circ> = \mathsf{0} & \text{otherwise} \end{cases}$$

2. *If $A = (s \in X)$ for the set variable X, then $A^\circ := (x(s^\circ) = \mathsf{0})$.*

3. *If $A = \neg B$, then $A^\circ := \neg B^\circ$.*

4. *If $A = B \vee C$, then $A^\circ := B^\circ \vee C^\circ$.*

5. *If $A = (\exists x)B$, then $A^\circ := (\exists x \in \mathsf{N})B^\circ$.*

6. *If $A = (\exists X)B$, then $A^\circ := (\exists x)[x \in \mathcal{P}(\mathsf{N}) \wedge B^\circ]$.*

7. *If $A = (\exists F)B$, then $A^\circ := (\exists f)[f \in (\mathsf{N} \to \mathsf{N}) \wedge B^\circ]$.*

As in the embedding of PA we have to ensure, that the translations of the number variables range over N. In addition, we have to make sure that the translations of the set and function variables range over $\mathcal{P}(\mathsf{N})$ and $(\mathsf{N} \to \mathsf{N})$, respectively.

Definition 3.2.3 ($\mathcal{L}$ translation of $\mathcal{L}_2$ formulas) *Let A be an arbitrary $\mathcal{L}_2$ formula. The $\mathcal{L}$ translation A_N of $A[\vec{U}, \vec{F}, \vec{v}]$ is the following formula:*

$$A_\mathsf{N}(\vec{u}, \vec{f}, \vec{v}) \quad := \quad \vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \wedge \vec{v} \in \mathsf{N} \to A^\circ(\vec{u}, \vec{f}, \vec{v})$$

Lemma 3.2.4 (Characteristic term I) *Let n be a natural number and $A[\vec{U}, \vec{F}, \vec{v}]$ be a Π^1_0 formula. Then there exists an individual $\mathcal{L}$ term $\mathsf{t}_A[\vec{u}, \vec{f}]$ without the free variables $v_0, \ldots, v_{n-1}$ so that $\mathsf{PRON}(\boldsymbol{\mu})$ proves the following formulas:*

$$n = 0: \quad (\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to \mathsf{t}_A = \mathsf{0} \vee \mathsf{t}_A = \mathsf{1}]$$

$$(\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (A^\circ(\vec{u}, \vec{f}) \leftrightarrow \mathsf{t}_A = \mathsf{0})]$$

$$n > 0: \quad (\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (\forall \vec{v} \in \mathsf{N})(\mathsf{t}_A{<}v_0, \ldots, v_{n-1}{>} = \mathsf{0} \vee \mathsf{t}_A{<}v_0, \ldots, v_{n-1}{>} = \mathsf{1})]$$

$$(\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (\forall \vec{v} \in \mathsf{N})(A^\circ(\vec{u}, \vec{f}, \vec{v}) \leftrightarrow \mathsf{t}_A{<}v_0, \ldots, v_{n-1}{>} = \mathsf{0})]$$

PROOF We define the term t_A by induction on the build-up of A.

1. If $m > 0$ and $A = \mathcal{R}(s_0, \ldots, s_{m-1})$ then t_A is defined to be the term
$$\begin{cases} \mathsf{pr}_{ch_\mathcal{R}}{<}(s_0)^\circ, \ldots, (s_{m-1})^\circ{>} & \text{if } n = 0 \\ (\lambda^* z.\mathsf{pr}_{ch_\mathcal{R}}{<}(s_0)^\circ, \ldots, (s_{m-1})^\circ{>}\ [(z)^n_0, \ldots, (z)^n_{n-1}/\vec{v}]) & \text{if } n > 0 \end{cases}$$

2. If $A = (s \in X)$ then t_A is defined to be the term
$$\begin{cases} \mathsf{pr}_{ch_=}{<}x(s^\circ), \mathsf{0}{>} & \text{if } n = 0 \\ (\lambda^* z.\mathsf{pr}_{ch_=}{<}x(s^\circ), \mathsf{0}{>}\ [(z)^n_0, \ldots, (z)^n_{n-1}/\vec{v}]) & \text{if } n > 0 \end{cases}$$

3. If $A = \neg B$ then t_A is defined to be the term
$$\begin{cases} \mathsf{pr}_{\dot{-}}{<}\mathsf{1}, \mathsf{t}_B{>} & \text{if } n = 0 \\ (\lambda^* z.\mathsf{pr}_{\dot{-}}{<}\mathsf{1}, \mathsf{t}_B z{>}) & \text{if } n > 0 \end{cases}$$

4. (a) If $A = B \vee C$ and B has the $m > 0$ free variables $v_{k_0}, \ldots, v_{k_{m-1}}$ and C has the $l > 0$ free variables $v_{k_m}, \ldots, v_{k_{m+l-1}}$, then t_A is defined to be the term
$(\lambda^* z.\mathsf{pr}_{min}{<}\mathsf{t}_B{<}(z)^n_{k_0}, \ldots, (z)^n_{k_{m-1}}{>}, \mathsf{t}_C{<}(z)^n_{k_m}, \ldots, (z)^n_{k_{m+l-1}}{>}{>})$.
   (b) If $A = B \vee C$ and at least one of the formulas B and C does not have any free variable, then t_A is defined to be the term
$$\begin{cases} \mathsf{pr}_{min}{<}\mathsf{t}_B, \mathsf{t}_C{>} & \text{if } n = 0 \\ (\lambda^* z.\mathsf{pr}_{min}{<}\mathsf{t}_B, \mathsf{t}_C z{>}) & \text{if } n > 0 \text{ and } B \text{ has no free variables} \\ (\lambda^* z.\mathsf{pr}_{min}{<}\mathsf{t}_B z, \mathsf{t}_C{>}) & \text{if } n > 0 \text{ and } C \text{ has no free variables} \end{cases}$$

5. If $A = (\exists x)B[x/v_n]$ then t_A is defined to be the term
$$\begin{cases} (\lambda^* y.\mathsf{t}_B(\mathsf{p}_1 y)){<}\mathsf{0}, \boldsymbol{\mu}(\lambda^* y.\mathsf{t}_B(\mathsf{p}_1 y))\mathsf{0}{>} & \text{if } n = 0 \\ (\lambda^* z.\mathsf{t}_B{<}z, \boldsymbol{\mu}\mathsf{t}_B z{>}) & \text{if } n > 0 \end{cases}$$
Note that for $n > 0$ the term t_B is in $(\mathsf{N}^{n+1} \to \mathsf{N})$ by induction hypothesis, because B has one more free variable than A. That is the reason why we can apply the μ operator to t_B.
For $n = 0$ the term t_B is in $(\mathsf{N} \to \mathsf{N})$ by induction hypothesis, so we

cannot apply the μ operator to t_B. By taking $\mathsf{0}$ as a dummy parameter, we can apply $\boldsymbol{\mu}$ to $(\lambda^* y.\mathsf{t}_B(\mathsf{p}_1 y))$.

It is an easy exercise to prove by induction on the build-up of A that $\mathsf{PRON}(\boldsymbol{\mu})$ proves the two formulas above. □

Theorem 3.2.5 (Embedding of ACA) *Let $B[\vec{U}, \vec{F}, \vec{v}]$ be an $\mathcal{L}_2$ formula so that ACA proves $B(\vec{U}, \vec{F}, \vec{v})$. Then we have*

$$\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N}) \ \vdash \ B_{\mathsf{N}}(\vec{u}, \vec{f}, \vec{v})$$

Proof We can prove this theorem by induction on the length of derivation.

1. Because $\cdot^{\circ}$ is an extension of $\cdot^{\diamond}$, we can say that the logical axioms, the rules of inference, and the axioms of PRA^- have already been verified in the proof of theorem 3.1.4.

2. The $\mathcal{L}$ translation of the induction scheme $(\Pi^1_\infty\text{-}\mathsf{I_N})$ is
$\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge \vec{v} \in \mathsf{N} \rightarrow$
$[A^{\circ}(\vec{u}, \vec{f}, \vec{v}, \mathsf{0}) \wedge (\forall x \in \mathsf{N})(A^{\circ}(\vec{u}, \vec{f}, \vec{v}, x) \rightarrow A^{\circ}(\vec{u}, \vec{f}, \vec{v}, \mathsf{pr}_{\mathcal{S}} x)) \rightarrow$
$(\forall x \in \mathsf{N}) A^{\circ}(\vec{u}, \vec{f}, \vec{v}, x)]$.
Suppose, that $\vec{u} \in \mathcal{P}(\mathsf{N})$, $\vec{f} \in (\mathsf{N} \rightarrow \mathsf{N})$, and $\vec{v} \in \mathsf{N}$. Then we have exactly the induction scheme $(\mathcal{L}\text{-}\mathsf{I_N})$ because PRON proves $(\mathsf{pr}_{\mathcal{S}} x = \mathsf{s_N} x)$ by definition 2.1.10.

3. The $\mathcal{L}$ translation of the comprehension scheme $(\Pi^1_0\text{-}\mathsf{CA})$ is
$\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge \vec{v} \in \mathsf{N} \rightarrow$
$(\exists x)[x \in \mathcal{P}(\mathsf{N}) \wedge (\forall y \in \mathsf{N})(xy = \mathsf{0} \leftrightarrow A^{\circ}(\vec{u}, \vec{f}, \vec{v}, y))]$.
Suppose, that $\vec{u} \in \mathcal{P}(\mathsf{N})$, $\vec{f} \in (\mathsf{N} \rightarrow \mathsf{N})$, and $\vec{v} \in \mathsf{N}$, and define x to be the term $(\lambda^* z.\mathsf{t}_A{<}v_0, \ldots, v_{n-1}, z{>})$. Then we can prove $x \in \mathcal{P}(\mathsf{N})$ and for $y \in \mathsf{N}$ we have $xy = \mathsf{0} \leftrightarrow \mathsf{t}_A{<}v_0, \ldots, v_{n-1}, y{>} = \mathsf{0} \leftrightarrow A^{\circ}(\vec{u}, \vec{f}, \vec{v}, y)$ by lemma 3.2.4.

4. The $\mathcal{L}$ translation of the graph principle axiom (GP) is
$(\forall x)[x \in \mathcal{P}(\mathsf{N}) \rightarrow [(\forall y \in \mathsf{N})(\exists! z \in \mathsf{N})(x{<}y, z{>} = \mathsf{0}) \rightarrow$
$(\exists f)(f \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge (\forall y \in \mathsf{N})(x{<}y, fy{>} = \mathsf{0}))]]$.
Suppose, that x is in $\mathcal{P}(\mathsf{N})$ and $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N})$ proves
$(\forall y \in \mathsf{N})(\exists! z \in \mathsf{N})(x{<}y, z{>} = \mathsf{0})$. Now let y be in N. Then we define t to be the term $(\lambda^* u.x{<}y, u{>})$, which is in $(\mathsf{N} \rightarrow \mathsf{N})$. We cannot apply the μ operator to t, so we have to define another term s in $(\mathsf{N}^2 \rightarrow \mathsf{N})$ by $(\lambda^* v.t(\mathsf{p}_1 v))$. Now we can define $f := (\lambda^* w.\boldsymbol{\mu} s w) =$
$(\lambda^* w.\boldsymbol{\mu}(\lambda^* v.x{<}y, \mathsf{p}_1 v{>})w)$, which is in $(\mathsf{N} \rightarrow \mathsf{N})$ by the axiom $(\boldsymbol{\mu}.1)$. Finally, we can prove $x{<}y, fy{>}) = x{<}y, \boldsymbol{\mu} sy{>} = s{<}\mathsf{0}, \boldsymbol{\mu} sy{>} = \mathsf{0}$ by the axiom $(\boldsymbol{\mu}.2)$.

The difficulty of this proof is the case of the graph principle axiom, because we cannot apply the μ operator to any $t \in (\mathsf{N} \to \mathsf{N})$. Therefore we have to define the convenient $s \in (\mathsf{N}^2 \to \mathsf{N})$. $\Box$

This theorem shows that $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N})$ is at least as strong as ACA. It is the purpose of section 4.2 to show that it has the same strength.

3.3 $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N})$ contains Π^1_1-CA

Theorem 3.3.1 (Π^1_1 normal forms) *For every Π^1_1 formula A there exists an arithmetic formula B_A which contains the free variables of A plus two fresh variables w_0 and w_1 so that:*

$$\mathsf{ACA_0} \;\vdash\; A(\vec{U}, \vec{F}, \vec{v}) \leftrightarrow \neg(\exists G)(\forall x) B_A(\vec{U}, \vec{F}, \vec{v}, G(\mathcal{S}(x)), G(x))$$

Proof The proof of this theorem is more or less basic and can be found at many places (for example in Simpson [14]). $\Box$

We are now ready for the preparation of our next embedding. Note that we can use the translation $\cdot^\circ$ from definition 3.2.1 and definition 3.2.2. In addition, the $\mathcal{L}$ translation A_N for every $\mathcal{L}_2$ formula A is the same as in definition 3.2.3.

Lemma 3.3.2 (Characteristic term II) *Let n be a natural number and $A[\vec{U}, \vec{F}, \vec{v}]$ be an arbitrary Π^1_1 formula. Then there exists an individual $\mathcal{L}$ term $\mathsf{ct}_A[\vec{u}, \vec{f}]$ without the free variables $v_0, \ldots, v_{n-1}$ so that* $\mathsf{PRON(SUS)}$ *proves the following formulas:*

$$n = 0: \quad (\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to \mathsf{ct}_A = 0 \vee \mathsf{ct}_A = 1]$$

$$(\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (A^\circ(\vec{u}, \vec{f}) \leftrightarrow \mathsf{ct}_A = 0)]$$

$$n > 0: \quad (\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (\forall \vec{v} \in \mathsf{N})(\mathsf{ct}_A{<}v_0, \ldots, v_{n-1}{>} = 0 \vee \mathsf{ct}_A{<}v_0, \ldots, v_{n-1}{>} = 1)]$$

$$(\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (\forall \vec{v} \in \mathsf{N})(A^\circ(\vec{u}, \vec{f}, \vec{v}) \leftrightarrow \mathsf{ct}_A{<}v_0, \ldots, v_{n-1}{>} = 0)]$$

Proof From theorem 3.3.1 we know that there is an arithmetic formula B_A so that $\mathsf{ACA_0}$ proves $A(\vec{U}, \vec{F}, \vec{v}) \leftrightarrow \neg(\exists G)(\forall x) B_A(\vec{U}, \vec{F}, \vec{v}, G(\mathcal{S}(x)), G(x))$. By lemma 3.2.4 we get the existence of an $\mathcal{L}$ term t_{B_A} so that $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N})$ proves the two following formulas (no problem if $n = 0$):

$$(\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (\forall \vec{v}, w_0, w_1 \in \mathsf{N})$$
$$(\mathsf{t}_{B_A}{<}v_0, \ldots, v_{n-1}, w_0, w_1{>} = \mathsf{0} \vee \mathsf{t}_{B_A}{<}v_0, \ldots, v_{n-1}, w_0, w_1{>} = \mathsf{1})]$$

$$(\forall \vec{u}, \vec{f})[\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \to (\forall \vec{v}, w_0, w_1 \in \mathsf{N})$$
$$(A^\circ(\vec{u}, \vec{f}, \vec{v}, w_0, w_1) \leftrightarrow \mathsf{t}_{B_A}{<}v_0, \ldots, v_{n-1}, w_0, w_1{>} = \mathsf{0})]$$

Now we define the term ct_A by:

$$\mathsf{ct}_A \;:=\; \begin{cases} \mathsf{pr}_{\dot{-}}{<}\mathsf{1}, \mathsf{E}_1(\lambda^* y.\mathsf{t}_{B_A}{<}(y)^3_1, (y)^3_2{>})\mathsf{0}{>} & \text{if } n = 0 \\ (\lambda^* z.\mathsf{pr}_{\dot{-}}{<}\mathsf{1}, \mathsf{E}_1 \mathsf{t}_{B_A} z{>}) & \text{if } n > 0 \end{cases}$$

Note that for $n > 0$ the term t_{B_A} is in $(\mathsf{N}^{n+2} \to \mathsf{N})$ by lemma 3.2.4, because B_A has two more free variables than A. That is the reason why we can apply the Suslin operator to t_{B_A}.
For $n = 0$ the term t_{B_A} is in $(\mathsf{N}^2 \to \mathsf{N})$ by lemma 3.2.4, so we cannot apply the Suslin operator to t_{B_A}. By taking $\mathsf{0}$ as a dummy parameter, we can apply E_1 to $(\lambda^* y.\mathsf{t}_{B_A}{<}(y)^3_1, (y)^3_2{>})$.
We will show the proof of this lemma for $n > 0$. Given $\vec{u} \in \mathcal{P}(\mathsf{N})$, $\vec{f} \in (\mathsf{N} \to \mathsf{N})$, and $\vec{v} \in \mathsf{N}$, we have to show that $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N})$ proves the following equation:

$$\mathsf{ct}_A{<}v_0, \ldots, v_{n-1}{>} \;=\; \begin{cases} \mathsf{0} & \text{if } A^\circ(\vec{u}, \vec{f}, \vec{v}) \\ \mathsf{1} & \text{otherwise} \end{cases}$$

We can prove in $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N})$ that:

$$\mathsf{ct}_A{<}v_0, \ldots, v_{n-1}{>} = \begin{cases} \mathsf{1} & \text{if } \mathsf{E}_1 \mathsf{t}_{B_A}{<}v_0, \ldots, v_{n-1}{>} = \mathsf{0} \\ \mathsf{0} & \text{otherwise} \end{cases}$$

$$= \begin{cases} \mathsf{1} & \text{if } (\exists g)[g \in (\mathsf{N} \to \mathsf{N}) \wedge (\forall x \in \mathsf{N})(\mathsf{t}_{B_A}{<}v_0, \ldots, v_{n-1}, g(\mathsf{s_N} x), gx{>} = \mathsf{0})] \\ \mathsf{0} & \text{otherwise} \end{cases}$$

$$= \begin{cases} \mathsf{1} & \text{if } (\exists g)[g \in (\mathsf{N} \to \mathsf{N}) \wedge (\forall x \in \mathsf{N}) B^\circ_A(\vec{u}, \vec{f}, \vec{v}, g(\mathsf{s_N} x), gx)] \\ \mathsf{0} & \text{otherwise} \end{cases}$$

$$= \begin{cases} \mathsf{1} & \text{if } \neg A^\circ(\vec{u}, \vec{f}, \vec{v}) \\ \mathsf{0} & \text{otherwise} \end{cases} \qquad \square$$

Theorem 3.3.3 (Embedding of Π^1_1-CA) *Let $B[\vec{U}, \vec{F}, \vec{v}]$ be an $\mathcal{L}_2$ formula so that Π^1_1-CA proves $B(\vec{U}, \vec{F}, \vec{v})$. Then we have*

$$\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N}) \;\vdash\; B_\mathsf{N}(\vec{u}, \vec{f}, \vec{v})$$

Proof We can prove this theorem by induction on the length of derivation. We only need to check the axiom (Π^1_1-CA), because this is the only change from ACA to Π^1_1-CA and we have the translation $\cdot^\circ$ from section 3.2.

The $\mathcal{L}$ translation of the comprehension scheme (Π^1_1-CA) is
$\vec{u} \in \mathcal{P}(\mathsf{N}) \wedge \vec{f} \in (\mathsf{N} \to \mathsf{N}) \wedge \vec{v} \in \mathsf{N} \to$
$(\exists x)[x \in \mathcal{P}(\mathsf{N}) \wedge (\forall y \in \mathsf{N})(xy = \mathsf{0} \leftrightarrow A^\circ(\vec{u}, \vec{f}, \vec{v}, y))]$.
Suppose, that $\vec{u} \in \mathcal{P}(\mathsf{N})$, $\vec{f} \in (\mathsf{N} \to \mathsf{N})$, and $\vec{v} \in \mathsf{N}$, and define x to be the term $(\lambda^* z.\mathsf{ct}_A < v_0, \ldots, v_{n-1}, z >)$. Then we can prove $x \in \mathcal{P}(\mathsf{N})$ and for $y \in \mathsf{N}$ we have $xy = \mathsf{0} \leftrightarrow \mathsf{ct}_A < v_0, \ldots, v_{n-1}, y > = \mathsf{0} \leftrightarrow A^\circ(\vec{u}, \vec{f}, \vec{v}, y)$ by lemma 3.3.2. □

This theorem shows that $\mathsf{PRON}(\mathsf{SUS}) + (\mathcal{L}\text{-}\mathsf{I}_\mathsf{N})$ is at least as strong as Π^1_1-CA. It is the purpose of section 4.3 to show that it has the same strength.

4 Upper proof-theoretic bounds

4.1 PA contains $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N})$

We already know that $\mathsf{BON} + (\hat{\mathcal{L}}\text{-}\mathsf{I_N}) \subseteq \mathsf{PA}$ from Feferman [3]. The only thing we have to show yet is that $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N}) \subseteq \mathsf{BON} + (\hat{\mathcal{L}}\text{-}\mathsf{I_N})$.
The embedding of PRON in BON is straightforward. We only have to define a convenient translation of $\mathcal{L}$ terms and $\mathcal{L}$ formulas to the language $\hat{\mathcal{L}}$ of BON.

Definition 4.1.1 *In the translation $\cdot^{\triangleright}$ of $\mathcal{L}$ terms we express the terms with use of full λ abstraction we have in BON. We define $\cdot^{\triangleright}$ by induction on the definition of $\mathcal{L}$ terms.*

1. *If t is a variable of $\mathcal{L}$ or the constant $\mathsf{0}$, then $t^{\triangleright}$ is defined to be the $\hat{\mathcal{L}}$ term t. Further, the following translations are also straightforward:*
   $\mathsf{k}^{\triangleright} := \hat{\mathsf{k}}$
   ${\mathsf{p_0}}^{\triangleright} := \hat{\mathsf{p}}_0$
   ${\mathsf{p_1}}^{\triangleright} := \hat{\mathsf{p}}_1$
   ${\mathsf{s_N}}^{\triangleright} := \hat{\mathsf{s}}_\mathsf{N}$
   ${\mathsf{p_N}}^{\triangleright} := \hat{\mathsf{p}}_\mathsf{N}$
2. $\mathsf{i}^{\triangleright} := \hat{\mathsf{s}}\hat{\mathsf{k}}\hat{\mathsf{k}}$
3. ${\mathsf{a_2}}^{\triangleright} := (\lambda xy.\hat{\mathsf{p}}(\hat{\mathsf{p}}_0 xy)(\hat{\mathsf{p}}_1 xy))$
4. ${\mathsf{b_2}}^{\triangleright} := (\lambda xy.\hat{\mathsf{p}}_0 x(\hat{\mathsf{p}}_1 xy))$
5. ${\mathsf{d_N}}^{\triangleright} := (\lambda x.\hat{\mathsf{d}}_\mathsf{N}(\hat{\mathsf{p}}_0(\hat{\mathsf{p}}_0(\hat{\mathsf{p}}_0 x)))(\hat{\mathsf{p}}_1(\hat{\mathsf{p}}_0(\hat{\mathsf{p}}_0 x)))(\hat{\mathsf{p}}_1(\hat{\mathsf{p}}_0 x))(\hat{\mathsf{p}}_1 x))$
6. $\mathsf{r}^{\triangleright} := (\lambda u.\mathsf{fix}(tu))$ *while*
   $t = (\lambda uhx.[\hat{\mathsf{d}}_\mathsf{N}(\lambda z.\hat{\mathsf{p}}_0 u(\hat{\mathsf{p}}_0 x))(\lambda z.\hat{\mathsf{p}}_1 u(\hat{\mathsf{p}}(sx)(h(sx))))(\hat{\mathsf{p}}_1 x)\mathsf{0}]\mathsf{0})$
   and $s = (\lambda y.\hat{\mathsf{p}}(\hat{\mathsf{p}}_0 y)(\hat{\mathsf{p}}_\mathsf{N}(\hat{\mathsf{p}}_1 y)))$
7. *If t is the $\mathcal{L}$ term rs, then $t^{\triangleright}$ is defined to be the $\hat{\mathcal{L}}$ term $r^{\triangleright}s^{\triangleright}$.*
8. *If t is the $\mathcal{L}$ term $<r, s>$, then $t^{\triangleright}$ is defined to be the $\hat{\mathcal{L}}$ term $\hat{\mathsf{p}}r^{\triangleright}s^{\triangleright}$.*

The translation $\cdot^{\triangleright}$ for $\mathcal{L}$ formulas is canonical. The most difficult work has been done in the previous definition.

Definition 4.1.2 *Let us define the translation $\cdot^{\triangleright}$ by induction on the definition of $\mathcal{L}$ formulas.*

$$\begin{array}{rcl}
(\mathsf{N}(t))^{\triangleright} & := & \mathsf{N}(t^{\triangleright}) \\
(t\downarrow)^{\triangleright} & := & t^{\triangleright}\downarrow \\
(s = t)^{\triangleright} & := & (s^{\triangleright} = t^{\triangleright})
\end{array}$$

$$\begin{array}{rcl}(\neg A)^{\triangleright} & := & \neg A^{\triangleright} \\ (A \vee B)^{\triangleright} & := & A^{\triangleright} \vee B^{\triangleright} \\ ((\exists x)A)^{\triangleright} & := & (\exists x)A^{\triangleright}\end{array}$$

Theorem 4.1.3 *Let A be an $\mathcal{L}$ formula and $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N}) \vdash A$. Then we can show that:*

$$\mathsf{BON} + (\hat{\mathcal{L}}\text{-}\mathsf{I_N}) \ \vdash \ A^{\triangleright}$$

PROOF We can prove this theorem by induction on the length of derivation. We are in the logic of partial terms and in BON we have the same logical axioms and the same rules of inference as in PRON. In addition, the translation of the induction scheme $(\mathcal{L}\text{-}\mathsf{I_N})$ is an instance of $(\hat{\mathcal{L}}\text{-}\mathsf{I_N})$. So we have to care only about the non-logical axioms. We show the proof for the axioms of the recursor r. The translations of the axioms (11) and (12) of PRON are the following $\hat{\mathcal{L}}$ formulas:

$(11)^{\triangleright}$ $\quad \mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)\downarrow \wedge\, \mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)(\hat{\mathsf{p}}a\mathsf{0}) \simeq fa$

$(12)^{\triangleright}$ $\quad b \in \mathsf{N} \rightarrow \mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)(\hat{\mathsf{p}}a(\hat{\mathsf{s}}_{\mathsf{N}}b)) \simeq g(\hat{\mathsf{p}}(\hat{\mathsf{p}}ab)(\mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)(\hat{\mathsf{p}}ab)))$

1. We have $\mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg) \simeq \mathsf{fix}(t(\hat{\mathsf{p}}fg))$ and due to theorem 2.3.6 we know that $\mathsf{fix}\, r\downarrow$ for every $\hat{\mathcal{L}}$ term r, so we have $\mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)\downarrow$.

2. $\mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)(\hat{\mathsf{p}}a\mathsf{0}) \simeq \mathsf{fix}(t(\hat{\mathsf{p}}fg))(\hat{\mathsf{p}}a\mathsf{0}) \simeq t(\hat{\mathsf{p}}fg)(\mathsf{fix}(t(\hat{\mathsf{p}}fg)))(\hat{\mathsf{p}}a\mathsf{0}) \simeq$
   $t(\hat{\mathsf{p}}fg)(r^{\triangleright}(\hat{\mathsf{p}}fg))(\hat{\mathsf{p}}a\mathsf{0}) \simeq (\lambda z.fa)\mathsf{0} \simeq fa$

3. Suppose, that b is in N. Then we can prove
   $\mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)(\hat{\mathsf{p}}a(\hat{\mathsf{s}}_{\mathsf{N}}b)) \simeq \mathsf{fix}(t(\hat{\mathsf{p}}fg))(\hat{\mathsf{p}}a(\hat{\mathsf{s}}_{\mathsf{N}}b)) \simeq$
   $t(\hat{\mathsf{p}}fg)(\mathsf{fix}(t(\hat{\mathsf{p}}fg)))(\hat{\mathsf{p}}a(\hat{\mathsf{s}}_{\mathsf{N}}b)) \simeq t(\hat{\mathsf{p}}fg)(r^{\triangleright}(\hat{\mathsf{p}}fg))(\hat{\mathsf{p}}a(\hat{\mathsf{s}}_{\mathsf{N}}b)) \simeq$
   $(\lambda z.g(\hat{\mathsf{p}}(s(\hat{\mathsf{p}}a(\hat{\mathsf{s}}_{\mathsf{N}}b)))(r^{\triangleright}(\hat{\mathsf{p}}fg)(s(\hat{\mathsf{p}}a(\hat{\mathsf{s}}_{\mathsf{N}}b))))))\mathsf{0} \simeq$
   $(\lambda z.g(\hat{\mathsf{p}}(\hat{\mathsf{p}}ab)(\mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)(\hat{\mathsf{p}}ab))))\mathsf{0} \simeq g(\hat{\mathsf{p}}(\hat{\mathsf{p}}ab)(\mathsf{r}^{\triangleright}(\hat{\mathsf{p}}fg)(\hat{\mathsf{p}}ab)))$ $\square$

Corollary 4.1.4 $\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N}) \equiv \mathsf{PA}$ *with theorem 3.1.4.* $\square$

4.2 ACA contains $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N})$

In this section, we will formalize a model of $\mathsf{PRON}(\boldsymbol{\mu})$ in ACA. To realize this model we only have to change the interpretations of the function symbol $\odot$ and the constants b_2, a_2, and r in the recursion theoretic model $\mathcal{M}$. Additionally, we have to define the interpretation of the constant $\boldsymbol{\mu}$.

Definition 4.2.1 *Let $\mathcal{M}_0$ be the model $\mathcal{M}$ with the following changes and addition:*

$$
\begin{array}{rcl}
\odot^{\mathcal{M}_0} & := & \mu Prim Ev \\
\mathsf{b_2}^{\mathcal{M}_0} & := & \langle 3, 1, s_1, \mathsf{p_0}^{\mathcal{M}}, \mathsf{p_1}^{\mathcal{M}} \rangle \textit{ while } s_1 \textit{ is the} \\
 & & \textit{index of the primitive recursive function} \\
 & & (x, y) \rightarrow \\
 & & \quad \begin{cases} \langle 3, 1, x, y \rangle & \textit{if } x \in \mu Prim \wedge (x)_1 = 1 \wedge \\ & \quad y \in \mu Prim \wedge (y)_1 = 1 \\ \langle 3, 1, x, \langle 1, 1, 0 \rangle \rangle & \textit{if } x \in \mu Prim \wedge (x)_1 = 1 \wedge \\ & \quad (y \notin \mu Prim \vee (y)_1 \neq 1) \\ \langle 1, 1, 0 \rangle & \textit{otherwise} \end{cases} \\
\mathsf{a_2}^{\mathcal{M}_0} & := & \langle 3, 1, s_2, \langle 1, 1, <>^{\mathcal{M}} \rangle, \mathsf{p_0}^{\mathcal{M}}, \mathsf{p_1}^{\mathcal{M}} \rangle \textit{ while } s_2 \textit{ is the} \\
 & & \textit{index of the primitive recursive function} \\
 & & (x, y, z) \rightarrow \\
 & & \quad \begin{cases} \langle 3, 1, x, y, z \rangle & \textit{if } y \in \mu Prim \wedge (y)_1 = 1 \wedge \\ & \quad z \in \mu Prim \wedge (z)_1 = 1 \\ \langle 3, 1, x, y, \langle 1, 1, 0 \rangle \rangle & \textit{if } y \in \mu Prim \wedge (y)_1 = 1 \wedge \\ & \quad (z \notin \mu Prim \vee (z)_1 \neq 1) \\ \langle 3, 1, x, \langle 1, 1, 0 \rangle, z \rangle & \textit{if } (y \notin \mu Prim \vee (y)_1 \neq 1) \wedge \\ & \quad z \in \mu Prim \wedge (z)_1 = 1 \\ \langle 3, 1, x, \langle 1, 1, 0 \rangle, \langle 1, 1, 0 \rangle \rangle & \textit{otherwise} \end{cases} \\
\mathsf{r}^{\mathcal{M}_0} & := & \textit{index of the primitive recursive function} \\
 & & x \rightarrow \\
 & & \quad \begin{cases} \langle 4, 2, (x)_0, \overline{\overline{\overline{(x)_1}}} \rangle' & \textit{if } (x)_0 \in \mu Prim \wedge (x)_{0,1} = 1 \wedge \\ & \quad (x)_1 \in \mu Prim \wedge (x)_{1,1} = 1 \\ \langle 4, 2, (x)_0, \langle 1, 3, 0 \rangle \rangle' & \textit{if } (x)_0 \in \mu Prim \wedge (x)_{0,1} = 1 \wedge \\ & \quad ((x)_1 \notin \mu Prim \vee (x)_{1,1} \neq 1) \\ \langle 4, 2, \langle 1, 1, 0 \rangle, \overline{\overline{\overline{(x)_1}}} \rangle' & \textit{if } ((x)_0 \notin \mu Prim \vee (x)_{0,1} \neq 1) \wedge \\ & \quad (x)_1 \in \mu Prim \wedge (x)_{1,1} = 1 \\ \langle 1, 1, 0 \rangle & \textit{otherwise} \end{cases} \\
\boldsymbol{\mu}^{\mathcal{M}_0} & := & \textit{index of the primitive recursive function} \\
 & & x \rightarrow \begin{cases} \langle 5, 1, \widetilde{x} \rangle & \textit{if } x \in \mu Prim \wedge (x)_1 = 1 \\ \langle 1, 1, 0 \rangle & \textit{otherwise} \end{cases}
\end{array}
$$

Theorem 4.2.2 *$\mathcal{M}_0$ is a model of* $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathsf{Tot}) + (\mathsf{Nat})$.

PROOF The proof of this theorem is almost the same as of theorem 2.2.2. We only have to check the axioms about the constant $\boldsymbol{\mu}$. The axiom $(\boldsymbol{\mu}.1)$

is satisfied because the universe of $\mathcal{M}_0$ is the set $\mathbb{N}$ of all natural numbers. We show that the axiom ($\boldsymbol{\mu}$.2) is satisfied, as well:
Suppose, that $f^{\mathcal{M}_0}$ is the index of a unary function of $PRIM(\mu)$. Then we can prove $(\boldsymbol{\mu} fa)^{\mathcal{M}_0} = [[\boldsymbol{\mu}^{\mathcal{M}_0}](f^{\mathcal{M}_0})](a^{\mathcal{M}_0}) = [\langle 5, 1, \widetilde{f^{\mathcal{M}_0}}\rangle](a^{\mathcal{M}_0}) = Zero^1(\widetilde{f^{\mathcal{M}_0}})(a^{\mathcal{M}_0})$. If there exists a natural number x so that $[f^{\mathcal{M}_0}](\langle a^{\mathcal{M}_0}, x\rangle) = 0$, we immediately know that $[\widetilde{f^{\mathcal{M}_0}}](a^{\mathcal{M}_0}, x) = 0$ and $(\boldsymbol{\mu} fa)^{\mathcal{M}_0} = \min\{y \mid [f^{\mathcal{M}_0}](\langle a^{\mathcal{M}_0}, y\rangle) = 0\}$, so we have $[f^{\mathcal{M}_0}](\langle a^{\mathcal{M}_0}, (\boldsymbol{\mu} fa)^{\mathcal{M}_0}\rangle) = 0$.
If $f^{\mathcal{M}_0} \notin \mu Prim$ or $(f^{\mathcal{M}_0})_1 \neq 1$ then $(\boldsymbol{\mu} fa)^{\mathcal{M}_0} = [\langle 1, 1, 0\rangle](a^{\mathcal{M}}) = 0$ and $\mu PrimEv(f^{\mathcal{M}_0}, x) = 0$ for every natural number x, which implies $\mu PrimEv(f^{\mathcal{M}_0}, \langle a^{\mathcal{M}_0}, (\boldsymbol{\mu} fa)^{\mathcal{M}_0}\rangle) = 0$. □

Let n, b_0, ..., b_{n-1}, c be arbitrary natural numbers and e be an n-ary index of $\mu Prim$. Our next goal is to formalize the expression $[e](\vec{b}) = c$ in ACA. For this purpose, we will construct triples $s := \langle e, \langle b_0, \ldots, b_{n-1}\rangle, c\rangle$ and for every natural number l we will define a set X so that $(X)_l$ contains all triples with $Lev(e) \leq l$. The slices $(X)_l$ of X are defined by $s \in (X)_l \leftrightarrow \langle s, l\rangle \in X$.
In the case of $\mathsf{BON}(\hat{\boldsymbol{\mu}})$ a hierarchy with finite levels is not sufficient. This is witnessed by the fact that the ω-jump (and much more) is definable in models of $\mathsf{BON}(\hat{\boldsymbol{\mu}})$. This latter fact is due to the presence of the $\hat{\mathsf{s}}$ combinator, which allows for diagonalisation at limit ordinals; this diagonalisation, however, cannot be obtained by primitive recursive means. For the model construction in the case of $\mathsf{BON}(\hat{\boldsymbol{\mu}})$, cf. Feferman and Jäger [5].
The following formulas A and B will prepare the construction of the intended hierarchy.

Definition 4.2.3 *For triples s with indices of level 0 we define the Π_0 formula $A[s]$ as follows:*

$$\begin{aligned} A &:= A_0 \land (s)_0 \in \mu Prim \land A_1 \\ A_0 &:= s \in Seq \land lh(s) = 3 \land (s)_1 \in Seq \land lh((s)_1) = (s)_{0,1} \\ A_1 &:= [(s)_{0,0} = 0 \land (\exists b \leq s)((s)_{1,0} = b \land (s)_2 = b+1)] \lor \\ &\quad\;\; [(s)_{0,0} = 1 \land (s)_2 = (s)_{0,2}] \lor [(s)_{0,0} = 2 \land (s)_2 = (s)_{1,(s)_{0,2}}] \end{aligned}$$

Definition 4.2.4 *For triples s with indices of level $l+1$ we define the arithmetic $\mathcal{L}_2$ formula $B[U, s, l]$ as follows:*

$$\begin{aligned} B &:= s \in U \lor [B_0 \land (s)_0 \in \mu Prim \land (B_1 \lor B_2 \lor B_3)] \\ B_0 &:= s \in Seq \land lh(s) = 3 \land (s)_1 \in Seq \land lh((s)_1) = (s)_{0,1} \land \\ &\quad\;\; Lev((s)_0) = l+1 \end{aligned}$$

$$
\begin{aligned}
B_1 \;\; := \;\; & (s)_{0,0} = 3 \wedge (\exists c_0, \ldots, c_{lh((s)_0) \dot- 4}) \\
& \quad [\langle (s)_{0,2}, \langle c_0, \ldots, c_{lh((s)_0) \dot- 4} \rangle, (s)_2 \rangle \in U \wedge \\
& \quad (\forall d < lh((s)_0) \dot- 3)(\langle (s)_{0,d+3}, (s)_1, c_k \rangle \in U)]
\end{aligned}
$$

$$
\begin{aligned}
B_2 \;\; := \;\; & (s)_{0,0} = 4 \wedge (\exists b)[b \in \mathit{Seq} \wedge lh(b) = (s)_{1, lh((s)_1) \dot- 1} + 1 \wedge \\
& \langle (s)_{0,2}, \langle (s)_{1,0}, \ldots, (s)_{1, lh((s)_1) \dot- 2} \rangle, (b)_0 \rangle \in U \wedge \\
& (s)_2 = (b)_{lh(b) \dot- 1} \wedge (\forall d < lh(b) \dot- 1) \\
& \quad (\langle (s)_{0,3}, \langle (s)_{1,0}, \ldots, (s)_{1, lh((s)_1) \dot- 2}, d, (b)_d \rangle, (b)_{d+1} \rangle \in U)]
\end{aligned}
$$

$$
\begin{aligned}
B_3 \;\; := \;\; & (s)_{0,0} = 5 \wedge \\
& [[(\exists b)(\langle (s)_{0,2}, \langle (s)_{1,0}, \ldots, (s)_{1, lh((s)_1) \dot- 1}, b \rangle, 0 \rangle \in U \wedge (s)_2 = b) \wedge \\
& \quad (\forall b)(\langle (s)_{0,2}, \langle (s)_{1,0}, \ldots, (s)_{1, lh((s)_1) \dot- 1}, b \rangle, 0 \rangle \in U \rightarrow b \geq (s)_2)] \vee \\
& [(\forall b)(\langle (s)_{0,2}, \langle (s)_{1,0}, \ldots, (s)_{1, lh((s)_1) \dot- 1}, b \rangle, 0 \rangle \notin U) \wedge (s)_2 = 0]]
\end{aligned}
$$

Now we are able to define the hierarchy $\mathcal{H}_0$ which collects all the triples $s = \langle e, \langle \vec{b} \rangle, c \rangle$ satisfying $[e](\vec{b}) = c$ up to a certain level.

Definition 4.2.5 $\mathcal{H}_0[W, u]$ *is the formula which tests if the slices* $(W)_l$ *contain the triples with indices of level* l *for all* $l \leq u$*:*

$$
\begin{aligned}
\mathcal{H}_0 \;\; := \;\; & (\forall s)[(s \in (W)_0 \leftrightarrow A(s)) \wedge \\
& (\forall l < u)(s \in (W)_{l+1} \leftrightarrow B((W)_l, s, l))]
\end{aligned}
$$

Note that $\mathcal{H}_0$ *is an arithmetic* $\mathcal{L}_2$ *formula.*

If we want to show in ACA the existence of a set X so that $\mathcal{H}_0(X, z)$ for a given z, we have to verify that $(X)_l$ is a set in ACA for every $l < u$.

Lemma 4.2.6 ACA *proves* $(\forall z)(\exists X)\mathcal{H}_0(X, z)$

Proof With Σ^1_1 induction on z.

$z = 0$: $(\exists Y)(\forall s)(s \in Y \leftrightarrow A(s))$ is an instance of arithmetic comprehension, because A is Π_0, so we get the desired set X with arithmetic comprehension by $(\exists X)(\forall p)[p \in X \leftrightarrow (\exists s \in Y)(p = \langle s, 0 \rangle)]$.

$z \rightarrow z + 1$: by induction hypothesis we have $(\exists Y)\mathcal{H}_0(Y, z)$, that is we know that $(Y)_z$ is a set. $(\exists Z)(\forall s)(s \in Z \leftrightarrow B((Y)_z, s, z))$ is an instance of arithmetic comprehension, because B is an arithmetic $\mathcal{L}_2$ formula, so we get the desired set X with arithmetic comprehension by
$(\exists X)(\forall p)[p \in X \leftrightarrow p \in Y \vee (\exists s \in Z)(p = \langle s, z + 1 \rangle)]$.
It is easy to see that $\mathcal{H}_0(X, z + 1)$. □

We have constructed the hierarchy $\mathcal{H}_0$ like that to make sure that it is unique and increasing. We will prove the first property in the following lemma, the second one is very easy to see.

Lemma 4.2.7 ACA *proves*

$$\mathcal{H}_0(W_0, u_0) \wedge \mathcal{H}_0(W_1, u_1) \wedge u_0 \leq u_1 \rightarrow (\forall l \leq u_0)((W_0)_l = (W_1)_l)$$

PROOF With arithmetic induction on l.

$l = 0$**:** Given $\mathcal{H}_0(W_0, u_0)$ and $\mathcal{H}_0(W_1, u_1)$ we know that $(\forall s)(s \in (W_0)_0 \leftrightarrow A(s))$ and $(\forall s)(s \in (W_1)_0 \leftrightarrow A(s))$, so we have $(W_0)_0 = (W_1)_0$.

$l \rightarrow l + 1$**:** If $l + 1 \leq u_0$ then $l < u_0$ and we have $(W_0)_l = (W_1)_l$ by induction hypothesis. Given $\mathcal{H}_0(W_0, u_0)$ and $\mathcal{H}_0(W_1, u_1)$ we know that $(\forall s)(s \in (W_0)_{l+1} \leftrightarrow B((W_0)_l, s, l))$ and $(\forall s)(s \in (W_1)_{l+1} \leftrightarrow B((W_1)_l, s, l))$, so we have $(W_0)_{l+1} = (W_1)_{l+1}$. □

We are now ready to define the Application for the embedding of $\mathsf{PRON}(\boldsymbol{\mu})+ (\mathcal{L}\text{-}\mathsf{I_N})$ in ACA. It must be a formula functional in the third argument and it must contain every triple of our hierarchy $\mathcal{H}_0$ with a unary index. In fact, the following definition is the precise formalization of $\mu PrimEv$ in arithmetics.

Definition 4.2.8 *We define* $\boldsymbol{\mu}\mathsf{App}[u, v, w]$ *to be the following formula:*

$$\begin{aligned} \boldsymbol{\mu}\mathsf{App} \;:=\; & [u \in \mu Prim \wedge (u)_1 = 1 \wedge \\ & \quad (\exists X)(\mathcal{H}_0(X, Lev(u)) \wedge \langle u, \langle v \rangle, w \rangle \in (X)_{Lev(u)})] \vee \\ & [(u \notin \mu Prim \vee (u)_1 \neq 1) \wedge w = 0] \end{aligned}$$

Note that $\boldsymbol{\mu}\mathsf{App}$ *is equivalent to a* Σ^1_1 *formula, because* $\mathcal{H}_0$ *is an arithmetic* $\mathcal{L}_2$ *formula. It is even equivalent to a* Π^1_1 *formula.*

We have made sure that we have in our application $\boldsymbol{\mu}\mathsf{App}$ every triple with a unary index of $\mu Prim$. It is also true that the application is functional in the third argument.

Lemma 4.2.9 ACA *proves*

$$(\forall x, y, z_0, z_1)(\boldsymbol{\mu}\mathsf{App}(x, y, z_0) \wedge \boldsymbol{\mu}\mathsf{App}(x, y, z_1) \rightarrow z_0 = z_1)$$

PROOF If $x \notin \mu Prim$ or $(x)_1 \neq 1$ then $\boldsymbol{\mu}\mathsf{App}(x, y, z)$ implies $z = 0$ and we have finished the proof.
If x is a unary index of $\mu Prim$, we can prove this lemma by induction on the level $l := Lev(x)$ of x:

$l = 0$**:** We have the three cases $(x)_0 = 0$ (successor), $(x)_0 = 1$ (constant function), and $(x)_0 = 2$ (projections), which are easy to prove.

$l \to l+1$**:** In the cases $(x)_0 = 3$ (composition) and $(x)_0 = 5$ (non-constructive μ operator) the proof immediately follows by induction hypothesis. In the case $(x)_0 = 4$ (primitive recursion) we need induction on the argument. $\Box$

We are going to define a translation $\cdot^\star$ of formulas from $\mathcal{L}$ to $\mathcal{L}_2$. First, we will define a translation from $\mathcal{L}$ terms to $\mathcal{L}_2$ formulas. The translation of a constant c is the numeral $\overline{m}$, if m is the interpretation $c^{\mathcal{M}_0}$ of c in $\mathcal{M}_0$.

Definition 4.2.10 *For every $\mathcal{L}$ term t we define the $\mathcal{L}_2$ formula $V_t^\star[x]$, so that the variable x does not occur in t, by induction on the build-up of t as follows:*

1. *If t is a variable of $\mathcal{L}$ then $V_t^\star := (x = t)$*
2. *If t is a constant of $\mathcal{L}$ then $V_t^\star := (x = \overline{t^{\mathcal{M}_0}})$*
3. *If t is the $\mathcal{L}$ term $<r, s>$ then*
   $V_t^\star := (\exists z_0, z_1)[V_r^\star(z_0) \wedge V_s^\star(z_1) \wedge x = \langle z_0, z_1\rangle]$
4. *If t is the $\mathcal{L}$ term (rs) then*
   $V_t^\star := (\exists z_0, z_1)[V_r^\star(z_0) \wedge V_s^\star(z_1) \wedge \boldsymbol{\mu}\mathsf{App}(z_0, z_1, x)]$

Note that every formula $V_t^\star$ is equivalent to a Σ_1^1 (Π_1^1) formula, because $\boldsymbol{\mu}\mathsf{App}$ is Σ_1^1 (Π_1^1), as well.

Remark 4.2.11 *Let t be an arbitrary $\mathcal{L}$ term. Then* ACA *proves $(\exists! x)V_t^\star(x)$ by a simple induction on the build-up of t.*

Definition 4.2.12 *For every $\mathcal{L}$ formula A we define the $\mathcal{L}_2$ formula $A^\star$ by induction on the build-up of A as follows:*

1. *If $A = \mathsf{N}(t)$ or $A = t\downarrow$ then $A^\star := (\exists x)V_t^\star(x)$*
2. *If $A = (s = t)$ then $A^\star := (\exists x)(V_s^\star(x) \wedge V_t^\star(x))$*
3. *If $A = \neg B$ then $A^\star := \neg B^\star$*
4. *If $A = B \vee C$ then $A^\star := B^\star \vee C^\star$*
5. *If $A = (\exists x)B$ then $A^\star := (\exists x)B^\star$*

The following lemma shows that we have defined a convenient translation.

Lemma 4.2.13 *Let A be an arbitrary $\mathcal{L}_1$ formula. Then we can prove the following assertion:*

$$\mathsf{ACA} \;\vdash\; A \leftrightarrow (A^\circ)^\star$$

Proof This is verified by a straightforward induction on the build-up of A with remark 4.2.11. $\square$

Now we can show that ACA proves the translation of an $\mathcal{L}$ formula provable in $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N})$.

Theorem 4.2.14 *Let A be an $\mathcal{L}$ formula and $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N}) \vdash A$. Then we can derive $A^\star$ in ACA.*

Proof We can prove this theorem by induction on the length of derivation. The logical axioms are easy to verify and in ACA we have the same rules of inference as in $\mathsf{PRON}(\boldsymbol{\mu})$. Because $A^\star$ is an $\mathcal{L}_2$ formula, the translation of the induction scheme $(\mathcal{L}\text{-}\mathsf{I_N})$ can be derived with $(\Pi^1_\infty\text{-}\mathsf{I_N})$.
Now let A be a non-logical axiom of $\mathsf{PRON}(\boldsymbol{\mu})$. This axiom is satisfied in the model $\mathcal{M}_0$ as we have proved in theorem 4.2.2. We have defined $\boldsymbol{\mu}\mathsf{App}$ in a way that the properties of indices are satisfied. Finally, the definition of the translation $\cdot^\star$ guarantees that $A^\star$ is derivable in ACA.
As an example we have written the exact translations of $(\boldsymbol{\mu}.1)$ and $(\boldsymbol{\mu}.2)$ in appendix B. $\square$

We know the strength of $\mathsf{PRON}(\boldsymbol{\mu})+(\mathcal{L}\text{-}\mathsf{I_N})$ because we have done embeddings in both directions.

Corollary 4.2.15 $\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N}) \equiv \mathsf{ACA}$ *with theorem 3.2.5.* $\square$

4.3 Π^1_1-CA contains PRON(SUS) + $(\mathcal{L}\text{-}\mathsf{I_N})$

For the last embedding we will formalize a model of $\mathsf{PRON(SUS)}$ in Π^1_1-CA. This time, we only have to change the interpretations of the function symbol $\odot$ and the constants b_2, a_2, r, and $\boldsymbol{\mu}$ in the model $\mathcal{M}_0$ of $\mathsf{PRON}(\boldsymbol{\mu})$. Additionally, we have to define the interpretation of the constant E_1.

Definition 4.3.1 *Let $\mathcal{M}_1$ be the model $\mathcal{M}_0$ with the following changes and addition:*

$$
\begin{array}{rcl}
\odot^{\mathcal{M}_1} & := & \mathit{SusPrimEv} \\
\mathsf{b_2}^{\mathcal{M}_1} & := & \langle 3,1,s_1,\mathsf{p_0}^{\mathcal{M}},\mathsf{p_1}^{\mathcal{M}}\rangle \textit{ while } s_1 \textit{ is the} \\
 & & \textit{index of the primitive recursive function} \\
 & & (x,y) \to \\
 & & \quad \begin{cases} \langle 3,1,x,y\rangle & \textit{if } x \in \mathit{SusPrim} \wedge (x)_1 = 1 \wedge \\ & \quad y \in \mathit{SusPrim} \wedge (y)_1 = 1 \\ \langle 3,1,x,\langle 1,1,0\rangle\rangle & \textit{if } x \in \mathit{SusPrim} \wedge (x)_1 = 1 \wedge \\ & \quad (y \notin \mathit{SusPrim} \vee (y)_1 \neq 1) \\ \langle 1,1,0\rangle & \textit{otherwise} \end{cases} \\
\mathsf{a_2}^{\mathcal{M}_1} & := & \langle 3,1,s_2,\langle 1,1,<>^{\mathcal{M}}\rangle,\mathsf{p_0}^{\mathcal{M}},\mathsf{p_1}^{\mathcal{M}}\rangle \textit{ while } s_2 \textit{ is the} \\
 & & \textit{index of the primitive recursive function} \\
 & & (x,y,z) \to \\
 & & \quad \begin{cases} \langle 3,1,x,y,z\rangle & \textit{if } y \in \mathit{SusPrim} \wedge (y)_1 = 1 \wedge \\ & \quad z \in \mathit{SusPrim} \wedge (z)_1 = 1 \\ \langle 3,1,x,y,\langle 1,1,0\rangle\rangle & \textit{if } y \in \mathit{SusPrim} \wedge (y)_1 = 1 \wedge \\ & \quad (z \notin \mathit{SusPrim} \vee (z)_1 \neq 1) \\ \langle 3,1,x,\langle 1,1,0\rangle,z\rangle & \textit{if } (y \notin \mathit{SusPrim} \vee (y)_1 \neq 1) \wedge \\ & \quad z \in \mathit{SusPrim} \wedge (z)_1 = 1 \\ \langle 3,1,x,\langle 1,1,0\rangle,\langle 1,1,0\rangle\rangle & \textit{otherwise} \end{cases} \\
\mathsf{r}^{\mathcal{M}_1} & := & \textit{index of the primitive recursive function} \\
 & & x \to \\
 & & \quad \begin{cases} \langle 4,2,(x)_0,\overline{\overline{\overline{(x)_1}}}\rangle' & \textit{if } (x)_0 \in \mathit{SusPrim} \wedge (x)_{0,1} = 1 \wedge \\ & \quad (x)_1 \in \mathit{SusPrim} \wedge (x)_{1,1} = 1 \\ \langle 4,2,(x)_0,\langle 1,3,0\rangle\rangle' & \textit{if } (x)_0 \in \mathit{SusPrim} \wedge (x)_{0,1} = 1 \wedge \\ & \quad ((x)_1 \notin \mathit{SusPrim} \vee (x)_{1,1} \neq 1) \\ \langle 4,2,\langle 1,1,0\rangle,\overline{\overline{\overline{(x)_1}}}\rangle' & \textit{if } ((x)_0 \notin \mathit{SusPrim} \vee (x)_{0,1} \neq 1) \wedge \\ & \quad (x)_1 \in \mathit{SusPrim} \wedge (x)_{1,1} = 1 \\ \langle 1,1,0\rangle & \textit{otherwise} \end{cases} \\
\boldsymbol{\mu}^{\mathcal{M}_1} & := & \textit{index of the primitive recursive function} \\
 & & x \to \begin{cases} \langle 5,1,\widetilde{x}\rangle & \textit{if } x \in \mathit{SusPrim} \wedge (x)_1 = 1 \\ \langle 1,1,0\rangle & \textit{otherwise} \end{cases} \\
\mathsf{E_1}^{\mathcal{M}_1} & := & \textit{index of the primitive recursive function} \\
 & & x \to \begin{cases} \langle 6,1,\overline{\overline{x}}\rangle & \textit{if } x \in \mathit{SusPrim} \wedge (x)_1 = 1 \\ \langle 1,1,0\rangle & \textit{otherwise} \end{cases}
\end{array}
$$

Remark 4.3.2 *We are not yet able to show that $\mathcal{M}_1$ is a model of* $\mathsf{PRON(SUS)}$*, because we have quantifiers over arbitrary functions in the evaluation function. Suppose, that $f^{\mathcal{M}_1}$ is a unary index of SusPrim. Then we can prove the following equation:* $(\mathsf{E}_1 fa)^{\mathcal{M}_1} = [\langle 6, 1, \overline{\overline{f}}\rangle](a^{\mathcal{M}_1})$

$$= \begin{cases} 0 & \textit{if } (\exists G)(\forall z)([\overline{\overline{f}}](a^{\mathcal{M}_1}, G(\mathcal{S}(z)), G(z)) = 0) \\ 1 & \textit{otherwise} \end{cases}$$

It was not possible to define $[\langle 6, n, f\rangle](\vec{x})$ as

$$\begin{cases} 0 & \textit{if } (\exists g \in \mathit{SusPrim})(\forall z)([f](\vec{x}, [g](\mathcal{S}(z)), [g](z)) = 0) \\ 1 & \textit{otherwise} \end{cases}$$

because definition 1.1.12 is inductive. After having formalized the application function in Π^1_1-CA*, we will show that both conditions are equivalent and that $\mathcal{M}_1$ is a model of* $\mathsf{PRON(SUS)} + (\mathsf{Tot}) + (\mathsf{Nat})$*.*

Let n, b_0, $\ldots$, b_{n-1}, c be arbitrary natural numbers and e be an n-ary index of *SusPrim*. Our next goal is to formalize the expression $[e](\vec{b}) = c$ in Π^1_1-CA. For this purpose, we will construct triples $s := \langle e, \langle b_0, \ldots, b_{n-1}\rangle, c\rangle$ and for every natural number l we will define a set X so that $(X)_l$ contains all triples with $\mathit{Lev}(e) \leq l$.
In the case of SUS a hierarchy with finite levels is not sufficient for similar reasons as in the case of $\mathsf{BON}(\hat{\boldsymbol{\mu}})$. In this case the ω-hyperjump (and much more) is definable in models of SUS. For the model construction in the case of SUS, cf. Jäger and Strahm [8].
The following formulas C and D will prepare the construction of the intended hierarchy.

Definition 4.3.3 *For triples s with indices of level 0 we define the Π_0 formula $C[s]$ as follows:*

$$C \quad := \quad A_0 \wedge (s)_0 \in \mathit{SusPrim} \wedge A_1$$

The formulas A_0 and A_1 are the same as in definition 4.2.3.

Definition 4.3.4 *For triples s with indices of level $l{+}1$ we define the formula $D[U, s, l]$ as follows:*

$$D \quad := \quad s \in U \vee [B_0 \wedge (s)_0 \in \mu\mathit{Prim} \wedge (B_1 \vee B_2 \vee B_3 \vee D_0 \vee D_1)]$$

$$D_0 \quad := \quad (s)_{0,0} = 6 \wedge (s)_2 = 0 \wedge (\exists G)(\forall b) \\ (\langle\langle (s)_{0,2}, \langle (s)_{1,0}, \ldots, (s)_{1,lh((s)_1) \dot{-} 1}, G(\mathcal{S}(b)), G(b)\rangle, 0\rangle \in U)$$

$$D_1 \quad := \quad (s)_{0,0} = 6 \wedge (s)_2 = 1 \wedge (\forall G)(\exists b) \\ (\langle\langle (s)_{0,2}, \langle (s)_{1,0}, \ldots, (s)_{1,lh((s)_1) \dot{-} 1}, G(\mathcal{S}(b)), G(b)\rangle, 0\rangle \notin U)$$

The formulas B_0, B_1, B_2, and B_3 are the same as in definition 4.2.4. Note that D is equivalent to a Σ^1_2 (Π^1_2) formula.

Now we are able to define the hierarchy $\mathcal{H}_1$ which collects all the triples $s = \langle e, \langle\vec{b}\rangle, c\rangle$ satisfying $[e](\vec{b}) = c$ up to a certain level.

Definition 4.3.5 $\mathcal{H}_1[W, u]$ *is the formula which tests if the slices $(W)_l$ contain the triples with indices of level l for all $l \leq u$:*

$$\begin{aligned}\mathcal{H}_1 \quad := \quad & (\forall s)[(s \in (W)_0 \leftrightarrow C(s)) \wedge \\ & (\forall l < u)(s \in (W)_{l+1} \leftrightarrow D((W)_l, s, l))]\end{aligned}$$

As a preparation for the next step, we have to prove that we can apply the comprehension scheme (Σ^1_1-CA) in Π^1_1-CA.

Lemma 4.3.6 *Let A be an arbitrary Σ^1_1 formula. Then Π^1_1-CA proves the following formula:*

$$(\exists X)(\forall y)[y \in X \leftrightarrow A(y)]$$

Proof The formula $\neg A$ is Π^1_1 and we can prove the existence of a set X satisfying $(\forall y)(y \in X \leftrightarrow \neg A(y))$ with the (Π^1_1-CA) comprehension scheme. Let us now take the formula $\neg(u \in X)$ which is arithmetic in X. With arithmetic comprehension we get the existence of a set Y satisfying $(\forall z)(z \in Y \leftrightarrow \neg(z \in X))$ and we can easily see that $(\forall z)(z \in Y \leftrightarrow A(z))$. □

If we want to show in Π^1_1-CA the existence of a set X so that $\mathcal{H}_1(X, z)$ for a given z, we have to verify that $(X)_l$ is a set in Π^1_1-CA for every $l < u$.

Lemma 4.3.7 Π^1_1-CA *proves* $(\forall z)(\exists X)\mathcal{H}_1(X, z)$.

Proof With induction on z.

$z = 0$**:** $(\exists Y)(\forall s)(s \in Y \leftrightarrow C(s))$ is an instance of arithmetic comprehension, because C is Π_0, so we get the desired set X with arithmetic comprehension by $(\exists X)(\forall p)[p \in X \leftrightarrow (\exists s \in Y)(p = \langle s, 0\rangle)]$.

$z \to z + 1$**:** by induction hypothesis we have $(\exists Y)\mathcal{H}_1(Y, z)$, that is we know that $(Y)_z$ is a set. The formula D is Σ^1_2, but we only have Π^1_1 and Σ^1_1 comprehension. So we have to use the fact that D is provably equivalent to $D_2 \vee D_3$, if we define D_2 and D_3 to be the following formulas:
$D_2 := s \in U \vee (B_0 \wedge (s)_0 \in \mu Prim \wedge (B_1 \vee B_2 \vee B_3 \vee D_0))$
$D_3 := s \in U \vee (B_0 \wedge (s)_0 \in \mu Prim \wedge D_1)$
We can immediately see that D_2 is Σ^1_1 and D_3 is Π^1_1. Now we can prove

the existence of two sets Y_0, Y_1 so that $(\forall s)(s \in Y_0 \leftrightarrow D_2((Y)_z, s, z))$ and $(\forall s)(s \in Y_1 \leftrightarrow D_3((Y)_z, s, z))$. The formula $(u \in Y_0) \vee (u \in Y_1)$ is arithmetic in Y_0 and Y_1, so we get a set Z by $(\forall s)(s \in Z \leftrightarrow (u \in Y_0) \vee (u \in Y_1)$ with arithmetic comprehension and we know that $(\forall s)(s \in Z \leftrightarrow D((Y)_z, s, z))$. The desired set X is obtained by $(\exists X)(\forall p)(p \in X \leftrightarrow p \in Y \vee (\exists s \in Z)(p = \langle s, z+1\rangle))$ with arithmetic comprehension, too. It is now clear that $\mathcal{H}_1(X, z+1)$. $\square$

We have constructed the hierarchy $\mathcal{H}_1$ like that to make sure that it is unique and increasing like the hierarchy $\mathcal{H}_0$ from definition 4.2.5.

Lemma 4.3.8 Π^1_1-CA *proves*

$$\mathcal{H}_1(W_0, u_0) \wedge \mathcal{H}_1(W_1, u_1) \wedge u_0 \leq u_1 \rightarrow (\forall l \leq u_0)((W_0)_l = (W_1)_l)$$

Proof The proof is exactly the same as the proof of lemma 4.2.7, with arithmetic induction on l. $\square$

We are ready to define the Application for the embedding of $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N})$ in Π^1_1-CA. It must be a formula functional in the third argument and it must contain every triple of our hierarchy $\mathcal{H}_1$ with a unary index. In fact, the following definition is the precise formalization of *SusPrimEv* in arithmetics.

Definition 4.3.9 *We define* $\mathsf{SusApp}[u, v, w]$ *to be the following formula:*

$$\begin{aligned} \mathsf{SusApp} \; := \; & [u \in \mathit{SusPrim} \wedge (u)_1 = 1 \wedge \\ & \quad (\exists X)(\mathcal{H}_1(X, \mathit{Lev}(u)) \wedge \langle u, \langle v\rangle, w\rangle \in (X)_{\mathit{Lev}(u)})] \vee \\ & [(u \notin \mathit{SusPrim} \vee (u)_1 \neq 1) \wedge w = 0] \end{aligned}$$

We have made sure that we have in our application SusApp every triple with a unary index of *SusPrim*. It is also true that the application is functional in the third argument.

Lemma 4.3.10 Π^1_1-CA *proves*

$$(\forall x, y, z_0, z_1)(\mathsf{SusApp}(x, y, z_0) \wedge \mathsf{SusApp}(x, y, z_1) \rightarrow z_0 = z_1)$$

Proof The proof is almost the same as of lemma 4.2.9. We only have additional indices s with $(s)_0 = 6$. $\square$

Before we can begin with the embedding, we have to prove that it is the same if we find a descending chain with an arbitrary function from outside the theory $\mathsf{PRON(SUS)}$ or with an index of *SusPrim*. For this purpose, we define the formalization of finding a descending chain.

Definition 4.3.11

$$\begin{array}{rcl} [\![e]\!](b) = c & := & \mathsf{SusApp}(e, b, c) \\ \mathsf{CDC} & := & (\forall x)([\![e]\!](\langle\langle a, [\![c]\!](\mathcal{S}(x))\rangle, [\![c]\!](x)\rangle) = 0) \\ \mathsf{FDC} & := & (\forall x)([\![e]\!](\langle\langle a, F(\mathcal{S}(x))\rangle, F(x)\rangle) = 0) \end{array}$$

Furthermore, we will need for every index a set which collects all the zeros of this index with respect to SusApp.

Lemma 4.3.12 Π^1_1-CA *proves*

$$(\exists X)(\forall y)(y \in X \leftrightarrow [\![e]\!](y) = 0)$$

In the following we call this set $ext(e)$.

PROOF Nevertheless we only have Π^1_1 comprehension, we can prove the existence of $ext(e)$ due to lemma 4.3.7. □

We prepare our proof of the equivalence of CDC and FDC with two technical lemmas.

Lemma 4.3.13 *Let $A[R, a, u, v_0, v_1]$ be an arithmetic $\mathcal{L}_2$ formula. Then there exist primitive recursive functions $\mathcal{F}$ and $\mathcal{G}$ so that* Π^1_1-CA *proves:*

$$\begin{array}{rcl} A(ext(e), a, u, v_0, v_1) & \leftrightarrow & [\![\mathcal{F}(e)]\!](\langle\langle\langle a, u\rangle, v_0\rangle, v_1\rangle) = 0 \\ [\![\langle 6, 1, \overline{\overline{\mathcal{F}(e)}}\rangle]\!](\langle a, u\rangle) = 0 & \leftrightarrow & [\![\mathcal{G}(e)]\!](\langle a, u\rangle) = 0 \end{array}$$

Lemma 4.3.14 *Let $B[R, S, a, u, v]$ be an arithmetic $\mathcal{L}_2$ formula. Further, assume that*

$$\Pi^1_1\text{-}\mathsf{CA} \ \vdash \ (\forall U, V)(\forall x, y)(\exists! z)B(U, V, x, y, z)$$

and let $\mathcal{F}_B$ denote the (class) function of B. Then there exists a primitive recursive function $\mathcal{H}$, so that Π^1_1-CA *proves:*

$$\begin{array}{rcl} [\![\mathcal{H}(e, f, a, u)]\!](0) & = & u \\ [\![\mathcal{H}(e, f, a, u)]\!](v + 1) & = & \mathcal{F}_B(ext(e), ext(f), a, [\![\mathcal{H}(e, f, a, u)]\!](v)) \end{array}$$

The proofs of these two lemmas are straightforward, although a bit tedious. We are now ready to turn to the theorem about the inside-outside property mentioned above.

Theorem 4.3.15 (Inside-outside property) Π^1_1-CA *proves*

$$(\exists c)\mathsf{CDC}(c, e, a) \ \leftrightarrow \ (\exists F)\mathsf{FDC}(F, e, a)$$

Proof Let us first prove the direction from the left to the right. Given a natural number c with $\mathsf{CDC}(c, e, a)$ we immediately have $(\forall x)(\exists! y)([\![c]\!](x) = y)$. Then we have to distinct the two following cases:

1. $c \notin \mathit{SusPrim} \vee (c)_1 \neq 1$ leads to $(\forall x)([\![c]\!](x) = 0)$ and $\mathsf{FDC}(F, e, a)$ holds for $F := Cs^1_0$.

2. $c \in \mathit{SusPrim} \wedge (c)_1 = 1$ leads to $(\exists X)\mathcal{H}_1(X, \mathit{Lev}(c))$ and $(\forall x)(\exists! y)(\langle c, \langle x\rangle, y\rangle \in X_{\mathit{Lev}(c)})$, so we know that $X_{\mathit{Lev}(c)}$ is a set. If we define the arithmetic formula $\varphi[u, v]$ by $\varphi := (\exists x, y)(\langle u, \langle x\rangle, y\rangle \in X_{\mathit{Lev}(u)} \wedge v = \langle x, y\rangle)$, we get $(\exists Z)(\forall z)(z \in Z \leftrightarrow \varphi(c, z))$ by arithmetic comprehension, so we have $(\forall x)(\exists! y)(\langle x, y\rangle \in Z)$. Using the graph principle we get $(\exists F)(\forall x)(\langle x, F(x)\rangle \in Z)$ and $(\forall x)(F(x) = [\![c]\!](x))$. It immediately follows that the function F satisfies the condition $\mathsf{FDC}(F, e, a)$.

The other direction is much more complicated and makes use of the two technical lemmas 4.3.13 and 4.3.14. Given a Function F with $\mathsf{FDC}(F, e, a)$ we can construct an index c satisfying $\mathsf{CDC}(c, e, a)$ with a very intuitive leftmost branch argument.
Let us first define the formula $A[R, a, u, v_0, v_1]$ as follows:

$$\begin{aligned} A \;:=\; & (\exists x, y)[x \in \mathit{Seq} \wedge \mathit{lh}(x) = y \wedge (x)_0 = u \wedge (x)_{y \dot- 1} = v_1 \;\wedge \\ & (\forall z < y \dot- 1)(\langle\langle a, (x)_{z+1}\rangle, (x)_z\rangle \in R)] \wedge \langle\langle a, v_0\rangle, v_1\rangle \in R \end{aligned}$$

$A(R, a, u, v_0, v_1)$ expresses that $\langle\langle a, v_0\rangle, v_1\rangle \in R$ and v_1 is accessible from u by means of R. From lemma 4.3.13 we know about the existence of two primitive recursive functions $\mathcal{F}$ and $\mathcal{G}$ so that

$$\begin{aligned} A(\mathit{ext}(e), a, u, v_0, v_1) \;&\leftrightarrow\; [\![\mathcal{F}(e)]\!](\langle\langle\langle a, u\rangle, v_0\rangle, v_1\rangle) = 0 \\ [\![\langle 6, 1, \overline{\overline{\mathcal{F}(e)}}\rangle]\!](\langle a, u\rangle) = 0 \;&\leftrightarrow\; [\![\mathcal{G}(e)]\!](\langle a, u\rangle) = 0 \end{aligned}$$

Further, we define the formula $B[R, S, a, u, v]$ as follows:

$$\begin{aligned} B \;:=\; & [\langle\langle a, v\rangle, u\rangle \in R \wedge \langle a, v\rangle \in S \;\wedge \\ & \quad (\forall w < v)(\langle\langle a, w\rangle, u\rangle \notin R \vee \langle a, w\rangle \notin S)] \;\vee \\ & [(\forall w)(\langle\langle a, w\rangle, u\rangle \notin R \vee \langle a, w\rangle \notin S) \wedge v = 0] \end{aligned}$$

We can immediately see that $(\forall U, V)(\forall x, y)(\exists! z)B(U, V, x, y, z)$.

From lemma 4.3.14 we get a primitive recursive function $\mathcal{H}$ so that

$$\begin{aligned} [\![\mathcal{H}(e, f, a, u)]\!](0) \;&=\; u \\ [\![\mathcal{H}(e, f, a, u)]\!](v + 1) \;&=\; \mathcal{F}_B(\mathit{ext}(e), \mathit{ext}(f), a, [\![\mathcal{H}(e, f, a, u)]\!](v)) \end{aligned}$$

We can finish the construction of the index c by defining
$c := \mathcal{H}(e, \mathcal{G}(e), a, F(0))$ and we can prove the following defining equations of $[\![c]\!]$ in Π^1_1-CA:

$$\begin{array}{rcl} [\![c]\!](0) & = & F(0) \\ [\![c]\!](v+1) & = & \mathcal{F}_B(ext(e), ext(\mathcal{G}(e)), a, [\![c]\!](v)) \end{array}$$

We can prove with induction on x that $\mathsf{CDC}(e, a, c)$, that is
$(\forall x)([\![e]\!](\langle\langle a, [\![c]\!](\mathcal{S}(x))\rangle, [\![c]\!](x)\rangle) = 0)$. □

Due to this theorem we can prove that $\mathcal{M}_1$ is a model of $\mathsf{PRON}(\mathsf{SUS})$.

Theorem 4.3.16 *$\mathcal{M}_1$ is a model of* $\mathsf{PRON}(\mathsf{SUS}) + (\mathsf{Tot}) + (\mathsf{Nat})$.

Proof The proof of this theorem is almost the same as of theorem 4.2.2. We only have to check the axioms about the constant E_1. The axiom (E_1.1) is satisfied because the universe of $\mathcal{M}_1$ is the set $\mathbb{N}$ of all natural numbers. We show that the axiom (E_1.2) is satisfied, as well:
Suppose, that $f^{\mathcal{M}_1}$ is the unary index of a function of $PRIM(SUS)$. Then we can prove that $(\mathsf{E}_1 fa)^{\mathcal{M}_1} = [[\mathsf{E}_1{}^{\mathcal{M}_1}](f^{\mathcal{M}_1})](a^{\mathcal{M}_1}) = [\langle 6, 1, \overline{\overline{f^{\mathcal{M}_1}}}\rangle](a^{\mathcal{M}_1}) = Sus^1(\overline{\overline{f^{\mathcal{M}_1}}})(a^{\mathcal{M}_1})$. Due to theorem 4.3.15 we know that $Sus^1(\overline{\overline{f^{\mathcal{M}_1}}})(a^{\mathcal{M}_1}) = 0$ if and only if there exists a unary index $c \in SusPrim$ so that
$[\overline{\overline{f^{\mathcal{M}_1}}}](x, [c](\mathcal{S}(x)), [c](x)) = 0$ for every natural number x. This is equivalent to the existence of a g so that $[f^{\mathcal{M}_1}](\langle\langle a^{\mathcal{M}_1}, [g^{\mathcal{M}_1}](\mathcal{S}(x))\rangle, [g^{\mathcal{M}_1}](x)\rangle) = 0$ for all a.
If $f^{\mathcal{M}_1} \notin SusPrim$ or $(f^{\mathcal{M}_1})_1 \neq 1$ then $(\mathsf{E}_1 fa)^{\mathcal{M}_1} = [\langle 1, 1, 0\rangle](a^{\mathcal{M}_1}) = 0$ and $SusPrimEv(f^{\mathcal{M}_1}, x) = 0$ for every natural number x, so we have $SusPrimEv(f^{\mathcal{M}_1}, \langle\langle a^{\mathcal{M}_1}, SusPrimEv(c, \mathcal{S}(x))\rangle, SusPrimEv(c, x)\rangle) = 0$ for every natural number c and x. □

We are going to define a translation $\cdot^*$ of formulas from $\mathcal{L}$ to $\mathcal{L}_2$. First, we will define a translation from $\mathcal{L}$ terms to $\mathcal{L}_2$ formulas. The translation of a constant c is the numeral $\overline{m}$, if m is the interpretation $c^{\mathcal{M}_1}$ of c in $\mathcal{M}_1$.

Definition 4.3.17 *For every $\mathcal{L}$ term t we define the $\mathcal{L}_2$ formula $V^*_t[x]$, so that the variable x does not occur in t, by induction on the build-up of t as follows:*

1. *If t is a variable of $\mathcal{L}$ then $V^*_t := (x = t)$*
2. *If t is a constant of $\mathcal{L}$ then $V^*_t := (x = \overline{t^{\mathcal{M}_1}})$*
3. *If t is the $\mathcal{L}$ term $<r, s>$ then*
   $V^*_t := (\exists z_0, z_1)[V^*_r(z_0) \wedge V^*_s(z_1) \wedge x = \langle z_0, z_1\rangle]$

4. *If t is the $\mathcal{L}$ term (rs) then*
   $V_t^* := (\exists z_0, z_1)[V_r^*(z_0) \wedge V_s^*(z_1) \wedge \mathsf{SusApp}(z_0, z_1, x)]$

Remark 4.3.18 *Let t be an arbitrary $\mathcal{L}$ term. Then Π_1^1-CA proves $(\exists! x)V_t^*(x)$ by a simple induction on the build-up of t.*

Definition 4.3.19 *For every $\mathcal{L}$ formula A we define the $\mathcal{L}_2$ formula A^* by induction on the build-up of A as follows:*

1. *If $A = \mathsf{N}(t)$ or $A = t\downarrow$ then $A^* := (\exists x)V_t^*(x)$*
2. *If $A = (s = t)$ then $A^* := (\exists x)(V_s^*(x) \wedge V_t^*(x))$*
3. *If $A = \neg B$ then $A^* := \neg B^*$*
4. *If $A = B \vee C$ then $A^* := B^* \vee C^*$*
5. *If $A = (\exists x)B$ then $A^* := (\exists x)B^*$*

The following lemma shows that we have defined a convenient translation.

Lemma 4.3.20 *Let A be an arbitrary $\mathcal{L}_1$ formula. Then we can prove the following assertion:*

$$\Pi_1^1\text{-}\mathsf{CA} \;\vdash\; A \leftrightarrow (A^\circ)^*$$

Proof This is verified by a straightforward induction on the build-up of A with remark 4.3.18. □

Now we can show that Π_1^1-CA proves the translation of an $\mathcal{L}$ formula provable in $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N})$.

Theorem 4.3.21 *Let A be an $\mathcal{L}$ formula and $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N}) \vdash A$. Then we can derive A^* in Π_1^1-CA.*

Proof We can prove this theorem by induction on the length of derivation. The logical axioms are easy to verify and in Π_1^1-CA we have the same rules of inference as in $\mathsf{PRON(SUS)}$. Because A^* is an $\mathcal{L}_2$ formula, the translation of the induction scheme $(\mathcal{L}\text{-}\mathsf{I_N})$ can be derived with $(\Pi_\infty^1\text{-}\mathsf{I_N})$.
Now let A be a non-logical axiom of $\mathsf{PRON(SUS)}$. This axiom is satisfied in the model $\mathcal{M}_1$ as we have proved in theorem 4.3.16. We have defined SusApp in a way that the properties of indices are satisfied. Finally, the definition of the translation $\cdot^*$ guarantees that A^* is derivable in Π_1^1-CA.
As an example we have written the exact translations of $(\mathsf{E_1}.1)$ and $(\mathsf{E_1}.2)$ in appendix B. □

We know the strength of $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N})$ because we have done embeddings in both directions.

Corollary 4.3.22 $\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N}) \equiv \Pi_1^1$-$\mathsf{CA}$ *with theorem 3.3.3.* □

5 Set induction

5.1 Lower bounds

It is obvious that formula induction is stronger than set induction: we can easily derive (S-$\mathsf{I_N}$) from ($\mathcal{L}$-$\mathsf{I_N}$). The induction scheme (S-$\mathsf{I_N}$) is even to weak to prove the existence of terms with the properties of $\mathsf{pr}_{\mathcal{F}}$, so we have to add totality to the axioms of primitive recursion. That is the reason why we replace the constant r by a new constant $\mathsf{r_N}$ in our applicative framework.

Definition 5.1.1 ($\mathsf{PRON^t}$) *We define* $\mathsf{PRON^t}$ *to be the* $\mathcal{L}$ *theory* PRON^- *plus the two following axioms about the new constant* $\mathsf{r_N}$*.*

XI. Primitive recursion on N

($\mathsf{r_N}$.1) $\quad f \in (\mathsf{N}^2 \to \mathsf{N}) \wedge a \in \mathsf{N} \to \mathsf{r_N}{<}f,a{>} \in (\mathsf{N} \to \mathsf{N}) \wedge \mathsf{r_N}{<}f,a{>}\mathsf{0} = a$

($\mathsf{r_N}$.2) $\quad f \in (\mathsf{N}^2 \to \mathsf{N}) \wedge a \in \mathsf{N} \wedge b \in \mathsf{N} \to$
$\quad\quad \mathsf{r_N}{<}f,a{>}(\mathsf{s_N}b) = f{<}b, \mathsf{r_N}{<}f,a{>}b{>}$

In $\mathsf{PRON^t}$ we can prove the existence of terms $\mathsf{pr}^{\mathsf{t}}_{\mathcal{F}}$ with the properties of $\mathsf{pr}_{\mathcal{F}}$. Due to the axioms about $\mathsf{r_N}$ we do not need induction for this proof.

Theorem 5.1.2 *Let $\mathcal{F}$ be an arbitrary primitive recursive function. Then there exists a closed $\mathcal{L}$ term $\mathsf{pr}^{\mathsf{t}}_{\mathcal{F}}$ so that the defining equations of $\mathcal{F}$ are derivable in $\mathsf{PRON^t}$ for $\mathsf{pr}^{\mathsf{t}}_{\mathcal{F}}$. Moreover, we can prove the following assertions:*

1. *If $\mathcal{F}$ is a constant, then $\mathsf{PRON^t} \vdash \mathsf{pr}^{\mathsf{t}}_{\mathcal{F}} \in \mathsf{N}$.*

2. *If the arity of $\mathcal{F}$ is $l > 0$, then $\mathsf{PRON^t} \vdash \mathsf{pr}^{\mathsf{t}}_{\mathcal{F}} \in (\mathsf{N}^l \to \mathsf{N})$.*

Proof We define the $\mathcal{L}$ term $\mathsf{pr}^{\mathsf{t}}_{\mathcal{F}}$ by induction on the build-up of $\mathcal{F}$ as follows:

1. If $\mathcal{F}$ is one of the functions $\mathcal{S}$, Cs^n_m, Pr^n_k, $Comp^n(\mathcal{G}, \mathcal{H}_0, \ldots, \mathcal{H}_{m-1})$, then $\mathsf{pr}^{\mathsf{t}}_{\mathcal{F}}$ is defined the same way as $\mathsf{pr}_{\mathcal{F}}$.

2. If $\mathcal{F} = Rec^{n+1}(\mathcal{G}, \mathcal{H})$ and $n = 0$ then $\mathsf{pr}^{\mathsf{t}}_{\mathcal{F}} := (\lambda^* z.\mathsf{r_N}{<}\mathsf{pr}^{\mathsf{t}}_{\mathcal{G}}, \mathsf{pr}^{\mathsf{t}}_{\mathcal{H}}{>}z)$

3. If $\mathcal{F} = Rec^{n+1}(\mathcal{G}, \mathcal{H})$ and $n > 0$ then
$\mathsf{pr}^{\mathsf{t}}_{\mathcal{F}} := (\lambda^* z.\mathsf{r_N}{<}\mathsf{pr}^{\mathsf{t}}_{\mathcal{G}}(\mathsf{p_0}z), (\lambda^* y.\mathsf{pr}^{\mathsf{t}}_{\mathcal{H}}{<}\mathsf{p_0}z, \mathsf{p_0}y, \mathsf{p_1}y{>}){>}(\mathsf{p_1}z))$

The proof of the two assertions above is straightforward and left as an exercise. □

We define the extensions of $\mathsf{PRON^t}$ the same way as the extensions of PRON.

Definition 5.1.3 *Like in definition 1.3.13 we add the same axioms to our basic theory with the same additional constants* $\boldsymbol{\mu}$ *and* E_1.

$$\begin{aligned} \mathsf{PRON^t}(\boldsymbol{\mu}) &:= \mathsf{PRON^t} + (\boldsymbol{\mu}.1) + (\boldsymbol{\mu}.2) \\ \mathsf{PRON^t}(\mathsf{SUS}) &:= \mathsf{PRON^t}(\boldsymbol{\mu}) + (\mathsf{E}_1.1) + (\mathsf{E}_1.2) \end{aligned}$$

In this case it is necessary to include the axioms about $\boldsymbol{\mu}$ in $\mathsf{PRON^t}(\mathsf{SUS})$, because there is no possibility to derive them from the axioms about E_1.

Theorem 5.1.4 *We have the following lower bounds:*

$$\begin{aligned} \mathsf{PRA} &\subseteq \mathsf{PRON^t} + (\mathsf{S\text{-}I_N}) \\ \mathsf{ACA_0} &\subseteq \mathsf{PRON^t}(\boldsymbol{\mu}) + (\mathsf{S\text{-}I_N}) \\ \Pi^1_1\text{-}\mathsf{CA_0} &\subseteq \mathsf{PRON^t}(\mathsf{SUS}) + (\mathsf{S\text{-}I_N}) \end{aligned}$$

PROOF For the first assertion, the translation $\cdot^\diamond$ from $\mathcal{L}_1$ terms and formulas to $\mathcal{L}$ must be modified. We replace every occurrence of $\mathsf{pr}_\mathcal{F}$ by $\mathsf{pr}^\mathsf{t}_\mathcal{F}$ in both definitions 3.1.1 and 3.1.2.
We can define a characteristic term $\mathsf{t}^\diamond_A$ for every quantifier-free $\mathcal{L}_1$ formula A. We can omit the sixth point and replace every occurrence of $\mathsf{pr}_\mathcal{F}$ by $\mathsf{pr}^\mathsf{t}_\mathcal{F}$ in the proof of lemma 3.2.4.
There is only one axiom that has not been proved in theorem 3.1.4, the induction scheme $(\mathsf{QF\text{-}I_N})$. The $\mathcal{L}$ translation of $(\mathsf{QF\text{-}I_N})$ is
$\vec{v} \in \mathsf{N} \to [A^\diamond(\vec{v}, \mathsf{0}) \wedge (\forall x \in \mathsf{N})(A^\diamond(\vec{v}, x) \to A^\diamond(\vec{v}, \mathsf{pr}^\mathsf{t}_\mathcal{S} x)) \to (\forall x \in \mathsf{N}) A^\diamond(\vec{v}, x)]$.
Suppose, that $\vec{v} \in \mathsf{N}$. Then we define t to be the term
$(\lambda^* y.\mathsf{t}^\diamond_A {<} v_0, \dots, v_{n-1}, y{>})$. We can prove $t \in \mathcal{P}(\mathsf{N})$ and $A(\vec{v}, x) \leftrightarrow tx = \mathsf{0}$ by lemma 3.2.4. With set induction we can now derive the $\mathcal{L}$ translation of $(\mathsf{QF\text{-}I_N})$.

For the second assertion, the translation $\cdot^\circ$ from $\mathcal{L}_2$ terms and formulas to $\mathcal{L}$ must be modified. We replace again every occurrence of $\mathsf{pr}_\mathcal{F}$ by $\mathsf{pr}^\mathsf{t}_\mathcal{F}$ in both definitions 3.2.1 and 3.2.2.
We can define a characteristic term t°_A for every arithmetic $\mathcal{L}_2$ formula A by replacing every occurrence of $\mathsf{pr}_\mathcal{F}$ by $\mathsf{pr}^\mathsf{t}_\mathcal{F}$ in the proof of lemma 3.2.4.
There is only one axiom that has not been proved in theorem 3.2.5, the induction scheme $(\mathsf{IA_N})$. The $\mathcal{L}$ translation of $(\mathsf{IA_N})$ is
$(\forall x)[x \in \mathcal{P}(\mathsf{N}) \to [x\mathsf{0} = \mathsf{0} \wedge (\forall y \in \mathsf{N})(xy = \mathsf{0} \to x(\mathsf{pr}^\mathsf{t}_\mathcal{S}) = 0) \to$
$(\forall y \in \mathsf{N})(xy = 0)]]$.
Suppose, that x is in $\mathcal{P}(\mathsf{N})$. Then we have exactly the induction scheme $(\mathsf{S\text{-}I_N})$ because PRON proves $(\mathsf{pr}_\mathcal{S} x = \mathsf{s_N} x)$ by definition 2.1.10.

For the third assertion, we can use the modified translation $\cdot^\circ$ and the characteristic term t°_A. For every Π^1_1 formula A we can define a characteristic term ct°_A the same way as in the proof of lemma 3.3.2. Then we have proved everything we need in the proof of theorem 3.3.3. $\square$

5.2 Upper bounds

Let us now continue the embeddings with the upper bounds.

Theorem 5.2.1 *We have the following upper bounds:*

$$\begin{array}{rcl} \mathsf{PRON}^{\mathsf{t}} + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}}) & \subseteq & \mathsf{PRA}^{-} + (\Sigma_1\text{-}\mathsf{I}_{\mathsf{N}}) \\ \mathsf{PRON}^{\mathsf{t}}(\boldsymbol{\mu}) + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}}) & \subseteq & \mathsf{ACA}_0 \\ \mathsf{PRON}^{\mathsf{t}}(\mathsf{SUS}) + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}}) & \subseteq & \Pi^1_1\text{-}\mathsf{CA}_0 \end{array}$$

Proof Due to the work of Feferman and Jäger [5], and Jäger and Strahm [8], it is only necessary to show the following assertions:

1. $\mathsf{PRON}^{\mathsf{t}} + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}}) \subseteq \mathsf{BON} + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$
2. $\mathsf{PRON}^{\mathsf{t}}(\boldsymbol{\mu}) + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}}) \subseteq \mathsf{BON}(\hat{\boldsymbol{\mu}}) + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$
3. $\mathsf{PRON}^{\mathsf{t}}(\mathsf{SUS}) + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}}) \subseteq \mathsf{SUS} + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$

Remember the translation $\cdot^{\triangleright}$ of section 4.1. The translation of the scheme $(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$ is an instance of $(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$.
All we have to do for the first assertion yet, is to replace the translation $\mathsf{r}^{\triangleright}$ of r by the new translation ${\mathsf{r}_{\mathsf{N}}}^{\triangleright}$ of r_{N}:

$${\mathsf{r}_{\mathsf{N}}}^{\triangleright} \quad := \quad (\lambda w.\hat{\mathsf{r}}_{\mathsf{N}}(\lambda uv.\hat{\mathsf{p}}_0 w(\hat{\mathsf{p}}uv))(\hat{\mathsf{p}}_1 w))$$

The translations of the axioms about r_{N} must be derivable in BON:

$(\mathsf{r}_{\mathsf{N}}.1)^{\triangleright}$ $\quad (\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}xy) \in \mathsf{N}) \wedge a \in \mathsf{N} \rightarrow$
$\quad {\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa) \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge {\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa)0 = a$

$(\mathsf{r}_{\mathsf{N}}.2)^{\triangleright}$ $\quad (\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}xy) \in \mathsf{N}) \wedge a \in \mathsf{N} \wedge b \in \mathsf{N} \rightarrow$
$\quad {\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa)(\hat{\mathsf{s}}_{\mathsf{N}}b) = f(\hat{\mathsf{p}}b({\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa)b))$

1. Suppose, that $(\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}xy) \in \mathsf{N})$ and $a \in \mathsf{N}$. Then we can prove $(\forall x, y \in \mathsf{N})((\lambda uv.f(\hat{\mathsf{p}}uv))xy \in \mathsf{N})$, so ${\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa) = \hat{\mathsf{r}}_{\mathsf{N}}(\lambda uv.f(\hat{\mathsf{p}}uv))a$ is in $(\mathsf{N} \rightarrow \mathsf{N})$ and ${\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa)0 = \hat{\mathsf{r}}_{\mathsf{N}}(\lambda uv.f(\hat{\mathsf{p}}uv))a0 = a$
2. Suppose, that $(\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}xy) \in \mathsf{N})$, $a \in \mathsf{N}$, and $b \in \mathsf{N}$. Then we can prove $(\forall x, y \in \mathsf{N})((\lambda uv.f(\hat{\mathsf{p}}uv))xy \in \mathsf{N})$ and ${\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa)(\hat{\mathsf{s}}_{\mathsf{N}}b) = \hat{\mathsf{r}}_{\mathsf{N}}(\lambda uv.f(\hat{\mathsf{p}}uv))a(\hat{\mathsf{s}}_{\mathsf{N}}b) = (\lambda uv.f(\hat{\mathsf{p}}uv))b(\hat{\mathsf{r}}_{\mathsf{N}}(\lambda uv.f(\hat{\mathsf{p}}uv))ab) = (\lambda uv.f(\hat{\mathsf{p}}uv))b({\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa)b = f(\hat{\mathsf{p}}b({\mathsf{r}_{\mathsf{N}}}^{\triangleright}(\hat{\mathsf{p}}fa)b))$

For the second assertion we can just take this embedding of $\mathsf{PRON^t}$ in BON and add the translation $\boldsymbol{\mu}^{\triangleright}$ of $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}^{\triangleright} \quad := \quad (\lambda fa.\hat{\boldsymbol{\mu}}(\lambda u.f(\hat{\mathsf{p}}au)))$$

The translations of the axioms about $\boldsymbol{\mu}$ must be derivable in $\mathsf{BON}(\hat{\boldsymbol{\mu}})$:

$(\boldsymbol{\mu}.1)^{\triangleright}$ $\quad (\forall x \in \mathsf{N})(f(\hat{\mathsf{p}}ax) \in \mathsf{N}) \leftrightarrow \boldsymbol{\mu}^{\triangleright} fa \in \mathsf{N}$

$(\boldsymbol{\mu}.2)^{\triangleright}$ $\quad (\forall x \in \mathsf{N})(f(\hat{\mathsf{p}}ax) \in \mathsf{N}) \rightarrow$
$[(\exists x \in \mathsf{N})(f(\hat{\mathsf{p}}ax) = 0) \rightarrow f(\hat{\mathsf{p}}a(\boldsymbol{\mu}^{\triangleright} fa)) = 0]$

1. $(\forall x \in \mathsf{N})(f(\hat{\mathsf{p}}ax) \in \mathsf{N}) \leftrightarrow (\lambda u.f(\hat{\mathsf{p}}au)) \in (\mathsf{N} \rightarrow \mathsf{N}) \leftrightarrow$
   $\hat{\boldsymbol{\mu}}(\lambda u.f(\hat{\mathsf{p}}au)) \in \mathsf{N} \leftrightarrow \boldsymbol{\mu}^{\triangleright} fa \in \mathsf{N}$

2. Suppose, that $(\forall x \in \mathsf{N})(f(\hat{\mathsf{p}}ax) \in \mathsf{N})$ and $(\exists x \in \mathsf{N})(f(\hat{\mathsf{p}}ax) = 0)$. Then we can prove $(\lambda u.f(\hat{\mathsf{p}}au)) \in (\mathsf{N} \rightarrow \mathsf{N})$ and $(\exists x \in \mathsf{N})[(\lambda u.f(\hat{\mathsf{p}}au))x = 0]$, so we have $f(\hat{\mathsf{p}}a(\boldsymbol{\mu}^{\triangleright} fa)) = (\lambda u.f(\hat{\mathsf{p}}au))(\boldsymbol{\mu}^{\triangleright} fa) =$
   $(\lambda u.f(\hat{\mathsf{p}}au))(\hat{\boldsymbol{\mu}}(\lambda u.f(\hat{\mathsf{p}}au))) = 0$

For the third assertion we can again take this embedding of $\mathsf{PRON^t}(\boldsymbol{\mu})$ in $\mathsf{BON}(\hat{\boldsymbol{\mu}})$ and add the translation ${\mathsf{E}_1}^{\triangleright}$ of E_1:

$${\mathsf{E}_1}^{\triangleright} \quad := \quad (\lambda fa.\hat{\mathsf{E}}_1(\lambda uv.f(\hat{\mathsf{p}}(\hat{\mathsf{p}}au)v)))$$

The translations of the axioms about E_1 must be derivable in SUS:

$(\mathsf{E}_1.1)^{\triangleright}$ $\quad (\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}(\hat{\mathsf{p}}ax)y) \in \mathsf{N}) \leftrightarrow {\mathsf{E}_1}^{\triangleright} fa \in \mathsf{N}$

$(\mathsf{E}_1.2)^{\triangleright}$ $\quad (\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}(\hat{\mathsf{p}}ax)y) \in \mathsf{N}) \rightarrow [(\exists g)[g \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge$
$(\forall x \in \mathsf{N})(f(\hat{\mathsf{p}}(\hat{\mathsf{p}}a(g(\hat{\mathsf{s}}_{\mathsf{N}}x)))(gx)) = 0)] \leftrightarrow {\mathsf{E}_1}^{\triangleright} fa = 0]$

1. $(\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}(\hat{\mathsf{p}}ax)y) \in \mathsf{N}) \leftrightarrow$
   $(\forall x, y \in \mathsf{N})((\lambda uv.f(\hat{\mathsf{p}}(\hat{\mathsf{p}}au)v))xy \in \mathsf{N}) \leftrightarrow$
   $\hat{\mathsf{E}}_1(\lambda uv.f(\hat{\mathsf{p}}(\hat{\mathsf{p}}au)v)) \in \mathsf{N} \leftrightarrow {\mathsf{E}_1}^{\triangleright} fa \in \mathsf{N}$

2. Suppose, that $(\forall x, y \in \mathsf{N})(f(\hat{\mathsf{p}}(\hat{\mathsf{p}}ax)y) \in \mathsf{N})$. Then we can prove
   $(\forall x, y \in \mathsf{N})((\lambda uv.f(\hat{\mathsf{p}}(\hat{\mathsf{p}}au)v))xy \in \mathsf{N})$ and
   $(\exists g)[g \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge (\forall x \in \mathsf{N})(f(\hat{\mathsf{p}}(\hat{\mathsf{p}}a(g(\hat{\mathsf{s}}_{\mathsf{N}}x)))(gx)) = 0)] \leftrightarrow$
   $(\exists g)[g \in (\mathsf{N} \rightarrow \mathsf{N}) \wedge (\forall x \in \mathsf{N})((\lambda uv.f(\hat{\mathsf{p}}(\hat{\mathsf{p}}au)v))(g(\hat{\mathsf{s}}_{\mathsf{N}}x))(gx) = 0)] \leftrightarrow$
   $\hat{\mathsf{E}}_1(\lambda uv.f(\hat{\mathsf{p}}(\hat{\mathsf{p}}au)v)) = 0 \leftrightarrow {\mathsf{E}_1}^{\triangleright} fa = 0$ $\square$

Corollary 5.2.2 *Together with theorem 5.1.4 we have proved that:*

1. $\mathsf{PRA} \subseteq \mathsf{PRON^t} + (\mathsf{S}\text{-}\mathsf{I_N}) \subseteq \mathsf{PRA^-} + (\Sigma_1\text{-}\mathsf{I_N})$
2. $\mathsf{PRON^t}(\boldsymbol{\mu}) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{ACA_0}$
3. $\mathsf{PRON^t}(\mathsf{SUS}) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \Pi^1_1\text{-}\mathsf{CA_0}$ □

Remark 5.2.3 *Due to Parsons [11] we know that* PRA *and* $\mathsf{PRA^-} + (\Sigma_1\text{-}\mathsf{I_N})$ *prove the same* Π_2 *sentences. So, if we restrict the relation* $\equiv$ *to* Π_2 *sentences, we can say that* $\mathsf{PRON^t} + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{PRA}$.

A Overview

The following table gives a short survey of proof-theoretic equivalences.

Applicative theories	Equivalent theories	Ordinal
$\mathsf{PRON^t} + (\mathsf{S\text{-}I_N})$ $\mathsf{BON} + (\mathsf{S\text{-}I_N})$ [1] $\mathsf{BON} + (\hat{\mathsf{V}}\text{-}\mathsf{I_N})$ [2,8]	PRA [1] $\mathsf{PRA} + (\Sigma_1\text{-}\mathsf{I_N})$ [1] $\mathsf{PEA^+} + (\mathsf{r}) + (\mathsf{S\text{-}I_N})$ [3]	ω^ω
$\mathsf{PRON} + (\mathcal{L}\text{-}\mathsf{I_N})$ $\mathsf{BON} + (\hat{\mathcal{L}}\text{-}\mathsf{I_N})$ [4] $\mathsf{PRON^t}(\boldsymbol{\mu}) + (\mathsf{S\text{-}I_N})$ $\mathsf{BON}(\hat{\boldsymbol{\mu}}) + (\mathsf{S\text{-}I_N})$ [1]	PA [4] $\mathsf{ACA_0}$ [1] $\mathsf{PA^r_\Omega}$ [1] $\mathsf{EPSON} + (\mathsf{r})$ [3]	ε_0
$\mathsf{PRON}(\boldsymbol{\mu}) + (\mathcal{L}\text{-}\mathsf{I_N})$	ACA	$\varepsilon_{\varepsilon_0}$ [5]
$\mathsf{BON}(\hat{\boldsymbol{\mu}}) + (\hat{\mathcal{L}}\text{-}\mathsf{I_N})$ [1]	$\mathsf{ACA}_{<\varepsilon_0}$ [1] $\mathsf{PA^w_\Omega}$ [1] $\widehat{\mathsf{ID}}_1$ [1]	$\varphi\varepsilon_0 0$ [1]
$\mathsf{PRON^t(SUS)} + (\mathsf{S\text{-}I_N})$ $\mathsf{SUS} + (\mathsf{S\text{-}I_N})$ [6]	$\Pi^1_1\text{-}\mathsf{CA_0}$ [6] $\Delta^1_2\text{-}\mathsf{CA_0}$ [6] $\mathsf{KPi^r}$ [6]	$\Psi 0 \Omega_\omega$ [7]
$\mathsf{PRON(SUS)} + (\mathcal{L}\text{-}\mathsf{I_N})$	$\Pi^1_1\text{-}\mathsf{CA}$	$\Psi 0(\Omega_\omega \cdot \varepsilon_0)$ [7]
$\mathsf{SUS} + (\hat{\mathsf{V}}\text{-}\mathsf{I_N})$ [6,8]	$(\Pi^1_1\text{-}\mathsf{CA})_{<\omega^\omega}$ [6] $\Delta^1_2\text{-}\mathsf{CR}$ [6] $\mathsf{KPi^r} + (\Sigma_1\text{-}\mathsf{I_N})$ [6]	$\Psi 0 \Omega_{\omega^\omega}$ [7]
$\mathsf{SUS} + (\hat{\mathcal{L}}\text{-}\mathsf{I_N})$ [6]	$(\Pi^1_1\text{-}\mathsf{CA})_{<\varepsilon_0}$ [6] $\Delta^1_2\text{-}\mathsf{CA}$ [6] $\mathsf{KPi^w}$ [6]	$\Psi 0 \Omega_{\varepsilon_0}$ [7]

[1] introduced and proved by Feferman and Jäger [5]
[2] a proof can be found in Strahm [15]
[3] introduced by Schlüter [12]
[4] introduced and proved by Feferman [3]
[5] proved by Schütte [13]
[6] introduced and proved by Jäger and Strahm [8]
[7] proved by Buchholz and Schütte [2]
[8] $(\hat{\mathsf{V}}\text{-}\mathsf{I_N}) : f0 \in \mathsf{N} \wedge (\forall x \in \mathsf{N})(fx \in \mathsf{N} \rightarrow f(\hat{\mathsf{s}}_{\mathsf{N}} x) \in \mathsf{N}) \rightarrow (\forall x \in \mathsf{N})(fx \in \mathsf{N})$

B Translations of some axioms

Translation $\cdot^\star$ of the axioms about $\boldsymbol{\mu}$:

$(\boldsymbol{\mu}.1)^\star$ $(\forall x)[(\exists y)(y = x) \rightarrow (\exists y, z_0, z_1)[z_0 = f \wedge (\exists z_2, z_3)(z_2 = a \wedge z_3 = x \wedge z_1 = \langle z_2, z_3\rangle) \wedge \boldsymbol{\mu}\mathsf{App}(z_0, z_1, y)]] \leftrightarrow (\exists x, z_0, z_1)[(\exists z_2, z_3)(z_2 = \overline{\boldsymbol{\mu}^{\mathcal{M}_0}} \wedge z_3 = f \wedge \boldsymbol{\mu}\mathsf{App}(z_2, z_3, z_0)) \wedge z_1 = a \wedge \boldsymbol{\mu}\mathsf{App}(z_0, z_1, x)]$

$(\boldsymbol{\mu}.2)^\star$ $(\forall x)[(\exists y)(y = x) \rightarrow (\exists y, z_0, z_1)[z_0 = f \wedge (\exists z_2, z_3)(z_2 = a \wedge z_3 = x \wedge z_1 = \langle z_2, z_3\rangle) \wedge \boldsymbol{\mu}\mathsf{App}(z_0, z_1, y)]] \rightarrow [(\exists x)[(\exists y)(y = x) \rightarrow (\exists y)[(\exists z_0, z_1)[z_0 = f \wedge (\exists z_2, z_3)(z_2 = a \wedge z_3 = x \wedge z_1 = \langle z_2, z_3\rangle) \wedge \boldsymbol{\mu}\mathsf{App}(z_0, z_1, y)] \wedge y = 0]] \rightarrow (\exists x)[(\exists z_0, z_1)[z_0 = f \wedge (\exists z_2, z_3)[z_2 = a \wedge (\exists z_4, z_5)[(\exists z_6, z_7)(z_6 = \overline{\boldsymbol{\mu}^{\mathcal{M}_0}} \wedge z_7 = f \wedge \boldsymbol{\mu}\mathsf{App}(z_6, z_7, z_4)) \wedge z_5 = a \wedge \boldsymbol{\mu}\mathsf{App}(z_4, z_5, z_3)] \wedge z_1 = \langle z_2, z_3\rangle] \wedge \boldsymbol{\mu}\mathsf{App}(z_0, z_1, x)] \wedge x = 0]]$

Translation $\cdot^*$ of the axioms about E_1:

$(\mathsf{E}_1.1)^*$ $(\forall x, y)[(\exists z)(z = x) \wedge (\exists z)(z = y) \rightarrow (\exists z, z_0, z_1)[z_0 = f \wedge (\exists z_2, z_3)[(\exists z_4, z_5)(z_4 = a \wedge z_5 = x \wedge z_2 = \langle z_4, z_5\rangle) \wedge z_3 = y \wedge z_1 = \langle z_2, z_3\rangle] \wedge \mathsf{SusApp}(z_0, z_1, z)]] \leftrightarrow (\exists x, z_0, z_1)[(\exists z_2, z_3)(z_2 = \overline{\mathsf{E_1}^{\mathcal{M}_1}} \wedge z_3 = f \wedge \mathsf{SusApp}(z_2, z_3, z_0)) \wedge z_1 = a \wedge \mathsf{SusApp}(z_0, z_1, x)]$

$(\mathsf{E}_1.2)^*$ $(\forall x, y)[(\exists z)(z = x) \wedge (\exists z)(z = y) \rightarrow (\exists z, z_0, z_1)[z_0 = f \wedge (\exists z_2, z_3)[(\exists z_4, z_5)(z_4 = a \wedge z_5 = x \wedge z_2 = \langle z_4, z_5\rangle) \wedge z_3 = y \wedge z_1 = \langle z_2, z_3\rangle] \wedge \mathsf{SusApp}(z_0, z_1, z)]] \rightarrow [(\exists g)[(\forall x)[(\exists y)(y = x) \rightarrow (\exists y, z_0, z_1)(z_0 = g \wedge z_1 = x \wedge \mathsf{SusApp}(z_0, z_1, y))] \wedge (\forall x)[(\exists y)(y = x) \rightarrow (\exists y)[(\exists z_0, z_1)[z_0 = f \wedge (\exists z_2, z_3)[(\exists z_4, z_5)[z_4 = a \wedge (\exists z_6, z_7)[z_6 = g \wedge (\exists z_8, z_9)(z_8 = \overline{\mathsf{s_N}^{\mathcal{M}_1}} \wedge z_9 = x \wedge \mathsf{SusApp}(z_8, z_9, z_7)) \wedge \mathsf{SusApp}(z_6, z_7, z_5)] \wedge z_2 = \langle z_4, z_5\rangle] \wedge (\exists z_4, z_5)(z_4 = g \wedge z_5 = x \wedge \mathsf{SusApp}(z_4, z_5, z_3)) \wedge z_1 = \langle z_2, z_3\rangle] \wedge \mathsf{SusApp}(z_0, z_1, y)] \wedge y = 0]]] \leftrightarrow (\exists x)[(\exists z_0, z_1)[(\exists z_2, z_3)(z_2 = \overline{\mathsf{E_1}^{\mathcal{M}_1}} \wedge z_3 = f \wedge \mathsf{SusApp}(z_2, z_3, z_0)) \wedge z_1 = a \wedge \mathsf{SusApp}(z_0, z_1, x)] \wedge x = 0]]$

References

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

[2] Buchholz, W., and Schütte, K. *Proof Theory of Impredicative Subsystems of Analysis*. Bibliopolis, Naples, 1988.

[3] Feferman, S. A language and axioms for explicit mathematics. In *Algebra and Logic*, J. Crossley, Ed., vol. 450 of *Lecture Notes in Mathematics*. Springer, Berlin, 1975, pp. 87–139.

[4] Feferman, S. Hilbert's program relativized: proof-theoretical and foundational studies. *Journal of Symbolic Logic 53*, 2 (1988), 364–384.

[5] 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.

[6] Feferman, S., Jäger, G., and Strahm, T. *Explicit Mathematics*. In preparation.

[7] Hinman, P. G. *Recursion-Theoretic Hierarchies*. Springer, Berlin, 1978.

[8] Jäger, G., and Strahm, T. The proof-theoretic strength of the Suslin operator in applicative theories. Preprint, Aug. 1999. To appear in *Festschrift for Solomon Feferman*.

[9] Kahle, R. Natural numbers and forms of weak induction in applicative theories. Tech. Rep. IAM-95-001, Institut für Informatik und angewandte Mathematik, Universität Bern, January 1995.

[10] Minari, P. Unpublished draft note, Sept. 1996.

[11] Parsons, C. On a number theoretic choice schema and its relation to induction. In *Intuitionism and Proof Theory, Proceedings of the Summer Conference at Buffalo N.Y., 1968*, J. Myhill, A. Kino, and R. Vesley, Eds. North Holland, Amsterdam, 1970, pp. 459–473.

[12] Schlüter, A. A theory of rules for enumerated classes of functions. *Archive for Mathematical Logic 34* (1995), 47–63.

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

[14] Simpson, S. G. *Subsystems of Second Order Arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, 1998.

[15] Strahm, T. *On the Proof Theory of Applicative Theories*. PhD thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1996.

[16] Troelstra, A., and van Dalen, D. *Constructivism in Mathematics*, vol. I. North-Holland, Amsterdam, 1988.

Address:
David Steiner, Institut für Informatik und angewandte Mathematik,
Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
dastein@iam.unibe.ch