Some Notes on Proof Theory and Elements of Ordinal Analysis

Daniel Rogozin

Contents

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

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

$$\{=, 0, S, P, +, \dot{-}, \cdot, exp_2\}$$

where S and P are successor and precessor functions respectively. Further, we will denote $S(x)$ and $P(x)$ as $x+1$ and $x\dot{-}1$ respectively. 2^x stands for $exp_2(x)$.

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

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

along with the bounded induction scheme:

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

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

- $B \coloneqq \forall x < tP(x) \equiv \forall x(x < t \to P(x))$
- $B \coloneqq \exists x < tP(x) \equiv \exists x(x < t \wedge P(x))$

A Σ_1-formula is a formula of the form:

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

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

Lemma 1.1. $\mathbf{I}\Delta_0(\exp)$ 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. $\mathbf{I}\Delta_0(\exp)$ 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. $x < y \to x \cdot (z+1) < y \cdot (z+1)$

13. $x < 2^x$

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

Proof. Straightforward induction. $\square$

Definition 1.1. A function $f : \mathbb{N}^k \to \mathbb{N}$ is *provably* Σ_1 or *provably recursive* in an arithmetical theory if there is a Σ_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. $T \vdash F(\vec{x}, y) \wedge F(\vec{x}, y') \to y = y'$

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 $T \vdash F(\vec{x}, y) \to y < t(\vec{x})$.

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 φ be a statement not involving f such that φ is true in $(\mathcal{M}, f)$, so φ is true in $\mathcal{M}$ as well. By compactness T proves φ. $\square$

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

Proof. Let f be a function defined by some $\mathbf{I}\Delta_0(\exp)$-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 $t(x_1, \ldots, x_n)$ built up from the constants $0, S, P, +, \dot{-}, \cdot, exp_2$ there exists $k < \omega$ such that:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

so we have (by first order logic):

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

Thus f is provably Σ_1 in $\mathbf{I}\Delta_0(\exp)$, so the rest is to find its bounding term. $\mathbf{I}\Delta_0(\exp)$ proves the following:

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

and

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

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

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

and so

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

$\square$

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 $\mathbf{I}\Delta_0(\exp)$. Then f is provably bounded in $\mathbf{I}\Delta_0(\exp)$.

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

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

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

We can prove:

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

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

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

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

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

Now we have to check that $\mathbf{I}\Delta_0(\exp) \vdash \exists y F(\vec{x}, z, y)$. 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 $\mathbf{I}\Delta_0(\exp) \vdash F(\vec{x}, 0, y)$ 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 $\mathbf{I}\Delta_0(\exp)$.

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, $\cdot$, $+$, exp_2, $\dot{-}$ 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 $\mathbf{I}\Delta_0(\exp)$. We have the following logical rules:

$$\frac{}{\Gamma, R\vec{t}, \neg R\vec{t}} \,\mathbf{Ax}$$

$$\frac{\Gamma, A_0, A_1}{\Gamma, A_0 \vee A_1} \vee \qquad\qquad \frac{\Gamma, A_0 \qquad \Gamma, A_1}{\Gamma, A_0 \wedge A_1} \wedge$$

$$\frac{\Gamma, A(t)}{\Gamma, \exists x A(x)} \exists \qquad\qquad \frac{\Gamma, A}{\Gamma, \forall x A} \forall$$

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

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

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

Of course, the cut rule is admissible:

$$\frac{\Gamma, A \qquad \Gamma, \neg A}{\Gamma}\ \mathbf{cut}$$

Definition 1.2. Let $\exists \vec{z} B(\vec{z})$ be a closed Σ_1-formula, then it is *true at* m, written as $m \models \exists \vec{z} B(\vec{z})$, if there exist natural numbers $m_1, \ldots, m_l$ such that each $m_i < m$ and $B(\vec{m})$ is true in the standard model.

A finite set Γ of closed Σ_1-formulas is true at m, written as $m \models \Gamma$ if at least one of them is true at m.

If $\Gamma(x_1, \ldots, x_k)$ is a finite set of Σ_1-formulas whose free variables occur amongst $x_1, \ldots, x_k$. Let $f : \mathbb{N}^k \to \mathbb{N}$, then $f \models \Gamma(x_1, \ldots, x_k)$ we have $f(\vec{n}) \models \Gamma(x_1 := n_1, \ldots, x_k := n_k)$ for each $\vec{n} = (n_1, \ldots, n_k)$.

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 $f(\vec{n}) \models \Gamma(x_1 := n_1, \ldots, x_k := n_k)$ implies $f'(\vec{n}) \models \Gamma(x_1 := n_1, \ldots, x_k := n_k)$.

Lemma 1.7. Let $\Gamma(\vec{x})$ be a finite set of Σ_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 Γ 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 Σ_1.

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

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

We can similarly cancel each of $\neg A_2, \ldots, \neg A_s$ and so obtain the proof of Γ with Σ_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 $f(\vec{n}) = n_1 + \cdots + n_k$.

2. If $\Gamma, B_0 \vee B_1$ is derivable, so is Γ, 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 Γ, B_0 and Γ, 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 Γ. Since all the formulas are Σ_1, $\forall x B(y)$ must be bounded, so $B(y) \coloneqq \neg(y < t) \vee B'(y)$ 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 Γ comes about by the cut rule with Σ_1 formula $C \coloneqq \exists\vec{z} B(\vec{z})$, 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:

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

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

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

   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 $B(\vec{n}, k) \to B(\vec{n}, k+1)$ 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 Σ_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 Σ_1 in $\mathbf{I}\Delta_0(\exp)$. Then we have a formula

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

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 $m_0, \ldots, m_k$ less that $g(n)$ satisfying the bounded formula $B(\vec{n}, m_0, m_1, \ldots, m_k)$. Apply the elementary sequence coding:

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

so that if $m = \langle m_0, m_1, \ldots, m_k\rangle$ 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 $m_1, \ldots, m_k$ satisfying $B(\vec{n}, m_0, \ldots, m_k)$, we define f as:

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

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 restructed to Σ_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 Σ_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. $\mathbf{I}\Sigma_1 \vdash F(\vec{x}, y) \wedge F(\vec{x}, y') \to y = y'$.

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 $f(x_0, \ldots, x_n) = x_i$ 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 thatb 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 ($\mathrm{len}(z) = 4$). 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:

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

By the induction hypothesis f_0 and f_1 are provably recursive and they have associated Δ_0-formulas C_0 and C_1. Define

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

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 $k = \langle k_0, \ldots, k_n, m\rangle$ 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 $\mathbf{I}\Delta_0$

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

and

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

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

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

We have assumed that f_0 and f_1 are primitive recursive, we can prove $\exists y F(\vec{v}, 0, y)$ and $\exists y F(\vec{v}, x, y) \to \exists y F(\vec{v}, x+1, y)$. Then we derive $\exists y F(\vec{v}, x, y)$ by using Σ_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 uniquness 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 $F(\vec{v}, x, y) \wedge F(\vec{v}, x, y') \to y = y'$.

$\square$

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

Definition 2.1. A closed Σ_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 $\vec{m} = (m_1, \ldots, m_l)$ such that all $m_i < m$ for $i \in \{1, \ldots, l\}$ such that $B(\vec{m})$ is true in the standard model.

A finite set of formulas Γ of closed Σ_1-formulas is "true at m" (denoted as $m \models \Gamma$) if at least one of them is true at m.

If $\Gamma(x_1, \ldots, x_k)$ is a finite set of Σ_1-formulas all of whose free variables occur amongst $x_1, \ldots, x_k$ and if $f : \mathbb{N}^k \to \mathbb{N}$, then we write $f \models \Gamma$ if for each assignments $\vec{n} = (n_1, \ldots, n_k)$ to the variables $x_1, \ldots, x_k$ 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

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

where y is not free in Γ, t is any term and A is any Σ_1-formula.

Lemma 2.2. (Σ_1-induction) Let $\Gamma(\vec{x})$ be a finite set of Σ_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 Γ is provable in this formalisation, then it has a proof in which all the non-atomic cut formulas are induction Σ_1-formulas. If Γ is classically derivable from non-logical axioms $A_1, \ldots, A_s$, then there is a cut-free proof (à la Tait) of $\neg A_1, \Delta, \Gamma$ where $\Delta = A_2, \ldots, A_s$. 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 Δ, Γ by cutting $F(t)$. One can detach $\neg A_2, \ldots, \neg A_s$, so we finally have a proof of Γ which uses cuts only on Σ_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, the for all $\vec{n}$ $\Gamma(\vec{n})$ contains a true atom. Choose $f(\vec{n}) = n_1 + \cdots + n_k$, and f is clearly primitive recursive, strictly incrasing 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

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

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 Γ. As far as all formulas are Σ_1, $\forall y B(y)$ must be $\mathbf{I}\Delta_0(\exp)$, so $B(y) \fallingdotseq \neg(y < t) \vee B'(y)$ for some elemetary 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 Σ_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

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

where $\exists\vec{z}B(\vec{z})$ is a cut Σ_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 $\Delta_0(\exp)$-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:

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

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

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

where $A(\vec{x}, y)$ is an induction Σ_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 Σ_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 $\vec{m}$

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

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

$$\begin{gathered} f(\vec{n}, 0) = f_0(\vec{n}) \\ f(\vec{n}, k+1) = f_1(\vec{n}, k, f(\vec{n}, k), \ldots, f(\vec{n}, k)) \end{gathered}$$

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 $g : \mathbb{N}^k \to \mathbb{N}$ is Σ_1 be a function defined by a Σ_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 ε_0-recursion in Peano Arithmetic

We show that the provably recursive functions of Peano arithmetic are ε_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, ε_0 can be defined as the least fixed point of the mapping $\alpha \mapsto \omega^\alpha$ where α is an ordinal.

Let us discuss first how one can represent ordinals below ε_0.

3.1 Ordinals below ε_0

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

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

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

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

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

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

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

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

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

Clearly

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

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

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

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 γ.

1. $\gamma = 0$, then $\gamma[n] = \emptyset$ and the right hand side is $\delta[n] \cap \{\delta\}$, which is the same as $\alpha[n] = (\delta + 1)[n]$.

2. If γ 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 γ is a successor. Then α 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

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

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

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

   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 $\beta = \omega^{\beta_1} \cdot b_1 + \cdots + \omega^{\beta_l} \cdot b_l$ is defined by induction to be the maximum of all the b_i's and all the maximum coefficients of the exponents β_i's.

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

Proof. By induction on α. 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

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

for $\alpha > \gamma > \beta_1 > \cdots > \beta_k$ 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

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

The hierarchy of functions G_α 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 occurence of ω in the Cantor normal form ω.

Proof. Induction on γ.

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

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

2. If γ 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 γ 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 $\beta = \omega^{\beta_1} \cdot b_1 + \ldots \omega^{\beta_l} \cdot b_l$ be an ordinal. A *coding ordinal* is the sequence number $\overline{\beta}$ constructed recursively the following way

$$\overline{\beta} = \langle \langle \overline{\beta_1}, b_1 \rangle, \ldots, \langle \overline{\beta_l}, b_l \rangle \rangle.$$

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

Lemma 3.4.

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

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

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

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

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 π_2 is a pair (m', n'). If so and $m' = 0$ and $n' \neq 0$, then β 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 β is a limit ordinal of then form $\delta + \omega^\gamma \cdot n'$ where $m' = \overline{\gamma}$. 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 β we have a "spare" ω^γ which must be either reduced to $\omega^{\gamma-1} \cdot (n+1)$ when $\mathrm{Succ}(\gamma)$ or to $\omega^{\gamma(n)}$ if $\mathrm{Lim}(\gamma)$. 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 additinal 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 $\mathrm{Succ}(\gamma)$ or $\langle h(m', n), 1 \rangle$ if $\mathrm{Lim}(\gamma)$.

This defines $h(m,n)$, but such a definition is actually primitive recursive so far. Let us chech 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 $\mathrm{Succ}(\gamma)$, 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 α 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 $g(0,n) = \overline{\alpha}$ 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_α 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 $b = \overline{\beta}$ and $c = \overline{\gamma}$ for some $\beta < \gamma < \alpha$.

Now assume $b = \overline{\beta}$, $c = \overline{\gamma}$ 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 α and at each stage descending from a δ to either $\delta - 1$ if $\mathrm{Succ}(\delta)$ or $\delta(\max(b,c))$ if $\mathrm{Lim}(\delta)$ necessarily passes through γ and then through β. 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 it. 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 ε_0 can be expressed by means of $\mathbf{I}\Delta_0(\exp)$.

3.2 Introducing the fast-growing hierarchy

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

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

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

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

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

Note that H_α and F_α 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_α and F_α are ε_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 β. If $\beta = 0$, then the equation trivially holds. Assume $\mathrm{Succ}(\beta)$ 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 $\mathrm{Lim}(\beta)$, 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 α. If $\alpha = 0$, then

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

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

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

The limit case is immediate. □

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

Proof. Induction on α. The case $\alpha = 0$ is trivial since H_0 is the identity function and $0[n] = \emptyset$. If $\mathrm{Succ}(\alpha)$, then H_α 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 $\mathrm{Lim}(\alpha)$ 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 $\omega^\beta \in \omega^\alpha[n]$. □

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 α. The zero case is trivial. If $\mathrm{Succ}(\alpha)$, then either $\beta \in (\alpha - 1)[n]$ or $\beta = \alpha - 1$. In either case we apply the induction hypothesis. If α 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 $F_{\alpha(n)}(m) \leq F_\alpha(m)$. □

3.3 α-recursion and ε_0-recursion

Definition 3.6 (**α-recursion).**

1. An *α-recursion* if a function definition of the following form, defining $f : \mathbb{N}^{k+1} \to \mathbb{N}$ from functions $g_0, g_1, \ldots, g_s$ by the following equations:

$$f(0, \vec{m}) = g_0(\vec{m})$$
$$f(n, \vec{m}) = T(g_1, \ldots, g_s, f_{\prec n}, n, \vec{m}) \text{ provided } n \geq 1.$$

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

$$f_{\prec n}(n', \vec{m}) = \begin{cases} f(n', \vec{m}), \text{if } n' \prec_\alpha n \\ 0, \text{otherwise} \end{cases}$$

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

2. An *unnested* α is one of the special form:

$$f(0, \vec{m}) = g_0(\vec{m})$$
$$f(n, \vec{m}) = g_1(n, \vec{m}, f(g_2(n, \vec{m}), \ldots, g_{k+1}(n, \vec{m})))$$

   with a single recursive call of f where $g_2(n, \vec{m}) \prec_\alpha n$ 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 α-recursions with $\alpha < \varepsilon_0(i)$. It is *unnested* $\varepsilon_0(i)$*-recursive* if all the α-recursions are unnested. It is ε_0*-recursive* if it is $\varepsilon_0(i)$-recursive for some (any) i.

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

$$f(0, \vec{m}) = g_0(\vec{m})$$
$$f(n, \vec{m}) = T(g_1, \ldots, g_s, f_{\prec n}, n, \vec{m})$$

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 β and k are arbitrary constants, but it is assumed that the last exponent in the Cantor normal form of β is $\geq$ the first exponent in the normal form of α, 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 have depth 0 and each compositional term $g(T_1, \ldots, T_l)$ 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 $n = \overline{\gamma}$ for some $\gamma < \alpha$. Let us claim by induction on γ 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_β is strictly increasing and bounded by $F_{\beta+1}$. Assume $n \neq 0$ and assume the claim for all $n' = \overline{\delta}$ where $\delta < \gamma$.

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

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

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

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

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

because v_1 is greater than the maximum coefficient of δ, 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 insertingn the assumed bounds for the v_j, we have

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

and thus we have

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

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

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

So we have proved the claim.

Now we derive the result of the lemma. Assume $n = \overline{\gamma}$ 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 α is non-zero. □

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_α for some $\alpha < \epsilon_0(i)$.

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_α, then g can be defined from F_α by the following substitution

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

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

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

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

Let us choose a large enough $\delta < \varepsilon_0(i)$ such that $\beta < \delta$, let us define $f(b,m)$ by δ-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 $\mathrm{Succ}(\alpha)$ or $\alpha(n)$ for $\mathrm{Lim}(\alpha)$:

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

Then f is $\varepsilon_0(i)$-recursive and $F_\alpha(m) = f(\overline{\alpha}, 1, m)$, so F_α 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 $g(T_1, \ldots, T_l)$ is the sum of the numbers of steps needed to compute subterms $T_1, \ldots, T_l$ plus the number of steps required to compute $g(v_1, \ldots, v_l)$ 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

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

is any recursion-step of an α-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 α-recursion from functions $\vec{g}$, a bounding function is also definable by α-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_γ 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 recusrively definable from H_{ω^α}. But H_{ω^α} itself is defined an unnested ω^α-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 $g_0, g_1, \ldots, g_{k+2}$ by an unnested α-recursion where $\alpha < \varepsilon_0(i+1)$:

$$f(0, \vec{m}) = g_0(\vec{m})$$
$$f(n, \vec{m}) = g_1(n, \vec{m}, f(g_2(n, \vec{m}), \ldots, g_{k+2}(n, \vec{m})))$$

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

$$f'(0, \vec{m}) = 0$$
$$f'(n, \vec{m}) = 1 + f'(g_2(n, \vec{m}), \ldots, g_{k+2}(n, \vec{m}))$$

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

3.4 Provable recursiveness of F_α and H_α

In this subsection we will show that for every $\alpha < \varepsilon_0(i)$ for $i < \omega$, the function F_α is provably recursive in the theory $\mathbf{I}\Sigma_{i+1}$.

The required machinery for coding ordinals below ε_0 is elementary, so one can assume that it can be defined in $\mathbf{I}\Delta_0(\exp)$. We will make use of the function h such that if a codes a successor ordinal α, then $h(a, n)$ codes $\alpha - 1$ and a codes a limit ordinal α, then $h(a, n)$ codes $\alpha(n)$. One can decide whether a codes a successor ordinal ($\mathrm{Succ}(\alpha)$) or a limit ordinal ($\mathrm{Lim}(\alpha)$) by asking whether $h(a, 0) = h(a, 1)$ or not. It is a bit easier to show the provable recursiveness of the Hardy functions H_α first of all since the Hardy functions are defined involving no nested recursion. After that one can conclude the provable recursiveness of the fast-growing hierarchy by using the equation $F_\alpha = H_{\omega^\alpha}$.

Definition 3.7. Let $H(a, x, y, z)$ be a $\Delta_0(exp)$-formula of the following form:

$$\begin{aligned}
&(z)_0 = \langle 0, y\rangle \wedge \pi_2(z) = \langle a, x\rangle \wedge \\
&\forall i < \mathrm{lh}(z)\ (\mathrm{lh}((z)_i) = 2 \wedge (i < 0 \to (z)_{i,0} > 0)) \wedge \\
&\forall 0 < i < \mathrm{lh}(z)\ (\mathrm{Succ}((z)_{i,0}) \to (z)_{i-1,0} = h((z)_{i,0}, (z)_{i,1}) \\
&\qquad \wedge (z)_{i-1,1} = (z)_{i,1} + 1) \wedge \\
&\forall 0 < i < \mathrm{lh}(z)\ (\mathrm{Lim}((z)_{i,0}) \to (z)_{i-1,0} = h((z)_{i,0}, (z)_{i,1}) \wedge (z)_{i-1,i} = (z)_{i,1})
\end{aligned}$$

Lemma 3.9 (Definability of H_α)**.** $H_\alpha(n) = m$ iff $\exists z H(\overline{\alpha}, n, m, z)$ is true. For each $\alpha < \varepsilon_0$ one show

$$\mathbf{I}\Sigma_1 \vdash \exists z H(\overline{\alpha}, x, y, z) \wedge \exists z H(\overline{\alpha}, x, y', z) \to y = y'.$$

Proof. The meaning of the formula $\exists z H(\overline{\alpha}, n, m, z)$ is that there is a finite sequence of pairs $\langle \alpha_i, n_i\rangle$, beginning with $\langle 0, m\rangle$ and ending with $\langle \alpha, n\rangle$ such that at each $i > 0$ if $\mathrm{Succ}(\alpha_i)$ then $\alpha_{i-1} = \alpha_i - 1$ and $ni - 1 = n_i + 1$ and if $\mathrm{Lim}(\alpha_i)$ then $\alpha_{i-1} = \alpha_i(n_i)$ and $n_{i-1} = n_i$.

Thus by induction up along the sequence and by using the original definition of H_α one can easily see that for each $i > 0$ $H_{\alpha_i}(n_i) = m$ and thus $H_\alpha(n) = m$. But if $H_\alpha(n) = m$, then there exists a required computation sequence, so the first part of the lemma is shown.

As regards the second part, notice that one can show the following by induction for each n, m, m', s, s'

$$H(\overline{\alpha}, n, m, s) \to H(\overline{\alpha}, n, m', s') \to s = s' \wedge m = m'$$

This proof can be formalised in $\mathbf{I}\Delta_0(\exp)$ to give

$$H(\overline{\alpha}, x, y, z) \to H(\overline{\alpha}, x, y', z') \to z = z' \wedge y = y'$$

and hence $\exists z H(\overline{\alpha}, x, y, z) \to \exists z H(\overline{\alpha}, x, y', z') \to z = z' \wedge y = y'$ □

Lemma 3.10. $\mathbf{I}\Delta_0(\exp)$ proves the following formula

$$\exists z H(\omega^a, x, y, z) \to \exists z H(\omega^a c, y, w, z) \to \exists z H(\omega^a(c+1), x, w, z)$$

where $\omega^a c$ is the elementary term $\langle\langle a, c\rangle\rangle$ which constructs, from the code of α the code for $\omega^\alpha \cdot c$.

Proof. Assume we have sequences s and s' satisfying $H(\omega^a, x, y, s)$ and $H(\omega^a c, x, y, s)$. Add $\omega^a c$ to the first component of each pair in s. Then the last pair in s' and the last pair in s are identical. We concatenate s and s' by taking the repeating pair only once and construct an elementary term $t(s, s')$ satisfying $H(\omega^a(c+1), x, w, t)$. Then one can show

$$H(\omega^a, x, y, s) \to H(\omega^a c, y, w, s') \to H(\omega^a(c+1), x, w, t)$$

in a conservative extension of $\mathbf{I}\Delta_0(\exp)$ and thus derive the following in $\mathbf{I}\Delta_0(\exp)$

$$\exists z H(\omega^a, x, y, z) \to \exists z H(\omega^a c, y, w, z) \to \exists z H(\omega^a(c+1), x, w, z).$$

$\square$

Lemma 3.11. Let $H(a)$ be the Π_2-formula $\forall x \exists y \exists z H(a, x, y, z)$, then one can show the following by Π_2-induction:

1. $H(\omega^0)$.
2. $\mathrm{Succ}(a) \to H(\omega^{h(a,0)}) \to H(\omega^a)$.
3. $\mathrm{Lim}(a) \to \forall x H(\omega^{h(a,x)}) \to H(\omega^a)$.

Proof. The term $t_0 = \langle\langle 0, x+1\rangle, \langle 1, x\rangle\rangle$ witnesses $H(\omega^0, x, x+1, t_0)$ in $\mathbf{I}\Delta_0(\exp)$, so we have $H(\omega^0)$.

Further one can derive the following

$$H(\omega^{h(a,0)}) \to H(\omega^{h(a,0)c}) \to H(\omega^{h(a,0)}(c+1)).$$

So we obtain by Π_2-induction

$$H(\omega^{h(a,0)}) \to H(\omega^{h(a,0)}(x+1))$$

and

$$H(\omega^{h(a,0)}) \to \exists y \exists z H(\omega^{h(a,0)}(x+1), x, y, z).$$

But there is an elementary term t_1 with the property

$$\mathrm{Succ}(a) \to H(\omega^{h(a,0)}(x+1), x, y, z) \to H(\omega^a, x, y, t_1)$$

as far as t_1 needs to tag on to the end of the sequence z the new pair $\langle \omega^a, x\rangle$ and thus $t_1 = \pi(z, \langle \omega^a, x\rangle)$. Thus

$$\mathrm{Succ}(a) \to H(\omega^{h(a,0)}) \to H(\omega^a).$$

The final case is straightforward, we have

$$\mathrm{Lim}(a) \to H(\omega^{h(a,x)}, x, y, z) \to H(\omega^a, x, y, t_1)$$

and so by the Bernays rules we have

$$\mathrm{Lim}(a) \to \forall x H(\omega^{a,x}) \to H(\omega^a).$$

□

Definition 3.8 (Structural transifinite induction)**.** The *structural progressiveness* of a formula $A(a)$ is expressed by $\mathrm{SProg}_a\, A$, which is the conjunction of the formulas $A(0)$, $\forall a(\mathrm{Succ}(a) \to A(h(a,0)) \to A(a))$ and $\forall a(\mathrm{Lim}(a) \to \forall x A(h(a,x)) \to A(a))$.

The principle of *structural transfinite induction* up to an ordinal α is the following axiom schema, for all formulas A:

$$\mathrm{SProg}_a\, A \to \forall a \prec \overline{\alpha} A(a)$$

where $a \prec \overline{\alpha}$ means a lies in the field of the well-ordering $\prec_\alpha$, i.e. $a = 0 \vee 0 \prec_\alpha a$.

In particular, the previous lemma shows that the Π_2-formula $H(\omega^a)$ is structurally progressive and one can show that with Π_2-induction.

Definition 3.9 (Transfinite Induction)**.** The (general) *progressiveness* of a formula $A(a)$ is

$$\mathrm{Prog}_a\, A := \forall a(\forall b \prec a A(b) \to A(a))$$

The principle of a *transfinite induction* up to an ordinal α is the schema

$$\mathrm{Prog}_a\, A \to \forall a \prec \overline{\alpha} A(a)$$

where $a \prec \overline{\alpha}$ means that a lies in the field of the well-ordering $\prec_\alpha$.

Lemma 3.12. Structural transfinite induction up to α is derivable from transfinite induction up to α.

Proof. Let A be an arbitrary formula and assume $\mathrm{SProg}_a\, A$. Let us show $\forall a \prec \overline{\alpha} A(a)$. Let us transfinite induction for the formula $a \prec \overline{\alpha} \to A(a)$, then it is sufficient to prove the following

$$\forall a(\forall b \prec a, \overline{\alpha} A(b) \to a \prec \overline{\alpha} \to A(a))$$

which is equivalent to

$$\forall a \prec \overline{\alpha}(\forall b \prec a A(b) \to A(a)).$$

The latter is proved from $\mathrm{SProg}_a\, A$ and the properties of the h function. □

We also have induction over an arbitrary well-ordered set as a consequence. Comparisons are made using a "measure function" μ into an initial segment of the ordinals. The principle of "general induction" up to α is

$$\mathrm{Prog}^\mu_x\, A(x) \to \forall x(\mu(x) \prec \overline{\alpha} \to A(x))$$

where $\mathrm{Prog}^\mu_x\, A(x)$ expresses "μ-progressiveness" with respect to the measure function μ and the ordering $\prec_\alpha$

$$\mathrm{Prog}^{\mu}_x A(x) := \forall a(\forall y(\mu(y) \prec a \to A(y) \to \forall x(\mu(x) = a \to A(x))))$$

We claim that general induction up to an ordinal α is provable from transfinite induction up to α. Indeed, assume $\mathrm{Prog}^{\mu}_x A(x)$. Let us show $\forall x(\mu(x) \prec \overline{\alpha} \to A(x))$. Consider $B(a) := \forall x(\mu(x) = a \to A(x))$. It it sufficient to prove $\forall a \prec \overline{\alpha} B(a)$, which is $\forall a \prec \overline{\alpha} \forall x(\mu(x) = a \to A(x))$. By transfinite induction it is sufficient to prove $\mathrm{Prog}_a B$, which is, in turn,

$$\forall a(\forall b \prec a \forall y(\mu(y) = b \to A(y) \to \forall x(\mu(x) = a \to A(x))))$$

But that follows from the assumption $\mathrm{Prog}^{\mu}_x A(x)$.

3.5 Gentzen's theorem on transfinite induction in PA

We make use of Gentzen's result on the provability of transfinite induction up ε_0 to complete provable recursiveness of H_α and F_α. We will need some properties of $\prec$ and $\oplus$, the elementary function on ordinal codes such that $\overline{\alpha} \oplus \overline{\beta} = \overline{\alpha + \beta}$.

Lemma 3.13. The following facts are provable in $\mathbf{I}\Delta_0(\exp)$:

1. $a \prec 0 \to A$,
2. $c \prec b \oplus \omega^0 \to (c \prec b \to A) \to (c = b \to A) \to A$,
3. $a \oplus 0 = 0 \oplus a = a$,
4. $a \oplus (b \oplus c) = (a \oplus b) \oplus c$,
5. $\omega^a 0 = 0$,
6. $\omega^a(x + 1) = \omega^a x \oplus \omega^a$,
7. $a \neq 0 \to c \prec b \oplus \omega^a \to c \prec b \oplus w^{\mathbf{e}(a,b,c)} \mathbf{m}(a, b, c)$,
8. $a \neq c \to c \prec b \oplus \omega^a \to \mathbf{e}(a, b, c) \prec a$.

where $\mathbf{e}$ and $\mathbf{m}$ denote appropriate elementary function constants.

Theorem 3.2 (Gentzen, Parsons)**.** For every Π_2-formula F and each $i > 0$ we can prove in $\mathbf{I}\Sigma_{i+1}$ the principle of transfinite induction up to α for all $\alpha < \varepsilon_0(i)$.

Proof. Let $A(a)$ be a Π_j formula, let

$$A^+(a) := \forall b(\forall c \prec b A(c) \to \forall c \prec b \oplus \omega^a A(c))$$

A is Π_j, then by reduction to prenex form, A^+ is equivalent to a Π_{j+1}-formula. The crucial point is that

$$\mathbf{I}\Sigma_j \vdash \mathrm{Prog}_a A(a) \to \mathrm{Prog}^+_a A(a).$$

Assume $\mathrm{Prog}_a A(a)$, i.e. $\forall a(\forall b \prec aA(b) \to A(a))$ and $\forall b \prec aA^+(b)$. We have got to show $A^+(a)$. So assume $\forall c \prec bA(c)$ and $c \prec b \oplus \omega^a$. Let us show $A(c)$.

Let $a = 0$, then $c \prec b \oplus \omega^0$. By Lemma 3.13. 2, it is sufficient to derive $A(c)$ from $c \prec b$ as well as from $a = b$. If $c \prec b$, then $A(c)$ follows from $\forall c \prec bA(c)$ by quantifier elimination. If $c = b$, then $A(c)$ follows from $\mathrm{Prog}_a A(a)$ and $\forall c \prec bA(c)$.

Assume $a \neq 0$ and $c \prec b \oplus \omega^a$, then we obtain the following by Lemma 3.13. 7

$$c \prec b \oplus \omega^{\mathbf{e}(a,b,c)}\mathbf{m}(a,b,c)$$

and $\mathbf{e}(a,b,c)$ by Lemma 3.13. 8. From $\forall b \prec aA^+(b)$ we get $A^+(\mathbf{e}(a,b,c))$. By the definition of A^+ we have

$$\forall u \prec b \oplus \omega^{\mathbf{e}(a,b,c)}x \;\; A(u) \to \forall u \prec (b \oplus \omega^{\mathbf{e}(a,b,c)}x) \oplus \omega^{\mathbf{e}(a,b,c)A(u)}.$$

By using Lemma 3.13. 4 and Lemma 3.13. 6 we obtain

$$\forall u \prec b \oplus \omega^{\mathbf{e}(a,b,c)}xA(u) \to \forall u \prec (b \oplus \omega^{\mathbf{e}(a,b,c)}x) \oplus \omega^{\mathbf{e}(a,b,c)}A(u).$$

So we obtain by $\forall c \prec bA(c)$, Lemma 3.13. 3 and Lemma 3.13. 5

$$\forall u \prec b \oplus \omega^{\mathbf{e}(a,b,c)}0 \;\; A(u)$$

So by Π_j-induction we conclude $\forall u \prec b \oplus \omega^{\mathbf{e}(a,b,c)}\mathbf{m}(a,b,c)A(u)$ and thus $A(c)$. Thus $\mathbf{I}\Sigma_j \vdash \mathrm{Prog}_a A(a) \to \mathrm{Prog}_a A^+(a)$.

Take $i > 0$ and let $\prec$ denote the well-ordering $\prec_{\varepsilon_0(i)}$. Let $F(v)$ be Π_2-formula, define $A(a)$ to be $\forall v \prec a\, F(v)$. Thus A is also Π_2 and also the implication $\mathrm{Prog}_v F(v) \to \mathrm{Prog}_a A(a)$ is derivable in $\mathbf{I}\Delta_0(\exp)$. Let us iterate the above procedure i times starting with $j = 2$, we obtain the formulas $A^+, A^{++}, \ldots, A^{(i)}$, where $A^{(i)}$ is Π_{i+2} and

$$\mathbf{I}\Sigma_{i+1} \vdash \mathrm{Prog}_v F(v) \to \mathrm{Prog}_u A^{(i)}(u).$$

Fix $\alpha < \varepsilon_0(i)$ and choose k such that $\alpha \leq \varepsilon_0(i)(k)$. Apply the progressiveness of $A^{(i)}(u)$ $k+1$ times, we obtain $A^{(i)}(\overline{k+1})$. Thus

$$\mathbf{I}\Sigma_{i+1} \vdash \mathrm{Prog}_v F(v) \to A^{(i)}(\overline{k+1}).$$

Let us instantiate the outermost universally quantified variable of $A^{(i)}$ to zero to obtain

$$A^{(i)}(\overline{k+1}) \to A^{(i-1)}(\omega^{\overline{k+1}}).$$

Again, let us instantiate the outermost universally quantified variable in $A^{(i-1)}$ to obtain

$$A^{(i-1)}(\omega^{\overline{k+1}}) \to A^{(i-2)}(\omega^{\omega^{\overline{k+1}}}).$$

Continue this way and note that $\varepsilon_0(i)(k)$ consists of an exponential stack of i ω's with $k+1$ on the top, we finally get down to

$$\mathbf{I}\Sigma_{i+1} \vdash \mathrm{Prog}_v F(v) \to A(\overline{\varepsilon_0(i)(k)}).$$

But $A(\overline{\varepsilon_0(i)(k)})$ is just

$$\forall v \prec \overline{\varepsilon_0(i)(k)} F(v).$$

We thus have proved the transfinite induction principle for F up to $\varepsilon_0(i)(k)$ in $\mathbf{I}\Sigma_{i+1}$ and thus up to α. $\square$

Theorem 3.3. For each i and for every $\alpha < \varepsilon_0(i)$, the fast-growing function F_α is provably recursive in $\mathbf{I}\Sigma_{i+1}$.

Proof. If $i = 0$, then α is finite and F_α is primitive recursive and thus F_α is provably recursive in $\mathbf{I}\Sigma_1$. Suppose $i > 0$. As far as $F_\alpha = H_{\omega^\alpha}$, we need to show that for every $\alpha < \varepsilon_0(i)$ that H_{ω^α} is provably recursive in $\mathbf{I}\Sigma_{i+1}$. The lemma above shows that the defining Π_2-formula $H(\omega^a)$ is provably (structurally) progressive in $\mathbf{I}\Sigma_2$. Thus, by, Gentzen's result

$$\mathbf{I}\Sigma_{i+1} \vdash \forall a \prec \overline{\alpha}\ H(\omega^a).$$

Apply the progressiveness and obtain

$$\mathbf{I}\Sigma_{i+1} \vdash H(\omega^{\overline{\alpha}})$$

Thus $\mathbf{I}\Sigma_{i+1}$ proves the Σ_1-definability of H_{ω^α}. $\square$

Corollary 3.3. Any $\varepsilon_0(i)$-recursive function is provably recursive in $\mathbf{I}\Sigma_{i+1}$.

Proof. We have already showed that each $\varepsilon_0(i)$-recursive function is register-machine computable in a number of steps bounded by some F_α with $\alpha < \varepsilon_0(i)$. Thus each such function is primitive recursively in $\mathbf{I}\Sigma_{i+1}$. Thus we can show the Σ_1-definability of all $\varepsilon_0(i)$-recursive functions. $\square$

4 $\mathbf{RCA}_0$

5 $\mathbf{WKL}_0$

6 $\mathbf{ACA}_0$

7 ATR

8 $\mathbf{\Pi^1_1}$-comprehension

9 Kripke-Platek Set Theory