

# Some Notes on Proof Theory and Elements of Ordinal Analysis

Daniel Rogozin

## Contents

<b>1</b>	<b>Provable Recursion in <math>\mathbf{I}\Delta_0(\text{exp})</math></b>	<b>2</b>
1.1	Proof-theoretic Characterisation . . . . .	7
<b>2</b>	<b>Primitive Recursion and <math>\mathbf{I}\Sigma_1</math></b>	<b>11</b>
2.1	$\mathbf{I}\Sigma_1$ provable functions are primitive recursive . . . . .	13
<b>3</b>	<b><math>\varepsilon_0</math>-recursion in Peano Arithmetic</b>	<b>17</b>
3.1	Ordinals below $\varepsilon_0$ . . . . .	18
3.2	Introducing the fast-growing hierarchy . . . . .	22
3.3	$\alpha$ -recursion and $\varepsilon_0$ -recursion . . . . .	24
3.4	Provable recursiveness of $F_\alpha$ and $H_\alpha$ . . . . .	28
<b>4</b>	<b><math>\mathbf{RCA}_0</math></b>	<b>28</b>
<b>5</b>	<b><math>\mathbf{WKL}_0</math></b>	<b>28</b>
<b>6</b>	<b><math>\mathbf{ACA}_0</math></b>	<b>28</b>
<b>7</b>	<b><math>\mathbf{ATR}</math></b>	<b>28</b>
<b>8</b>	<b><math>\Pi^1_1</math>-comprehension</b>	<b>28</b>
<b>9</b>	<b>Kripke-Platek Set Theory</b>	<b>28</b>

# 1 Provable Recursion in $\mathbf{I}\Delta_0(\text{exp})$

$\mathbf{I}\Delta_0(\text{exp})$  is a theory in first-order logic in the language:

$$\{=, 0, S, P, +, \dot{-}, \cdot, \text{exp}_2\}$$

where  $S$  and  $P$  are successor and predecessor functions respectively. Further, we will denote  $S(x)$  and  $P(x)$  as  $x+1$  and  $x\dot{-}1$  respectively.  $2^x$  stands for  $\text{exp}_2(x)$ .

The non-logical axioms of  $\mathbf{I}\Delta_0(\text{exp})$  are the following list:

- $x+1 \neq 0$
- $0\dot{-}1 = 0$
- $x+0 = x$
- $x\dot{-}0 = x$
- $x \cdot 0 = 0$
- $2^0 = 1$
- $x+1 = y+1 \rightarrow x = y$
- $(x+1)\dot{-}1 = x$
- $x+(y+1) = (x+y)+1$
- $x\dot{-}(y+1) = x\dot{-}y\dot{-}1$
- $x \cdot (y+1) = x \cdot y + x$
- $2^{x+1} = 2^x + 2^x$

along with the bounded induction scheme:

$$B(0) \wedge \forall x(B(x) \rightarrow B(x+1)) \rightarrow \forall x B(x)$$

where  $B$  is a  $\Delta$ -formula, that is a formula one of the following forms (with bounded quantifiers only):

- $B \equiv \forall x < t P(x) \equiv \forall x(x < t \rightarrow P(x))$
- $B \equiv \exists x < t P(x) \equiv \exists x(x < t \wedge P(x))$

A  $\Sigma_1$ -formula is a formula of the form:

$$\exists \vec{x} B(\vec{x})$$

where  $B(\vec{x}) \in \Delta_0$ .

**Lemma 1.1.**  $\mathbf{I}\Delta_0(\text{exp})$  proves (the universal closures of):

1.  $x = 0 \vee x = (x\dot{-}1) + 1$
2.  $x + (y + z) = (x + y) + z$
3.  $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
4.  $x \cdot (y + z) = x \cdot y + x \cdot z$
5.  $x + y = y + x$
6.  $x \cdot y = y \cdot x$
7.  $x\dot{-}(y + z) = (x\dot{-}y)\dot{-}z$

$$8. 2^{x+y} = 2^x \cdot 2^y$$

*Proof.*

1. This is self-evident.
2. If  $z = 0$ , then  $x + y = x + y$ . If  $z = z' + 1$ , then, by applying the IH and the relevant axioms:

$$(x + (y + (z' + 1))) = (x + ((y + z') + 1)) = (x + (y + z')) + 1 = ((x + y) + z') + 1 = (x + y) + (z' + 1)$$

3. If  $z = 0$ , then  $x \cdot (y \cdot 0) = (x \cdot y) \cdot 0$ . If  $z = z' + 1$ , then:

$$x \cdot (y \cdot (z' + 1)) = x \cdot (y \cdot z' + y) = x \cdot (y \cdot z') + x \cdot y = (x \cdot y) \cdot z' + x \cdot y = (x \cdot y) \cdot (z' + 1)$$

4. The rest of the cases are shown by induction on  $z$ . Consider the exponentiation law. If  $y = 0$ , then

$$2^{x+0} = 2^x = 0 + 2^x = 2^x \cdot 0 + 2^x = 2^x \cdot (0 + 1) = 2^x \cdot 2^0$$

If  $y = y' + 1$ , then:

$$2^{x+(y'+1)} = 2^{(x+y')+1} = 2^x \cdot 2^{y'} + 2^x \cdot 2^{y'} = 2^x \cdot 2^{y'+1}$$

□

**Lemma 1.2.**  $\mathbf{I}\Delta_0(\text{exp})$  proves (the universal closures of):

1.  $\neg x < 0$
2.  $x \leq 0 \leftrightarrow x = 0$
3.  $0 \leq x$
4.  $x \leq x$
5.  $x < x + 1$
6.  $x < y + 1 \leftrightarrow x \leq y$
7.  $x \leq y \leftrightarrow x < y \vee x = y$
8.  $x \leq y \wedge y \leq z \rightarrow x \leq z$
9.  $x < y \wedge y < z \rightarrow x < z$
10.  $x \leq y \vee y < x$
11.  $x < y \rightarrow x + z < y + z$

$$12. x < y \rightarrow x \cdot (z + 1) < y \cdot (z + 1)$$

$$13. x < 2^x$$

$$14. x < y \rightarrow 2^x < 2^y$$

*Proof.* Straightforward induction.  $\square$

**Definition 1.1.** A function  $f : \mathbb{N}^k \rightarrow \mathbb{N}$  is *provably  $\Sigma_1$*  or *provably recursive* in an arithmetical theory if there is a  $\Sigma_1$  formula  $F(\vec{x}, y)$ , a “defining formula” of  $f$ , such that:

1.  $f(\vec{n}) = m$  iff  $\omega \models f(\vec{n}) = m$
2.  $T \vdash \exists y F(\vec{x}, y)$
3.  $T \vdash F(\vec{x}, y) \wedge F(\vec{x}, y') \rightarrow y = y'$

If a defining formula  $F \in \Delta_0$ , then a function  $f$  is *provably bounded* in  $T$  if there is a term  $t(\vec{x})$  such that  $T \vdash F(\vec{x}, y) \rightarrow y < t(\vec{x})$ .

**Theorem 1.1.** Let  $f$  be a provably recursive in  $T$ , then we can conservatively extend  $T$  by adding a new function symbol  $f$  along with the defining axiom  $F(\vec{x}, f(\vec{x}))$ .

*Proof.* Let  $\mathcal{M} \models T$ ,  $\mathcal{M}$  can be made into a model  $(\mathcal{M}, f)$  where we interpret  $f$  as the function which is uniquely determined by the second and third conditions of the definitions above. Let  $\varphi$  be a statement not involving  $f$  such that  $\varphi$  is true in  $(\mathcal{M}, f)$ , so  $\varphi$  is true in  $\mathcal{M}$  as well. By compactness  $T$  proves  $\varphi$ .  $\square$

**Lemma 1.3.** Each term defines a provably bounded function of  $\mathbf{I}\Delta_0(\text{exp})$ .

*Proof.* Let  $f$  be a function defined by some  $\mathbf{I}\Delta_0(\text{exp})$ -term  $t$ , that is,  $f(\vec{x}) = t(\vec{x})$ . Take  $y = t(\vec{x})$  as the defining formula for  $f$  since  $\exists y (y = t(\vec{x}))$  is derivable. If  $y' = t(\vec{x}) \wedge y = t(\vec{x})$ , then  $y = y'$  by transitivity. A formula  $y = t(\vec{x})$  is bounded and  $y = t$  implies  $y < t + 1$ . Thus  $f$  is provably bounded.  $\square$

**Lemma 1.4.** Define  $2_k(x)$  as  $2_0(x) = x$  and  $2_{n+1}(x) = 2^{2_n(x)}$ . Then for every term  $t(x_1, \dots, x_n)$  built up from the constants  $0, S, P, +, \cdot, \dot{-}, \cdot, \text{exp}_2$  there exists  $k < \omega$  such that:

$$\mathbf{I}\Delta_0(\text{exp}) \vdash t(x_1, \dots, x_n) < 2_k\left(\sum_{k=0}^n x_k\right)$$

*Proof.* Let  $t$  be a term constructed from subterms  $t_0$  and  $t_1$  by using one of the function constants. Assume that inductively  $t_0 < 2_{k_0}(s_0)$  and  $t_1 < 2_{k_1}(s_1)$  are both provable for some  $k_0, k_1 < \omega$ , where  $s_i$  is the sum of the variables of  $t_i$  for  $i = 0, 1$ .

Let  $s$  be the sum of all variables appearing in either  $t_0$  or  $t_1$  and let  $k = \max(k_0, k_1)$ . Then one can prove  $t_0 < 2_k(s)$  and  $t_1 < 2_k(s)$ . So one needs to show the following:

1.  $t_0 + 1 < 2_{k+1}(s)$
2.  $t_0 \dot{-} 1 < 2_k(s)$
3.  $t_0 \dot{-} t_1 < 2_k(s)$
4.  $t_0 \cdot t_1 < 2_k(s)$
5.  $t_0 + t_1 < 2_k(s)$
6.  $2^{t_0} < 2_k(s)$

So  $\mathbf{I}\Delta_0(\text{exp}) \vdash t < 2_{k+1}(s)$ . □

**Lemma 1.5.** Let  $f$  be a function defined by composition:

$$f(\vec{x}) = g_0(g_1(\vec{x}), \dots, g_m(\vec{x}))$$

where  $g_0, g_1, \dots, g_m$  are functions each of which is provably bounded in  $\mathbf{I}\Delta_0(\text{exp})$ . Then  $f$  is provably bounded in  $\mathbf{I}\Delta_0(\text{exp})$ .

*Proof.* Each  $g_i$  has a defining formula  $G_i$  and, by Lemma 1.4, there is a number  $k_i < \omega$  such that:

$$\mathbf{I}\Delta_0(\text{exp}) \vdash \exists y < 2_{k_i}(s) G_i(\vec{x}, y)$$

where  $s$  is the sum of elements of  $\vec{x}$ . And for  $i = 0$  one has:

$$\mathbf{I}\Delta_0(\text{exp}) \vdash \exists y < 2_{k_0}(s_0) G_0(y_1, \dots, y_m, y)$$

where  $s_0$  is the sum of  $y_1, \dots, y_m$ .

Let  $k = \max\{k_i < \omega \mid i < m + 1\}$  and let  $F(\vec{x}, y)$  be the bounded formula:

$$\exists y_1 < 2_k(s) \dots \exists y_m < 2_k(s) C(\vec{x}, y_1, \dots, y_m, y)$$

where  $C(\vec{x}, y_1, \dots, y_m, y)$  is the conjunction:

$$G_1(\vec{x}, y_1) \wedge \dots \wedge G_m(\vec{x}, y_m) \wedge G_0(y_1, \dots, y_m, y)$$

$F$  is clearly a defining formula for  $f$  such that  $\mathbf{I}\Delta_0(\text{exp}) \vdash \exists y F(\vec{x}, y)$ .

Moreover, each  $G_i$  is unique, so  $\mathbf{I}\Delta_0(\text{exp})$  also proves:

$$\begin{aligned} & C(\vec{x}, y_1, \dots, y_m, y) \wedge C(\vec{x}, z_1, \dots, z_m, z) \rightarrow \\ & \rightarrow \bigwedge_{j=1}^m y_j = z_j \wedge G_0(y_1, \dots, y_m, y) \wedge G_0(y_1, \dots, y_m, z) \rightarrow \\ & \rightarrow y = z \end{aligned}$$

so we have (by first order logic):

$$\mathbf{I}\Delta_0(\text{exp}) \vdash F(\vec{x}, y) \wedge F(\vec{x}, z) \rightarrow y = z$$

Thus  $f$  is provably  $\Sigma_1$  in  $\mathbf{I}\Delta_0(\text{exp})$ , so the rest is to find its bounding term.  $\mathbf{I}\Delta_0(\text{exp})$  proves the following:

$$C(\vec{x}, y_1, \dots, y_m, y) \rightarrow \bigwedge_{j=1}^m y_j < 2_k(s) \wedge y < 2_k(y_1 + \dots + y_m)$$

and

$$\bigwedge_{j=1}^m y_j < 2_k(s) \rightarrow y_1 + \dots + y_m < 2_k(s) \cdot m$$

Put  $t(\vec{x}) = 2_k(2_k(s) \cdot m)$ , then we obtain

$$\mathbf{I}\Delta_0(\text{exp}) \vdash C(\vec{x}, y_1, \dots, y_m, y) \rightarrow y < t(\vec{x})$$

and so

$$\mathbf{I}\Delta_0(\text{exp}) \vdash F(\vec{x}, y) \rightarrow y < t(\vec{x})$$

□

**Lemma 1.6.** Suppose  $f$  is defined by bounded minimisation

$$f(\vec{n}, m) = \mu_{k < m}(g(\vec{n}, k) = 0)$$

from a function  $g$  which is provably bounded in  $\mathbf{I}\Delta_0(\text{exp})$ . Then  $f$  is provably bounded in  $\mathbf{I}\Delta_0(\text{exp})$ .

*Proof.* Let  $G$  be a defining formula for  $g$ . Let  $F(\vec{x}, z, y)$  be the bounded formula

$$y \leq z \wedge \forall i < y \neg G(\vec{x}, i, 0) \wedge (y = z \vee G(\vec{x}, y, 0))$$

$\omega \models F(\vec{n}, m, k)$  iff either  $k$  is the least number less than  $m$  such that  $g(\vec{n}, k) = 0$  or there is no such and  $k = m$ . Thus it means that  $k$  is the value of  $f(\vec{n}, m)$ , so  $F$  is a defining formula for  $f$ .

Furthermore

$$\mathbf{I}\Delta_0(\text{exp}) \vdash F(\vec{x}, z, y) \rightarrow y < z + 1$$

so  $t(\vec{x}, z) = z + 1$  can be taken as a bounding term for  $f$ .

We can prove:

$$F(\vec{x}, z, y) \wedge F(\vec{x}, z, y') \wedge y < y' \rightarrow G(\vec{x}, y, 0) \wedge \neg G(\vec{x}, y', 0)$$

and similarly for interchanged  $y$  and  $y'$ . So we can prove:

$$F(\vec{x}, z, y) \wedge F(\vec{x}, z, y') \rightarrow \neg y < y' \wedge \neg y' < y$$

As far as  $y < y' \vee y' < y \vee y = y'$ , we have

$$F(\vec{x}, z, y) \wedge F(\vec{x}, z, y') \rightarrow y = y'$$

Now we have to check that  $\mathbf{I}\Delta_0(\text{exp}) \vdash \exists y F(\vec{x}, z, y)$ . We construct such  $y$  by bounded induction on  $z$ .

1.  $z = 0$ .

$F(\vec{x}, 0, 0)$  is provable since  $y = 0 \leftrightarrow y \leq 0$  and  $\neg i < 0$ . So  $\mathbf{I}\Delta_0(\text{exp}) \vdash F(\vec{x}, 0, y)$  is provable.

2. Assume  $\exists y F(\vec{x}, z, y)$  is provable, let show that that  $\exists y F(\vec{x}, z + 1, y)$  is provable.

We can show  $y \leq z \rightarrow y + 1 \leq z + 1$  and, via  $i < y + 1 \leftrightarrow i < y \vee i = y$ ,

$$\forall i < y \neg G(\vec{x}, i, 0) \wedge ((y = z) \wedge \neg G(\vec{x}, y, 0)) \rightarrow \forall i < y + 1 \neg G(\vec{x}, i, 0) \wedge y + 1 = z + 1$$

Therefore

$$F(\vec{x}, z, y) \rightarrow F(\vec{x}, z + 1, y + 1) \vee F(\vec{x}, z + 1, y)$$

and thus:

$$\exists y F(\vec{x}, z, y) \rightarrow \exists y F(\vec{x}, z + 1, y)$$

□

**Theorem 1.2.** Every elementary function is provably bounded in  $\mathbf{I}\Delta_0(\text{exp})$ .

*Proof.* As we know from recursion theory, the class of elementary functions can be characterised as those functions which are definable from 0,  $S$ ,  $P$ ,  $\cdot$ ,  $+$ ,  $\text{exp}_2$ ,  $-$  and  $\cdot$  by composition and minimisation. And then we apply above lemmas. □

### 1.1 Proof-theoretic Characterisation

For this section we shall be using a Tait-style formalisation of  $\mathbf{I}\Delta_0(\text{exp})$ . We have the following logical rules:

$$\begin{array}{c} \frac{}{\Gamma, R\vec{t}, \neg R\vec{t}} \mathbf{Ax} \\[10pt] \frac{\Gamma, A_0, A_1}{\Gamma, A_0 \vee A_1} \vee \qquad \frac{\Gamma, A_0 \quad \Gamma, A_1}{\Gamma, A_0 \wedge A_1} \wedge \\[10pt] \frac{\Gamma, A(t)}{\Gamma, \exists x A(x)} \exists \qquad \frac{\Gamma, A}{\Gamma, \forall x A} \forall \end{array}$$

where  $R\vec{t}$  is an atomic formula and  $x$  is not free in  $A$  in the  $\forall$  rule. Here  $\Gamma$  stores all non-logical axioms of  $\mathbf{I}\Delta_0(\text{exp})$  along with its negations. We also have the bounded induction rule:

$$\frac{\Gamma, B(0) \quad \Gamma, \neg B(n), B(n+1)}{\Gamma, B(t)} \mathbf{BInd}$$

where  $B$  is a bounded formula and  $t$  is any term.

Of course, the cut rule is admissible:

$$\frac{\Gamma, A \quad \Gamma, \neg A}{\Gamma} \text{ cut}$$

**Definition 1.2.** Let  $\exists \vec{z}B(\vec{z})$  be a closed  $\Sigma_1$ -formula, then it is *true at  $m$* , written as  $m \models \exists \vec{z}B(\vec{z})$ , if there exist natural numbers  $m_1, \dots, m_l$  such that each  $m_i < m$  and  $B(\vec{m})$  is true in the standard model.

A finite set  $\Gamma$  of closed  $\Sigma_1$ -formulas is true at  $m$ , written as  $m \models \Gamma$  if at least one of them is true at  $m$ .

If  $\Gamma(x_1, \dots, x_k)$  is a finite set of  $\Sigma_1$ -formulas whose free variables occur amongst  $x_1, \dots, x_k$ . Let  $f : \mathbb{N}^k \rightarrow \mathbb{N}$ , then  $f \models \Gamma(x_1, \dots, x_k)$  we have  $f(\vec{n}) \models \Gamma(x_1 := n_1, \dots, x_k := n_k)$  for each  $\vec{n} = (n_1, \dots, n_k)$ .

**Fact 1.1. (Persistence)**

1. If  $m \leq m'$ , then  $m \models \exists \vec{z}B(\vec{z})$  implies  $m' \models \exists \vec{z}B(\vec{z})$ .
2. If  $\forall \vec{n} \in \mathbb{N}^k$   $f(\vec{n}) \leq f'(\vec{n})$ , then  $f(\vec{n}) \models \Gamma(x_1 := n_1, \dots, x_k := n_k)$  implies  $f'(\vec{n}) \models \Gamma(x_1 := n_1, \dots, x_k := n_k)$ .

**Lemma 1.7.** Let  $\Gamma(\vec{x})$  be a finite set of  $\Sigma_1$  formulas such that

$$\mathbf{I}\Delta_0(exp) \vdash \bigvee_{\gamma(\vec{x}) \in \Gamma(\vec{x})} \gamma(\vec{x}).$$

Then there is an elementary function  $f$  such that  $f \models \Gamma(\vec{x})$  and  $f$  is strongly increasing on its variables.

*Proof.* If  $\Gamma$  is provable in  $\mathbf{I}\Delta_0(exp)$ , then it is provable in the Tait-style version of  $\mathbf{I}\Delta_0(exp)$ , where all cut formulas are  $\Sigma_1$ .

If  $\Gamma$  is classically derivable from non-logical axioms  $A_1, \dots, A_s$ , then there is a cut-free proof in the Tait calculus of  $\neg A_1, \Delta, \Gamma$ , where  $\Delta = \neg A_2, \dots, \neg A_s$ . Let us show how to cancel  $\neg A_1$  using a  $\Sigma_1$ -cut.

If  $A_1$  is an induction axiom on some formula  $B$ , then we have a cut-free proof of:

$$B(0) \wedge \forall y(\neg B(y) \vee B(y+1)) \wedge \exists x \neg B(x), \Delta, \Gamma$$

Thus we also have cut-free proofs of  $B(0), \Delta, \Gamma, \neg B(y), B(y+1), \Delta, \Gamma$  and  $\exists x \neg B(x), \Delta, \Gamma$ . So we have

$$\frac{\frac{\Delta, \Gamma, B(0) \quad \Delta, \Gamma, \neg B(y), B(y+1)}{\Delta, \Gamma, B(x)} \mathbf{BInd} \quad \frac{\Delta, \Gamma, \forall x B(x)}{\Delta, \Gamma} \forall \quad \frac{\exists x \neg B(x), \Delta, \Gamma}{\Delta, \Gamma} \Sigma_1\text{-cut}$$

We can similarly cancel each of  $\neg A_2, \dots, \neg A_s$  and so obtain the proof of  $\Gamma$  with  $\Sigma_1$ -cuts only.

Now we choose a proof of  $\Gamma(\vec{x})$  and proceed by induction on the height of the proof and determine an elementary function  $f$  such that  $f \models \Gamma$ .



1. If  $\Gamma(\vec{x})$  is an axiom, then for all  $\vec{n}$   $\Gamma(\vec{n})$  contains a true atom. So for any  $f$   $f \models \Gamma$ . Let us choose  $f(\vec{n}) = n_1 + \dots + n_k$ .
2. If  $\Gamma, B_0 \vee B_1$  is derivable, so is  $\Gamma, B_0, B_1$ . Note that  $B_0$  and  $B_1$  are both bounded. Let  $f \models \Gamma, B_0, B_1$ , then  $f \models \Gamma, B_0 \vee B_1$ .
3. Assume  $\Gamma, B_0 \wedge B_1$  is derivable, then  $\Gamma, B_0$  and  $\Gamma, B_1$ . By the induction hypothesis we have  $f_0 \models \Gamma, B_0$  and  $f_1 \models \Gamma, B_1$ , so, by persistence, we have  $\lambda \vec{n}. f_0(\vec{n}) + f_1(\vec{n}) \models \Gamma, B_0 \wedge B_1$ .
4. Assume  $\Gamma, \forall y B(y)$  is derivable, then  $\Gamma, B(y)$  is derivable and  $y$  is not free in  $\Gamma$ . Since all the formulas are  $\Sigma_1$ ,  $\forall x B(y)$  must be bounded, so  $B(y) = \neg(y < t) \vee B'(y)$  for some term  $t$  and for some bounded formula  $B'$ . By the induction hypothesis, assume  $f_0 \models \Gamma, \neg(y < t), B'(y)$  for some increasing elementary function  $f_0$ . Then we have:

$$f_0(\vec{n}, k) \models \Gamma(\vec{n}), \neg(k < t(\vec{n})), B'(\vec{n}, k)$$

Let  $g$  be an increasing elementary function bounding  $t$ , define

$$f(\vec{n}) = \sum_{k < g(\vec{n})} f(\vec{n}, k)$$

We have either  $f(\vec{n}) \models \Gamma(\vec{n})$  or, by persistence,  $B'(\vec{n}, k)$  is true for every  $k < t(\vec{n})$ . So  $f \models \Gamma, \forall y B(y)$  and  $f$  is elementary.

5. Assume  $\Gamma, \exists y A(y, \vec{x})$  is derivable, so  $\Gamma, A(t, \vec{x})$  is derivable for some term  $t$ . By the IH, there is elementary  $f_0$  such that for all  $\vec{n}$  one has

$$f_0(\vec{n}) \models \Gamma(\vec{n}), A(t(\vec{n}), \vec{n})$$

Then either  $f_0(\vec{n}) \models \Gamma(\vec{n})$  or else  $f_0(\vec{n})$  bounds true witnesses for all existential quantifiers in  $A(t(\vec{n}), \vec{n})$ . Choose an elementary function  $g$  which is bounding for  $t$ . Define  $f(\vec{n}) = f_0(\vec{n}) + g(\vec{n})$ , then for all  $\vec{n}$  either  $f(\vec{n}) \models \Gamma(\vec{n})$  or  $f(\vec{n}) \models \exists y A(y, \vec{n})$ .

6. Assume  $\Gamma$  comes about by the cut rule with  $\Sigma_1$  formula  $C = \exists \vec{z} B(\vec{z})$ , so the premises are  $\Gamma, \forall \vec{z} \neg B(\vec{z})$  and  $\Gamma, \exists \vec{z} B(\vec{z})$ .

Without increasing the height of a proof, we can invert all universal quantifiers in the first premise. So we have  $\neg B(\vec{z})$ .  $B$  is bounded, so the induction hypothesis can be applied to this formula to obtain an elementary function  $f_0$  such that, for all assignments  $[\vec{x} := \vec{n}]$  and  $[\vec{z} := \vec{m}]$

$$f_0(\vec{n}, \vec{m}) \models \Gamma(\vec{n}), \neg B(\vec{n}, \vec{m})$$

Now we apply the induction hypothesis to the second premise of the cut rule, so we have an elementary function  $f_1$  such that for all  $\vec{n}$  either  $f_1(\vec{n}) \models \Gamma(\vec{n})$  or there are fixed witnesses  $\vec{m} < f_1(\vec{n})$  such that  $B(\vec{n}, \vec{m})$  is true.

Define  $f$  the following way:

$$f(\vec{n}) = f_0(\vec{n}, f_1(\vec{n}), \dots, f_1(\vec{n}))$$

Furthermore  $f \models \Gamma$ . For otherwise there would be a tuple  $\vec{n}$  such that  $\Gamma(\vec{n})$  is not true at  $f(\vec{n})$ , so, by persistence,  $\Gamma(\vec{n})$  is not true at  $f_1(\vec{n})$ . Thus  $B(\vec{n}, \vec{m})$  is true for particular numbers  $\vec{m} < f_1(\vec{n})$ . But then  $f_0(\vec{n}, \vec{m}) < f(\vec{n})$ , so, by persistence,  $\Gamma(\vec{n})$  cannot be true at  $f_0(\vec{n}, \vec{m})$ . Thus  $B(\vec{n}, \vec{m})$  is false, so we have a contradiction.

7. Finally suppose  $\Gamma(\vec{x}), B(\vec{x}, t)$  comes from the induction rule on a bounded formula  $B$ . The premises of the rule  $\Gamma(\vec{x}), B(\vec{x}, 0)$  and  $\Gamma(\vec{x}), \neg B(\vec{x}, y), B(\vec{x}, y+1)$ .

Let us apply the induction hypothesis to each of the premises, and then we obtain increasing elementary functions  $f_0$  and  $f_1$  such that for all  $\vec{n}$  and for all  $k$

$$\begin{aligned} f_0(\vec{n}) &\models \Gamma(\vec{n}), B(\vec{n}, 0) \\ f_1(\vec{n}, k) &\models \Gamma(\vec{n}), \neg B(\vec{n}, k), B(\vec{n}, k+1) \end{aligned}$$

Now let

$$f(\vec{n}) = f_0(\vec{n}) + \sum_{k < g(\vec{n})} f_1(\vec{n}, k)$$

where  $g$  is an increasing elementary function which is bounding for the term  $t$ .  $f$  is elementary and increasing, and, by persistence for  $f_0$  and  $f_1$ , we have either  $f(\vec{n}) \models \Gamma(\vec{n})$  or else  $B(\vec{n}, 0)$  and  $B(\vec{n}, k) \rightarrow B(\vec{n}, k+1)$  are true for all  $k < t(\vec{n})$ . In either case, we have  $f \models \Gamma(\vec{x}), B(\vec{x}, t(\vec{x}))$ .

□

**Theorem 1.3.** A number-theoretic function is elementary iff  $f$  is provably  $\Sigma_1$  in  $\mathbf{I}\Delta_0(exp)$ .

*Proof.* The only if part is in Theorem 1.2, so we show the if part only. Assume  $f$  is provably  $\Sigma_1$  in  $\mathbf{I}\Delta_0(exp)$ . Then we have a formula

$$F(\vec{x}, y) = \exists z_1 \dots \exists z_k B(\vec{x}, y, z_1, \dots, z_k)$$

which defines  $f$  and such that

$$\mathbf{I}\Delta_0(exp) \models \exists y F(\vec{x}, y)$$

By Lemma 1.7, there exists an elementary function  $g$  such that for every tuple of arguments  $\vec{n}$  there are numbers  $m_0, \dots, m_k$  less than  $g(n)$  satisfying the bounded formula  $B(\vec{n}, m_0, m_1, \dots, m_k)$ . Apply the elementary sequence coding:

$$h(\vec{n}) = \langle g(\vec{n}), g(\vec{n}), \dots, g(\vec{n}) \rangle$$

so that if  $m = \langle m_0, m_1, \dots, m_k \rangle$  where  $m_i < g(\vec{n})$  for each  $i \in n+1$ , so  $m < h(\vec{n})$ .

As far as  $f(\vec{n})$  is the unique  $m_0$  for which there are  $m_1, \dots, m_k$  satisfying  $B(\vec{n}, m_0, \dots, m_k)$ , we define  $f$  as:

$$f(\vec{n}) = (\mu_{m < h(\vec{n})} B(\vec{n}, (m)_0, (m)_1, \dots, (m)_k))_0.$$

$B$  is a bounded formula of  $\mathbf{I}\Delta_0(exp)$ ,  $B$  is elementarily decidable. Moreover, elementary functions are closed under composition and bounded minimisation, so  $f$  is elementary.  $\square$

## 2 Primitive Recursion and $\mathbf{I}\Sigma_1$

$\mathbf{I}\Sigma_1$  is an arithmetical theory where the induction scheme is restricted to  $\Sigma_1$  formulas.

**Lemma 2.1.** Every primitive recursion is provably recursive in  $\mathbf{I}\Sigma_1$ .

*Proof.* We have to show represent each primitive recursive function  $f$  with a  $\Sigma_1$  formula  $F(\vec{x}, y) := \exists z C(\vec{x}, y, z)$  such that:

1.  $f(\vec{n}) = m$  iff  $\omega \models F(\vec{x}, y)$ .
2.  $\mathbf{I}\Sigma_1 \vdash \exists y F(\vec{x}, y)$ .
3.  $\mathbf{I}\Sigma_1 \vdash F(\vec{x}, y) \wedge F(\vec{x}, y') \rightarrow y = y'$ .

In each case  $C(\vec{x}, y, z)$  will be a  $\Delta_0(exp)$ -formula constructed via sequence encoding in  $\mathbf{I}\Delta_0(exp)$ . Such a formula expresses that  $z$  is a uniquely determined sequence number encoding the computation of  $f(\vec{x}) = y$  and containing the output value  $y$  as its final element, so  $y = \pi_2(z)$ .

Condition 1 will hold by the definition of  $C$ . Condition 3 will be satisfied by the uniqueness of  $z$ . We consider five definitional schemes by which  $f$  could be introduced:

1.  $f$  is the constant-zero function, that is,  $f(x) = 0$ , no matter what  $x$  is. Then we take  $C := y = 0 \wedge z = \langle 0 \rangle$ . All the conditions are obviously satisfied.
2. If  $f$  is the successor function  $f(x) = x + 1$ , we let

$$C(x, y, z) := y = x + 1 \wedge z = \langle x + 1 \rangle$$

All the conditions are obvious.

3. Now assume  $f$  is the projection function  $f(x_0, \dots, x_n) = x_i$  for some  $i \in n + 1$ . We let

$$C(\vec{x}, y, z) := y = x_i \wedge z = \langle x_i \rangle$$

4. Now assume  $f$  is defined by substitution from previously generated primitive recursive functions  $f_0, f_1, f_2$ :

$$f(\vec{x}) = f_0(f_1(\vec{x}), f_2(\vec{x}))$$

By the induction hypothesis, assume that  $f_0, f_1, f_2$  are provably recursive and we have  $\Delta_0(exp)$ -formulas  $C_0, C_1, C_2$  encoding their computations ( $\text{len}(z) = 4$ ). For the function  $f$  define:

$$C(\vec{x}, y, z) := \bigwedge_{i \in \{1, 2\}} C_i(\vec{x}, \pi_2((z)_i), (z)_i) \wedge C_0(\pi_2((z)_1), \pi_2((z)_2), y, (z)_0) \wedge (z)_3 = y.$$

Let us check the required conditions:

- (a) Condition 1 holds since  $f(\vec{n}) = m$  iff there are numbers  $m_1$  and  $m_2$  such that  $f_1(\vec{n}) = m_1$ ,  $f_2(\vec{n}) = m_2$  and  $f_0(m_1, m_2) = m$ . These hold if and only if there are number  $k_1, k_2, k_0$  such that  $C_1(\vec{n}, m_1, k_1)$ ,  $C_2(\vec{n}, m_2, k_2)$  and  $C_0(m_1, m_2, m, k_0)$  are all true. And these hold if and only if  $C(\vec{n}, m, \langle k_0, k_1, k_2, m \rangle)$  is true. Thus  $f(\vec{n}) = m$  iff and only if  $F(\vec{n}, m) = \exists z C(\vec{n}, m, z)$  is true.
  - (b) Condition 2 holds since from  $C_1(\vec{x}, y_1, z_1)$ ,  $C_2(\vec{x}, y_2, z_2)$  and  $C(y_1, y_2, y, z_0)$  we can derive  $C(\vec{x}, y, \langle z_0, z_1, z_2, y \rangle)$  in  $\mathbf{I}\Delta_0$ . So provided  $\exists y \exists z C_1(\vec{x}, y, z)$ ,  $\exists y \exists z C_2(\vec{x}, y, z)$  and  $\forall y_1 \forall y_2 \exists y \exists z C(y_1, y_2, y, z)$ , we can prove  $\exists y F(\vec{x}, y) := C(\vec{x}, y, z)$ .
  - (c) Condition 3 is self-evident.
5. Now assume that  $f$  is defined from  $f_1$  and  $f_2$  by primitive recursion:

$$\begin{aligned} f(\vec{v}, 0) &= f_0(\vec{v}) \\ f(\vec{v}, x+1) &= f_1(\vec{v}, x, f(\vec{v}, x)) \end{aligned}$$

By the induction hypothesis  $f_0$  and  $f_1$  are provably recursive and they have associated  $\Delta_0$ -formulas  $C_0$  and  $C_1$ . Define

$$\begin{aligned} C(\vec{v}, x, y, z) &:= C_0(\vec{v}, \pi_2((z)_0), (z)_0) \wedge \\ &\quad \forall i < x \ (C_i(\vec{v}, i, \pi_2((z)_i), \pi_2((z)_{i+1}))) \wedge \\ &\quad (z)_{x+1} = y \wedge \pi_2((z)_x) = y \end{aligned}$$

Let us check that all the conditions are satisfied:

- (a) Condition 1 holds since  $f(\vec{l}, n) = m$  if and only if there is a sequence number  $k = \langle k_0, \dots, k_n, m \rangle$  such that  $k_0$  encodes the computation of  $f(\vec{l}, 0)$  with the value  $\pi_2(k_0)$ , and for each  $i < n$ ,  $k_{i+1}$  codes the computation of  $f(\vec{l}, i+1) = f_1(\vec{l}, i, \pi_2(k_i))$  with values  $\pi_2(k_{i+1})$  and  $\pi_2(k_n) = m$ . This is equivalent to  $\models F(\vec{l}, n, m) \leftrightarrow \exists z C(\vec{l}, n, m, z)$ .

(b) To show Condition 2 we have to prove the following in  $\mathbf{I}\Delta_0$

$$C_0(\vec{v}, y, z) \rightarrow C(\vec{v}, 0, y, \langle z, y \rangle)$$

and

$$C(\vec{v}, x, y, z) \wedge C_1(\vec{v}, x, y, y', z') \rightarrow C(\vec{v}, x+1, y', t)$$

for a suitable term  $t$  which removes the end component  $y$  of  $z$  and replaces it by  $z'$ , and then adds the final component  $y'$ . More specifically

$$t = \pi(\pi(\pi_1(z), z'), y')$$

Hence from  $\exists y \exists z C_0(\vec{v}, y, z)$  we obtain  $\exists y \exists z C(\vec{v}, 0, y, z)$ , and from  $\forall y \exists y' \exists z' C_1(\vec{v}, x, y, y', z')$  one can derive

$$\exists y \exists z C(\vec{v}, x, y, z) \rightarrow \exists y \exists z C(\vec{v}, x+1, y, z)$$

We have assumed that  $f_0$  and  $f_1$  are primitive recursive, we can prove  $\exists y F(\vec{v}, 0, y)$  and  $\exists y F(\vec{v}, x, y) \rightarrow \exists y F(\vec{v}, x+1, y)$ . Then we derive  $\exists y F(\vec{v}, x, y)$  by using  $\Sigma_1$ -induction.

(c) To show Condition 3 assume  $C(\vec{v}, x, y, z)$  and  $C(\vec{v}, x, y', z')$ , where  $z$  and  $z'$  are sequence numbers of the same length  $x+2$ . Furthermore we have  $C_0(\vec{v}, \pi_2((z)_0), (z)_0)$  and  $C_0(\vec{v}, \pi_2((z')_0), (z')_0)$ , so we have  $(z)_0 = (z')_0$ .

Similarly we have  $\forall i < x \ C_1(\vec{v}, i, \pi_2((z)_i), \pi_2((z)_{i+1}), (z)_{i+1})$  and the same formula where  $z$  is replaced by  $z'$ . So if  $(z)_i = (z')_i$ , and one can deduce  $(z)_{i+1} = (z')_{i+1}$  using the uniqueness assumption for  $C_1$ . By  $\Delta_0(exp)$ -induction we obtain  $\forall i \leq x \ ((z)_i = (z')_i)$ .

The final conjuncts in  $C$  give  $(z)_{x+1} = \pi_2((z)_x) = y$  and the same formulas where  $z$  is replaced by  $z'$  and where  $y$  is replaced by  $y'$ . But since  $(z)_x = (z')_x$  we have  $y = y'$ , since all the components are equal,  $z = z'$ . Thus we have  $F(\vec{v}, x, y) \wedge F(\vec{v}, x, y') \rightarrow y = y'$ .

□

## 2.1 $\mathbf{I}\Sigma_1$ provable functions are primitive recursive

**Definition 2.1.** A closed  $\Sigma_1$ -formula  $\exists \vec{z} B(z)$  with  $B \in \Delta_0(exp)$  is said to be “true at  $m$ ” (denoted as  $m \models \exists \vec{z} B(z)$ ) if there are numbers  $\vec{m} = (m_1, \dots, m_l)$  such that all  $m_i < m$  for  $i \in \{1, \dots, l\}$  such that  $B(\vec{m})$  is true in the standard model.

A finite set of formulas  $\Gamma$  of closed  $\Sigma_1$ -formulas is “true at  $m$ ” (denoted as  $m \models \Gamma$ ) if at least one of them is true at  $m$ .

If  $\Gamma(x_1, \dots, x_k)$  is a finite set of  $\Sigma_1$ -formulas all of whose free variables occur amongst  $x_1, \dots, x_k$  and if  $f : \mathbb{N}^k \rightarrow \mathbb{N}$ , then we write  $f \models \Gamma$  if for each assignments  $\vec{n} = (n_1, \dots, n_k)$  to the variables  $x_1, \dots, x_k$  we have  $f(\vec{n}) \models \Gamma(\vec{n})$ .

Note that we have the persistence property for  $\models$  which completely repeats persistence for  $\mathbf{I}\Delta_0(exp)$ .

We shall be using a Tait-style formalisation of  $\mathbf{I}\Sigma_0$  where the induction rule

$$\frac{\Gamma, A(0) \quad \Gamma, \neg A(y), A(y+1)}{\Gamma, A(t)}$$

where  $y$  is not free in  $\Gamma$ ,  $t$  is any term and  $A$  is any  $\Sigma_1$ -formula.

**Lemma 2.2.** ( $\Sigma_1$ -induction) Let  $\Gamma(\vec{x})$  be a finite set of  $\Sigma_1$ -formulas such that

$$\mathbf{I}\Sigma_1 \vdash \bigvee \Gamma(\vec{x})$$

then there is a primitive recursive function  $f$  such that  $f \models \Gamma$  and  $f$  is strictly increasing on its variables.

*Proof.* We note that if  $\Gamma$  is provable in this formalisation, then it has a proof in which all the non-atomic cut formulas are induction  $\Sigma_1$ -formulas. If  $\Gamma$  is classically derivable from non-logical axioms  $A_1, \dots, A_s$ , then there is a cut-free proof (à la Tait) of  $\neg A_1, \Delta, \Gamma$  where  $\Delta = A_2, \dots, A_s$ . Then if  $A_1$  is an induction axiom on a formula  $F$ , then we have have a cut-free proof of

$$F(0) \wedge \forall y (\neg F(y) \vee F(y+1)) \wedge \neg F(t), \Delta, \Gamma$$

and thus, by inversion, we have cut-free proofs of  $F(0), \Delta, \Gamma$ ,  $\neg F(y), F(y+1), \Delta, \Gamma$  and  $\neg F(t), \Delta, \Gamma$ .

So we obtain  $F(t), \Delta, \Gamma$  by the induction rule and then we obtain  $\Delta, \Gamma$  by cutting  $F(t)$ . One can detach  $\neg A_2, \dots, \neg A_s$ , so we finally have a proof of  $\Gamma$  which uses cuts only on  $\Sigma_1$ -induction formulas or on atoms arising from non-logical axioms. Such proofs are said to be “free-cut” free.

Let us choose such a proof for  $\Gamma(\vec{x})$  and show by induction on the height of a proof that there exists a primitive recursive function satisfying  $f \models \Gamma$ .

1. Let  $\Gamma(\vec{x})$  be an axiom, then for all  $\vec{n}$   $\Gamma(\vec{n})$  contains a true atom. Choose  $f(\vec{n}) = n_1 + \dots + n_k$ , and  $f$  is clearly primitive recursive, strictly increasing and  $f \models \Gamma$ .
2. Assume we have

$$\frac{\Gamma, B_0, B_1}{\Gamma, B_0 \vee B_1} \vee$$

Then both  $B_0$  and  $B_1$  are both  $\Delta_0(exp)$ -formulas, so any function  $f$  satisfying  $f \models \Gamma, B_0, B_1$  also satisfies  $\Gamma, B_0 \vee B_1$ .

3. Assume we have

$$\frac{\Gamma, B_0 \quad \Gamma, B_1}{\Gamma, B_0 \wedge B_1} \wedge$$

By the induction hypothesis we have  $f_i(\vec{n}) \models \Gamma(\vec{n}), B_i(\vec{n})$  where  $i \in \{0, 1\}$  for all  $\vec{n}$ . By the persistence property,  $\lambda\vec{n}.f_0(\vec{n}) + f_1(\vec{n}) \models \Gamma, B_0 \wedge B_1$ .

4. Assume we have

$$\frac{\Gamma, B(y)}{\Gamma, \forall y B(y)} \forall$$

where  $y$  is not free in  $\Gamma$ . As far as all formulas are  $\Sigma_1$ ,  $\forall y B(y)$  must be  $\mathbf{I}\Delta_0(exp)$ , so  $B(y) = \neg(y < t) \vee B'(y)$  for some elementary or primitive recursive term  $t$ . Assume we have  $f_0 \models \Gamma, \neg(y < t) \vee B'(y)$  for some increasing primitive recursive function  $f_0$ . Then, for any assignments  $\vec{x} \mapsto \vec{n}$  and  $y \mapsto k$ , we have

$$f_0(\vec{n}, k) \models \Gamma(\vec{n}), \neg(k < t(\vec{n})), B'(\vec{n}, k).$$

We let

$$f(\vec{n}) = \sum_{k < g(\vec{n})} f_0(\vec{n}, k)$$

for some function  $g$ , which is increasing primitive recursive bounding the values of term  $t$ . So we have either  $f(\vec{n}) \models \Gamma$  or  $B'(\vec{n}, k)$  is true for every  $k < t(\vec{n})$ . Thus  $f \models \Gamma, \forall y B(y)$  as required.

5. Suppose we have

$$\frac{\Gamma, A(t)}{\Gamma, \exists y A(y)} \exists$$

where  $A$  is a  $\Sigma_1$ -formula. By the induction hypothesis we have a function  $f_0$  such that for all  $\vec{n}$

$$f_0(\vec{n}) \models \Gamma(\vec{n}), A(t(\vec{n}), \vec{n})$$

Then either  $f_0(\vec{n}) \models \Gamma(\vec{n})$  or otherwise  $f_0(\vec{n})$  bounds true witnesses for all the existential quantifiers already in  $A(t(\vec{n}), \vec{n})$ . Choose an elementary bounding function  $g$  for the term  $t$  and define  $f(\vec{n}) = f_0(\vec{n}) + g(\vec{n})$ , so we have either  $f(\vec{n}) \models \Gamma(\vec{n})$  or  $f(\vec{n}) \models \exists y A(y, \vec{n})$  for all  $\vec{n}$ .

6. Assume we have

$$\frac{\Gamma, \forall \vec{z} \neg B(\vec{z}) \quad \Gamma, \exists \vec{z} B(\vec{z})}{\Gamma} \text{cut}$$

where  $\exists \vec{z} B(\vec{z})$  is a cut  $\Sigma_1$ -formula.

Note that we have

$$\frac{\Gamma, \neg B(\vec{z})}{\Gamma, \forall \vec{z} \neg B(\vec{z})} \forall$$

Note  $B$  is a  $\Delta_0(\text{exp})$ -formula, so let us apply the induction hypothesis to obtain a primitive recursive function  $f_0$  such that for each assignments  $\vec{x} \mapsto \vec{n}$  and  $\vec{z} \mapsto \vec{m}$

$$f_0(\vec{n}, \vec{m}) \models \Gamma(\vec{n}), \neg B(\vec{n}, \vec{m}).$$

We apply the induction hypothesis to the second premise to obtain a primitive recursive function  $f_1$  such that for all  $\vec{n}$  either  $f_1(\vec{n}) \models \Gamma(\vec{n})$  or otherwise there are fixed witnesses  $\vec{m} < f_1(\vec{n})$  s.t.  $B(\vec{n}, \vec{m})$  is true. Let us define  $f$  by substitution:

$$f(\vec{n}) = f_0(\vec{n}, f_1(\vec{n}), \dots, f_1(\vec{n}))$$

where  $f$  is primitive recursive, greater or equal that  $f_1$  (pointwise) and strictly increasing. Furthermore  $f \models \Gamma$ .

For otherwise, let us suppose there exists a tuple  $\vec{n}$  such that  $\Gamma(\vec{n})$  is not true  $f(\vec{n})$  and, thus, by persistence at  $f_1(\vec{n})$ . So  $B(\vec{n}, \vec{m})$  is true for some  $\vec{m} < f_1(\vec{n})$ . Thus  $f_0(\vec{n}, \vec{m}) < f(\vec{n})$ , and then, by persistence,  $\Gamma(\vec{n})$  cannot be true at  $f_0(\vec{n}, \vec{m})$ . Then  $B(\vec{n}, \vec{m})$ , so we have a contradiction.

7. Suppose we have

$$\frac{\Gamma(\vec{x}), A(\vec{x}, 0) \quad \Gamma, \neg A(\vec{x}, y), A(\vec{x}, y+1)}{\Gamma, A(\vec{x}, t)}$$

where  $A(\vec{x}, y)$  is an induction  $\Sigma_1$ -formula of the form  $\exists \vec{z} B(\vec{x}, y, \vec{z})$ . Let us invert universal quantifiers in  $\neg A(\vec{x}, y)$ , the second premise of the rule becomes

$$\Gamma(\vec{x}), \neg B(\vec{x}, y, \vec{z}), A(\vec{x}, y+1)$$

which is now a set  $\Sigma_1$ -formulas. We can apply the induction hypothesis to each of the premises to have primitive recursive function  $f_0$  and  $f_1$  such that for each  $\vec{n}$ ,  $k$  and  $\vec{m}$

$$\begin{aligned} f_0(\vec{n}) &\models \Gamma(\vec{n}), A(\vec{n}, 0) \\ f_1(\vec{n}, k, \vec{m}) &\models \Gamma(\vec{n}), \neg B(\vec{n}, k, \vec{m}), A(\vec{n}, k+1) \end{aligned}$$

Define  $f$  by primitive recursion from  $f_0$  and  $f_1$  the following way

$$\begin{aligned} f(\vec{n}, 0) &= f_0(\vec{n}) \\ f(\vec{n}, k+1) &= f_1(\vec{n}, k, f(\vec{n}, k), \dots, f(\vec{n}, k)) \end{aligned}$$



Then for all  $\vec{n}$  and for all  $\vec{k}$  one has  $f(\vec{n}, k) \models \Gamma(\vec{n}), A(\vec{n}, k)$  which is shown by induction on  $k$ . The base case holds by the definition of  $f_0(\vec{n})$ . For the induction step assume that  $f(\vec{n}, k) \models \Gamma(\vec{n}), A(\vec{n}, k)$ . If  $\Gamma(\vec{n})$  is not true at  $f(\vec{n}, k + 1)$ . By persistence it is not true at  $f(\vec{n}, k)$  and thus  $f(\vec{n}, k) \models A(\vec{n}, k)$ . Therefore there are numbers  $\vec{m} < f(\vec{n}, k)$  such that  $B(\vec{n}, k, \vec{m})$  is true. Thus  $f_1(\vec{n}, k, \vec{m}) \models \Gamma(\vec{n}), A(\vec{n}, k + 1)$  and since  $f_1(\vec{n}, k, \vec{m}) \leq f(\vec{n}, k + 1)$  we have, by persistence,  $f(\vec{n}, k + 1) \models \Gamma(\vec{n}), A(\vec{n}, k + 1)$  as required.

So we substitute for the final argument  $k$  in  $f$  an elementary (or primitive recursive) function  $g$  which bounds the values of  $t$ , so that  $f'(\vec{n}) = f(\vec{n}, g(\vec{n}))$ , and thus we have  $f(\vec{n}, t(\vec{n})) \models \Gamma(\vec{n}), A(\vec{n}, t(\vec{n}))$  for all  $\vec{n}$  and thus, by persistence,  $f' \models \Gamma(\vec{x}), A(\vec{x}, t)$ .

□

**Theorem 2.1.** The provably recursive functions of  $\mathbf{I}\Sigma_1$  are exactly primitive recursive functions.

*Proof.* We have already shown that all primitive recursive functions are provably recursive in  $\mathbf{I}\Sigma_1$ , so let us show the converse.

Let  $g : \mathbb{N}^k \rightarrow \mathbb{N}$  be a function defined by a  $\Sigma_1$ -formula  $F(\vec{x}, y) := \exists z C(\vec{x}, y, z)$  where  $C$  is  $\Delta_0(exp)$  and  $\mathbf{I}\Sigma_1 \models \exists y F(\vec{x}, y)$ . By the lemma above, there exists a primitive recursive function  $f$  such that for all  $n \in \mathbb{N}^k$

$$f(\vec{n}) \models \exists y \exists z C(\vec{n}, y, z).$$

That is, for every  $\vec{n}$  there is an  $m < f(\vec{n})$  and a  $k < f(\vec{n})$  such that  $C(\vec{n}, m, k)$  is true and this  $m$  is the value of  $g(\vec{n})$ .

$g$  can be defined by primitive recursion from  $f$  the following way:

$$g(\vec{n}) = (\mu_{m < h(\vec{n})} C(\vec{n}, (m)_0, (m)_1))$$

where  $h(\vec{n}) = \langle f(\vec{n}), f(\vec{n}) \rangle$ .

□

### 3 $\varepsilon_0$ -recursion in Peano Arithmetic

We show that the provably recursive functions of Peano arithmetic are  $\varepsilon_0$ -recursive functions, that is, functions definable from the primitive recursive functions by substitutions and recursion over well-orderings of natural numbers with order types strictly less than the ordinal

$$\varepsilon_0 = \sup\{\omega, \omega^\omega, \omega^{\omega^\omega}, \dots\}$$

Equivalently,  $\varepsilon_0$  can be defined as the least fixed point of the mapping  $\alpha \mapsto \omega^\alpha$  where  $\alpha$  is an ordinal.

Let us discuss first how one can represent ordinals below  $\varepsilon_0$ .

### 3.1 Ordinals below $\varepsilon_0$

Every ordinal  $\alpha < \varepsilon_0$  is either 0 or  $\alpha$  can be represented uniquely in *Cantor normal form*:

$$\alpha = \omega^{\gamma_1} \cdot c_1 + \omega^{\gamma_2} \cdot c_2 + \cdots + \omega^{\gamma_k} \cdot c_k$$

where  $k < \omega$ ,  $\gamma_k < \cdots < \gamma_2 < \gamma_1 < \alpha$  and  $c_1, \dots, c_k < \omega$  are coefficients. If  $\gamma_k = 0$ , then  $\alpha$  is a successor ordinal, written  $\text{Succ}(\alpha)$ , and its predecessor  $\alpha - 1$  the representation

$$\alpha = \omega^{\gamma_1} \cdot c_1 + \omega^{\gamma_2} \cdot c_2 + \cdots + \omega^{\gamma_{k-1}} \cdot c_{k-1}.$$

Otherwise  $\alpha$  is a limit ordinal, written  $\text{Lim}(\alpha)$ , and it has infinitely many possible increasing sequences of smaller ordinals whose limit is  $\alpha$ .

We shall pick out one concrete sequence  $\{\alpha(n) \mid n < \omega\}$  for each limit ordinal  $\alpha$  the following way. First write  $\alpha$  as  $\delta + \omega^\gamma$  where

$$\begin{aligned} \delta &= \omega^{\gamma_1} \cdot c_1 + \cdots + \omega^{\gamma_k} \cdot (c_k - 1) \\ \gamma &= \gamma_k. \end{aligned}$$

By induction we can assume that when  $\gamma$  is a limit ordinal, its fundamental sequence  $\{\gamma(n) \mid n < \omega\}$  has been already specified. We let for each  $n < \omega$

$$\alpha(n) = \begin{cases} \delta + \omega^{\gamma^{-1}} \cdot (n + 1), & \text{if } \text{Succ}(\gamma) \\ \delta + \omega^{\gamma(n)}, & \text{if } \text{Lim}(\gamma). \end{cases}$$

Clearly

$$\alpha = \lim_{n \rightarrow \omega} \alpha(n).$$

**Definition 3.1.** Let  $\alpha < \varepsilon_0$  and  $n < \omega$ , define a finite set of ordinals  $\alpha[n]$  the following way:

$$\alpha[n] = \begin{cases} \emptyset, & \text{if } \alpha = 0 \\ (\alpha - 1)[n] \cup \{\alpha - 1\}, & \text{if } \text{Succ}(\alpha) \\ \alpha(n)[n], & \text{if } \text{Lim}(\alpha) \end{cases}$$

**Lemma 3.1.** For each  $\alpha = \delta + \omega^\gamma$  and for each  $n < \omega$

$$\alpha[n] = \delta[n] \cup \{\delta + \omega^{\gamma_1} \cdot c_1 + \cdots + \omega^{\gamma_k} \cdot c_k \mid \forall i (\gamma_i \in \gamma[n] \wedge c_i \leq n)\}.$$

*Proof.* Induction on  $\gamma$ .

1.  $\gamma = 0$ , then  $\gamma[n] = \emptyset$  and the right hand side is  $\delta[n] \cup \{\delta\}$ , which is the same as  $\alpha[n] = (\delta + 1)[n]$ .
2. If  $\gamma$  is limit, then  $\gamma[n] = \gamma(n)[n]$ , so the right hand side is the same as the one with  $\gamma(n)[n]$  instead of  $\gamma[n]$ . By the induction hypothesis applied to  $\alpha(n) = \delta + \omega^{\gamma(n)}$ , which is equal to  $\alpha(n)[n]$ , which is  $\alpha[n]$  by definition.

3. Suppose  $\gamma$  is a successor. Then  $\alpha$  is a limit and  $\alpha[n] = \alpha(n)[n]$ , where  $\alpha(n) = \delta + \omega^{\gamma-1} \cdot (n+1)$ . So we can write  $\alpha(n) = \alpha(n-1) + \omega^{\gamma-1}$ , where  $\alpha(-1) = \delta$  when  $n = 0$ . By the induction hypothesis for  $\gamma-1$ , the set  $\alpha[n]$  equals

$$\alpha(n-1)[n] \cup \{\alpha(n-1) + \omega^{\gamma_1} \cdot c_1 + \dots + \omega^{\gamma_k} \cdot c_k \mid \forall i (\gamma_i \in (\gamma-1)[n] \wedge c_i \leq n)\}$$

and similarly for each  $\alpha(n-1)[n], \alpha(n-2)[n], \dots, \alpha(1)[n]$ . For each  $m \leq n$ ,  $\alpha(m-q) = \delta + \omega^{\gamma-1} \cdot m$ . In turn, this last set is the same as

$$\delta[n] \cup \{\delta + \omega^{\gamma-1} \cdot m + \omega^{\gamma_1} \cdot c_1 + \dots + \omega^{\gamma_k} \cdot c_k \mid \forall i (\gamma_i \in (\gamma-1)[n] \wedge c_i \leq n) \wedge m \leq n\}$$

and this is the set since  $\gamma[n] = (\gamma-1)[n] \cup \{\gamma-1\}$ .

□

**Corollary 3.1.** Let  $\alpha < \varepsilon_0$  be a limit ordinal, then for every  $0 \neq n < \omega$   $\alpha(n) \in \alpha[n+1]$ . Furthermore if  $\beta \in \gamma[n]$ , then  $\omega^\beta \in \omega^\gamma[n]$ .

**Definition 3.2.** The *maximum coefficient* of  $\beta = \omega^{\beta_1} \cdot b_1 + \dots + \omega^{\beta_l} \cdot b_l$  is defined by induction to be the maximum of all the  $b_i$ 's and all the maximum coefficients of the exponents  $\beta_i$ 's.

**Lemma 3.2.** If  $\beta < \alpha$  and the maximum coefficient of  $\beta$  is  $\leq n$ , so  $\beta \in \alpha[n]$ .

*Proof.* By induction on  $\alpha$ . Let  $\alpha = \delta + \omega^\gamma$ . If  $\beta < \delta$ , then  $\beta \in \delta[n]$  by the induction hypothesis and  $\delta[n] \subseteq \alpha[n]$  by Lemma 3.1. Otherwise

$$\beta = \delta + \omega^{\beta_1} \cdot b_1 + \dots + \omega^{\beta_k} \cdot b_k$$

for  $\alpha > \gamma > \beta_1 > \dots > \beta_k$  and  $b_i \leq n$ . By induction hypothesis  $\beta_i \in \gamma[n]$ , so  $\beta \in \alpha[n]$  by Lemma 3.1. □

**Definition 3.3.** Let  $G_\alpha(n)$  denote the cardinality of the finite set  $\alpha[n]$ . We have

$$G_\alpha(n) = \begin{cases} 0, & \text{if } \alpha = 0 \\ G_{\alpha-1}(n+1), & \text{if } \text{Succ}(\alpha) \\ G_{\alpha(n)}(n), & \text{if } \text{Lim}(\alpha) \end{cases}$$

The hierarchy of functions  $G_\alpha$  is the *slow-growing* hierarchy.

**Lemma 3.3.** If  $\alpha = \delta + \omega^\gamma$ , then for all  $n < \omega$

$$G_\alpha(n) = G_\delta(n) + (n+1)^{G_\gamma(n)}.$$

Thus for each  $\alpha < \varepsilon_0$ ,  $G_\alpha(n)$  is the elementary function which results by substituting  $n+1$  for every occurrence of  $\omega$  in the Cantor normal form  $\omega$ .

*Proof.* Induction on  $\gamma$ .

1. If  $\gamma = 0$ , then  $\alpha = \delta + 1$ , thus

$$G_\alpha(n) = G_\delta(n) + 1 = G_\delta(n) + (n + 1)^0.$$

2. If  $\gamma$  is a successor, then  $\alpha = \delta + \omega^\gamma$  is limit and  $\alpha(n) = \delta + \omega^{\gamma-1} \cdot (n + 1)$ , so we apply the induction hypothesis for  $\gamma - 1$   $n + 1$  times and thus we have

$$G_\alpha(n) = G_{\alpha(n)}(n) = G_\delta(n) + (n + 1)^{G_{\gamma-1}(n) \cdot (n + 1)} = G_\delta(n) + (n + 1)^{G_\gamma(n)}$$

since  $G_{\gamma-1}(n) + 1 = G_\gamma(n)$ .

3. If  $\gamma$  is a limit ordinal, then  $\alpha(n) = \delta + \omega^{\gamma(n)}$ , so let us apply the induction hypothesis to  $\gamma(n)$ , then we have

$$G_\alpha(n) = G_{\alpha(n)}(n) = G_\delta(n) + (n + 1)^{G_{\gamma(n)}(n)}$$

which gives the result since  $\Gamma_{\gamma(n)}(n) = G_\gamma(n)$ .

□

**Definition 3.4. (Coding ordinals)**

Let  $\beta = \omega^{\beta_1} \cdot b_1 + \dots \omega^{\beta_l} \cdot b_l$  be an ordinal. A *coding ordinal* is the sequence number  $\bar{\beta}$  constructed recursively the following way

$$\bar{\beta} = \langle \langle \bar{\beta}_1, b_1 \rangle, \dots, \langle \bar{\beta}_l, b_l \rangle \rangle.$$

where 0 is coded by the empty sequence number.  $\bar{\beta}$  is numerically greater than the maximum coefficient of  $\beta$  and greater than the codes  $\bar{\beta}_i$  of all its exponents and their exponents, etc.

**Lemma 3.4.**

1. There exists an elementary function  $h : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  such that, for each ordinal  $\beta = \omega^{\beta_1} \cdot b_1 + \dots \omega^{\beta_l} \cdot b_l$ :

$$h(\bar{\beta}, n) = \begin{cases} 0, & \text{if } \beta = 0 \\ \bar{\beta} - 1, & \text{if } \text{Succ}(\beta) \\ \overline{\beta(n)}, & \text{if } \text{Lim}(\beta) \end{cases}$$

2. For each ordinal  $\alpha < \varepsilon_0$  there exists an elementary well-ordering  $\prec_\alpha \subset \mathbb{N} \times \mathbb{N}$  such that

$$\forall b, c \in \mathbb{N} \quad b \prec_\alpha c \leftrightarrow \exists \beta, \gamma < \alpha \quad \beta < \gamma \ \& \ b = \bar{\beta} \ \& \ c = \bar{\gamma}.$$

*Proof.*

1. First let

$$h(0, n) = 0$$

for any  $n$ . Then let  $0 < m < \omega$  be a non-zero sequence number. We first should see if the rightmost component  $\pi_2$  is a pair  $(m', n')$ . If so and  $m' = 0$  and  $n' \neq 0$ , then  $\beta$  is a successor and the code of its predecessor,  $h(m, n)$ , is defined as the new sequence number that we obtain by reducing  $n'$  by one or by removing this final component if  $n' = 1$ .

If  $\pi_2(m) = \langle m', n' \rangle$  where both  $m'$  and  $n'$  are non-zero, then  $\beta$  is a limit ordinal of the form  $\delta + \omega^\gamma \cdot n'$  where  $m' = \bar{\gamma}$ . Let  $k$  be the code of  $\delta + \omega^\gamma \cdot (n' - 1)$ , which is obtained by reducing  $n'$  by one inside  $m$  (or by deleting the final component from  $m$  when  $n' = 1$ ).

At the “right hand end” of  $\beta$  we have a “spare”  $\omega^\gamma$  which must be either reduced to  $\omega^{\gamma-1} \cdot (n + 1)$  when  $\text{Succ}(\gamma)$  or to  $\omega^{\gamma(n)}$  if  $\text{Lim}(\gamma)$ . In either case we are able to produce  $\beta(n)$ . Thus the required code  $h(m, n)$  of  $\beta(n)$  will be obtained by tagging on to the end of the sequence number  $k$  one additional pair encoding this additional term.

If we assume inductively that  $h(m', n)$  has been already defined for  $m' < m$ , then such an additional component is either  $\langle h(m', n), n + 1 \rangle$  if  $\text{Succ}(\gamma)$  or  $\langle h(m', n), 1 \rangle$  if  $\text{Lim}(\gamma)$ .

This defines  $h(m, n)$ , but such a definition is actually primitive recursive so far. Let us check that  $h$  is elementarily bounded, i.e.  $h$  is defined by limited recursion from elementary functions. Note that  $h(m, n) < m$  whenever  $m$  codes a successor ordinal. If  $m$  codes a limit ordinal,  $h(m, n)$  is obtained from the sequence number  $k < m$  by adding a new pair on the end. An extra item  $i$  is tagged on the end of a sequence number  $k$  by the function  $\pi(k, i)$  which is quadratic in both argument. If the item added is the pair  $\langle h(m', n), n + 1 \rangle$  where  $\text{Succ}(\gamma)$ , then  $h(m', n) < m$ , so  $h(m, n)$  is numerically bounded by some fixed polynomial in  $m$  and  $n$ . In the other case, we can say that  $h(m, n)$  is numerically bounded by some fixed polynomial of  $m$  and  $h(m', n)$ . Since  $m'$  codes an exponent in the Cantor normal form encoded by  $m$ , the second polynomial is iterated at most  $d$  times, where  $d$  is the “exponential height” of the normal form. Thus  $h(m, n)$  is bounded by some  $d$ -times iterated polynomial of  $m + n$ .  $d < m$ , so  $h(m, n)$  is bounded by the elementary function  $2^{2^{c \cdot (m+n)}}$  for some  $c < \omega$ . Therefore  $h$  is elementary as it is defined by bounded recursion.

2. Let  $\alpha < \varepsilon_0$  and let  $d$  be the exponential height of its Cantor normal form. We use the function  $h$  from the previous part, but we apply it to codes below  $\alpha$  only. They have the exponential height  $\leq d$ , so we can consider  $h$  as being bounded by some fixed polynomial of its two arguments. Define  $g(0, n) = \bar{\alpha}$  and  $g(i + 1, n) = h(g(i, n), n)$  and notice that  $g$  is therefore bounded by an  $i$ -times iterated polynomial, so  $g$  is defined by an elementarily limited recursion from  $h$ , so it is elementary.

Define  $b \prec_\alpha c$  if and only if  $c \neq 0$  and there are  $i$  and  $j$  such that  $0 < i < j \leq G_\alpha(\max(b, c) + 1)$  and  $g(i, \max(b, c)) = c$  and  $g(j, \max(b, c)) = b$ .

The function  $g$  and  $G_\alpha$  are elementary, so is the relation  $\prec_\alpha$  since the quantifiers are bounded. By the properties of  $h$  it is clear that if  $i < j$  then  $g(j, \max(b, c))$  codes an ordinal greater than  $g(j, \max(b, c))$ . Hence  $b \prec_\alpha c$ , then  $b = \bar{\beta}$  and  $c = \bar{\gamma}$  for some  $\beta < \gamma < \alpha$ .

Now assume  $b = \bar{\beta}$ ,  $c = \bar{\gamma}$  and  $\beta < \gamma < \alpha$ . The code of an ordinal is greater than its maximal coefficient, so we have  $\beta \in \alpha[\max(b, c)]$  and  $\gamma \in \alpha[\max(b, c)]$ . Thus the sequence starting with  $\alpha$  and at each stage descending from a  $\delta$  to either  $\delta - 1$  if  $\text{Succ}(\delta)$  or  $\delta(\max(b, c))$  if  $\text{Lim}(\delta)$  necessarily passes through  $\gamma$  and then through  $\beta$ . In turn, it means there are  $i, j < \omega$  such that  $0 < i < j$ ,  $g(i, \max(b, c)) = c$ ,  $g(j, \max(b, c)) = b$ . So  $b \prec_\alpha c$  holds if we can show that  $j \leq G_\alpha(\max(b, c) + 1)$ . In the sequence described above, only the successor stages contribute an element  $\delta - 1$  to  $\alpha[\max(b, c)]$ . At the limit stages  $\delta(\max(b, c))$  does not get put in. Although  $\delta(n)$  does not belong to  $\delta[n]$ , it does belong to  $\delta[n + 1]$ . Therefore all the ordinals in the descending sequence lie in  $\alpha[\max(b, c) + 1]$ , so  $j$  can not be bigger than the cardinality of this set, which is  $G_\alpha(\max(b, c) + 1)$ .

□

The moral is that the principles of transfinite induction and recursion over the initials segments of ordinals below  $\varepsilon_0$  can be expressed by means of  $\mathbf{ID}_0(\text{exp})$ .

### 3.2 Introducing the fast-growing hierarchy

**Definition 3.5.** The *Hardy hierarchy*  $\{H_\alpha\}_{\alpha < \varepsilon_0}$  is defined by recursion on  $\alpha$ :

$$H_\alpha(n) = \begin{cases} n, & \text{if } \alpha = 0 \\ H_{\alpha-1}(n+1), & \text{if } \text{Succ}(\alpha) \\ H_{\alpha(n)}(n), & \text{if } \text{Lim}(\alpha) \end{cases}$$

The *fast-growing hierarchy*  $\{F_\alpha\}_{\alpha < \varepsilon_0}$  is defined by recursion on  $\alpha$ :

$$F_\alpha(n) = \begin{cases} n + 1, & \text{if } \alpha = 0 \\ F_{\alpha-1}^{n+1}(n), & \text{if } \text{Succ}(\alpha) \\ F_{\alpha(n)}(n), & \text{if } \text{Lim}(\alpha) \end{cases}$$

where  $F_{\alpha-1}^{n+1}(n)$  is the  $(n+1)$ -times iteration of  $F_{\alpha-1}$  on  $n$ .

Note that  $H_\alpha$  and  $F_\alpha$  could be equivalently defined by purely number-theoretic means by working over the well-orderings  $\prec_\alpha$  instead of working over ordinals directly. So  $H_\alpha$  and  $F_\alpha$  are  $\varepsilon_0$ -recursive.

**Lemma 3.5.** For all  $\alpha, \beta < \varepsilon_0$  and for all  $n < \omega$ ,

1.  $H_{\alpha+\beta}(n) = H_\alpha(H_\beta(n))$ ,
2.  $H_{\omega^\alpha}(n) = F_\alpha(n)$ .

*Proof.* The first part is proved by induction on  $\beta$ . If  $\beta = 0$ , then the equation trivially holds. Assume  $\text{Succ}(\beta)$  and the induction hypothesis for  $\beta - 1$ , then we have:

$$H_{\alpha+\beta}(n) = H_{\alpha+(\beta-1)}(n+1) = H_{\alpha}(H_{\beta-1}(n+1)) = H_{\alpha}(H_{\beta}(n)).$$

If  $\text{Lim}(\beta)$ , then we have (by using the induction hypothesis for  $\beta(n)$ ):

$$H_{\alpha+\beta}(n) = H_{\alpha+\beta(n)}(n) = H_{\alpha}(H_{\beta(n)}(n)) = H_{\alpha}(H_{\beta}(n)).$$

The second part is proven by induction on  $\alpha$ . If  $\alpha = 0$ , then

$$H_{\omega^0}(n) = H_1(n) = n+1 = F_0(n)$$

If  $\text{Succ}(\alpha)$ , then

$$H_{\omega^\alpha}(n) = H_{\omega^{\alpha-1} \cdot (n+1)}(n) = H_{\omega^{\alpha-1}}^{n+1}(n) = F_{\alpha-1}^{n+1}(n) = F_{\alpha}(n).$$

The limit case is immediate.  $\square$

**Lemma 3.6.** For each  $\alpha < \varepsilon_0$ ,  $H_{\alpha}$  is strictly increasing and  $H_{\beta}(n) < H_{\alpha}(n)$  for  $\beta \in \alpha[n]$ . The same holds for  $F_{\alpha}$  for  $n \neq 0$ , for when  $n = 0$  we have  $F_{\alpha}(0) = 1$  for each  $\alpha$ .

*Proof.* Induction on  $\alpha$ . The case  $\alpha = 0$  is trivial since  $H_0$  is the identity function and  $0[n] = \emptyset$ . If  $\text{Succ}(\alpha)$ , then  $H_{\alpha}$  is  $H_{\alpha-1}$  composed with the successor function, it is strictly increasing by the induction hypothesis. Take  $\beta \in \alpha[n]$ , then either  $\beta \in (\alpha-1)[n]$  or  $\beta = \alpha-1$ , thus, by using the induction hypothesis

$$H_{\beta}(n) \leq H_{\alpha-1}(n) < H_{\alpha-1}(n+1) = H_{\alpha}(n).$$

If  $\text{Lim}(\alpha)$  then

$$H_{\alpha}(n) = H_{\alpha(n)}(n) < H_{\alpha(n)}(n+1)$$

but  $\alpha(n) \in \alpha[n+1] = \alpha(n+1)[n+1]$ , thus

$$H_{\alpha(n)}(n+1) < H_{\alpha(n+1)}(n+1) = H_{\alpha}(n+1)$$

Thus  $H_{\alpha}(n) < H_{\alpha}(n+1)$ . Furthermore if  $b \in \alpha[n]$ , then  $\beta \in \alpha(n)[n]$  so  $H_{\beta}(n) < H_{\alpha(n)}(n) = H_{\alpha}(n)$  by the induction hypothesis for  $\alpha(n)$ .

The same holds for  $F_{\alpha} = H_{\omega^\alpha}$  since if  $\beta \in \alpha[n]$  we then have  $\omega^\beta \in \omega^\alpha[n]$ .  $\square$

**Lemma 3.7.** If  $\beta \in \alpha[n]$ , then  $F_{\beta+1}(m) \leq F_{\alpha}(m)$  for all  $m \geq n$ .

*Proof.* Induction on  $\alpha$ . The zero case is trivial. If  $\text{Succ}(\alpha)$ , then either  $\beta \in (\alpha-1)[n]$  or  $\beta = \alpha-1$ . In either case we apply the induction hypothesis. If  $\alpha$  is a limit, then we have  $\beta \in \alpha(n)[n]$ , so by induction hypothesis  $F_{\beta+1}(m) \leq F_{\alpha(n)}(m)$ , but  $F_{\alpha(n)}(m) \leq F_{\alpha}(m)$ .  $\square$

### 3.3 $\alpha$ -recursion and $\varepsilon_0$ -recursion

**Definition 3.6** ( $\alpha$ -recursion).

1. An  $\alpha$ -recursion is a function definition of the following form, defining  $f : \mathbb{N}^{k+1} \rightarrow \mathbb{N}$  from functions  $g_0, g_1, \dots, g_s$  by the following equations:

$$\begin{aligned} f(0, \vec{m}) &= g_0(\vec{m}) \\ f(n, \vec{m}) &= T(g_1, \dots, g_s, f_{<_n}, n, \vec{m}) \text{ provided } n \geq 1. \end{aligned}$$

where  $T(g_1, \dots, g_s, f_{<_n}, n, \vec{m})$  is a fixed term built up from the number variables  $n$  and  $\vec{m}$  by applying functions  $g_1, \dots, g_s$  and the function  $f_{<_n}$  defined as

$$f_{<_n}(n', \vec{m}) = \begin{cases} f(n', \vec{m}), & \text{if } n' <_\alpha n \\ 0, & \text{otherwise} \end{cases}$$

Note that it is assumed that  $\alpha > 0$ .

2. An *unnested*  $\alpha$  is one of the special form:

$$\begin{aligned} f(0, \vec{m}) &= g_0(\vec{m}) \\ f(n, \vec{m}) &= g_1(n, \vec{m}, f(g_2(n, \vec{m}), \dots, g_{k+1}(n, \vec{m}))) \end{aligned}$$

with a single recursive call of  $f$  where  $g_2(n, \vec{m}) <_\alpha n$  for all  $n$  and  $\vec{m}$ .

3. Let  $\varepsilon_0(0) = \omega$  and  $\varepsilon_0(i+1) = \omega^{\varepsilon_0(i)}$ . For each particular  $i$ , a function is  $\varepsilon_0(i)$ -recursive if it can be defined from primitive recursive functions by successive substitutions and  $\alpha$ -recursions with  $\alpha < \varepsilon_0(i)$ . It is *unnested*  $\varepsilon_0(i)$ -recursive if all the  $\alpha$ -recursions are unnested. It is  $\varepsilon_0$ -recursive if it is  $\varepsilon_0(i)$ -recursive for some (any)  $i$ .

**Lemma 3.8 (Bounds for  $\alpha$ -recursion).** Let  $f$  be a function defined from  $g_1, \dots, g_s$  by an  $\alpha$ -recursion:

$$\begin{aligned} f(0, \vec{m}) &= g_0(\vec{m}) \\ f(n, \vec{m}) &= T(g_1, \dots, g_s, f_{<_n}, n, \vec{m}) \end{aligned}$$

where for each  $i \leq s$   $g_i(\vec{a}) < F_\beta(k + \max \vec{a})$  for all numerical arguments  $\vec{a}$ . Then there is a constant  $d$  such that for all  $n, \vec{m}$

$$f(n, \vec{m}) < F_{\alpha+\beta}(k + 2d + \max(n, \vec{m})).$$

Note that  $\beta$  and  $k$  are arbitrary constants, but it is assumed that the last exponent in the Cantor normal form of  $\beta$  is  $\geq$  the first exponent in the normal form of  $\alpha$ , so that  $\beta + \alpha$  is in Cantor normal form by default.



*Proof.* The constant  $d$  will be actually the depth of nesting of the term  $T$ , where variables have depth 0 and each compositional term  $g(T_1, \dots, T_l)$  has depth greater than the maximum depth of nesting of the subterms  $T_j$ .

Assume  $n$  lies in the field of the well-ordering  $\prec_\alpha$ . Then  $n = \bar{\gamma}$  for some  $\gamma < \alpha$ . Let us claim by induction on  $\gamma$  that

$$f(n, \vec{m}) < F_{\beta+\gamma+1}(k + 2d + \max(n, \vec{m})).$$

This is immediate when  $n = 0$ , because  $g_0(\vec{m}) < F_\beta(k + \max \vec{m})$  and  $F_\beta$  is strictly increasing and bounded by  $F_{\beta+1}$ . Assume  $n \neq 0$  and assume the claim for all  $n' = \bar{\delta}$  where  $\delta < \gamma$ .

Let  $T'$  be any subterm of  $T(g_1, \dots, g_s, f_{\prec_n}, n, \vec{m})$  with depth of nesting  $d'$ , built up by application of one of the functions  $g_1, \dots, g_s$  or  $f_{\prec_n}$  to subterms  $T_1, \dots, T_l$ . Assume (for a sub-induction on  $d'$ ) that each of these  $T_j$ 's has numerical value  $v_j$  less than  $F_{\beta+\gamma}^{2(d'-1)}(k + 2d + \max(n, \vec{m}))$ .

If  $T'$  is obtained by application of one of the functions  $g_i$  then its numerical value will be

$$g_i(v_1, \dots, v_l) < F_\beta(k + 2^{d'-1}_{\beta+\gamma})(k + 2d + \max(n, \vec{m})) < F_{\beta+\gamma}^{2d'}(k + 2d + \max(n, \vec{m}))$$

since  $k < u$  then  $F_\beta(k + u) < F_\beta(2u) < F_\beta^2(u)$  provided  $\beta \neq 0$ . On the other hand, if  $T'$  is obtained by application of the function  $f_{\prec_n}$ , its value will be  $f(v_1, \dots, v_l)$  if  $v_1 \prec_\alpha n$  or 0 otherwise. Suppose  $v_1 = \bar{\delta} \prec_\alpha \bar{\gamma}$ . So by the induction hypothesis:

$$f(v_1, \dots, v_l) < F_{\beta+\delta+1}(k + 2d + \max \vec{v}) \leq F_{\beta+\gamma}(k + 2d + \max \vec{v})$$

because  $v_1$  is greater than the maximum coefficient of  $\delta$ , so  $\delta \in \gamma[v_1]$ , so  $\beta + \delta \in (\beta + \gamma)[v_1]$  and hence  $F_{\beta+\gamma+1}$  is bounded by  $F_{\beta+\gamma}$  on arguments  $\geq v_1$ . Therefore inserting the assumed bounds for the  $v_j$ , we have

$$F(v_1, \dots, v_l) < F_{\beta+\gamma}(k + 2d + F_{\beta+\gamma}^{2(d'-1)}(k + 2d + \max(n, \vec{m})))$$

and thus we have

$$f(v_1, \dots, v_l) < F_{\beta+\gamma}^{2d'}(k + 2d + \max(n, \vec{m})).$$

So we have just shown that the value of every subterms of  $T$  with depth of nesting  $d'$  is less than  $F_{\beta+\gamma}^{2d'}(k + 2d + \max(n, \vec{m}))$ . Applying this to  $T$  itself with depth of nesting  $d$  we obtain

$$f(n, \vec{m}) < F_{\beta+\gamma}^{2d}(k + 2d + \max(n, \vec{m})) < F_{\beta+\gamma+1}(k + 2d + \max(n, \vec{m}))$$

So we have proved the claim.

Now we derive the result of the lemma. Assume  $n = \bar{\gamma}$  lies in the field of  $\prec_\alpha$ , then  $\beta + \gamma \in (\beta + \alpha)[n]$  and thus

$$f(n, \vec{m}) < F_{\beta+\gamma+1}(k + 2d + \max(n, \vec{m})) \leq F_{\beta+\alpha}(k + 2d + \max(n, \vec{m})).$$

If  $n$  does not lie in the field of  $\prec_\alpha$ , then  $f_{\prec_n}$  is the constant zero function, and thus in evaluating  $f(n, \vec{m})$  by the term  $T$  only applications of the  $g_i$ -functions are required. Thus we have

$$f(n, \vec{m}) < F_{\beta}^{2d}(k + 2d + \max(n, \vec{m})) < F_{\beta+\alpha}(k + 2d + \max(n, \vec{m})).$$

since  $\alpha$  is non-zero.  $\square$

**Theorem 3.1.** For each  $i$ , a function is  $\varepsilon_0(i)$ -recursive if and only if it is a register-machine computable in a number of steps bounded by  $F_{\alpha}$  for some  $\alpha < \varepsilon_0(i)$ .

*Proof.* 1. The "if" part.

If a function  $g$  is register-machine computable, then there is an elementary function  $U$  such that for all arguments  $\vec{m}$ , if  $s(\vec{m})$  bounds the number of steps required to compute  $g(\vec{m})$ , then  $g(\vec{m}) = U(\vec{m}, s(\vec{m}))$ . So if  $g$  is computable in a number of steps bounded by  $F_{\alpha}$ , then  $g$  can be defined from  $F_{\alpha}$  by the following substitution

$$g(\vec{m}) = U(\vec{m}, F_{\alpha}(\max \vec{m})).$$

So if  $F_{\alpha}$  is  $\varepsilon_0(i)$ -recursive, so is  $g$ . Let us show that if  $\alpha < \varepsilon_0(i)$  then  $F_{\alpha}$  is  $\varepsilon_0(i)$ -recursive.

The claim holds for  $i = 0$  since then all  $\alpha$ 's are finite, but the finite levels of  $F$  hierarchy are primitive recursive and thus  $\varepsilon_0(0)$ -recursive. Since  $i > 0$  and  $\alpha = \omega^{\gamma_1} \cdot c_1 + \dots + \omega^{\gamma_k} \cdot c_k < \varepsilon_0(i)$ .

Let us add one to each exponent and insert a successor term at the end, so we produce the ordinal  $\beta = \alpha' + n$ , where  $\alpha'$  is the limit  $\omega^{\gamma_1+1} \cdot c_1 + \dots + \omega^{\gamma_k+1} \cdot c_k$ .  $i > 0$ , so we have  $\beta < \varepsilon_0(i)$ . From the code of  $\alpha$ , denoted as  $a$ , we can compute the code for  $\alpha$ , denoted as  $a'$ . So  $b = \pi(a', \langle 0, n \rangle)$  is the code for  $\beta$ . And conversely, we are able to decode  $\alpha$ ,  $\alpha'$  and  $n$  from  $\beta$ .

Let us choose a large enough  $\delta < \varepsilon_0(i)$  such that  $\beta < \delta$ , let us define  $f(b, m)$  by  $\delta$ -recursion such that if  $b$  is the code for  $\beta = \alpha' + n$ , then  $f(b, m) = F_{\alpha}^n(m)$ . Let us expose the components from which  $b$  is constructed as  $b = (a, n)$ , so we can define  $f(a, n, m)$  using the elementary function  $h(a, n)$  that returns the code for  $\alpha - 1$  for  $\text{Succ}(\alpha)$  or  $\alpha(n)$  for  $\text{Lim}(\alpha)$ :

$$f(a, n, m) = \begin{cases} m + n, a = 0 \text{ or } n = 0 \\ f(h(a, m), m + 1, m), \text{ if } \text{Succ}(a) \text{ and } n = 1 \\ f(h(a, m), 1, m), \text{ if } \text{Lim}(a) \text{ and } n = 1 \\ f(a, 1, f(a, n - 1, m)), \text{ if } n > 1 \\ 0, \text{ otherwise} \end{cases}$$

Then  $f$  is  $\varepsilon_0(i)$ -recursive and  $F_{\alpha}(m) = f(\bar{\alpha}, 1, m)$ , so  $F_{\alpha}$  is  $\varepsilon_0(i)$ -recursive for every  $\alpha < \varepsilon_0(i)$ .

2. The "only if" part.

Note that the number of steps needed to compute a compositional term  $g(T_1, \dots, T_l)$  is the sum of the numbers of steps needed to compute sub-terms  $T_1, \dots, T_l$  plus the number of steps required to compute  $g(v_1, \dots, v_l)$  where  $v_j$  is the value of  $T_j$ .

Furthermore, the values  $v_j$  are bounded by the number of computation steps plus the maximal input. So we can compute a bound on the computation steps for any such term. Moreover, we can do that elementarily from given bounds for the input data. Now suppose

$$f(n, \vec{m}) = T(g_1, \dots, g_s, f_{\prec_n}, n, \vec{m})$$

is any recursion-step of an  $\alpha$ -recursion. So if we have bounding functions on the numbers of steps to compute each of the  $g_i$ 's and we assume inductively that we already have a bound on the number of steps to compute  $f(n', -)$  for  $n' \prec_\alpha n$ . So we can elementarily estimate a bound on the steps to compute  $f(n, \vec{m})$ . So for any function defined by an  $\alpha$ -recursion from functions  $\vec{g}$ , a bounding function is also definable by  $\alpha$ -recursion by bounding functions for  $\vec{g}$ . We have the same for primitive recursion. All in all, every  $\varepsilon_0(i)$ -function is register-machine computable in a number of steps bounded by some  $F_\gamma$  for  $\gamma < \varepsilon_0(i)$ . □

**Corollary 3.2.** For each  $i$ , a function is  $\varepsilon_0(i)$ -recursive if and only if it is unnested  $\varepsilon_0(i+1)$ -recursive.

*Proof.* Every  $\varepsilon_0(i)$ -recursive function is computable in the number of steps bounded by  $F_\alpha = H_{\omega^\alpha}$  where  $\alpha < \varepsilon_0(i)$ . Thus it is primitive recursively definable from  $H_{\omega^\alpha}$ . But  $H_{\omega^\alpha}$  itself is defined an unnested  $\omega^\alpha$ -recursion and  $\omega^\alpha < \varepsilon_0(i+1)$ . So arbitrarily nested  $\varepsilon_0(i)$ -recursions are reducible to unnested  $\varepsilon_0(i+1)$ -recursions.

Conversely, assume  $f$  is defined from functions  $g_0, g_1, \dots, g_{k+2}$  by an unnested  $\alpha$ -recursion where  $\alpha < \varepsilon_0(i+1)$ :

$$\begin{aligned} f(0, \vec{m}) &= g_0(\vec{m}) \\ f(n, \vec{m}) &= g_1(n, \vec{m}, f(g_2(n, \vec{m}), \dots, g_{k+2}(n, \vec{m}))) \end{aligned}$$

with  $g_2(n, \vec{m}) \prec_\alpha n$  for all  $n$  and  $\vec{m}$ . Then the number of recursion-steps needed to compute  $f(n, \vec{m})$  is  $f'(n, \vec{m})$  where

$$\begin{aligned} f'(0, \vec{m}) &= 0 \\ f'(n, \vec{m}) &= 1 + f'(g_2(n, \vec{m}), \dots, g_{k+2}(n, \vec{m})) \end{aligned}$$

and  $f$  is thus definable from  $g_2, \dots, g_{k+2}$  by primitive recursion and bound for  $f'$ . Assume that the given functions  $g_j$  are all primitive recursively definable from, and bounded by,  $H_\beta$  where  $\beta < \varepsilon_0(i+1)$ . Now let us provide bounds for  $\alpha$ -recursion and show that  $f'(n, \vec{m})$  is bounded by  $H_{\beta \cdot \gamma}$  where  $n = \bar{\gamma}$  since

$$H_{\beta \cdot (\gamma+1)}(x) = H_{\beta \cdot \gamma + \beta}(x) = H_{\beta \cdot \gamma}(H_\beta(x)).$$

Thus  $f$  is definable from  $H_\beta$  and  $H_{\beta.\alpha}$ . Clearly since  $\beta, \alpha < \varepsilon_0(i+1)$  we can choose  $\beta = \omega^{\beta'}$  and  $\alpha = \omega^{\alpha'}$  for  $\alpha' \leq \beta' < \varepsilon_0(i)$ . Thus  $H_\beta = H_{\beta'}$  and  $H_{\beta.\alpha} = F_{\beta'+\alpha'}$  where  $\beta' + \alpha' < \varepsilon_0(i)$ . Therefore  $f$  is  $\varepsilon_0(i)$ -recursive.  $\square$

### 3.4 Provable recursiveness of $F_\alpha$ and $H_\alpha$

4 **RCA<sub>0</sub>**

5 **WKL<sub>0</sub>**

6 **ACA<sub>0</sub>**

7 **ATR**

8  **$\Pi_1^1$ -comprehension**

9 **Kripke-Platek Set Theory**