Characterizing the elementary recursive functions by a fragment of Gödel's T

Arnold Beckmann* and Andreas Weiermann†‡
Institut für mathematische Logik und Grundlagenforschung
der Westfälischen Wilhelms-Universität Münster
Einsteinstr. 62
D-48149 Münster
Germany

February 23, 2004

Abstract

Let T be Gödel's system of primitive recursive functionals of finite type in a combinatory logic formulation. Let $T^\star$ be the subsystem of T in which the iterator and recursor constants are permitted only when immediately applied to type 0 arguments. By a Howard-Schütte-style argument the $T^\star$-derivation lengths are classified in terms of an iterated exponential function. As a consequence a constructive strong normalization proof for $T^\star$ is obtained. Another consequence is that every $T^\star$-representable number-theoretic function is elementary recursive. Furthermore, it is shown that, conversely, every elementary recursive function is representable in $T^\star$.

The expressive weakness of $T^\star$ compared to the full system T can be explained as follows: In contrast to T, computation steps in $T^\star$ never increase the nesting-depth of $\mathcal{I}_\rho$ and $\mathcal{R}_\rho$ at recursion positions.

1 Introduction and motivation

This article is part of a general investigation of (sub-)recursive function theory and the theory of primitive recursive functionals of finite type by using methods borrowed from term rewriting theory. The general idea behind this approach is as follows. Assume that we are interested in a complexity class $\mathcal{C}$ of number-theoretic functions which is defined equationally. First, one attempts to assign to $\mathcal{C}$

*email: Arnold.Beckmann@math.uni-muenster.de
†Research partially supported by a Heisenberg fellowship of the DFG
‡email: weierma@math.uni-muenster.de

a canonical rewrite system $\mathcal{R}_{\mathcal{C}}$ that computes every function in $\mathcal{C}$. Second, one classifies the derivation lengths in $\mathcal{R}_{\mathcal{C}}$. If this can be done, we obtain information about the computational nature of $\mathcal{C}$. For example, we obtain time bounds for computing the elements of $\mathcal{C}$ on a Turing or register machine. This approach has been applied successfully to various small and large complexity classes (See, for example, [1, 5, 10, 14, 15, 16, 17]). It is particularly suited when $\mathcal{C}$ is defined via a (typed or untyped) reduction system, since then the choice of $\mathcal{R}_{\mathcal{C}}$ is clear in advance. In [17, 18] a classification of derivation lengths for Gödel's system T of primitive recursive functionals of finite types in the CL and the lambda formulation have been given by explicitly constructing an ε_0-recursive function $\mathcal{D} : T \to \omega$ so that a reduces to b implies $\mathcal{D}a > \mathcal{D}b$ for all terms $a, b \in T$. $\mathcal{D}$ may be considered as witnessing constructively the strong normalization of T. In the construction of $\mathcal{D}$ transfinite ordinals come into play via a variant of the Howard-Schütte-function $[\ ]_0 : T \to \varepsilon_0$ (see [7, 13]) which witnesses (constructively) the normalization property of T. In fact, $\mathcal{D}$ and $[\ ]_0$ have to be defined simultaneously. At this point it seems quite natural to pin down exactly the rôle of transfinite ordinals in the original Howard-Schütte proof and in its modification based on $\mathcal{D}$.

Of course, since T represents exactly the $< \varepsilon_0$-recursive functions the use of some form of transfinite induction up to ε_0 is necessary for proving the normalization property for T. Here we will find a natural subsystem $T^\star$ of T for which the strong normalization proof via $[\ ]_0$ does not require transfinite ordinals.

For this purpose let us take a closer look at the Howard-Schütte weak normalization proof for T given in [13]. The infinite ordinals are used when one deals with unrestricted occurrences of iterators $\mathcal{I}_\rho$. For the subsystem $\tilde{T}$ of T, in which iterator operators can occur only when applied to a numeral, the Howard-Schütte function $[\ ]_0$, restricted to $\tilde{T}$ witnesses constructively the strong normalization of $\tilde{T}$. In particular, $[\ ]_0$ also witnesses the strong normalization of the simple typed λ-calculus – i.e. the subsytem of T without iterators – and $[\ ]_0$ also provides non trivial upper bounds for the resulting derivation lengths.

In the Howard-Schütte approach iterator occurrences of the form $\mathcal{I}_\rho t^0$ where t^0 is not a numeral are dealt with by transfinite ordinals. It will be shown here that this is not necessary, and following [16, 17] we propose a modified definition of $[\ ]_0$ (that will also be denoted by $[\ ]_0$) that assigns finite ordinals to $\mathcal{I}_\rho t^0$. Let $T^\star$ be the subsystem of T in which the iterator operators $\mathcal{I}_\rho$ can occur only when applied to a (type 0) argument.

Thus, for example, the T-term $\mathcal{K}_{\rho\rho}\mathcal{I}_\rho\mathcal{I}_\rho$ is not a $T^\star$-term. Then the modified Howard-Schütte assignment $[\ ]_0$ witnesses the strong normalization for $T^\star$ without using transfinite ordinals.

But this proof yields much more interesting information! Indeed $[\cdot]_0$ can be defined in terms of elementary functions such as $0, +, \cdot$ and $\lambda x.2^x$ for any term $a \in T^\star$. Thus, the derivation lengths function for $T^\star$ is in the fourth Grzegorczyk class $\mathcal{E}_4$. Moreover, if $f : \mathbb{N}^k \to \mathbb{N}$ is representable in $T^\star$ then a look at the corresponding derivation lengths shows that f is computable in

elementary recursive time, hence is elementary recursive.

The expressive weakness of $T^\star$ can be explained as follows: Let a be a term of $T^\star$. Then the nesting-depth of iterator operators in an arbitrary computation of a is bounded by the nesting-depth of iterator operators occuring in a itself, i.e. is bounded a priori. This is not the case for unrestricted iterator occurrences $\mathcal{I}_\rho$, since we can not predict a priori the recursion arguments of $\mathcal{I}_\rho$ which may appear during a compution of a term a (in which $\mathcal{I}_\rho$ occurs not restrictedly). This is the deeper reason to assign the infinite ordinal ω in some way to $\mathcal{I}_\rho$, since ω is larger than all the finite ordinals which are assigned to members of the familiy $(\mathcal{I}_\rho t^0)_{t^0 \in T}$

However, it is shown that random-access machine transitions can be simulated in the subsystem of $T^\star$ in which the iterator is dropped. This is utilized for showing that $T^\star$ represents any elementary recursive function, hence $T^\star$ represents exactly the elementary recursive functions.

It turns out that $T^\star$ defines the same class of numbertheoretic functions as Leivant's calculus of predicative recurrence in higher types, namely the class of elementary recursive functions. Nevertheless it seems that Leivant's result is not directly comparable to the characterization of $T^\star$ obtained here. We do not need a tiering operator on finite types. Furthermore, in $T^\star$ any elementary recursive function $f : \mathbb{N}^k \to \mathbb{N}$ is represented by a type one functional, whereas in Leivant's approach one has to find an object type τ_f so that f can be skewly represented by a term of type $\tau_f^k \to 0$. In this paper the use of iterators is essential whereas in Leivant's approach reccurrence can be eliminated by using an interpretation which simulates higher type Church numerals.

The methods of this paper can also be applied to a λ-formulation of $T^\star$. The restriction on terms consists in allowing only iterators of the form $\mathcal{I}_\rho t^0$ and in disallowing λ-abstraction of the form $\lambda x \ldots \mathcal{I}_\rho t^0 \ldots$ where x occurs in t^0. A somewhat related restriction on reductions is already contained in [7], where Howard deals with contracting subterm occurences instead of subform occurences (cf. p. 446 in [7]).

In Section 2 we define $T^\star$ in detail. In Section 3 we classify the $T^\star$-derivation lengths and show that every $T^\star$-representable function is elementary recursive. In Section 4 we prove that every elementary recursive function is representable in $T^\star$. Section 5 contains some applications of the machinery developed in Section 3. In particular, by a uniform Howard-Schütte style argument we show that $T^\star$ is closed under several schemes of recursion like recursion with parameter substitution, simple nested recursion and unnested multiple recursion.

2 Basic definitions

We follow essentially the notation of chapter VI of [13].

Definition 1 *Inductive definition of types.*

1. *The symbol* 0 *is a type.*

2. *If* σ *and* τ *are types than* $(\sigma)\tau$ *is also a type.*

For the sake of brevity we write $\sigma\tau$ for $(\sigma)\tau$.

We posit of the following variables and constants.

1. Denumerably infinitely many *variables* of each type.

2. The *arithmetic* constants $\mathcal{O}$ (for the natural number zero), $\mathcal{S}^+$ (for the successor function) and $\mathcal{P}$ (for the predecessor function).

3. The *combinators* $\mathcal{K}_{\sigma\tau}$ and $\mathcal{S}_{\rho\sigma\tau}$ for each type ρ, σ, τ.

4. The *discriminator* $\mathcal{D}_\sigma$ for each type σ.

5. The *iterator functional* $\mathcal{I}_\rho$ for each type ρ.

Compared with Schütte's approach we have added constants for the predecessor function and the discriminator. (In T these functions are definable anyway.) In contrast to the other constants the iterator functionals will not be considered as terms.

Definition 2 *Inductive definition of the terms of* $T^\star$ *and their types*

1. *Every variable of type* τ *is a* term *of type* τ.

2. $\mathcal{O}$ *is a* term *of type* 0.

3. $\mathcal{S}^+$ *is a* term *of type* 00.

4. $\mathcal{P}$ *is a* term *of type* 00.

5. $\mathcal{K}_{\sigma\tau}$ *is a* term *of type* $\tau\sigma\tau$.

6. $\mathcal{S}_{\rho\sigma\tau}$ *is a* term *of type* $(\rho\sigma\tau)(\rho\sigma)\rho\tau$.

7. $\mathcal{D}_\sigma$ *is a* term *of type* $0\sigma\sigma\sigma$.

8. *If* t^0 *is a term of type* 0 *then* $\mathcal{I}_\rho t^0$ *is a* term *of type* $(\rho\rho)\rho\rho$.

9. *If* $a^{\sigma\tau}$ *is a term of type* $\sigma\tau$ *and* b^σ *is a term of type* σ *then* $a^{\sigma\tau}(b^\sigma)$ *is a* term *of type* τ.

For the sake of brevity we write $a^{\sigma\tau}b^\sigma$ for $a^{\sigma\tau}(b^\sigma)$ if the term b^σ is not written as a composite term. x^τ, y^τ, z^τ range over variables of type τ. $a^\tau, b^\tau, c^\tau, d^\tau$ range over terms of type τ. In particular, t^0 ranges over terms of type 0. For avoiding annoying case distinctions we allow $a^{\sigma\tau}b^\sigma$ also to denote terms of the form $\mathcal{I}_\rho t^0$. In this case b^σ and $a^{\sigma\tau}b^\sigma$ are terms but not $a^{\sigma\tau}$.

Definition 3 *Inductive definition of the numerals* $\underline{n}$

1. $\underline{0} := \mathcal{O}$.
2. $\underline{n+1} := \mathcal{S}^+\underline{n}$.

In order to allow general unrestricted reductions we modify Schütte's definition of the one step reduction $\rhd^1$ as follows.

Definition 4 $\rhd^1$ *is the least binary relation on terms of* $T^\star$ *so that*

1. $\mathcal{P}(\mathcal{O}) \rhd^1 \mathcal{O}$.
2. $\mathcal{P}(\mathcal{S}^+ t^0) \rhd^1 t^0$.
3. $\mathcal{K}_{\sigma\tau} a^\tau b^\sigma \rhd^1 a^\tau$.
4. $\mathcal{S}_{\rho\sigma\tau} a^{\rho\sigma\tau} b^{\rho\sigma} c^\rho \rhd^1 a^{\rho\sigma\tau} c^\rho (b^{\rho\sigma} c^\rho)$.
5. $\mathcal{D}_\sigma \mathcal{O} a^\sigma b^\sigma \rhd^1 a^\sigma$.
6. $\mathcal{D}_\sigma (\mathcal{S}^+ t^0) a^\sigma b^\sigma \rhd^1 b^\sigma$.
7. $\mathcal{I}_\tau 0 a^{\tau\tau} b^\tau \rhd^1 b^\tau$.
8. $\mathcal{I}_\tau (\mathcal{S}^+ t^0) a^{\tau\tau} b^\tau \rhd^1 a^{\tau\tau} (\mathcal{I}_\tau t^0 a^{\tau\tau} b^\tau)$.
9. *If* $b^\sigma \rhd^1 c^\sigma$ *then* $a^{\sigma\tau} b^\sigma \rhd^1 a^{\sigma\tau} c^\sigma$.
10. *If* $a^{\sigma\tau} \rhd^1 b^{\sigma\tau}$ *then* $a^{\sigma\tau} c^\sigma \rhd^1 b^{\sigma\tau} c^\sigma$.

By our convention, 9. includes the case: if $s^0 \rhd^1 t^0$ then $\mathcal{I}_\tau s^0 \rhd^1 \mathcal{I}_\tau t^0$.

Let $\rhd$ be the reflexive and transitive closure of $\rhd^1$.

Definition 5
1. *A term* a *is called* normal *or in* normal form *if there is no term* b *such that* $a \rhd^1 b$.
2. *A term* a *is called* strongly normalizable *if there does not exist an infinite sequence of terms* $\langle a_i : i < \omega \rangle$ *so that* $a = a_0$ *and* $a_i \rhd^1 a_{i+1}$ *for all* $i < \omega$.
3. *The relation* $\rhd$ *is called confluent, if for all terms* a, b, c *such that* $a \rhd b$ *and* $a \rhd c$ *there exists a term* d *such that* $b \rhd d$ *and* $c \rhd d$.

Lemma 1
1. *Every term is strongly normalizable.*
2. *The relation* $\rhd$ *is confluent.*
3. *If* a *is a closed and normal term of type* 0 *then there exists an* $n \in \mathbb{N}$ *so that* a *is equal to* $\underline{n}$.

Proof. These results can be adapted, for example, from [13]. Additionally assertion 1 is proved contructively in Section 3. □

Definition 6 *Let* $f : \mathbb{N}^k \to \mathbb{N}$. *We say that* f *is* representable *in* $T^\star$ *if there is a term* a^0 *with variables among* $x_1^0, \ldots, x_k^0$ *so that*

$$f(m_1, \ldots, m_k) = l \iff a[x_1 := \underline{m_1}, \ldots, x_k := \underline{m_k}] \rhd \underline{l}.$$

Note that Lemma yields that for each closed term a of type 0 there is a uniquely determined numeral $\underline{l}$ such that $a \rhd \underline{l}$. We would like to remark that it is not justified to define the representing term for f via the usual CL-translation of $\lambda x_1 \ldots x_k.a$, since – in contrast to T – the system $T^\star$ is not closed under translated λ-abstraction.

3 A constructive strong normalization proof for $T^\star$

Based on previous work by Howard and Schütte [7, 13] we define a mapping $[\cdot]_0$ such that $a \rhd^1 b$ implies $[a]_0 > [b]_0$. As motivated in [7] this requires an assignment of a sequence of numbers to terms that is needed for deeling with the higher types. The definition of the vector of a term a is based on the notion of the level, $g(a^\tau)$, of a term a which itself is given by the level $g(\tau)$ of τ.

Definition 7 *Recursive definition of* $g\tau$.

1. $g0 := 0$.
2. $g(\sigma\tau) := \max\{g\sigma + 1, g\tau\}$.

Definition 8

1. $[0]_i := 0$.
2. $[\mathcal{S}^+]_0 := 1$.
3. $[\mathcal{S}^+]_i := 0$ *if* $i > 0$.
4. $[\mathcal{P}]_0 := 1$.
5. $[\mathcal{P}]_i := 0$ *if* $i > 0$.
6. $[\mathcal{D}_\sigma]_0 := 1$.
7. $[\mathcal{D}_\sigma]_i := 0$ *if* $i > 0$.
8. $[a^\tau]_i := 1$ *if* $i \leq g\tau$ *and* a^τ *is a combinator.*
9. $[a^\tau]_i := 0$ *if* $i > g\tau$ *and* a^τ *is a combinator.*

10. $[\mathcal{I}_\rho t^0]_0 := [t^0]_0$.

11. $[\mathcal{I}_\rho t^0]_i := 1$ *if* $1 \leq i \leq g\rho + 1$.

12. $[\mathcal{I}_\rho t^0]_i := [t^0]_0$ *if* $i = g\rho + 2$.

13. $[\mathcal{I}_\rho t^0]_i := 0$ *if* $i > g\rho + 2$.

14. $[a^{\sigma\tau} b^\sigma]_i := 2^{[a^{\sigma\tau} b^\sigma]_{i+1}} \cdot ([a^{\sigma\tau}]_i + [b^\sigma]_i)$ *if* $i \leq g\sigma$ *and* $a^{\sigma\tau}$ *is not an iterator.*

15. $[a^{\sigma\tau} b^\sigma]_i := [a^{\sigma\tau}]_i$ *if* $i > g\sigma$ *and* $a^{\sigma\tau}$ *is not an iterator.*

The crucial point in Definition 8 is case 12 (cf. the proof of Lemma 13). In Lemma 13 we also need that $[s^0]_0 > [t^0]_0$ implies $[\mathcal{I}_\rho s^0]_0 > [\mathcal{I}_\rho t^0]_0$. This is the reason for putting $[\mathcal{I}_\rho t^0]_0 := [t^0]_0$ in case 10. Assigning a 1 to $\mathcal{S}^+$ in case 2 guarantees that $[\underline{n}]_0 = n$.

Lemma 2 *1. For each term* a *there is a* k *such that* $[a]_i = 0$ *for all* $i > k$.

2. $[a]_i < \omega$. □

Lemma 3 *1.* $[\underline{n}]_0 = n$.

2. $i \geq 1 \Rightarrow [\underline{n}]_i = 0$. □

Proof. By induction on n. □

Lemma 4 *1. If* $a^{\sigma\tau} b^\sigma$ *has not the form* $\mathcal{I}_\rho t^0$ *then* $[a^{\sigma\tau}]_i \leq [a^{\sigma\tau} b^\sigma]_i$.

2. If $i \leq g\sigma$ *then* $[b^\sigma]_i \leq [a^{\sigma\tau} b^\sigma]_i$.

Proof. 1. This follows immediately from case (4) of definition 7. (Cf. [13]).
2. The assertion is clear if $a^{\sigma\tau} b^\sigma \neq \mathcal{I}_\rho t^0$. If $a^{\sigma\tau} b^\sigma = \mathcal{I}_\rho t^0$ then $g\sigma = 0$ and $[\mathcal{I}_\rho t^0]_0 = [t^0]_0$ holds by definition. □

Lemma 5 *1.* $[a^{\sigma\tau} b^\sigma]_i \leq 2^{[a^{\sigma\tau} b^\sigma]_{i+1}} \cdot ([a^{\sigma\tau}]_i + [b^\sigma]_i)$ *if* $i \leq g\sigma$.

2. $[a^{\sigma\tau} b^\sigma]_i \leq [a^{\sigma\tau}]_i$ *if* $i > g\sigma$ *and* $a^{\sigma\tau} b^\sigma$ *has not the form* $\mathcal{I}_\rho t^0$.

Proof. If $a^{\sigma\tau} b^\sigma \neq \mathcal{I}_\rho t^0$ then equality holds by definition. Assume that $a^{\sigma\tau} b^\sigma = \mathcal{I}_\rho t^0$. Then $\sigma = 0$ and $g\sigma = 0$. Let $i = 0$. Then $[\mathcal{I}_\rho t^0]_0 := [t^0]_0 \leq 2^{[\mathcal{I}_\rho t^0]_1} \cdot ([\mathcal{I}_\rho]_0 + [t^0]_0)$. □

Definition 9 $a^\tau \gg b^\tau :\iff [a^\tau]_0 > [b^\tau]_0 \;\&\; (\forall i \leq g\tau)\, [a^\tau]_i \geq [b^\tau]_i$.

The proofs of the following lemmas 6, 7, 8, 9 and 10 are straightforward by simple monotonicity arguments. We only prove Lemma 10.

Lemma 6 $\mathcal{P}(\mathcal{O}) \gg \mathcal{O}$. □

Lemma 7 $\mathcal{P}(\mathcal{S}^+ t^0) \gg t^0$. □

Lemma 8 $\mathcal{D}_\sigma \mathcal{O} a^\sigma b^\sigma \gg a^\sigma$. □

Lemma 9 $\mathcal{D}_\sigma(\mathcal{S}^+ t^0) a^\sigma b^\sigma \gg b^\sigma$. □

Lemma 10 $\mathcal{K}_{\sigma\tau} a^\tau b^\sigma \gg a^\tau$.

Proof. As in [13] we have $[\mathcal{K}_{\sigma\tau} a^\tau b^\sigma]_i \geq [a^\tau]_i$ by Lemma 3. For $i = 0$ we obtain $[a^\tau]_0 < 2^{[\mathcal{K}_{\sigma\tau} a^\tau b^\sigma]_1} \cdot (2^{[\mathcal{K}_{\sigma\tau} a^\sigma]_1} \cdot ([\mathcal{K}_{\sigma\tau}]_0 + [a^\tau]_0) + [b^\sigma]_0) = [\mathcal{K}_{\sigma\tau} a^\sigma b^\tau]_0$. □

Lemma 11 $\mathcal{S}_{\rho\sigma\tau} a^{\rho\sigma\tau} b^{\rho\sigma} c^\rho \gg a^{\rho\sigma\tau} c^\rho (b^{\rho\sigma} c^\rho)$

Proof. See [13]. □

Lemma 12 $\mathcal{I}_\tau 0 a^{\tau\tau} b^\tau \gg b^\tau$.

Proof. Lemma 3 yields $[\mathcal{I}_\tau 0 a^{\tau\tau} b^\tau]_i \geq [b^\tau]_i$. Furthermore, we have $[b^\tau]_0 < [\mathcal{I}_\tau 0 a^{\tau\tau} b^\tau]_0 = 2^{[\mathcal{I}_\tau 0 a^{\tau\tau} b^\tau]_1} \cdot ([\mathcal{I}_\tau 0 a^{\tau\tau}]_0 + [b^\tau]_0)$. □

Lemma 13 $\mathcal{I}_\tau(\mathcal{S}^+ t^0) a^{\tau\tau} b^\tau \gg a^{\tau\tau}(\mathcal{I}_\tau t^0 a^{\tau\tau} b^\tau)$.

Proof. We follow the argument of [13].
Note that $[\mathcal{S}^+ t^0]_0 = [t^0]_0 + 1$.

Let $\alpha_i := [\mathcal{I}_\tau(\mathcal{S}^+ t^0) a^{\tau\tau}]_i$, $\beta_i := [\mathcal{I}_\tau(\mathcal{S}^+ t^0) a^{\tau\tau} b^\tau]_i$, $\xi_i := [\mathcal{I}_\tau t^0 a^{\tau\tau}]_i$, $\eta_i := [\mathcal{I}_\tau t^0 a^{\tau\tau} b^\tau]_i$, and $\zeta_i := [a^{\tau\tau}(\mathcal{I}_\tau t^0 a^{\tau\tau} b^\tau)]_i$. The proof given in [13] yields

$$(*) \qquad \xi_i < \alpha_i$$

and

$$(**) \qquad \eta_i + \zeta_i + 1 \leq \beta_i$$

whenever $1 \leq i \leq g\tau$. We now prove $\zeta_0 = [a^{\tau\tau}(\mathcal{I}_\tau t^0 a^{\tau\tau} b^\tau)]_0 < \beta_0$.
Since $a^{\tau\tau}$ is not an iterator we have $[a^{\tau\tau}(\mathcal{I}_\tau t^0 a^{\tau\tau} b^\tau)]_0 =$
$2^{\zeta_1} \cdot ([a]_0 + \eta_0) = 2^{\zeta_1} \cdot ([a]_0 + 2^{\eta_1} \cdot (\xi_0 + [b]_0)) =$

$$\zeta_0 = 2^{\zeta_1} \cdot ([a]_0 + 2^{\eta_1} \cdot (2^{\xi_1} \cdot ([t^0]_0 + [a]_0) + [b]_0))$$

By $(*)$ and $(**)$ ζ_0 is smaller than

$$\beta_0 = 2^{\beta_1} \cdot (\alpha_0 + [b]_0) = 2^{\beta_1} \cdot (2^{\alpha_1} \cdot ([t^0]_0 + 1 + [a]_0) + [b]_0).$$

□

Lemma 14 $b^\sigma \gg c^\sigma \Rightarrow a^{\sigma\tau} b^\sigma \gg a^{\sigma\tau} c^\sigma$

Proof. Assume first that $a^{\sigma\tau} b^\sigma$ has not the form $\mathcal{I}_\rho t^0$. Then

1. $[a^{\sigma\tau} b^\sigma]_i = 2^{[a^{\sigma\tau} b^\sigma]_{i+1}} \cdot ([a^{\sigma\tau}]_i + [b^\sigma]_i)$ if $i \leq g\sigma$.

2. $[a^{\sigma\tau}b^{\sigma}]_i = [a^{\sigma\tau}]_i$ if $i > g\sigma$.

3. $[a^{\sigma\tau}c^{\sigma}]_i = 2^{[a^{\sigma\tau}c^{\sigma}]_{i+1}} \cdot ([a^{\sigma\tau}]_i + [c^{\sigma}]_i)$ if $i \leq g\sigma$.

4. $[a^{\sigma\tau}c^{\sigma}]_i = [a^{\sigma\tau}]_i$ if $i > g\sigma$.

In this case the assertion follows immediately from the assumption.

Assume now that $a^{\sigma\tau}b^{\sigma} = \mathcal{I}_\rho t^0$. Then c^{σ} is also of type 0. $b^{\sigma} \gg c^{\sigma}$ yields $[\mathcal{I}_\rho b^{\sigma}]_0 > [\mathcal{I}_\rho c^{\sigma}]_0$. Let $i > 0$. If $i \neq g\rho + 2$ then $[\mathcal{I}_\rho b^{\sigma}]_i = 1 = [\mathcal{I}_\rho c^{\sigma}]_i$. Finally, for the critical case, suppose $i = g\rho + 2$. Then $b^{\sigma} \gg c^{\sigma}$ yields $[\mathcal{I}_\rho b^{\sigma}]_i = [b^{\sigma}]_0 > [c^{\sigma}]_0 = [\mathcal{I}_\rho c^{\sigma}]_i$. □

Lemma 15 *Assume that neither $a^{\sigma\tau}c^{\sigma}$ nor $b^{\sigma\tau}c^{\sigma}$ have the form $\mathcal{I}_\rho t^0$.*
$a^{\sigma\tau} \gg b^{\sigma\tau} \Rightarrow a^{\sigma\tau}c^{\sigma} \gg b^{\sigma\tau}c^{\sigma}$.

Proof. The assumptions yield

1. $[a^{\sigma\tau}c^{\sigma}]_i = 2^{[a^{\sigma\tau}c^{\sigma}]_{i+1}} \cdot ([a^{\sigma\tau}]_i + [c^{\sigma}]_i)$ if $i \leq g\sigma$.

2. $[a^{\sigma\tau}c^{\sigma}]_i = [a^{\sigma\tau}]_i$ if $i > g\sigma$.

3. $[b^{\sigma\tau}c^{\sigma}]_i = 2^{[b^{\sigma\tau}c^{\sigma}]_{i+1}} \cdot ([b^{\sigma\tau}]_i + [c^{\sigma}]_i)$ if $i \leq g\sigma$.

4. $[b^{\sigma\tau}c^{\sigma}]_i = [b^{\sigma\tau}]_i$ if $i > g\sigma$.

Thus the assertion follows immediately from $a^{\sigma\tau} \gg b^{\sigma\tau}$. □

Theorem 1 *If $a \rhd^1 b$ then $a \gg b$, hence $[a]_0 > [b]_0$.*

Proof. By induction on the definition of $\rhd^1$. The base cases are covered by lemmas 6 through 13.

Lemma 14 and 15 yield closure under application. □

Let $lh(a)$ denote the number of constants and variables in a. Thus $lh(0) = lh(\mathcal{S}^+) = lh(\mathcal{P}) = lh(\mathcal{D}_\sigma) = lh(\mathcal{K}_{\sigma\tau}) = lh(\mathcal{S}_{\rho\sigma\tau}) = lh(\mathcal{I}_\rho) = lh(x^\tau) = 1$ and $lh(ab) = lh(a) + lh(b)$. Let $G(a) := g(\tau)$ if a is a constant of type τ and set $G(ab) := \max\{G(a), G(b)\}$ otherwise. Then $G(a)$ is the maximal type level in a. An inspection of the definition of $[a]_i$ yields $[a]_i = 0$ for $i > G(a)$. Let $2_0(\alpha) := \alpha$ and $2_{n+1}(\alpha) := 2^{2_n(\alpha)}$.

Let T^- be $T^\star$ without the iterator constants $\mathcal{I}_\rho$. An immediate verification yields that $[a]_i \leq 2_{G(a)+1 \dot{-} i}(G(a) + 1 \dot{-} i + 2 \cdot lh(a))$ holds for every $a \in T^-$. Thus $[a]_0 \leq 2_{G(a)+1}(G(a) + 1 + 2 \cdot lh(a))$.

If $a_1 \rhd^1 \ldots \rhd^1 a_l$ then, by Theorem 1, $l \leq [a_1]_0 \leq 2_{G(a_1)+1}(G(a_1) + 1 + 2 \cdot lh(a_1))$. Let $\mathcal{DER}_{T^-}$, the derivation lengths function for T^-, be defined as follows: $\mathcal{DER}_{T^-}(m)$ is the maximal length of a reduction chain starting with any T^--term a, such that the depths of a is less than or equal to m. Then $\mathcal{DER}_{T^-}$ is an element of the fourth Grzegorczyk class $\mathcal{E}_4$ (See, for example, [11]

for a definition). Furthermore, let $a \in T^-$ be of type 00. Then the height of the reduction tree for $a\underline{n}$ is bounded by $2_k(n)$ where k only depends on a.

These observations can be extended to $T^\star$ as follows.

Definition 10 *Definition of the iterator nesting degree, $d(a)$, of a $T^\star$-term a.*

1. $d(0) = d(\mathcal{S}^+) = d(\mathcal{P}) = d(\mathcal{D}_\sigma) = d(\mathcal{K}_{\sigma\tau}) = d(\mathcal{S}_{\rho\sigma\tau}) = d(x^\tau) := 0$.
2. *If* $a^{\sigma\tau}b^\sigma = \mathcal{I}_\rho t^0$ *then* $d(a^{\sigma\tau}b^\sigma) := d(t^0) + 1$.
3. *If* $a^{\sigma\tau}b^\sigma \neq \mathcal{I}_\rho t^0$ *then* $d(a^{\sigma\tau}b^\sigma) := \max\{d(a^{\sigma\tau}), d(b^\sigma)\}$.

Lemma 16

$$[a]_i \leq 2_{(G(a)+1)\cdot d(a)+G(a)+1\dot{-}i}((G(a)+1)\cdot d(a) + G(a) + 1 \dot{-} i + 2\cdot lh(a)).$$

Proof. By induction on $lh(a)$ and subsidiary induction on $i \leq G(a)$. Recall that $[a]_i = 0$ for $i > G(a)$. If $lh(a) = 1$ then $[a]_i \leq 1$ and the assertion is clear. If a has the form bc but not the form $\mathcal{I}_\rho t^0$ then the assertion follows immediately from the induction hypothesis, since

$$2_m(x+1)\cdot(2_m(x) + 2_m(x)) \leq 2_m(x+2)$$

holds for all x and $m \geq 2$.

For the critical case assume that $a = \mathcal{I}_\rho t^0$. The induction hypothesis yields

$$\begin{aligned}
[\mathcal{I}_\rho t^0]_0 &= [\mathcal{I}_\rho t^0]_{\rho+2} \\
&= [t^0]_0 \\
&\leq 2_{(G(t^0)+1)\cdot d(t^0)+G(t^0)+1}((G(t^0)+1)\cdot d(t^0) + G(t^0) + 1 + 2\cdot lh(t^0)) \\
&\leq 2_{(G(t^0)+1)\cdot d(a)}((G(t^0)+1)\cdot d(a) + 2\cdot lh(t^0)) \\
&\leq 2_{(G(a)+1)\cdot d(a)}((G(a)+1)\cdot d(a) + 2\cdot lh(a))
\end{aligned}$$

Thus we obtain

$$[a]_i \leq 2_{(G(a)+1)\cdot d(a)+G(a)+1\dot{-}i}((G(a)+1)\cdot d(a) + G(a) + 1 \dot{-} i + 2\cdot lh(a))$$

for any $i \leq \rho + 2$. □

Lemma 17 *Let $a \in T^\star$ be a term of type 0 such that all the variables of a are contained in $\{x_1^0, \ldots, x_k^0\}$. Then $G(a) = G(a[x_1^0 := \underline{m_1}, \ldots, x_k^0 := \underline{m_k}])$ and $d(a) = d(a[x_1^0 := \underline{m_1}, \ldots, x_k^0 := \underline{m_k}]$. Moreover, if every variable from $\{x_1^0, \ldots, x_k^0\}$ occurs at most once in a then $lh(a[x_1^0 := \underline{m_1}, \ldots, x_k^0 := \underline{m_k}]) \leq lh(a) + m_1 + \cdots + m_k$.*

Proof. By a straightforward induction on $lh(a)$. □

Corollary 1 *Let $f : \mathbb{N}^k \to \mathbb{N}$ be representable in $T^\star$. Then f is elementary recursive.*

Proof. Let $f : \mathbb{N}^k \to \mathbb{N}$ be represented by a term $a \in T^\star$ whose variables are among $\{x_1^0, \ldots, x_k^0\}$ and no variable occurs more than once. Then by Lemma 16 and Lemma 17 $[a[x_1^0 := \underline{m_1}, \ldots, x_k^0 := \underline{m_k}]]_0 \leq 2_{(G(a)+1)\cdot d(a)+G(a)+1}((G(a)+1) \cdot d(a) + G(a) + 1 + 2 \cdot (lh(a) + m_1 + \cdots + m_k))$. Thus by Theorem 1 the number of steps needed for rewriting $a[x_1^0 := \underline{m_1}, \ldots, x_k^0 := \underline{m_k}]$ by any reasonable rewriting strategy, for example leftmost reduction, into the numeral representing its value is bounded elementary recursive in $m_1, \ldots, m_k$. Hence, f is computable in elementary recursive time and thus is elementary recursive. □

4 Representing the elementary recursive functions

In this section we prove in detail that every elementary recursive function is representable in $T^\star$. It is obvious that random–access machine (RAM)–computations can be simulatet in Gödel's T. Although it is known in the literature (see, for example, Leivant [8]) that this can also be done in weaker systems the proofs known to us are somehow sketchy and do not clarify the rôle of iterators in such a simulation. Therefore, we will define explicitly the transition function of a RAM in T^-, the subsystem of $T^\star$ in which no term occurrences of (restricted) iterators are allowed. (T^- can be viewed as a CL-formulation of the simply typed λ-calculus.) We will see that only iterators of type 0 and 00 are needed to represent the elementary recursive functions. Then we can apply the following characterization of the elementary recursive functions (cf., for example, [11]).

Fact 1 *$f : \mathbb{N}^m \to \mathbb{N}$ is elementary recursive iff there are natural numbers $k > m, c_1, c_2$ and a program P for a RAM with k registers such that*

$$f(\vec{a}) = OUT^k(i\tau_P(c_1 + 2_{c_2}(\max\{\vec{a}\}), IN_m^k(\vec{a})))$$

and

$$i\tau_P(c_1 + 2_{c_2}(\max\{\vec{a}\}), IN_m^k(\vec{a})) \text{ is an end–configuration}$$

for all $\vec{a} \in \mathbb{N}^m$.

Recall that $2_m(b)$ is defined by $2_0(b) = b$ and $2_{c+1}(b) = 2^{2_c(b)}$. The notions $IN_m^k, OUT^k, i\tau_P$ are defined explicitly in the following.

To fix the context we recall the notion of a $k+1$–RAM in detail. A *random–access machine with $k+1$ registers* ($k+1$–RAM) is a tuple

$$(\mathbb{N}^{k+1}, \{INC(r), DEC(r) : r \leq k\}, \{BEQ(r) : r \leq k\}).$$

A *program* for a $k+1$–RAM is a finite set $P = \{p_0, \ldots, p_l\}$ of instructions where an *instruction* p_i is a tuple $(i, inst, m)$ with $m \in \{0, \ldots, l+1\}$ and $inst \in \{INC(r), DEC(r) : r \leq k\}$ or a tuple $(i, inst, m, n)$ with $m, n \in \{0, \ldots, l+1\}$ and $inst \in \{BEQ(r) : r \leq k\}$.

The *transition function* for a program $P = \{p_0, \ldots, p_l\}$ is the map

$$\tau_P : \{0, \ldots, l+1\} \times \mathbb{N}^{k+1} \to \{0, \ldots, l+1\} \times \mathbb{N}^{k+1}$$

which is defined by

$$\tau_P(i, (z_0, \ldots, z_k)) = \begin{cases} (m, (z_0, \ldots, z_r + 1, \ldots, z_k)) & \text{if } (i, INC(r), m) \in P \\ (m, (z_0, \ldots, z_r \dot{-} 1, \ldots, z_k)) & \text{if } (i, DEC(r), m) \in P \\ (m, (z_0, \ldots, z_r, \ldots, z_k)) & \text{if } (i, BEQ(r), m, n) \in P \\ & \text{and } z_r = 0 \\ (n, (z_0, \ldots, z_r, \ldots, z_k)) & \text{if } (i, BEQ(r), m, n) \in P \\ & \text{and } z_r \neq 0 \\ (i, (z_0, \ldots, z_k)) & \text{if } i = l+1. \end{cases}$$

An element from $\{0, \ldots, l+1\} \times \mathbb{N}^{k+1}$ is called a *configuration*, one from $\{l+1\} \times \mathbb{N}^{k+1}$ an *end–configuration*. The *iterated transition function*

$$i\tau_P : \mathbb{N} \times \mathbb{N}^{k+1} \to \{0, \ldots, l+1\} \times \mathbb{N}^{k+1}$$

is defined by

$$i\tau_P(0, \vec{z}) = (0, \vec{z}), \qquad i\tau_P(n+1, \vec{z}) = \tau_P(i\tau_P(n, \vec{z})).$$

We set

$$IN_m^{k+1} : \mathbb{N}^m \to \mathbb{N}^{k+1}, \qquad (z_1, \ldots, z_m) \mapsto (z_1, \ldots, z_m, 0, \ldots, 0)$$

and

$$OUT^{k+1} : \mathbb{N} \times \mathbb{N}^{k+1} \to \mathbb{N}, \qquad (i, (z_0, \ldots, z_k)) \mapsto z_0.$$

To simulate transitions of a k–RAM in T^- we recall the standard definition of λ–abstraction for T^--terms. (See, for example, [13] for a definition.)

Definition 11 *1.* $\lambda x^\rho.x^\rho := \mathcal{ID}_\rho := \mathcal{S}_{\rho(\rho\rho)\rho}\mathcal{K}_{(\rho\rho)\rho}\mathcal{K}_{\rho\rho}$

2. If x^ρ *does not occur in* c^τ*, then* $\lambda x^\rho.c^\tau := \mathcal{K}_{\rho\tau}c^\tau$

3. If x^ρ *occurs in* $(a^{\sigma\tau}b^\sigma)$*, then* $\lambda x^\rho.(a^{\sigma\tau}b^\sigma) := \mathcal{S}_{\rho\sigma\tau}(\lambda x^\rho.a^{\sigma\tau})(\lambda x^\rho.b^\sigma)$

Then $\lambda x^\rho.c^\tau \in T^-$ is a term of type $\rho\tau$ for any term $c^\tau \in T^-$ of type τ and x^ρ does not occur in $\lambda x^\rho.c^\tau$.

In the sequel we write $\lambda x^\rho.f[x]$ in short for $\lambda x^\rho.f[x^\rho]$. As usual we put $\lambda x_1 \ldots x_n.a := \lambda x_1(\ldots(\lambda x_n.a)\ldots)$. It should be noticed that the definition of λ–abstraction can not be extended to all $T^\star$-terms, because we then would have $\lambda x^0(\mathcal{I}_\rho x)$ as a term, hence $\mathcal{I}_\rho$ has to be a term.

Theorem 2 *Assume that $c \in T^-$ is of type τ and $d \in T^\star$ is of type ρ. Then*

$$(\lambda x^\rho.c^\tau)d^\rho \rhd c^\tau[x^\rho := d^\rho].$$

Proof by induction on the generation of $c \in T^-$. □

In order to deal with formal tuple generation and decomposition we first extend the discriminator as follows.

$$\begin{aligned}
&\mathcal{D}^2_\sigma := \mathcal{D}_\sigma \\
&\mathcal{D}^{n+1}_\sigma := \lambda x^0 y^\sigma_0 \dots y^\sigma_n.\ \mathcal{D}_\sigma x y_0(\mathcal{D}^n_\sigma(\mathcal{P}x)y_1 \dots y_n).
\end{aligned}$$

Then $\mathcal{D}^n_\sigma \in T^-$ is a term of type $0\sigma^{n+1}$ such that

$$\mathcal{D}^{n+1}_\sigma \underline{m} s_0 \dots s_n \rhd \begin{cases} s_m & \text{if } m \le n \\ s_n & \text{else} \end{cases}$$

Let

$$\begin{aligned}
dec^\sigma_i &= \lambda x^{0\sigma}.x\underline{i} \\
tu^\sigma_k &= \lambda x^\sigma_1 \dots x^\sigma_k y^0.(\mathcal{D}^k_\sigma y x_1 \dots x_k)
\end{aligned}$$

and observe that $dec^\sigma_i,\ tu^\sigma_k \in T^-$ are of type $(0\sigma)\sigma$ resp. $\sigma^k 0\sigma$. For terms $s_1, \dots, s_k \in T^\star$ of type σ we compute

$$\begin{aligned}
dec^\sigma_i(tu^\sigma_k s_1 \dots s_k) &\rhd tu^\sigma_k s_1 \dots s_k \underline{i} \\
&\rhd \mathcal{D}^k_\sigma \underline{i} s_1 \dots s_k \\
&\rhd s_i
\end{aligned}$$

[Of course we do **not** have $tu(dec_0 s)(dec_1 s) \rhd s$]

For a fixed k we abbreviate $tu^0_k s_1 \dots s_k$ by $\langle s_1, \dots, s_k \rangle$ and $dec^0_i s$ by $(s)_i$. For a given program $P = \{p_0, \dots, p_l\}$ we formalize the k–RAM–transition by a term of T^- as follows. We interpret a configuration as a $k+1$–tuple, i.e. as a term of type 00. For $0 \le i \le k$ and $0 \le j \le l$ let $A_{ij} \in T^-$ be a term of type 0 which modifies the i–th position of a configuration x^{00} according to the instruction p_j:

$$A_{0j} = \begin{cases} \underline{b} & \text{if } p_j = (j, inst, b) \\ \mathcal{D}(x^{00})_{r+1} \underline{b}\, \underline{c} & \text{if } p_j = (j, BEQ(r), b, c) \end{cases}$$

and for $r < k$

$$A_{(r+1)j} = \begin{cases} \mathcal{P}((x^{00})_{r+1}) & \text{if } p_j = (j, DEC(r), b) \\ \mathcal{S}^+((x^{00})_{r+1}) & \text{if } p_j = (j, INC(r), b) \\ (x^{00})_{r+1} & \text{else.} \end{cases}$$

Then the transition function for the i–th position of a configuration x^{00}, $0 \le i \le k$ is defined by

$$tr^i_P = \mathcal{D}^{l+2}_0 (x^{00})_0 A_{i0} \dots A_{il} I_0.$$

Let

$$tr_P = \lambda x^{00}.\langle tr_P^0, \ldots, tr_P^k\rangle.$$

Obviously, $tr_P^i \in T^-$ is a term of type 0, hence $tr_P \in T^-$ is of type (00)00. By construction we get

Lemma 18

$$\begin{aligned}&\tau_P(a,(z_1,\ldots,z_k)) = (b,(z_1',\ldots,z_k'))\\ &\iff tr_P\langle \underline{a}, \underline{z_1}, \ldots, \underline{z_k}\rangle \rhd \langle \underline{b}, \underline{i}z_1', \ldots, \underline{i}z_k'\rangle\end{aligned}$$

hence

$$\begin{aligned}&i\tau_P(n,(z_1,\ldots,z_k)) = (b,(z_1',\ldots,z_k'))\\ &\iff \mathcal{I}_{00}\underline{n}(tr_P)\langle \underline{0}, \underline{z_1}, \ldots, \underline{z_k}\rangle \rhd \langle \underline{b}, \underline{z_1'}, \ldots, \underline{z_k'}\rangle.\end{aligned}$$

□

Theorem 3 *The elementary recursive functions are representable in $T^\star$.*

Proof. Let $f : \mathbb{N}^m \to \mathbb{N}$ be elementary recursive. By Fact 1 there are natural numbers $k > m, l, c_1, c_2$ and some program $P = \{p_0, \ldots, p_l\}$ for a k–RAM such that

$$f(\vec{q}) = OUT^k(i\tau_P(c_1 + 2_{c_2}(\max\{\vec{q}\}), IN_m^k(\vec{q})))$$

and

$$(i\tau_P(c_1 + 2_{c_2}(\max\{\vec{q}\}), IN_m^k(\vec{q})))_0 > l$$

for all $\vec{q} \in \mathbb{N}^m$.

We define a term $t \in T^\star$ in the variables $x_1^0, \ldots, x_m^0$ which satisfies $\forall \vec{q} \in \mathbb{N}^m \exists r \in \mathbb{N}$

$$t[x_1 := \underline{q_1}, \ldots, x_m := \underline{q_m}] \rhd \underline{r} \quad \text{and} \quad r \geq c_1 + 2_{c_2}(\max\{\vec{q}\}). \tag{1}$$

Let

$$\tilde{t_0} := x_1 + (x_2 + \ldots (x_m + \underline{c_1}) \ldots)$$

where $u + v := \mathcal{I}_0 u(\mathcal{S}^+)v$, and

$$\tilde{t_1} := \mathcal{I}_{00} z^0(\mathit{double})(\mathcal{S}^+)$$

where $\mathit{double} := \lambda y_1^{00} y_2^0.(y_1(y_1 y_2))$. By induction on q one easily verifies that $(\tilde{t_1} u)[z := \underline{q}] \rhd \mathcal{S}^{+(2^q)} u$, hence $(\tilde{t_1}\underline{0})$ represents the exponential function (cf. [8]). We define

$$t_0 := \tilde{t_0}, \quad t_{n+1} := (\tilde{t_1}\underline{0})[z := t_n], \quad t := t_{c_2}$$

and observe that t fulfills the assertion (1).

Now we define

$$R = \mathit{dec}_1^0(\mathcal{I}_{00} t(tr_P)\langle \underline{0}, x_1^0, \ldots, x_m^0\rangle)$$

which is a term of $T^\star$ of type 0. We claim that R represents f.

Indeed, let $\vec{q} \in \mathbb{N}^m$, then there is an $r \in \mathbb{N}$ such that $t[x_1 := \underline{q_1}, \ldots, x_m := \underline{q_m}] \rhd \underline{r}$ and $r \geq c_1 + 2_{c_2}(\max\{\vec{q}\})$. There is also some $\vec{q'} \in \mathbb{N}^k$ with $i\tau_P(r, \vec{q}) = (l+1, \vec{q'})$. We compute

$$f(\vec{q}) = OUT^k(i\tau_P(c_1 + 2_{c_2}(\max\{\vec{q}\}), IN^k_m(\vec{q}))) = OUT^k(l+1, \vec{q'}) = q'_1.$$

Hence

$$\begin{array}{rcl} R[x_1 := \underline{q_1}, \ldots, x_m := \underline{q_m}] & \rhd & dec^0_1(\mathcal{I}_{00}\underline{r}(tr_P)\langle \underline{0}, \underline{q_1}, \ldots, \underline{q_m}\rangle) \\ & \rhd & dec^0_1(\langle \underline{l+1}, \underline{q'_1}, \ldots, \underline{q'_m}\rangle) \\ & \rhd & \underline{q'_1} \equiv \underline{f(\vec{q})}. \end{array}$$

□

5 Further applications

In this section we indicate how the Howard-Schütte style derivation lengths classification of Section 3 can easily be adapted for showing that $T^\star$ is closed under various schemes of recursion like recursion with parameter substitution, simple nested recursion and unnested multiple recursion. For his calculus of predicative recurrence of finite types Leivant has established similar looking closure properties in [8]. Related closure properties of the primitive recursive functions have been shown by Simmons in [12].

First we consider the system in which the iterator is replaced by a constant for the recursor. For any type τ let $\mathcal{R}_\tau$ be a new constructor constant. We enlarge the term definition [Def. 2] by the following rule:
If t^0 is a term of type 0 then $\mathcal{R}_\tau t^0$ is a term of type $(0\tau\tau)\tau\tau$.
The definition of the reduction relation $\rhd^1$ [Def. 4] is extended by the following clauses:

$$\begin{array}{rcl} \mathcal{R}_\tau 0 a^{0\tau\tau} b^\tau & \rhd^1 & b^\tau, \\ \mathcal{R}_\tau(\mathcal{S}^+ t^0) a^{0\tau\tau} b^\tau & \rhd^1 & (a^{0\tau\tau} t^0)(\mathcal{R}_\tau t^0 a^{0\tau\tau} b^\tau). \end{array}$$

The definition of $[\,]_i$ is extended as follows:

1. $[\mathcal{R}_\tau t^0]_i := [t^0]_i$ if $i = 0$.
2. $[\mathcal{R}_\tau t^0]_i := 1$ if $1 \leq i \leq g\rho + 1$.
3. $[\mathcal{R}_\tau t^0]_i := [t^0]_0$ if $i = g\rho + 2$.
4. $[\mathcal{R}_\tau t^0]_i := 0$ if $i > g\rho + 2$.

Then, for the extended calculus one shows that $a \rhd^1 b$ implies $[a]_0 > [b]_0$ and as in Section 3 we see that the derivation lengths function for any term representing

a number-theoretic function in the extended calculus is elementary recursive in the input arguments.

The treatment of recursion with parameter substitution, simple nested recursion and unnested multiple recursion is similar. Let us start with parameter recursion. For any type τ let $\mathcal{PR}_\tau$ be a new constructor constant. We enlarge the term definition by the following rule:
If t^0 is a term of type 0 and if $c^{\tau\tau}$ is a term of type $\tau\tau$ then $\mathcal{PR}_\tau c^{\tau\tau} t^0$ is a term of type $(0\tau\tau)\tau\tau$.
The defining reduction rules for recursion with parameter substitution are:

$$\begin{aligned} \mathcal{PR}_\tau c^{\tau\tau} 0 a^{0\tau\tau} b^\tau &\rhd^1 b^\tau, \\ \mathcal{PR}_\tau c^{\tau\tau} (\mathcal{S}^+ t^0) a^{0\tau\tau} b^\tau &\rhd^1 (a^{0\tau\tau} t^0)(\mathcal{PR}_\tau c^{\tau\tau} t^0 a^{0\tau\tau} (c^{\tau\tau} b^\tau)) \end{aligned}$$

For treating these rules the definition of $[\,]_i$ is extended as follows:

1. $[\mathcal{PR}_\tau c^{\tau\tau} t^0]_i := [c^{\tau\tau}]_i + [t^0]_i$ if $i = 0$.
2. $[\mathcal{PR}_\tau c^{\tau\tau} t^0]_i := 1$ if $1 \leq i \leq g\rho + 1$.
3. $[\mathcal{PR}_\tau c^{\tau\tau} t^0]_i := [c^{\tau\tau}]_0 + [t^0]_0$ if $i = g\rho + 2$.
4. $[\mathcal{PR}_\tau c^{\tau\tau} t^0]_i := 0$ if $i > g\rho + 2$.

Then, for the extended calculus one shows that $a \rhd^1 b$ implies $[a]_0 > [b]_0$ and as before we see that the derivation lengths function for any term representing a number-theoretic function in the extended calculus is elementary recursive in the input arguments.

Now we deal with simple nested recursion. For any type τ let $\mathcal{SNR}_\tau$ be a new constructor constant. The term definition is extended as follows:
If t^0 is a term of type 0 then $\mathcal{SNR}_\tau t^0$ is a term of type $(0\tau\tau)\tau\tau$.
The defining reduction rules for simple nested recursion are:

$$\begin{aligned} \mathcal{SNR}_\tau 0 a^{0\tau\tau} b^\tau &\rhd^1 b^\tau, \\ \mathcal{SNR}_\tau (\mathcal{S}^+ t^0) a^{0\tau\tau} b^\tau &\rhd^1 (a^{0\tau\tau} t^0)(\mathcal{SNR}_\tau t^0 a^{0\tau\tau} (\mathcal{SNR}_\tau t^0 a^{0\tau\tau} b)) \end{aligned}$$

For treating these rules the definition of $[\,]_i$ is changed as follows:

1. $[\mathcal{SNR}_\tau t^0]_i := [t^0]_i$ if $i = 0$.
2. $[\mathcal{SNR}_\tau t^0]_i := 1$ if $1 \leq i \leq g\rho + 1$.
3. $[\mathcal{SNR}_\tau t^0]_i := [t^0]_0$ if $i = g\rho + 2$.
4. $[\mathcal{SNR}_\tau t^0]_i := 0$ if $i > g\rho + 2$.
5. $[a^{\sigma\tau} b^\sigma]_i := 4^{[a^{\sigma\tau} b^\sigma]_{i+1}} \cdot ([a]_i + [b]_i)$ if $a^{\sigma\tau}$ is not an iterator or recursor.

Again, for the extended calculus one shows that $a \rhd^1 b$ implies $[a]_0 > [b]_0$ and as before we see that the derivation lengths function for any any term representing a number-theoretic function in the extended calculus is elementary recursive in the input arguments.

Finally, let us treat unnested multiple recursion. For any type τ let $\mathcal{UMR}_\tau$ be a new constructor constant. The term definition is extended as follows:
If t^0 and s^0 are terms of type 0 and c^{00} is of type 00 then $\mathcal{UMR}_\tau c^{00} s^0 t^0$ is a term of type $(0\tau\tau\tau)\tau\tau$.
The defining reduction rules for unnested multiple recursion are:

$$\begin{aligned} \mathcal{UMR}_\tau c^{00}(s^0)0a^{0\tau\tau\tau}b^\tau &\rhd^1 b^\tau, \\ \mathcal{UMR}_\tau c^{00}0(t^0)a^{0\tau\tau\tau}b^\tau &\rhd^1 b^\tau, \\ \mathcal{UMR}_\tau c^{00}(\mathcal{S}^+ s^0)(\mathcal{S}^+ t^0)a^{0\tau\tau\tau}b^\tau &\rhd^1 \\ (a^{0\tau\tau\tau}t^0)(\mathcal{UMR}_\tau c^{00}s^0(c^{00}t^0)a^{0\tau\tau\tau}b^\tau)&(\mathcal{UMR}_\tau c^{00}(\mathcal{S}^+ s^0)t^0 a^{0\tau\tau\tau}b^\tau) \end{aligned}$$

For treating these rules the definition of $[\,]_i$ is changed as follows:

1. $[\mathcal{UMR}_\tau c^{00}s^0t^0]_i := [t^0]_i$ if $i = 0$.
2. $[\mathcal{UMR}_\tau c^{00}s^0t^0]_i := 1$ if $1 \le i \le g\rho + 1$.
3. $[\mathcal{UMR}_\tau c^{00}s^0t^0]_i := 8^{([c]_0+1)^{3\cdot([s]_0+1)}\cdot([t^0]_0+1)}$ if $i = g\rho + 2$.
4. $[\mathcal{UMR}_\tau c^{00}s^0t^0]_i := 0$ if $i > g\rho + 2$.
5. $[a^{\sigma\tau}b^\sigma]_i := 8^{[a^{\sigma\tau}b^\sigma]_{i+1}} \cdot ([a]_i + [b]_i)$ if $a^{\sigma\tau}$ is not an iterator or recursor.

Then, for the extended calculus one shows that $a \rhd^1 b$ implies $[a]_0 > [b]_0$ and as before we see that the derivation lengths function for any any term representing a number-theoretic function in the extended calculus is elementary recursive in the input arguments.

Acknowledgements. The authors would like to thank the referee for valuable comments.

References

[1] Beckmann, A. and Weiermann, A.: *A term rewriting characterization of the polytime functions and related complexity classes.* Archive for Mathematical Logic 36 (1996), 11-30.

[2] Beckmann, A. and Weiermann, A.: *Investigations on subrecursion via logic programming.* Preprint, Münster (1995).

[3] Beckmann, A.: *Exact bounds for lengths of reductions in typed λ-calculus.* Preprint, Münster (1998) (submitted)

[4] Bellantoni, S. and Cook, S.: *A new recursion-theoretic characterization of the polytime functions.* Comput. Complexity 2, No. 2 (1992), 97-110.

[5] Cichon, E.A. and Weiermann, A.: *Term rewriting theory for the primitive recursive functions.* Annals of Pure and Applied Logic 83 (1997), 199-223.

[6] Hofbauer, D.: *Termination proofs by multiset path orderings imply primitive recursive derivation lengths.* Proc. 2nd ALP. Lecture Notes in Computer Science 463 (1990), 347-358.

[7] W. Howard: *Assignment of ordinals to terms for primitive recursive functionals of finite type.* Intuitionism and Proof Theory. North-Holland, Amsterdamm 1970, 443-458.

[8] D. Leivant: *Predicative recurrence in finite type.* In A. Nerode and Y.V. Matiyasevich (eds.), Logical Foundations of Computer Science. Springer Lecture Notes in Computer Science 813 (1994), 227-239.

[9] D. Leivant: *Ramified recurrence and computational complexity I: Word recurrence and poly-time.* Feasible Mathematics II, P. Clote and J. Remmel (eds.), Perspectives in Computer Science, Birkhäuser (1995).

[10] Möllerfeld, M. and Weiermann, A.: *A uniform approach to $\prec$-recursion.* Preprint, Münster (1995) (submitted).

[11] H.E. Rose: Subrecursion: Functions and Hierarchies. Oxford University Press 1984.

[12] Simmons, H.: *The realm of primitive recursion.* Arch. Math. Logic 27 (1988), 177-188.

[13] K. Schütte: *Proof Theory.* Springer 1977.

[14] Weiermann, A.: *Termination proofs for term rewriting systems by lexicographic path orderings yield multiply recursive derivation lengths.* Theoretical Computer Science 139 (1995), 355-362.

[15] Weiermann, A.: *Rewriting theory for the Hydra battle and the extended Grzegorczyk hierarchy*, Preprint, Nancy and Münster (1995) The Journal of Symbolic Logic (to appear).

[16] Weiermann, A.: *A proof of strongly uniform termination for Gödel's T by methods from local predicativity.* Archive for Mathematical Logic 36 (1997), 445-460.

[17] Weiermann, A.: *How is it that infinitary methods can be applied to finitary mathematics. Gödel's T: a case study.* The Journal of Symbolic Logic 63, Number 4, (1998), 1348-1370.

[18] Wilken, G. and Weiermann, A.: *Sharp upper bounds for the depths of reduction trees of typed lambda calculus with recursors.* Preprint, Münster (1998) (submitted).