An Exercise in Programming in PA: Fast Growing Hierarchy Functions

Paul J. Voda

Institute of Informatics, Comenius University Bratislava,
Mlynská dolina, 842 15 Bratislava, Slovakia,
email: voda@fmph.uniba.sk homepage: www.fmph.uniba.sk/∼voda

1 Introduction

That mathematical logic is extremely relevant to computer programming is widely accepted. For some years now we were claiming that Peano Arithmetic (PA), actually its fragment $I\Sigma_1$ containing primitive recursive functions, suffices for the derivation and proofs of correctness of computer programs. We were doing this in the PA-based programming language and proof system CL. See [KV98] for the logical description of CL and [Ko01] for an extensive selection of programs which our undergraduate students are expected to design and verify in CL.

Coding is the daily bread to computer programmers and they have developed a wide range of techniques to make it simple. On the other hand, logicians consider coding as almost trivial. They usually just outline the proofs of theorems involving coding because they view the proofs as tedious.

The purpose of this paper is to illustrate that the techniques of computer programming can be used in mathematical logic. We apply the well-known list processing techniques to the task of introducing into PA (i.e. programming) the functions F_α of the Fast Growing Hierarchy (see [LW70,Wa70,Wa72,Ro84]). Although computationally unfeasible (already F_ω grows as Ackermann), the functions are important because all $\prec \epsilon_0$-ordinal recursive functions, which are the provably recursive functions of PA, can be defined from them by propositional connectives and bounded quantification, i.e. by $\Sigma_0(\{F_\alpha \mid \alpha \prec \epsilon_0\})$ formulas.

The introduction of F_α into PA is far from trivial. In this paper we follow Fairtlough and Wainer [FW98] where we streamline their presentation by employing some computer programming tricks. The introduction becomes an exercise in list processing. What logicians see as a mechanical reflection of properties of ordinals we view as an intricate interplay between the depth and length of lists which is fascinating on its own. We carry out the proofs in detail in order to demonstrate that programs can be rigidly proved in a very natural way. We think that despite the detail, the proofs are not tedious.

Our further contribution is the representation of ordinals by natural numbers in an extremely simple way in which every natural number represents. This permits the view of natural numbers as a special 'data structure' with its own induction on the depth of the structure. Properties of the data structure become interesting on their own and are not perceived as mere arithmetization of ordinal properties.

Another contribution is the introduction of the fast growing *Hardy* functions H_α (which define the functions F_α explicitly [Wa72,Ro84]) via very natural sequences descending in depth. The sequences are akin to the sequences of Goodstein [Goo44,FW98] generally perceived by logicians as being mathematical rather than metamathematical.

The terminology of this paper is based on [HP93,FW98]. We extensively use the closure of $I\Sigma_1$ under course of values recursion.

2 Ordinal-like Data Structure and its Induction

2.1 Pairing. We assume a $I\Sigma_1$ *pairing* function x, y satisfying:

$$\begin{aligned}
&\vdash x_1, y_1 = x_2, y_2 \rightarrow x_1 = x_2 \wedge y_1 = y_2 \\
&\vdash x < y \rightarrow z, x < z, y \wedge x, z < y, z \\
&\vdash x < x, y \wedge y < x, y \\
&\vdash x = 0 \vee \exists v \exists w \, x = v, w \; .
\end{aligned}$$

The *modified diagonal* function of Cantor $x, y = (x+y) \cdot (x+y+1) \div 2 + x + 1$ suffices for the purposes of this paper where the modification assures that $0 < x, y$. It was shown in [Vo94] that this function is not good when we are interested in questions of computational complexity.

We adopt a convention that whenever the infix pairing comma in (x, y) cannot be mistaken for a comma separating arguments we may drop the parentheses around pairing. Thus, for instance, if f is a binary function then $f((v, w), x, y)$ abbreviates $f((v, w), (x, y))$. Notice that the first set of parentheses cannot be omitted. We let pairing group to the right where x, y, z abbreviates $x, (y, z)$. We also assign to the pairing function the lowest precedence. Thus, for instance, $x, y + z$ stands for $x, (y + z)$.

Every pairing function with above properties establishes a one-to-one correspondence between $\mathbb{N}$ and finite sequences over $\mathbb{N}$. The empty sequence is coded by 0 and $a_1 \dots a_n$ by $a_1, \dots, a_n, 0$. Codes of finite sequences are called *lists* in computer programming.

2.2 List processing in PA. We introduce into $I\Sigma_1$ two basic list processing functions: *concatenation* ($\oplus$) and *duplication* ($\otimes$) both binary infix and satisfying:

$$\begin{aligned}
&\vdash 0 \oplus z = z \qquad &&\vdash (x, y) \oplus z = x, y \oplus z \\
&\vdash x \otimes 0 = 0 \qquad &&\vdash x \otimes (i + 1) = x, x \otimes i \; .
\end{aligned}$$

The following properties are proved in $I\Sigma_1$ by complete induction on x except (4) which is proved by induction on i:

$$\vdash x \oplus 0 = x \tag{1}$$
$$\vdash (x \oplus y) \oplus z = x \oplus (y \oplus z) \tag{2}$$
$$\vdash x > 0 \to \exists y{<}x \exists z{<}x\, x = y \oplus (z, 0) \tag{3}$$
$$\vdash x \otimes (i + j) = x \otimes i \oplus x \otimes j \tag{4}$$

For instance, we prove (3) by assuming $x > 0$ and considering two cases. If $x = u, 0$ then it suffices to take $y = 0$ and $z = u$, both $< x$. If $x = u, v, w$ then, since $x > v, w > 0$, we get $v, w = y_0 \oplus (z, 0)$ for some $y_0 < v, w$ and $z < v, w$ from IH and it suffices to take $y = u, y_0$ because $u, v, w = u, y_0 \oplus (z, 0) = (u, y_0) \oplus (z, 0)$ and $u, y_0 < u, v, w = x$.

2.3 Natural numbers as an ordinal-like data structure. We can ambiguously view natural numbers as representants of ordinal numbers $\prec \epsilon_0$. The number 0 represents the ordinal 0 and a, b represents $\beta \# \omega^\alpha$ if a, b represent α, β respectively. Here $\#$ is the natural sum of ordinals. Note that if α is represented by a and $i \prec \omega$ then $\alpha + i$ is represented by $0 \otimes i \oplus a$. When we present an ordinal $\alpha \prec \epsilon_0$ in Cantor's notation, replace in it $\beta \cdot n$ by repeated addition, replace every $+$ by $\#$, and represent this by a we obtain the *canonical* representant of α.

We introduce into $I\Sigma_1$ by primitive recursion a binary function w_n^k yielding for each $n, k \in \mathbb{N}$ the canonical representant of the ordinal w_n^k. The function satisfies:

$$\vdash \omega_0^k = 0 \otimes k \qquad \vdash \omega_{n+1}^k = \omega_n^k, 0 \ .$$

The numbers ω_n^k have the *depth* n. All numbers are classified by the unary *depth* predicates $a \prec \omega_n^k$ in a for all $n \geq 0$ and $k \geq 1$ which are introduced into $I\Sigma_1$ to satisfy:

$$\vdash a \prec \omega_0^k \leftrightarrow \exists i (a = 0 \otimes i \wedge i < \overline{k})$$
$$\vdash a \prec \omega_{n+1}^k \leftrightarrow a = 0 \vee \exists b \exists c (a = b, c \wedge b \prec \omega_n^k \wedge c \prec \omega_{n+1}^k) \ .$$

Here the *numerals* $\overline{k}$ are terms satisfying $\overline{0} \equiv 0$ and $\overline{k+1} \equiv S(\overline{k})$.

The mention of the ordinal ordering symbol $\prec$ in the symbols for the depth predicates obviates its arithmetization which is far from trivial, especially with the ambiguous coding of ordinals. $I\Sigma_1$ proves:

$$\vdash a \prec \omega_1^1 \leftrightarrow \exists i\, a = 0 \otimes i \tag{1}$$
$$\vdash a \oplus b \prec \omega_{n+1}^k \leftrightarrow a \prec \omega_{n+1}^k \wedge b \prec \omega_{n+1}^k \tag{2}$$
$$\vdash a \prec \omega_n^k \to a \prec \omega_{n+j}^{k+i} \ . \tag{3}$$

(1)(2): By complete induction on a. (3): By inductions on i and j in the metatheory.

2.4 Arithmetization of fundamental sequences. We need to arithmetize the fundamental sequences $\{\lambda\}(i)$ for limit ordinals λ satisfying:

$$\{\omega^{\alpha+1}{\cdot}(\beta+1)\}(i) = \omega^{\alpha+1}{\cdot}\beta + \omega^{\alpha}{\cdot}i \qquad \{\omega^{\lambda}{\cdot}(\beta+1)\}(i) = \omega^{\lambda}{\cdot}\beta + \omega^{\{\lambda\}(i)} \ .$$

It is of advantage for technical reasons to assign fundamental sequences also to 0 and successor ordinals as $\{0\}(i) = 0$ and $\{\alpha+1\}(i) = \alpha$ respectively. The arithmetization can be done in $I\Sigma_1$ to satisfy:

$$\begin{aligned}
&\vdash \{0\}(i) = 0 \land \{0,c\}(i) = c\\
&\vdash \{(0,b),c\}(i) = b{\otimes}i \oplus c\\
&\vdash a > 0 \rightarrow \{(a,b),c\}(i) = \{a,b\}(i), c \ .
\end{aligned}$$

2.5 Induction on depth. The function $\{a\}(i)$ can be viewed as trading the depth of a for its *length* $L(a)$ where $L(0) = 0$ and $L(x,y) = L(y)+1$. This is captured in the principle of *induction on depth*:

$$\phi[0] \land \forall a \forall b(\forall i \phi[\{a,b\}(i)] \rightarrow \phi[a,b]) \rightarrow \phi[\tau] \tag{1}$$

designated by $D^{\tau}_{x}\phi$ where $\phi[x]$ is a formula with the distinguished variable x and τ a term. We designate by $Prog_x\phi$ the antecedent of (1). We have

$$I\Sigma_1 \vdash Prog_x\phi \land \phi[a] \rightarrow \phi[0,a] \tag{2}$$

because $\{0,a\}(i) = a$ and so from $\phi[a]$ we get $\forall i \phi[\{0,a\}(i)]$ and from $Prog_x\phi$ we get $\phi[0,a]$.

The principle $D^{\tau}_{x}\phi$ is valid in the standard model of PA for every ϕ and τ because whenever $0 \prec \alpha \prec \epsilon_0$ we have $\{\alpha\}(i) \prec \alpha$. However, the best one can prove in fragments of PA is expressed by the following theorem.

2.6 Theorem (Induction on depth). *For every $m, k \geq 1$, $n \geq 0$, and every Π_m or Σ_m formula $\phi[x]$ we have:*

$$I\Sigma_1 \vdash a \prec \omega^k_0 \rightarrow D^a_x\phi \tag{1}$$

$$I\Sigma_{n+m} \vdash a \prec \omega^k_{n+1} \rightarrow D^a_x\phi \ . \tag{2}$$

Proof. We prove (1) by metatheoretical induction on k. For $k=1$ we get $a=0$ from $a \prec \omega^1_0$ and $\phi[0]$ from $Prog_x\phi$. In the inductive case we assume $Prog_x\phi$ and from $a \prec \omega^{k+1}_0$ we have either $a=0$ and so $\phi[0]$ from $Prog_x\phi$, or else $a = 0,b$ where $b \prec \omega^k_0$. We then get $\phi[b]$ from IH and $\phi[0,b]$ from 2.5(2).

The proof of (2) is based on a jump construction of Gentzen [Ge43] (see also Takeuti [Tak75]). For a given $m \geq 1$ we take any Π_m or Σ_m formula $\phi[x]$. We set $\phi_0[x] \equiv \phi[x]$ and

$$\phi_{n+1}[x] \equiv \forall a(\phi_n[a] \rightarrow \phi_n[x,a]) \ .$$

Clearly, $\phi_{n+1}[x]$ is Π_{n+m+1} in $I\Sigma_1$ and so we can use ϕ_n as induction formulas in $I\Sigma_{n+m}$ because Π_n-induction is admissible in $I\Sigma_n$ for all n. For any $n \geq 0$

we prove first:

$$I\Sigma_{n+m} \vdash Prog_x \phi_n \to Prog_x \phi_{n+1} \tag{3}$$

by working in $I\Sigma_{n+m}$. We assume $Prog_x \phi_n$. For the proof of the first conjunct $\phi_{n+1}[0]$ we take any b s.t. $\phi_n[b]$ and obtain $\phi_n[0,b]$ by 2.5(2). The second conjunct is $\forall a \forall b(\forall i \phi_{n+1}[\{a,b\}(i)] \to \phi_{n+1}[a,b])$, we take any a, b and assume $\forall i \phi_{n+1}[\{a,b\}(i)]$. For the proof of $\phi_{n+1}[a,b]$ it suffices to take any i and c s.t. $\phi_n[c]$ and derive $\phi_n[\{(a,b),c\}(i)]$ from which we get the desired $\phi_n[(a,b),c]$ from the assumption $Prog_x \phi_n$.

For $\phi_n[\{(a,b),c\}(i)]$ we consider two cases. If $a > 0$ then, since $\{(a,b),c\}(i) = \{a,b\}(i),c$, we get $\phi_n[\{a,b\}(i),c]$ from $\phi_{n+1}[\{a,b\}(i)]$ and $\phi_n[c]$. If $a = 0$ then, since $\{(0,b),c\}(i) = b \otimes i \oplus c$, we prove $\phi_n[b \otimes i \oplus c]$ by induction on i. In the base case, we have $b \otimes 0 \oplus c = c$ and $\phi_n[c]$ is an assumption. In the inductive case we get $\phi_{n+1}[b]$ from $\phi_{n+1}[\{a,b\}(i)]$ because $\{a,b\}(i) = \{0,b\}(i) = b$. But then $\phi_n[b, b \otimes i \oplus c]$ by IH and we have

$$b, b \otimes i \oplus c = (b, b \otimes i) \oplus c = b \otimes (i+1) \oplus c\ .$$

For any $n, j \geq 0$ and $k \geq 1$ we now prove the jump property:

$$I\Sigma_{n+m} \vdash \forall a(a \prec \omega_j^k \to D_x^a \phi_{n+1}) \to \forall a(a \prec \omega_{j+1}^k \to D_x^a \phi_n) \tag{4}$$

by working in $I\Sigma_{n+m}$. We assume the antecedent, take any a, assume $a \prec \omega_{j+1}^k$, $Prog_x \phi_n$, and prove $\phi_n[a]$ by complete induction on a where we consider two cases implied by $a \prec \omega_{j+1}^k$. Either $a = 0$ and we get $\phi_n[0]$ from $Prog_x \phi_n$, or else $a = b, c$ for some b, c s.t. $b \prec \omega_j^k$ and $c \prec \omega_{j+1}^k$. From the latter, since $c < a$, we obtain $\phi_n[c]$ by IH. From the former, since $Prog_x \phi_{n+1}$ holds by (3), we obtain $\phi_{n+1}[b]$ from the antecedent. Combining the two yields $\phi_n[b,c]$.

Finally, for any $n \geq 0$ and $k \geq 1$ we get

$$I\Sigma_{n+m} \vdash \forall a(a \prec \omega_0^k \to D_x^a \phi_{n+1})$$

from (1) and derive (2) by applying modus ponens $(n+1)$-times to (4). $\square$

3 Fast Growing Hierarchy Functions

3.1 Functions H_α and F_α. The *Hardy* functions H_α (see [Wa72]) satisfy:

$$H_0(x) = x \qquad H_{\alpha+1}(x) = H_\alpha(x+1) \qquad H_\lambda(x) = H_{\{\lambda\}(x)}(x)\ .$$

They can be used to define the fast growing functions $F_\alpha(x) = H_{w^\alpha}(x)$ such that:

$$F_0(x) = x+1 \qquad F_{\alpha+1}(x) = F_\alpha^x(x) \qquad F_\lambda(x) = F_{\{\lambda\}(x)}(x)\ .$$

A very simple definition of Hardy functions is as 'limits' of ordinal sequences $(\alpha)_j$ descending to finite ordinals. The sequences for $i \prec \omega$ and λ a limit satisfy:

$$(i)_j = i \qquad (\lambda + i)_0 = \lambda + i \qquad (\lambda+i)_{j+1} = (\{\lambda\}(i) + i)_j\ .$$

It is not difficult to see that for every $\alpha \prec \epsilon_0$ and $x \prec \omega$ there is a $j \in \mathbb{N}$ such that $H_\alpha(x) = (\alpha + x)_j \prec \omega$.

3.2 Arithmetization of $(\alpha)_j$. We easily get

$$I\Sigma_1 \vdash \exists! i \exists! b (a = 0 \otimes i \oplus b \wedge \forall b_1 \forall b_2 (b = b_1, b_2 \to b_1 > 0)) \tag{1}$$

and so we can arithmetize in $I\Sigma_1$ the ordinal sequences from Par. 3.1 to satisfy:

$$\vdash (a)_0 = a$$

$$\vdash (a)_j = 0 \otimes i \to (a)_{j+1} = 0 \otimes i \tag{2}$$

$$\vdash (a)_j = 0 \otimes i \oplus (b_1, b_2) \wedge b_1 > 0 \to (a)_{j+1} = 0 \otimes i \oplus \{b_1, b_2\}(i) \ . \tag{3}$$

$I\Sigma_1$ proves:

$$\vdash ((a)_i)_j = (a)_{i+j} \tag{4}$$

$$\vdash \exists i \, (a)_j \oplus b = (a \oplus b)_i \ . \tag{5}$$

(4): By induction on j. (5): This follows from $\forall a \exists i {\le} j \, (a)_j \oplus b = (a \oplus b)_i$ proved by Π_1-induction on j. In the base case take $i = 0$. In the inductive case we have for $i_0 \le j$ obtained by IH and $i_1 \le 1$ obtained from $\exists i_1 {\le} 1 \, (a)_1 \oplus b = (a \oplus b)_{i_1}$ which is easily proved by case analysis on $(a)_0$ implied by (2) and (3):

$$(a)_{j+1} \oplus b \overset{(4)}{=} ((a)_1)_j \oplus b \overset{IH}{=} ((a)_1 \oplus b)_{i_0} = ((a \oplus b)_{i_1})_{i_0} \overset{(4)}{=} (a \oplus b)_{i_0 + i_1} \ .$$

Although $\forall a \exists j \, (a)_j \prec \omega_1^1$ is true in the standard model of PA, the best one can prove in fragments of PA is expressed by the following theorem.

3.3 Convergence theorem. *For every $n, k \ge 1$ we have*

$$I\Sigma_n \vdash a \prec \omega_n^k \to \exists j \, (a)_j \prec \omega_1^1 \ . \tag{1}$$

Proof. We cannot prove the progressivity in x of the formula $\exists j \, (x)_j \prec \omega_1^1$ except in a Π_2-form $\forall i \exists j \, (0 \otimes i \oplus x)_j \prec \omega_1^1$ but then the induction on depth would yield only $I\Sigma_{n+1} \vdash a \prec \omega_n^k \to \forall i \exists j \, (0 \otimes i \oplus a)_j \prec \omega_1^1$ which is not deep enough. This can be remedied by taking any $n, k \ge 1$ and proving below $I\Sigma_1 \vdash Prog_x \phi$ for a Π_2-formula $\phi[x] \equiv \forall i \exists j \, (0 \otimes i \oplus (x, 0))_j \prec \omega_1^1$. We then get by depth induction:

$$I\Sigma_n \vdash a \prec \omega_{n-1}^k \to \forall i \exists j \, (0 \otimes i \oplus (a, 0))_j \prec \omega_1^1 \tag{2}$$

where if $n = 1$ we use 2.6(1) and 2.6(2) otherwise. We extend (2) to

$$I\Sigma_n \vdash a \prec \omega_n^k \to \exists j \, (0 \otimes i \oplus a)_j \prec \omega_1^1 \tag{3}$$

by complete Σ_1-induction on a where we consider two cases. If $a = 0$ then we take $j = 0$. If $a > 0$ we assume $a \prec \omega_n^k$ and get $a = b \oplus (c, 0)$ for some $b < a$ and c by 2.2(3) and $b \prec \omega_n^k$, $c \prec \omega_{n-1}^k$ by 2.3(2). We then get $(0 \otimes i \oplus b)_{j_0} = 0 \otimes i_0$ for some j_0, i_0 from IH and 2.3(1) and $(0 \otimes i_0 \oplus (c, 0))_{j_1} \prec \omega_1^1$ for some j_1 from (2). For some j_2 obtained from 3.2(5) we get:

$$(0 \otimes i \oplus a)_{j_1 + j_2} = (0 \otimes i \oplus b \oplus (c, 0))_{j_1 + j_2} \overset{3.2(4)}{=} ((0 \otimes i \oplus b \oplus (c, 0))_{j_2})_{j_1} \overset{3.2(5)}{=}$$
$$((0 \otimes i \oplus b)_{j_0} \oplus (c, 0))_{j_1} = (0 \otimes i_0 \oplus (c, 0))_{j_1} \prec w_1^1 \ .$$

To conclude the proof of (1) we take any a s.t. $a \prec \omega_n^k$, obtain $a = 0{\otimes}i \oplus b$ for some i, b by 3.2(1) with $b \prec \omega_n^k$ by 2.3(2), and apply (3).

We need another 'extension' property:

$$I\Sigma_1 \vdash \forall i \exists j\, (0{\otimes}i \oplus (a,0))_j \prec \omega_1^1 \to \exists j\, (0{\otimes}i \oplus a{\otimes}m)_j \prec \omega_1^1 \tag{4}$$

which is proved similarly as the extension of (2) to (3). We assume the antecedent and prove the consequent by Σ_1-induction on m. In the base case we take $j = 0$. In the inductive case we have $(0{\otimes}i \oplus a{\otimes}m)_{j_0} = 0{\otimes}i_0$ for some j_0, i_0 by IH and 2.3(1) and $(0{\otimes}i_0 \oplus (a,0))_{j_1} \prec \omega_1^1$ for some j_1 from the assumption. We then get for some j_2:

$$\begin{aligned}(0{\otimes}i \oplus a{\otimes}(m+1))_{j_1+j_2} &\overset{2.2(4)}{=} (0{\otimes}i \oplus a{\otimes}m \oplus (a,0))_{j_1+j_2} \overset{3.2(4)}{=}\\ &((0{\otimes}i \oplus a{\otimes}m \oplus (a,0))_{j_2})_{j_1} \overset{3.2(5)}{=} ((0{\otimes}i \oplus a{\otimes}m)_{j_0} \oplus (a,0))_{j_1} =\\ &(0{\otimes}i_0 \oplus (a,0))_{j_1} \prec w_1^1\ .\end{aligned}$$

We now prove $\vdash \mathit{Prog}_x \phi$ by working in $I\Sigma_1$. For $\phi[0]$ it suffices to take $j = 0$. For the second conjunct we take any a, b, i, assume $\forall i_0 \phi[\{a,b\}(i_0)]$, and consider two cases. If $a = 0$ we have:

$$(0{\otimes}i \oplus ((0,b),0))_1 = 0{\otimes}i \oplus \{(0,b),0\}(i) = 0{\otimes}i \oplus b{\otimes}i \oplus 0 = 0{\otimes}i \oplus b{\otimes}i\ .$$

Because $\{0,b\}(i_0) = b$, the assumption becomes $\forall i \exists j\, (0{\otimes}i \oplus (b,0))_j \prec \omega_1^1$ and we use it in (4) to obtain $(0{\otimes}i \oplus b{\otimes}i)_j \prec \omega_1^1$ for some j. Thus $((0{\otimes}i \oplus ((0,b),0))_1)_j \prec \omega_1^1$ and we use 3.2(4) to obtain $\phi[0,b]$. If $a > 0$ we have

$$(0{\otimes}i \oplus ((a,b),0))_1 = 0{\otimes}i \oplus \{(a,b),0\}(i) = 0{\otimes}i \oplus (\{a,b\}(i),0)$$

and $(0{\otimes}i \oplus (\{a,b\}(i),0))_j \prec \omega_1^1$ for some j from $\forall i_0 \phi[\{a,b\}(i_0)]$. We use 3.2(4) again to obtain $\phi[a,b]$ from $((0{\otimes}i \oplus ((a,b),0))_1)_j \prec \omega_1^1$. $\square$

3.4 Theorem (Introduction of H_α and F_α). *For any $n \geq 1$ and ordinal α such that $\alpha \prec \omega_{n+1}^1$ we can introduce into $I\Sigma_n$ the function H_α to satisfy:*

$$I\Sigma_n \vdash H_\alpha(x) = y \leftrightarrow \exists j\, (0{\otimes}x \oplus \overline{a})_j = 0{\otimes}y \tag{1}$$

where a is the canonical representant of α. Consequently, if $\alpha \prec \omega_n^1$ we can explicitly define in $I\Sigma_n$ the function F_α from the previously introduced H_{ω^α} to satisfy $I\Sigma_n \vdash F_\alpha(x) = H_{\omega^\alpha}(x)$.

Proof. From $\alpha \prec \omega_{n+1}^1$ we get $\alpha \prec \omega_n^k$ for some $k \prec \omega$ and this is arithmetized in $I\Sigma_1$ by $\vdash \overline{a} \prec \omega_n^k$. From $\vdash 0{\otimes}x \prec \omega_n^k$ holding by 2.3(1)(3) we then get $\vdash 0{\otimes}x \oplus \overline{a} \prec \omega_n^k$ by 2.3(2). $I\Sigma_n$ then proves $\vdash \exists j\, (0{\otimes}x \oplus \overline{a})_j \prec \omega_1^1$ by the Convergence theorem from which the existence condition for (1) follows. The uniqueness condition follows from 3.2(2). $\square$

3.5 Theorem (Limits of provability in PA). *For any $n \geq 1$ we have*

$$I\Sigma_n \nvdash \exists j\,(0\otimes i \oplus w^{k}_{\overline{n}})_j \prec \omega^1_1 \qquad PA \nvdash \exists j\,(0\otimes i \oplus w^{k}_{n})_j \prec \omega^1_1 \qquad (1)$$

although $I\Sigma_n \vdash \exists j\,(0\otimes i \oplus w^{\overline{k}}_{\overline{n}})_j \prec \omega^1_1$ *for any* $k \geq 0$ *and* $PA \vdash \exists j\,(0\otimes i \oplus w^{k}_{\overline{n}})_j \prec \omega^1_1$ *for any* $n \geq 0$.

Proof. $I\Sigma_1$ proves $0\otimes i \oplus w^{\overline{k}}_{\overline{n}} \prec \omega^{k+1}_n$ and $0\otimes i \oplus w^{k}_{\overline{n}} \prec \omega^1_{n+1}$ and so the positive results follow by the Convergence theorem. This is the best, since we have $H_{\omega^1_{n+1}}(k) = H_{\omega^k_n}(k)$ and $H_{\epsilon_0}(n) = H_{\omega^1_n}(n)$, and so the two functions would be provably recursive in $I\Sigma_n$ and PA respectively if the formulas in (1) were provable. But that is not the case (see [FW98]). □

4 Conclusions and Future Work

We are currently preparing a paper concerned with the harder direction of characterizing provably recursive functions of PA where we will deal with the witnessing functions to $\vdash \forall x \exists y \phi[x, y]$, $\phi \in \Sigma_0$ completely within a 'functional programming language' without the usual detour through ordinals or functionals.

References

[FW98] M. Fairtlough and S. S. Wainer. *Hierarchies of Provably Recursive Functions*, in Handbook of Proof Theory, ed. S. Buss, Elsevier Science BV, 1998.

[Ge43] G. Gentzen. *Beweisbarkeit und Unbeweisbarkeit von Anfangfällen der Transfiniten Induktion in der Reiner Zahlentheorie*, Mathematische Annalen, vol. 119, 1943, pp. 140-161,

[Goo44] R. L. Goodstein. *On the Restricted Ordinal Theorem*, JSL vol. 9, 1944, pp. 33-41.

[HP93] P. Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetic*. Springer Verlag, 1993.

[KV98] J. Komara, P. J. Voda. *Theorems of Péter and Parson in Computer Programming*. Computer Science Logic Conf. (CSL-98), Brno The Czech Rep. 1998. Lecture Notes in Computer Science vol 1584, Springer Verlag 1999.

[Ko01] J. Komara. *Specification and Verification of Computer Programs*, www.ii.fmph.uniba.sk/~komara, 2001.

[LW70] M. Löb and S. S. Wainer. *Hieratchies of Number Theoretic Functions I and II*, Archiv für Mathematische Logik und Grundlagenforschung, vol. 14, 1970, pp. 39-51 and 97-113.

[Ro84] H. E. Rose. *Subrecursion: Functions and Hierarchies.* Number 9 in Oxford Logic Guides. Clarendon Press, Oxford, 1982.

[Tak75] G. Takeuti. *Proof Theory*. North-Holland, 1975.

[Vo94] P. J. Voda. *Subrecursion as a basis for a feasible programming language*, in L. Pacholski and J. Tiuryn, editors, Proceedings of CSL'94, number 933 in LNCS Springer Verlag, 1995, pages 324-338.

[Wa70] S. S. Wainer. *A classification of the ordinal recursive functions*, Archiv für mathematische Logik und Grundlagen Forschung, vol. 13, 1970, pp. 136-153.

[Wa72] S. S. Wainer. *Ordinal Recursion, and a Refinement of the Extended Grzegorczyk Hierarchy*, JSL vol. 37, 1972, pp. 281-292.