Annals of Pure and Applied Logic 160 (2009) 255–288

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

journal homepage: www.elsevier.com/locate/apal

A foundation for real recursive function theory

José Félix Costa [a,b,*], Bruno Loff [a,b], Jerzy Mycka [c]

[a] Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal
[b] Centro de Matemática e Aplicações Fundamentais do Complexo Interdisciplinar, University of Lisbon, Lisbon, Portugal
[c] Institute of Mathematics, University of Maria Curie-Sklodowska, Lublin, Poland

ARTICLE INFO

Article history:
Available online 19 March 2009

MSC:
03D99
68Q05
28A05

Keywords:
Real recursive functions
Infinite limits
Differential recursion
Computability on reals

ABSTRACT

The class of recursive functions over the reals, denoted by REC($\mathbb{R}$), was introduced by Cristopher Moore in his seminal paper written in 1995. Since then many subsequent investigations brought new results: the class REC($\mathbb{R}$) was put in relation with the class of functions generated by the General Purpose Analogue Computer of Claude Shannon; classical digital computation was embedded in several ways into the new model of computation; restrictions of REC($\mathbb{R}$) were proved to represent different classes of recursive functions, e.g., recursive, primitive recursive and elementary functions, and structures such as the Ritchie and the Grzergorczyk hierarchies.

The class of real recursive functions was then stratified in a natural way, and REC($\mathbb{R}$) and the analytic hierarchy were recently recognised as two faces of the same mathematical concept.

In this new article, we bring a strong foundational support to the Real Recursive Function Theory, rooted in Mathematical Analysis, in a way that the reader can easily recognise both its intrinsic mathematical beauty and its extreme simplicity. The new paradigm is now robust and smooth enough to be taught. To achieve such a result some concepts had to change and some new results were added.

1. Introduction

In 1996 Cris Moore published a seminal paper, *Recursive theory on the reals and continuous-time computation* [24], where he defines an inductive class of vector-valued functions over $\mathbb{R}$, aiming to provide a framework to study continuous-time phenomena from a computational perspective. This class was defined as the closure of some basic functions for the operators of composition, solving of first-order differential equations and a kind of minimalisation.

Some work was done since then, using Moore's definition [8,9,26,25], but unfortunately some of Moore's assumptions were not very consensual among people interested in the field. Most of these controversial assumptions were consequences of Moore's attempt to bring the minimalisation operator – used in the classical recursive functions – into a continuous context. So in their paper [27], Jerzy Mycka and José Félix Costa gave a similar definition of Moore's inductive class of functions, replacing minimalisation with the taking of infinite limits. We will cite both papers [24,27] when appropriate.

Restrictions of this inductive scheme have given rise to several interesting characterisations of computability [17,2,4,10] and complexity [9,7,3,29,28] of real functions; this is an analogue of the studies in *sub-recursion* of classical recursion theory. Connections with other areas have appeared, e.g., the study of periodic real recursive functions and the connections between infinite time Turing machines and real recursive functions [14,15].

* Corresponding address: Departamento de Matemática, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001, Lisboa, Portugal.
E-mail addresses: felixgomescosta@gmail.com (J.F. Costa), bruno.loff@gmail.com (B. Loff), Jerzy.Mycka@umcs.lublin.pl (J. Mycka).

0168-0072/$ – see front matter
doi:10.1016/j.apal.2009.01.013

We will, however, be concerned with the more general aspects of the theory. In their papers, Mycka and Costa [27,30,31] have laid out the basics for our study. The most important or distinguishing feature of the more *general* theory, opposed to what we have called sub-recursion, is the access to unrestricted infinite limits, which can be made to work as a search operator over $\mathbb{R}$.

The purpose of this text is to lay out a solid foundation for the theory of real recursive functions, and survey the known results in the more general aspects of the theory. We will begin in Section 2 by studying function algebras and differential equations; after this preparation, we introduce the inductive definition for the class of real recursive functions, and finish the section with some further considerations on differential equations and infinite limits. In Section 3, we prove the most fundamental results in the theory, and show how we may stratify the class of real recursive functions into a hierarchy. Section 4 will solve the problem of universality, with a proof that there is no universal real recursive function. In Section 5 we show that there is an exact correspondence between the class of real recursive functions and the analytical hierarchy of predicates. Finally, in Section 6, we attempt some advance in the problem of collapse; this problem was believed to have been solved in [22], but an error turned up after a solid foundation was put in place — we expound the current status of this open problem.

2. Introduction to real recursive functions

This section will establish the theoretical basis for the study of real recursive functions. We begin by studying function algebras and their related problems in Section 2.1. Some preliminaries on the Cauchy problem will be given in Section 2.2. The *primordial* function algebra which characterises the class of real recursive functions will be presented in Section 2.3. Finally, we will make some general considerations on two specific operators in Sections 2.4 and 2.5.

2.1. Function algebras

A function algebra is a characterisation of a set of functions by the inductive closure, for some operators, of another set of functions. This concept is frequently used in recursion theory, and more recently to obtain characterisations of complexity classes [11].

Definition 2.1. Let $\mathscr{F}$ be a class of functions, $\mathcal{F} \subseteq \mathscr{F}$ be a set of such functions, and $\mathcal{O} \subseteq \cup_{k\in\mathbb{N}}\{O : \mathscr{F}^k \to \mathscr{F}\}$ be a set of operators. The **inductive closure** of $\mathcal{F}$ for $\mathcal{O}$, written $\mathcal{A} = [\mathcal{F} ; \mathcal{O}]$, is the smallest set containing $\mathcal{F}$, such that if $f_1, \ldots, f_k \in \mathcal{A}$ are in the domain of the k-ary $O \in \mathcal{O}$, then $O(f_1, \ldots, f_k) \in \mathcal{A}$. When taken together with $\mathcal{O}$, the inductive closure $[\mathcal{F} ; \mathcal{O}]$ is called a **function algebra**.

A function algebra is said to be **enumerable** if both $\mathcal{F}$ and $\mathcal{O}$ are enumerable.

We will simplify the notation by writing $\mathcal{A} = [\mathcal{F} ; \mathcal{O}]$ to let $\mathcal{A}$ designate both the inductive closure $[\mathcal{F} ; \mathcal{O}]$ and the function algebra $([\mathcal{F} ; \mathcal{O}], \mathcal{O})$, when appropriate.[1]

Remark 2.2. We show that the definition is well-founded, i.e., for any such $\mathcal{F}$ and $\mathcal{O}$ there is a unique smallest $\mathcal{A}$ that satisfies the conditions of closure. This follows easily from Kleene's fixed point theorem. Take the complete lattice of sets of functions in $\mathscr{F}$, $L = \wp(\mathscr{F})$, under the partial order of inclusion. It is easy to see that the function $\mathfrak{f} : L \to L$, given by $\mathfrak{f}(F) = F \cup \mathcal{F} \cup \{O(f_1, \ldots, f_k) : f_1, \ldots, f_k \in F$ are in the domain of $O \in \mathcal{O}\}$, is continuous. We may conclude, by the well-known result of Knaster and Tarski, that there is a least fixed point of f. By Kleene's fixed point theorem,

$$\mathcal{A} = \cup_{k\in\mathbb{N}}\, \mathfrak{f}^k(\varnothing)$$

is this least fixed point. Such an $\mathcal{A}$ satisfies the closure condition since it is a fixed point of $\mathfrak{f}$, and the uniqueness condition since it is the least fixed point. □

Notation 2.3. *We make a liberal use of the square bracket notation, e.g., if f, g are functions, $\mathcal{F}$ is a class of functions, and $O_1, \ldots, O_k$ are operators, then we set*

$$[f, g, \mathcal{F} ; O_1, \ldots, O_n] = [\{f, g\} \cup \mathcal{F} ; \{O_1, \ldots, O_n\}]. \quad \square$$

Example 2.4. Consider the class $\mathscr{F}$ of partial, scalar, multiple-argument functions over $\mathbb{N} = \{0, 1, 2, \ldots\}$. Examples of functions in $\mathscr{F}$ are the zero function z, such that $z(x) = 0$; the successor function s, given by $s(x) = x+1$; and the projection functions u_i^n, $1 \le i \le n$, where each u_i^n obeys $u_i^n(x_1, \ldots, x_n) = x_i$. Take $\boldsymbol{x}$ to designate an arbitrary sequence $\boldsymbol{x} = x_1, \ldots, x_n$. We may consider the composition operators c^m, such that for every $g : \mathbb{N}^m \to \mathbb{N}$, $h_1, \ldots, h_m : \mathbb{N}^n \to \mathbb{N}$, the function $c^m(g, h_1, \ldots, h_m) : \mathbb{N}^n \to \mathbb{N}$ is given by

$$c^m(g, h_1, \ldots, h_m)(\boldsymbol{x}) = g(h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})).$$

[1] The distinction between the inductive closure $[\mathcal{F} ; \mathcal{O}]$ and the function algebra $([\mathcal{F} ; \mathcal{O}], \mathcal{O})$ is not always made in the literature (e.g., in [11]). However, we felt it was an important distinction to make, since certain concepts (such as inclusion of some function) relate to the inductive closure, as a set, and others (such as the rank of a function) relate to the function algebra, with its specific structure.

We will also consider the primitive recursion operator p, which, for every given $g : \mathbb{N}^n \to \mathbb{N}$ and $h : \mathbb{N}^{n+2} \to \mathbb{N}$, sets

$$p(g, h)(\boldsymbol{x}, 0) = g(\boldsymbol{x}) \text{ and } p(g, h)(\boldsymbol{x}, y + 1) = h(\boldsymbol{x}, y, p(g, h)(\boldsymbol{x}, y)).$$

The class PRIM of **primitive recursive functions** may then be given by the function algebra PRIM $= [z, s, u_i^n; c^m, p]$. □

It is important, in the study of a function algebra, to consider self-referential properties of the inductive closure. Function algebras provide a natural way of doing so, in particular for the case when the set of initial functions and the set of operators are countable.

Definition 2.5. Let $\mathcal{A} = [\mathcal{F}; \mathcal{O}]$ be an enumerable function algebra, for a set $\mathcal{F} = \{f_1, f_2, \ldots\}$ of functions and a set $\mathcal{O} = \{O_1, O_2, \ldots\}$ of operators. The set $D_{\mathcal{A}}$ of **descriptions** of $\mathcal{A}$, is the smallest set of words in $\{\texttt{fun}, \texttt{Op}, \langle, \rangle, \textbf{,}, 0, 1, \ldots, 9\}^*$, such that

(i) $\langle \texttt{fun}, n\rangle \in D_{\mathcal{A}}$ for all n, and
(ii) if $d_1, \ldots, d_k \in D_{\mathcal{A}}$, then $\langle \texttt{Op}, n, d_1, \ldots, d_k\rangle \in D_{\mathcal{A}}$.

We write $\mathcal{D}_{\mathcal{A}}$ to stand for the set of **good descriptions** in $D_{\mathcal{A}}$; by good description we mean any description d such that

(i) d is $\langle \texttt{fun}, n\rangle$ and $f_n \in \mathcal{F}$ (i.e., $\mathcal{F}$ has at least n functions), and in this case d is said to **describe** f_n, or
(ii) d is $\langle \texttt{Op}, n, d_1, \ldots, d_k\rangle$, for some good descriptions $d_1, \ldots, d_k$ which describe $g_1, \ldots, g_k$ in the domain of $O_n \in \mathcal{O}$; then d is said to describe $O_n(g_1, \ldots, g_k)$.

Notation 2.6. *With $\mathcal{A} = [\mathcal{F}; \mathcal{O}]$ given as above, let f denote a function $f_n \in \mathcal{F}$ and O denote an operator $O_n \in \mathcal{O}$. Then we write $\texttt{fun}_f$ to denote the pair $\texttt{fun}, n$, and $\texttt{Op}_O$ to denote the pair $\texttt{Op}, n$. In this sense, $\langle \texttt{fun}_f\rangle$ is the description $\langle \texttt{fun}, n\rangle$ and $\langle \texttt{Op}_O, d_1, \ldots, d_k\rangle$ is the description $\langle \texttt{Op}, n, d_1, \ldots, d_k\rangle$.* □

When studying a function algebra one wishes to understand the extent of functions which it contains. One then often considers a particular operator $O : \mathscr{F}^k \to \mathscr{F}$, and attempts to determine whether the function algebra is closed for O or not (meaning that the respective inductive closure is closed for O or not). Many important problems in computer science may be equated to the proof or disproof that certain function algebras are closed for certain operators, e.g., P = NP if and only if P is closed for a bounded minimalisation operator $\bar{\mu}$ [cf. 1,11], i.e., if P = [P; $\bar{\mu}$].[2]

The proof that a function algebra $\mathcal{A}$ is closed for an operator may or may not be constructive. In the former case, one considers that the closure is effective.

Definition 2.7. An enumerable function algebra $\mathcal{A}$ is said to be **effectively closed** under an operator $O : \mathscr{F}^k \to \mathscr{F}$ if there is an effective procedure which, given good descriptions of functions $f_1, \ldots, f_k \in \mathcal{A}$ in the domain of O, obtains a good description of the function $O(f_1, \ldots, f_k)$.

Function algebras usually offer natural measures of complexity, simply by looking at the descriptions which describe a certain function. One such syntactic measure, which is frequently considered, is the number of nested applications of a certain operator, or set of operators. This is called the rank.

Definition 2.8. Let $\mathcal{A} = [\mathcal{F}; \mathcal{O}]$ be an enumerable function algebra, with $\mathcal{O} = \{O_1, O_2, \ldots\}$, and let $\tilde{\mathcal{O}}$ be a subset of $\mathcal{O}$. The **rank of a good description** $d \in \mathcal{D}_{\mathcal{A}}$ for the set of operators $\tilde{\mathcal{O}}$ under the function algebra $\mathcal{A}$, rk(d), is inductively defined as:

(i) $\text{rk}(\langle \texttt{fun}, n\rangle) = 0$,
(ii) if $O_n \notin \tilde{\mathcal{O}}$, then $\text{rk}(\langle \texttt{Op}, n, d_1, \ldots, d_k\rangle) = \max(\text{rk}(d_1), \ldots, \text{rk}(d_k))$, and
(iii) if $O_n \in \tilde{\mathcal{O}}$, then $\text{rk}(\langle \texttt{Op}, n, d_1, \ldots, d_k\rangle) = \max(\text{rk}(d_1), \ldots, \text{rk}(d_k)) + 1$.

The **rank of a function** $f \in \mathcal{A}$ for $\tilde{\mathcal{O}}$ under $\mathcal{A}$, rk(f), is given by:

$$\text{rk}(f) = \min\{\text{rk}(d) : d \text{ is a good description which describes } f\}.$$

Notation 2.9. *The manner of denoting the rank for $\tilde{\mathcal{O}}$ under $\mathcal{A}$, using the word* rk, *does not make the dependency in $\tilde{\mathcal{O}}$ and $\mathcal{A}$ explicit. We chose to do this so as not to over-encumber the notation. When it becomes necessary to disambiguate between different possible function algebras or operator sets, we will then denote the rank by $\text{rk}^{\mathcal{A}}_{\tilde{\mathcal{O}}}$.* □

Example 2.10. One may consider, in the function algebra PRIM $= [z, s, u_i^n; c^m, p]$ for primitive recursive functions, the rank for the primitive recursion operator. Intuitively, the rank $\text{rk}_p^{\text{PRIM}}(f) \equiv \text{rk}(f)$ of a function $f : \mathbb{N} \to \mathbb{N}$ is the smallest number of nested *for loops* necessary for any program to compute f without *while loops* (from the basic functions z, s, and u_i^n). □

---

2 Note that P and NP are often taken as a class of subsets of $\mathbb{N}$, and not a class of functions. The problem P = NP for sets is equivalent to the problem "P = NP" for functions.

The rank for a set of operators, as a measure of complexity, is rich enough to stratify function algebras into interesting hierarchies.

Definition 2.11. Let $\mathcal{A} = [\mathcal{F}; \mathcal{O}]$ be an enumerable function algebra, with $\mathcal{O} = \{O_1, O_2, \ldots\}$, and let $\tilde{\mathcal{O}}$ be a subset of $\mathcal{O}$. The **rank hierarchy** for $\tilde{\mathcal{O}}$ under $\mathcal{A}$ is the $\mathbb{N}$-indexed family of functions:

$$H_n = \{f \in \mathcal{A} : \mathrm{rk}(f) \leqslant n\}.$$

Notation 2.12. *We take the same precautions of Notation 2.9 so as not to burden the notation for the rank hierarchy. When we wish to disambiguate between hierarchies for different operator sets or under different function algebras, we will either index our denotation as in $H_n^{\mathcal{A},\tilde{\mathcal{O}}}$, or assign different letters to the different hierarchies.* □

An important problem for such a hierarchy is whether it collapses, or degenerates, i.e., whether there is a number k such that $\mathcal{A} \subseteq H_k$. Again, many problems in computer science can be seen as a problem of rank-hierarchy degeneration. Recycling the previous example on P *vs.* NP, it can be shown that the polynomial time hierarchy collapses if and only if the rank hierarchy for a bounded minimalisation operator $\bar{\mu}$, under the function algebra [P; $\bar{\mu}$], collapses.

Example 2.13. The rank hierarchy for the primitive recursion operator under the function algebra $\mathrm{PRIM} = [z, s, u_i^n; c^m, p]$, $H_n \equiv H_n^{\mathrm{PRIM}, p}$, can be intuitively understood as the stratification of the primitive recursive functions by the number of nested *for loops* needed to compute each function. This hierarchy is known not to collapse (cf. [23]). □

Example 2.14. The function algebra $\mathrm{PRIM}^+ = [z, s, u_i^n, +; c^m, p]$, where $+$ represents the 2-ary sum, is an alternative function algebra for the class of primitive recursive functions. This algebra gives Kalmár's class of elementary functions, $\mathcal{E}$, at the second level of the rank hierarchy for primitive recursion. i.e., $\mathcal{E}$ is equal to $H_2 \equiv H_2^{\mathrm{PRIM}^+, p}$. □

Proposition 2.15. *Let $\mathcal{A} = [\mathcal{F}; \mathcal{O}]$ be an enumerable function algebra, let $\tilde{\mathcal{O}}$ be a subset of $\mathcal{O}$, and set $\mathcal{V} = \mathcal{O} - \tilde{\mathcal{O}}$. The rank hierarchy for $\tilde{\mathcal{O}}$ can be inductively defined by:*

(i) $H_0 = [\mathcal{F}; \mathcal{V}]$,
(ii) $I_n = H_n \cup \{\tilde{O}(f_1, \ldots, f_k) : \tilde{O} \in \tilde{\mathcal{O}}$ *and* $f_1, \ldots, f_k \in H_n \cap Dom(\tilde{O})\}$, *and*
(iii) $H_{n+1} = [I_n; \mathcal{V}]$.

The previous proposition better illustrates the idea of the rank hierarchy: the next level of the hierarchy is obtained by allowing one further application of the operators in the operator set. We will skip the proof, which is obtained by a simple induction.

There are many typical problems of interest to a function algebra. We have already mentioned the problem of closure and the problem of collapse. Another important problem is the problem of universality. In the following definition and in the later sections, we use $\simeq$ to denote equality whenever both sides of the equation may occur undefined; $f(\boldsymbol{x}) \simeq g(\boldsymbol{y})$ means that f and g are both either undefined, or defined and equal, respectively for $\boldsymbol{x}$ and $\boldsymbol{y}$.

Definition 2.16. A binary function $f : D \times D \to R$ in an enumerable function algebra $\mathcal{A}$ is said to be **universal** if there is an effective procedure which, given a good description d of a function $g : D \to R$ in $\mathcal{A}$, will construct an element $\tilde{d} \in D$ such that, for every $x \in D, f(\tilde{d}, x) \simeq g(x)$.[3]

It is thus important, in the study of a function algebra, to know if it has a universal function. The last problem which we will refer that generally concerns function algebras, is the problem of alternative characterisation. In order to understand the function algebra more clearly, it is important to find alternative ways to obtain the same class of functions, or discover that previously known classes of functions are exactly given by a function algebra.

Example 2.17. As we have seen, Meyer and Ritchie have solved the problem of collapse. It is interesting to note that the class of primitive recursive functions does not have a universal function: a simple argument based on the second recursion theorem would show that if this was the case, then PRIM would be closed under the minimalisation operator, which we know is false. An alternative characterisation can be obtain by substituting the recursion operator by an iteration operator, and adding a few basic functions [13]. □

Example 2.18. To realise that the collapse of a rank hierarchy may depend on the specific function algebra, and not only on the relevant operator, consider the class of partial recursive functions. If we take μ to stand for the unbounded minimalisation operator, the class of partial recursive functions, PREC, can be given by $\mathrm{PREC} = [z, s, u_i^n; c, p, \mu]$. Kleene's normal form theorem implies that any partial recursive function can be obtained using a fixed number of primitive recursions, and so, despite the fact that the rank hierarchy for p does not collapse under PRIM, the same rank hierarchy *does* collapse under the function algebra PREC. □

We will below study a specific class of real-valued vector functions, which we call real recursive functions. This class is an analogue of Kleene's partial recursive functions, and it was first conceived, in a primitive form, by Cris Moore [24]. We will give a complete definition of this class in Section 2.3, using a function algebra. In the following section, however, we will review some basic properties of ordinary differential equations.

[3] Notice that this definition is only suitable given certain assumptions on the nature of the functions in $\mathcal{A}$.

2.2. Weak conditions for existence and uniqueness

In the next subsection we will discuss a partial operator, called the differential recursion operator, which will give us solutions to very simple differential equations. It is thus in our interest to study conditions for existence and uniqueness of such solutions. Let f denote a *total* function from $\mathbb{R}^{n+1} \to \mathbb{R}^n$, and J represent a (possibly unbounded) open interval (A, B) of the real line, with $t_0 \in J$. Consider the Cauchy problem of the form

$$g(t_0) = \boldsymbol{g_0} \quad \partial_t g(t) = f(t, g(t)). \tag{1}$$

A **solution** of (1) on J is a continuous function $g : \mathbb{R} \to \mathbb{R}^n$ which takes the value $\boldsymbol{g_0}$ at $t = t_0$ and satisfies the differential equation for every t in J. We will show that a few straightforward properties of f will ensure that the solution of (1) exists and is unique. The proofs will require some basic knowledge of Banach spaces; the unfamiliarised reader will find a good reference in [20].

Definition 2.19 (*see, e.g, [33,35]*). A total function $f : \mathbb{R}^m \to \mathbb{R}^n$ is called **locally Lipschitz** if for every compact set $C \subset \mathbb{R}^m$ there is a constant K such that all $\boldsymbol{x}, \boldsymbol{y} \in C$ verify the Lipschitz condition

$$\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \leqslant K\|\boldsymbol{x} - \boldsymbol{y}\|. \tag{2}$$

The smallest such K is called the **Lipschitz constant** of f for C.

The local Lipschitz property implies other weaker properties, such as continuity. In fact, letting $B(\boldsymbol{x}, r)$ (or $\overline{B}(\boldsymbol{x}, r)$) denote the open (resp., closed) ball of radius r around $\boldsymbol{x}$, should we take an arbitrary $\varepsilon > 0$ and point $\boldsymbol{x} \in \mathbb{R}^m$, and let K satisfy (2) for the compact $C = \bar{B}(\boldsymbol{x}, 1)$; then $\|\boldsymbol{x} - \boldsymbol{y}\| < \frac{\varepsilon}{K}$ implies that $\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| < K\|\boldsymbol{x} - \boldsymbol{y}\| < \varepsilon$; so f is continuous. This continuity ultimately implies that the concept of Lipschitz constant is well-defined.

The name *locally Lipschitz* is motivated because a total function $f : \mathbb{R}^m \to \mathbb{R}^n$ is locally Lipschitz if and only if around every point $\boldsymbol{z} \in \mathbb{R}^m$ there is a neighbourhood V of $\boldsymbol{z}$ and a constant K such that all $\boldsymbol{x}, \boldsymbol{y} \in V$ satisfy (2). Clearly, if F is locally Lipschitz, then we may take $V = \overline{B}(\boldsymbol{z}, 1)$ as this neighbourhood. Now suppose that around every point $\boldsymbol{z} \in \mathbb{R}^m$ there is a neighbourhood $V_{\boldsymbol{z}}$ of $\boldsymbol{z}$ where (2) is satisfied — this implies that f is continuous, by a similar argument to the above paragraph. Now suppose that f is not locally Lipschitz, and take a compact set $C \subset \mathbb{R}^m$, and two sequences $\boldsymbol{x}_i, \boldsymbol{y}_i$ in C such that

$$\|f(\boldsymbol{x}_i) - f(\boldsymbol{y}_i)\| > 2^i\|\boldsymbol{x}_i - \boldsymbol{y}_i\|. \tag{3}$$

By the compactness of C, further suppose that $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ converge as $i \to \infty$, respectively to $\boldsymbol{x}$ and $\boldsymbol{y}$. Then the continuity of f implies that $f(\boldsymbol{x}_i) \to f(\boldsymbol{x})$ and $f(\boldsymbol{y}_i) \to f(\boldsymbol{y})$, and so

$$\|\boldsymbol{x}_i - \boldsymbol{y}_i\| < 2^{-i}\|f(\boldsymbol{x}_i) - f(\boldsymbol{y}_i)\| \to 0;$$

which means $\boldsymbol{x} = \boldsymbol{y}$. But then for every large enough i both $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ will be in the corresponding neighbourhood $V_{\boldsymbol{x}}$ of $\boldsymbol{x}$, and because $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ satisfy (3), then (2) cannot be satisfied in $V_{\boldsymbol{x}}$, a contradiction.

The following simpler conditions imply the local Lipschitz property.

Theorem 2.20. *Let $f : \mathbb{R}^m \to \mathbb{R}^n$ be a total function.*

(a) *The function f is locally Lipschitz if and only if every closed ball verifies (2).*
(b) *The function f is locally Lipschitz if and only if every closed m-cube verifies (2).*
(c) *If f is everywhere differentiable and $\mathrm{D}f$ is bounded in every compact set, then f is locally Lipschitz.*
(d) *If f is continuously differentiable, then it is locally Lipschitz.*

Proof. (a) The necessity is obvious, since a closed ball is compact. Now take any compact set C. Since C is bounded, some closed ball will contain C, and the constant K which ensures that (2) is satisfied in this closed ball will trivially suffice to ensure that (2) is also satisfied for C. The equivalence (b) is proved similarly.

(c) Suppose that f is differentiable, and $\mathrm{D}f$ is bounded in every compact set. Let C be an arbitrary compact set. Recall that, for any $\boldsymbol{x} \in \mathbb{R}^m$, $\mathrm{D}f(\boldsymbol{x}) \in \mathcal{L}(\mathbb{R}^m \to \mathbb{R}^n)$ is a bounded linear operator, and its norm is given by

$$\|\mathrm{D}f(\boldsymbol{x})\| = \sup_{\boldsymbol{y} \in \mathbb{R}^m} \frac{\|\mathrm{D}f(\boldsymbol{x})(\boldsymbol{y})\|}{\|y\|}.$$

Then let M_1 be a bound for $\|\mathrm{D}f\|$ in C, which exists by hypothesis.

Set $s : \mathbb{R}^m \times \mathbb{R}^m \to [0, +\infty)$ so that

$$s(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \dfrac{\|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \mathrm{D}f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})\|}{\|\boldsymbol{y} - \boldsymbol{x}\|} & \text{if } \boldsymbol{x} \neq \boldsymbol{y} \\ 0 & \text{otherwise.} \end{cases}$$

Then s is bounded in $C \times C$, because f is differentiable and $\mathrm{D}f$ is bounded in C. So let M_2 be an upper bound for s in $C \times C$. Thus,

$$\|f(\boldsymbol{y}) - f(\boldsymbol{x})\| \leqslant \|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \mathrm{D}f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})\| + \|\mathrm{D}f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})\| \leqslant (M_2 + M_1)\|\boldsymbol{y}-\boldsymbol{x}\|,$$

and f obeys the Lipschitz condition in C. By the arbitrary nature of C, f must be locally Lipschitz.

(d) If f is continuously differentiable, then Df is continuous and therefore bounded on any compact set: (c) gives us the result. □

Henceforth, in this section, f will always designate a total function.

Theorem 2.21 (*Picard*). *If f is locally Lipschitz, then there exists a solution g of* (1) *in a neighbourhood of t_0.*

Proof. Let $I = [t_0 - a, t_0 + a]$ for some $a > 0$, and let $R \subset \mathbb{R}^n$ be the closed n-cube centred on $\boldsymbol{g}_0$ with side of length $b > 0$. Since f is continuous and $I \times R$ is compact, let M be the maximum of f over $I \times R$, and let K be its Lipschitz constant for $I \times R$. We choose some $0 < \varepsilon < \min\{a, \frac{b}{M}, \frac{1}{K}\}$ and show that we may obtain a solution g of (1) in the interval $J = [t - \varepsilon, t + \varepsilon]$. Use the letter $\mathcal{C}$ to denote the space of continuous functions from J to R, under the supremum norm:

$$\|f\| = \sup_{t \in I} \|f(t)\|.$$

$\mathcal{C}$ is a Banach space, and so by the theorem of Banach, any strict contraction operator has a fixed point. We show that the operator T, given by

$$(Tg)(t) = g_0 + \int_{t_0}^{t} f(s, g(s))\mathrm{d}s,$$

is a strict contraction on $\mathcal{C}$, for the constant $k = \varepsilon K < 1$. See that if $g \in \mathcal{C}$, then $Tg \in \mathcal{C}$, since for any $t \in J$ we have $(Tg)(t) \in R$:

$$\|(Tg)(t) - \boldsymbol{g}_0\| \leqslant \|g(t_0) - g_0\| + \left| \int_{t_0}^{t} \|f(s, g(s))\|\mathrm{d}s \right| \leqslant \varepsilon M < b.$$

Additionally, T is a strict contraction, because

$$\|Tg - Th\| \leqslant \sup_{t \in I} \int_{t_0}^{t} \|f(s, g(s)) - f(s, h(s))\|\mathrm{d}s \leqslant \varepsilon K\|g - h\| = k\|g - h\|.$$

We conclude that if we let g denote the fixed point of T, then g is a solution of (1) on the interval J. □

We will make use of the following uniqueness theorem in the remaining sections.

Theorem 2.22. *If f is locally Lipschitz, and g is a solution of* (1) *on the interval J, then g is the unique solution of* (1) *on J.*

Proof. Suppose that there were two solutions g and $\tilde{g}$ to (1) on the interval J. Set $h(t) = \tilde{g}(t) - g(t)$. Let $t_0 \in [a, b] \subset J$ and let $C \subset R^n$ be an arbitrary compact set such that $g([a, b]) \subset C$ and $\tilde{g}([a, b]) \subset C$. Let K be the Lipschitz constant for f in $[a, b] \times C$, according to (2).

Clearly we have $h(t_0) = 0$, because both $g(t_0) = \tilde{g}(t_0) = \boldsymbol{g_0}$. If we denote the scalar product with $\cdot$, then, for every $t \in [a, b]$,

$$\partial_t \|h(t)\|^2 = 2h(t) \cdot \partial_t h(t) = 2(\tilde{g}(t) - g(t)) \cdot (f(t, \tilde{g}(t)) - f(t, g(t))).$$

From (2) and the Cauchy–Schwarz inequality, we obtain

$$|(\tilde{g}(t) - g(t)) \cdot (f(t, \tilde{g}(t)) - f(t, g(t)))| \leqslant K\|\tilde{g}(t) - g(t)\| \times \|\tilde{g}(t) - g(t)\| = K\|\tilde{g}(t) - g(t))\|^2.$$

This gives us $\partial_t \|h(t)\|^2 \leqslant 2K\|h(t)\|^2$, and so we must conclude that

$$\partial_t \left(\|h(t)\|^2 \mathrm{e}^{-2Kt}\right) = (\partial_t \|h(t)\|^2)\mathrm{e}^{-2Kt} - 2K\|h(t)\|^2 \mathrm{e}^{-2Kt} \leqslant 0.$$

So we see that $\|h(t)\|^2 \mathrm{e}^{-2Kt}$ does not increase on $[a, b]$, and since $\|h(0)\| = 0$, then it follows that $\|h(t)\| = 0$ for every $t \in [a, b]$, i.e., $g = \tilde{g}$ on $[a, b]$. The proof is done for an arbitrary compact set C such that $g([a, b]) \subset C$ and $\tilde{g}([a, b]) \subset C$. Now, should we take

$$C = \{\boldsymbol{y} : \|\boldsymbol{y}\| \leqslant \max_{t \in [a,b]} \|g(t)\| + \|\tilde{g}(t)\|\},$$

then C is well-defined by the continuity of g and $\tilde{g}$, and will be compact. Also, C contains $g([a, b])$ and $\tilde{g}([a, b])$. We then conclude by the previous argument that g and $\tilde{g}$ are equal on $[a, b]$, and everywhere on J, since a and b are arbitrary. □

The uniqueness theorem ensures immediately the following.

Corollary 2.23. *If f is locally Lipschitz, and $g, \tilde{g}$ are two solutions of* (1) *respectively in the intervals $J, \tilde{J}$, then $g = \tilde{g}$ on $J \cap \tilde{J}$.*

For some locally Lipschitz function f, let S denote the set of solutions of (1). We set A to be the infimum of $Dom(g)$ for every $g \in S$, and B will denote the supremum of $Dom(g)$ over every $g \in S$. The previous corollary provides that the following concept is well-defined.

Definition 2.24. Let f be locally Lipschitz. Then the **maximal solution** of (1) is the function g defined in (A, B) such that if $\tilde{g}$ is a solution of (1) over some interval (a, b), then $g(t) = \tilde{g}(t)$ for all $t \in (a, b)$. The interval (A, B) is called the **maximal interval** of (1).

It will be in our interest to show that the solutions of (1) are well-behaved in A and B.

Theorem 2.25. *Let f be locally Lipschitz, and let g denote the maximal solution of (1), defined on $J = (A, B)$. Then $B < +\infty$ (or $A > -\infty$) if and only if $\lim_{t\to\tilde{t}^-} \|g(t)\| = +\infty$ (resp. $\lim_{t\to\tilde{t}^+} \|g(t)\| = +\infty$) for some finite $\tilde{t} \in \mathbb{R}$, and in this case we have $B = \tilde{t}$ (resp. $A = \tilde{t}$).*

Proof. If $\lim_{t\to\tilde{t}^-} \|g(t)\| = +\infty$ for some finite $\tilde{t}$, then g cannot respect the equality (1) at point $\tilde{t}$, and thus $\tilde{t} = B < +\infty$. For the converse, suppose that $B < +\infty$, but that $\lim_{t\to B^-} g(t) = \boldsymbol{g}_1 \in \mathbb{R}^n$. Then the Cauchy problem

$$\tilde{g}(B) = \boldsymbol{g}_1 \quad \partial_y \tilde{g}(t) = f(t, \tilde{g}(t)) \tag{4}$$

would have a solution $\tilde{g}$ on some neighbourhood $[B - \varepsilon, B + \varepsilon]$ of B, by Theorem 2.21. But then, the function $\hat{g}$ given by $\hat{g}(t) = g(t)$ for $t < B$, $\hat{g}(t) = \tilde{g}(t)$ for $B \leqslant t < B + \varepsilon$ testifies that g is not the maximal solution. By contradiction, we conclude that either $B = +\infty$, or that $\lim_{t\to B^-} g(t)$ is undefined.

But it cannot be the case that $\lim_{t\to B^-} g(t)$ is undefined, while $\|g(t)\|$ is bounded for t sufficiently close to B, as we will now show, again by *reductio ad absurdum*. Suppose that these were the case, i.e., let $\tilde{J} = (t_0, B)$ be an open interval where g is bounded in the norm, and suppose that for some sequence t_i in this interval we would have $t_i \to B$ as $i \to \infty$, but $g(t_i)$ would diverge. By the boundedness of $\tilde{J}$ and of $\|g(t)\|$ in $\tilde{J}$, take some compact interval C containing $\tilde{J} \times g(\tilde{J})$, and let $M = \max_{(t,\boldsymbol{z})\in C} \|f(t, \boldsymbol{z})\|$. M is well-defined by continuity of f and compactness of C. Choose some sub-sequence $\tilde{t}_i$ of t_i such that

$$\sum_{i=1}^{\infty} |\tilde{t}_{i+1} - \tilde{t}_i| < +\infty, \tag{5}$$

(the existence of which is ensured by convergence of t_i), and also such that

$$\sum_{i=1}^{\infty} \|g(\tilde{t}_{i+1}) - g(\tilde{t}_i)\| = +\infty; \tag{6}$$

(this is ensured by the divergence of $g(t_i)$). The mean value theorem tells us that

$$\|g(\tilde{t}_{i+1}) - g(\tilde{t}_i)\| \leqslant M|\tilde{t}_{i+1} - \tilde{t}_i|.$$

But taking the sum with the proper indexes, this contradicts (5) and (6). So we conclude that divergence of $g(t_i)$ implies unboundedness of $\|g(t_i)\|$, and so $\lim_{t\to B^-} \|g(t)\| = +\infty$. We proceed in a similar way for A. □

2.3. The class of real recursive functions

We are now prepared to describe our function algebra. We take $\mathscr{F}$ to be the class of partial, vector-valued, multiple-argument functions over $\mathbb{R}$, i.e., the class of partial functions $f : \mathbb{R}^m \to \mathbb{R}^n$ for some $m, n \in \mathbb{N}$. We accept functions of arity 0, and call them **constants**, or **values**.

There will be two kinds of basic functions: the constant functions, and the projections. The constant functions are denoted -1^n, 0^n, and 1^n, for every $n = 0, 1, 2, \ldots$, and are given by $-1^n(x_1, \ldots, x_n) = -1$, $0^n(x_1, \ldots, x_n) = 0$ and $1^n(x_1, \ldots, x_n) = 1$. The projections are denoted by U_i^n, for each $n = 1, 2, \ldots$ and $1 \leqslant i \leqslant n$; they are given by $\mathrm{U}_i^n(x_1, \ldots, x_n) = x_i$.

The class will be closed under a countable number of partial operators over $\mathscr{F}$. The first operator is the **composition operator**, denoted by **C**. Given two functions $f : \mathbb{R}^k \to \mathbb{R}^n$ and $g : \mathbb{R}^m \to \mathbb{R}^k$, the function $\mathbf{C}(f, g)$ goes from $\mathbb{R}^m$ to $\mathbb{R}^n$, and is given by

$$\mathbf{C}(f, g)(\boldsymbol{x}) = f(g(\boldsymbol{x})), \quad \text{for every } \boldsymbol{x} \in \mathbb{R}^m.$$

The domain of $\mathbf{C}(f, g)$ is $Dom(\mathbf{C}(f, g)) = \{\boldsymbol{x} \in Dom(g) : g(\boldsymbol{x}) \in Dom(f)\}$.

Our second operator is the **differential recursion operator**, denoted with **R**. Let $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ be a total locally Lipschitz function. Consider, for a fixed $\boldsymbol{x} \in \mathbb{R}^m$, the Cauchy problem

$$g(\boldsymbol{x}, 0) = \boldsymbol{x} \quad \partial_t g(\boldsymbol{x}, t) = f(t, g(\boldsymbol{x}, t)). \tag{7}$$

Then $\mathbf{R}(f)$ is a function from $\mathbb{R}^{n+1}$ to $\mathbb{R}^n$ such that for every fixed $\boldsymbol{x} \in \mathbb{R}^n$, $\mathbf{R}(f)(\boldsymbol{x}, t) = g(\boldsymbol{x}, t)$, where $g(\boldsymbol{x}, \cdot)$ is the maximal solution of (7). The domain of $\mathbf{R}(f)$ is $Dom(\mathbf{R}(f)) = \{(\boldsymbol{x}, t) : \boldsymbol{x} \in \mathbb{R}^m, A(\boldsymbol{x}) < t < B(\boldsymbol{x})\}$, where A, B give the extrema of the maximal interval. In the next section we will show that the concept of differential recursion is well-founded, and provide a few examples.

Following this we have the **infinite supremum limit operator**, denoted by **Ls**. This operator takes any function $f : \mathbb{R}^{m+1} \to \mathbb{R}^n$, and maps it into the component-wise infinite supremum limit, i.e., for every $i = 1, \dots, n$,

$$(\mathbf{Ls}(f)(\boldsymbol{x}))_i = \limsup_{y\to\infty} (f(\boldsymbol{x}, y))_i \,.$$

For the sake of abbreviation, we write simply

$$\mathbf{Ls}(f)(\boldsymbol{x}) = \limsup_{y\to\infty} f(\boldsymbol{x}, y).$$

Then $Dom(\mathbf{Ls}(f)) = \{\boldsymbol{x} \in \mathbb{R}^m : \limsup_{y\to\infty} f(\boldsymbol{x}, y) \text{ exists}\}$. We will further discuss the **Ls** operator in Section 2.5.

The final operator is called the **aggregation operator**, denoted by the symbol **V**. The aggregation operator takes two functions $f : \mathbb{R}^m \to \mathbb{R}^k$ and $g : \mathbb{R}^m \to \mathbb{R}^n$ and joins them into a single vector function $\mathbf{V}(f, g) : \mathbb{R}^m \to \mathbb{R}^{k+n}$. As expected, this is given by

$$\mathbf{V}(f, g)(\boldsymbol{x}) = (f(\boldsymbol{x}), g(\boldsymbol{x})),$$

and $Dom(\mathbf{V}(f, g)) = Dom(f) \cap Dom(g)$.

We end this subsection with the central definition of this text.

Definition 2.26. The class of real recursive functions, denoted by REC($\mathbb{R}$), is given by the function algebra

$$\mathrm{REC}(\mathbb{R}) = [-1^n, 0^n, 1^n, \mathrm{U}_i^n; \mathbf{C}, \mathbf{R}, \mathbf{Ls}, \mathbf{V}].$$

We will use the letter Η (capital eta — not to be confused with the Latin capital h: $H \neq$ Η) to denote REC($\mathbb{R}$) given by this specific structure. We will, below, give alternative characterisations of REC($\mathbb{R}$), using other algebraic structures; these will also be denoted by capital Greek letters.

2.4. More on differential recursion

The operator of differential recursion is an attempt to mimic the functioning of an idealised disk-and-wheel integrator. Such integrators have been invented in the nineteenth century by Lord Kelvin [cf. [36]], and have been used to implement the famous differential analyser of Vannevar Bush [6]. To those who have studied real recursive functions, this has always given a certain sensation of security: the well-foundedness and good-behaviour of the differential recursion operator were freely assumed.

Recently, however, the well-foundedness of differential recursion was no longer a matter of consensual agreement. Authors, such as Akitoshi Kawamura [19], and referees of ours, have remarked that differential recursion needs a more thorough and precise treatment. The paper [19] provides a successful attempt, with a much stronger differential recursion operator than that shown above. In this paper, we have opted to limit the scope of the differential recursion operator. As stated in the previous section, $\mathbf{R}(f)$ will only be defined when f is a total locally Lipschitz function. This will ensure that whenever $f \in Dom(\mathbf{R})$, the solution $\mathbf{R}(f)$ exists (Theorem 2.21), is unique (Theorem 2.22), and has a good-behaviour in the extremities of its domain of definition (Theorem 2.25). We then see that the operator **R** is well-defined and well-behaved.

We *could* have imposed weaker conditions and obtain similar properties, but in our every attempt these would become too technical and complex. Local Lipschitz conditions, on the other hand, are considered in any standard text on ordinary differential equations [e.g. 12,34,35]. Furthermore, we have a few simple ways to assess whether f will be in the domain of **R** (Theorem 2.20).

The definition of solution ensures us that $\mathbf{R}(f)$ will always be continuously differentiable in the last variable, i.e., $\mathbf{R}(f)(\boldsymbol{x}, \cdot)$ is continuously differentiable for every fixed $\boldsymbol{x}$. We show the following stronger result.

Theorem 2.27. *For any total locally Lipschitz $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$, $g = \mathbf{R}(f)$ is locally Lipschitz in its domain.*

By *locally Lipschitz in its domain*, we mean that any compact set $C \subset Dom(g)$ must obey the Lipschitz condition. The proof will require the use of the following lemma.

Lemma 2.28 (*Gronwall–Reid*). *Let C be a given constant and $k : J \to [0, +\infty)$ a given non-negative continuous function on an interval $J \subseteq \mathbb{R}$ with $t_0 \in J$. Then, if $v : J \to [0, +\infty)$ is continuous and, for all $t \in J$,*

$$v(t) \leqslant C + \left| \int_{t_0}^{t} k(s)v(s)\mathrm{d}s \right| ; \qquad (8)$$

it follows that, again for all $t \in J$,

$$v(t) \leqslant C \exp\left(\left| \int_{t_0}^{t} k(s)\mathrm{d}s \right| \right) .$$

Proof. With $t \geqslant t_0$, the inequality (8) implies that

$$k(t)v(t) - k(t)\left(C + \int_{t_0}^{t} k(s)v(s)\mathrm{d}s\right) \leqslant 0.$$

Taking $Q(t) = C + \int_{t_0}^{t} k(s)v(s)\mathrm{d}s$, we rewrite this as $\partial_t Q(t) - k(t)Q(t) \leqslant 0$. Multiplying by $\exp(-\int_{t_0}^{t} k(s)\mathrm{d}s)$, we arrive at

$$\partial_t\left(Q(t)\exp\left(-\int_{t_0}^{t} k(s)\mathrm{d}s\right)\right) \leqslant 0.$$

Seeing that $Q(t_0) = C$, and integrating this last inequality from t_0 to t, we get

$$Q(t)\exp\left(-\int_{t_0}^{t} k(s)\mathrm{d}s\right) - C \leqslant 0 \iff Q(t) \leqslant C\exp\left(\int_{t_0}^{t} k(s)\mathrm{d}s\right).$$

From (8) and the definition of $Q(t)$, we obtain the intended result. For $t < t_0$, the result is proved similarly. □

Proof of Theorem 2.27. Choose an arbitrary $(\boldsymbol{x}, u) \in Dom(g)$. Let $J \subset Dom(g(\boldsymbol{x}, \cdot))$ be a closed interval with diameter d_J containing 0 and u as interior points. Use $S_{\boldsymbol{x}}$ to denote the unit neighbourhood of the graph $(t, g(\boldsymbol{x}, t))$ when t ranges over J:

$$S_{\boldsymbol{x}} = \{(t, \boldsymbol{z}) : t \in J, \|\boldsymbol{z} - g(\boldsymbol{x}, t)\| \leqslant 1\}.$$

By the continuity of g in its last variable, $S_{\boldsymbol{x}}$ is compact, and so f has a Lipschitz constant in $S_{\boldsymbol{x}}$, say $K_{\boldsymbol{x}}$. Suppose that for some $(\boldsymbol{y}, v) \in Dom(g)$, $0 < \delta < 1$, we have $\|\boldsymbol{y} - \boldsymbol{x}\| + |v - u| < \delta$, and that δ is small enough so that $v \in J$. Then for any $t \leqslant v$, $(\boldsymbol{y}, t)$ is in $Dom(g)$, and

$$\|g(\boldsymbol{y}, t) - g(\boldsymbol{x}, t)\| \leqslant \|\boldsymbol{y} - \boldsymbol{x}\| + \left|\int_0^t \|f(s, g(\boldsymbol{y}, s)) - f(s, g(\boldsymbol{x}, s))\|\mathrm{d}s\right|.$$

As long as $g(\boldsymbol{y}, t)$ remains on $S_{\boldsymbol{x}}$, we have, by the Gronwall–Reid Lemma,

$$\|g(\boldsymbol{y}, t) - g(\boldsymbol{x}, t)\| \leqslant \delta + \left|\int_0^t K_{\boldsymbol{x}}\|g(\boldsymbol{y}, s) - g(\boldsymbol{x}, s)\|\mathrm{d}s\right| \leqslant \delta\exp(K_{\boldsymbol{x}}|t|).$$

Now, $g(\boldsymbol{y}, t)$ is a continuous function of t, and $g(\boldsymbol{y}, 0) \in S_{\boldsymbol{x}}$. So $g(\boldsymbol{y}, t)$ must be in $S_{\boldsymbol{x}}$ for some open interval. So if we choose $\delta < \exp(-K_{\boldsymbol{x}}d_J)$, we must conclude by this last equation that $g(\boldsymbol{y}, t)$ remains in $S_{\boldsymbol{x}}$ for all $t \in J$, $t \leqslant v$. Furthermore, if for any given $\varepsilon > 0$ we set $\delta < \min(\varepsilon/2, 1)\exp(-K_{\boldsymbol{x}}d_J)$, then we get

$$\|g(\boldsymbol{y}, t) - g(\boldsymbol{x}, t)\| < \frac{\varepsilon}{2} \text{ for all } t \in J, t \leqslant v.$$

If, furthermore, $\delta < \frac{\log 2}{K_{\boldsymbol{x}}}$, then using the triangle inequality and the Gronwall–Reid Lemma,

$$\|g(\boldsymbol{y}, v) - g(\boldsymbol{x}, u)\| \leqslant \frac{\epsilon}{2} + \left|\int_u^v K_{\boldsymbol{x}}\|g(\boldsymbol{x}, s) - g(\boldsymbol{x}, s)\|\mathrm{d}s\right| \leqslant \frac{\varepsilon}{2}\exp(K_{\boldsymbol{x}}|v - u|) < \frac{\varepsilon}{2}\exp(K_{\boldsymbol{x}}\delta) < \varepsilon.$$

With this, we have proved that g is continuous in its domain. So let $C \subset Dom(g) \subseteq \mathbb{R}^{n+1}$ be a compact set; let J be a closed interval containing every t such that $(\boldsymbol{x}, t) \in C$. Because g is continuous, $D = J \times g(C)$ is compact, and so let K_f be the Lipschitz constant of f in D, and M_f be the maximum of $\|f\|$ in D. We may repeat the previous argument using the Lipschitz constant K_f which does not depend on $\boldsymbol{x}$. For any $(\boldsymbol{x}, u), (\boldsymbol{y}, v) \in C$, and assuming without loss of generality that $v \leqslant u$, we find

$$\|g(\boldsymbol{y}, v) - g(\boldsymbol{x}, v)\| \leqslant \|\boldsymbol{y} - \boldsymbol{x}\| + \left|\int_0^v K_f\|g(\boldsymbol{y}, s) - g(\boldsymbol{x}, s)\|\mathrm{d}s\right| \leqslant \|\boldsymbol{y} - \boldsymbol{x}\|\exp(K_f d_J).$$

Also,

$$\|g(\boldsymbol{x}, v) - g(\boldsymbol{x}, u)\| \leqslant \left|\int_v^u \|f(s, g(\boldsymbol{x}, s))\|\mathrm{d}s\right| \leqslant |u - v|M_f.$$

And so $K_g = 2(\exp(K_f d_J) + M_f)$ bounds the Lipschitz constant of g for C, because for our arbitrary $(\boldsymbol{x}, u), (\boldsymbol{y}, v) \in C$, by the triangle inequality,

$$\|g(\boldsymbol{y}, v) - g(\boldsymbol{x}, u)\| \leqslant \|\boldsymbol{y} - \boldsymbol{x}\|\exp(K_f d_J) + |u - v|M_f \leqslant \|(\boldsymbol{y}, u) - (\boldsymbol{x}, v)\|K_g. \quad \square$$

We conclude that the values of the solutions have a continuous dependence on the initial condition. At first sight, it might appear that we would prefer a more general differential recursion, where f would be allowed to depend on $\boldsymbol{x}$, or where the initial condition can be given at any point t_0. However, we will show in the next section that the Cauchy problem

$$g(\boldsymbol{x}, t_0) = g_0(\boldsymbol{x}) \quad \partial_t g(\boldsymbol{x}, t) = f(t, g(\boldsymbol{x}, t), \boldsymbol{x}) \tag{9}$$

can be reduced to the form (7).

A trivial example is the exponential function.

Example 2.29. The exponential function, exp, is trivially given by the differential recursion

$$\exp(0) = 1 \quad \partial_t \exp(t) = \exp(t).$$

In the notation of (9), we have $g_0 = 1$ and $f(t, x) = x$. □

We provide another simple example to give sin and cos, which is nevertheless appropriate to show the expressive power of differential recursion. In the next section, we will see that every function needed for the example is real recursive, and conclude that sin and cos are real recursive.

Example 2.30. Consider the differential recursion schema

$$g(0) = (0, 1) \qquad \partial_t g(t) = (g_2(t), -g_1(t)).$$

With the notation of (9), we have $g_0 = (0, 1)$ and $f(t, \boldsymbol{z}) = ((\boldsymbol{z})_2, -(\boldsymbol{z})_1)$. Easily, $f \in Dom(\mathbf{R})$. The solution can be recognised as $g = (\sin, \cos)$. □

2.5. More on the supremum limit operator

The class $\mathrm{REC}(\mathbb{R})$ is a subset of $\mathscr{F}$, i.e., it is composed of partial, multiple-argument vector functions over $\mathbb{R}$. This may cause confusion, because it is not immediately obvious how the concept of infinite supremum limit applies to such functions. We will give a rigorous, yet simple characterisation of the concept.

Any partial function $f : \mathbb{R}^m \to \mathbb{R}^n$ can be uniquely identified with its **graph**. The graph of f, G_f, is a predicate over $\mathbb{R}^{m+n}$, given by:

$$G_f(\boldsymbol{x}, \boldsymbol{z}) \iff \boldsymbol{x} \in Dom(f) \quad \text{and} \quad \boldsymbol{z} = f(\boldsymbol{x}).$$

Given any function $f : \mathbb{R}^{m+1} \to \mathbb{R}^n$, and setting $g(\boldsymbol{x}, w) = \sup_{y>w} f(\boldsymbol{x}, y)$, the graph of g will be given by:

$$G_g(\boldsymbol{x}, w, \boldsymbol{z}) \iff (\forall y > w)\ \boldsymbol{z}_i \geqslant (f(\boldsymbol{x}, y))_i \wedge (\forall u_i < z_i)(\exists y > w)\ u_i < (f(\boldsymbol{x}, y))_i, \quad \text{for } i = 1, \ldots, m.$$

In words, $\boldsymbol{z}$ is the supremum of $f(\boldsymbol{x}, \cdot)$ for $y > w$ if it is, component-wise, the least upper bound of $f(\boldsymbol{x}, y)$ for every $y > w$. We could further extend this symbolic expression to explicitly show its dependence on G_f:

$$\begin{gathered} G_g(\boldsymbol{x}, w, \boldsymbol{z}) \\ \iff \\ \underline{(\forall y > w)\exists \boldsymbol{v}}\ \Big[\underline{G_f(\boldsymbol{x}, y, \boldsymbol{v})} \wedge \boldsymbol{z}_i \geqslant \boldsymbol{v}_i\Big] \wedge (\forall u_i < \boldsymbol{z}_i)(\exists y > w)\exists \boldsymbol{v}\ \left[G_f(\boldsymbol{x}, y, \boldsymbol{v}) \wedge u_i < \boldsymbol{v}_i\right], \quad \text{for } i = 1, \ldots, m. \end{gathered} \tag{10}$$

See that G_g is still the graph of a function, for if some $(\boldsymbol{x}, w, \boldsymbol{z})$, $G_g(\boldsymbol{x}, w, \boldsymbol{v})$, then $\boldsymbol{z} = \boldsymbol{v}$. Now, we have that

$$(\boldsymbol{x}, w) \in Dom(g) \iff \exists \boldsymbol{z}\ G_g(\boldsymbol{x}, w, \boldsymbol{z}). \tag{11}$$

From (10) and (11), considering most especially the underlined part of (10), we may see that if for some particular $(\boldsymbol{x}, w)$, we have $(\boldsymbol{x}, y) \notin Dom(f)$ for some $y > w$, then $(\boldsymbol{x}, w) \notin Dom(g)$. Thus, in order for $g(\boldsymbol{x}, w)$ to be defined, $f(\boldsymbol{x}, y)$ will have to be defined for every $y > w$. If $f(\boldsymbol{x}, y)$ is defined for all $y > w$, for some $(\boldsymbol{x}, w)$, then the predicate (10) gives the component-wise supremum.

The same will apply to the supremum limit. Given $f : \mathbb{R}^{m+1} \to \mathbb{R}^n$, setting $h = \mathbf{Ls}f$ (i.e., $h(\boldsymbol{x}) = \limsup_{y\to\infty} f(\boldsymbol{x}, y)$), we get

$$G_h(\boldsymbol{x}, \boldsymbol{z}) \iff \boldsymbol{z} = \lim_{w\to+\infty} \sup_{y>w} f(\boldsymbol{x}, y) = \lim_{w\to+\infty} g(\boldsymbol{x}, w).$$

Making the dependence on G_g explicit, we find

$$G_h(\boldsymbol{x}, \boldsymbol{z}) \iff (\forall \varepsilon > 0)(\exists \tilde{w} > 0)(\forall w > \tilde{w})\underline{\exists \boldsymbol{v}\ G_g(\boldsymbol{x}, w, \boldsymbol{v})} \wedge \|\boldsymbol{v} - \boldsymbol{z}\| < \varepsilon.$$

As we have seen, the underlined sub-predicate will not be valid unless $f(\boldsymbol{x}, y)$ is total for all $y > w$. Since w is universally quantified, we conclude the following.

Remark 2.31. Take any fixed $\boldsymbol{x} \in \mathbb{R}^m$. If $f(\boldsymbol{x}, y)$ is undefined for arbitrarily large y, then $\mathbf{Ls}(f)(\boldsymbol{x})$ will be undefined, i.e., $\boldsymbol{x} \notin Dom(\mathbf{Ls}(f))$.

Furthermore, in order for the supremum limit to be defined, it is required that every one of its components is defined. The following two remarks are in order:

Remark 2.32. Take any fixed $\boldsymbol{x} \in \mathbb{R}^m$. If *at least one component* of $f(\boldsymbol{x}, y)$ is undefined for arbitrarily large y, then $\mathbf{Ls}(f)(\boldsymbol{x})$ will be undefined, i.e., $\boldsymbol{x} \notin Dom(\mathbf{Ls}(f))$.

Remark 2.33. Take any fixed $\boldsymbol{x} \in \mathbb{R}^m$. If *at least one of* $\limsup_{y\to\infty}(f(\boldsymbol{x}, y))_i$ is undefined, then $\mathbf{Ls}(f)(\boldsymbol{x})$ will also be undefined, i.e., $\boldsymbol{x} \notin Dom(\mathbf{Ls}(f))$.

The study we have made here also applies to the remaining operators. Regarding undefinedness and partiality, we use the same principle as classical recursion theory:

Strict undefinedness. *If a function is given an undefined parameter, or results in an undefined component, then the function will be undefined.*

E.g., $0 \times \bot = \bot$.

3. Basic theory

This section is devoted to the basic results of real recursive function theory. We will study the most elementary real recursive functions and operators in Section 3.1; in Section 3.2 we show a naturally-arising hierarchy of real recursive functions. More complex real recursive operators will be given in Section 3.3. Finally, in Section 3.4, we will consider the relationship between real recursive functions and partial recursive functionals.

3.1. A variety of real recursive functions and operators

We begin with the most basic operations over $\mathbb{R}$.

Proposition 3.1 ([24,27]). *The binary addition, subtraction and multiplication are real recursive.*

Proof. For addition, consider the following differential recursion scheme:

$$+(x,0)=x \quad \partial_y+(x,y)=1^2(y,+(x,y))=1.$$

Subtraction is obtained by replacing 1 with −1. For multiplication, set

$$g(x_1,x_2,0)=(x_1,x_2) \qquad \partial_y g(x_1,x_2,y)=\mathbf{V}(U_3^3,0^3)(t,g(x_1,x_2,t))=((g(x_1,x_2,y))_2,0).$$

Then $g(x_1,x_2,y)=(x_1+x_2y,x_2)$ is the solution, and so

$$\times(x,y)=\mathbf{C}(U_1^2,\mathbf{C}(g,\mathbf{V}(0^2,\mathbf{V}(U_1^2,U_2^2))))(x,y)=(g(0,x,y))_1. \quad \square$$

The expression for multiplication is not very simple, because we have not allowed ourselves to specify initial conditions that depend on the parameter $\boldsymbol{x}$. However, we said in the end of Section 2.4 that REC($\mathbb{R}$) is closed for a more general form of differential recursion. We will soon prove this rigorously, but just now we will continue with a few more very basic facts.

Proposition 3.2. *Take k scalar functions $f_1,\ldots,f_k:\mathbb{R}^m\to\mathbb{R}$. Then the function $F=(f_1,\ldots,f_k)$ is real recursive.*

Proof. Set $F=\mathbf{V}(f_1,\mathbf{V}(f_2,\ldots,\mathbf{V}(f_{k-1},f_k)\ldots))$. $\square$

This will allow us to simplify our notation. As a corollary we get:

Corollary 3.3. *For any $m,n\in\mathbb{N}$, the m-ary, n-component constants -1_n^m, 0_n^m, 1_n^m are real recursive.*

We could invent more similar assertions, such as the following two.

Proposition 3.4. *If $f:\mathbb{R}^2\to\mathbb{R}$ is real recursive, then so is $\tilde{f}:\mathbb{R}^2\to\mathbb{R}$, given by $\tilde{f}(x,y)=f(y,x)$.*

Proof. Take $\tilde{f}=\mathbf{C}(f,\mathbf{V}(U_2^2,U_1^2))$. $\square$

Proposition 3.5. *If $f:\mathbb{R}\to\mathbb{R}^2$ is real recursive, then so is $\tilde{f}:\mathbb{R}\to\mathbb{R}^2$, given by $\tilde{f}(x)=((f(x))_2,(f(x))_1)$.*

Proof. Take $\tilde{f}=\mathbf{C}(\mathbf{V}(U_2^2,U_1^2),f)$. $\square$

The main point, which we will not rigorously prove to avoid the tedious details, is that any fixed switching of components, or of the order of the arguments, or any selection of components, or a mixture of all of these things can be obtained in a straightforward way by using projections, composition and aggregation. We will take this for granted from this point forward. We may now prove the following, without excessive detail.

Proposition 3.6. *Let $t_0\in\mathbb{R}$ be a real recursive constant, let $g_0:\mathbb{R}^m\to\mathbb{R}^n$ be an arbitrary real recursive function, and let $f:\mathbb{R}^{m+n+1}\to\mathbb{R}^n$ be a total locally Lipschitz real recursive function. Then the maximal solution $g:\mathbb{R}^{m+1}\to\mathbb{R}^n$ of the differential equation*

$$g(\boldsymbol{x},t_0)=g_0(\boldsymbol{x}) \quad \partial_t g(\boldsymbol{x},t)=f(t,g(\boldsymbol{x},t),\boldsymbol{x}), \tag{12}$$

is real recursive.

Proof. Begin by considering the following differential recursion, where $\boldsymbol{z}$ ranges over $\mathbb{R}^n$, and $\boldsymbol{x}$ over $\mathbb{R}^m$:

$$\tilde{g}(\boldsymbol{z},\boldsymbol{x},0)=(\boldsymbol{z},\boldsymbol{x}) \quad \partial_t\tilde{g}(\boldsymbol{z},\boldsymbol{x},t)=(f(t+t_0,\tilde{g}(\boldsymbol{z},\boldsymbol{x},t)),\underbrace{0,\ldots,0}_{m}).$$

The solution exists and is unique, because the function given by $(f(t+t_0,\boldsymbol{v}),0,\ldots,0)$ is locally Lipschitz. Then the solution $\tilde{g}$ verifies $\tilde{g}(g_0(\boldsymbol{x}),\boldsymbol{x},t-t_0)=(g(\boldsymbol{x},t),\boldsymbol{x})$ (this is derived by a simple calculation), and so g is real recursive using composition and projections. $\square$

By using only the addition and subtraction functions, we have obtained the more general form of differential recursion (12). We could now show that multiplication is real recursive, simply by displaying the differential recursion scheme $\times(x,0)=0$, $\partial_y\times(x,y)=x$. By showing that REC($\mathbb{R}$) is closed under a differential recursion operator of the form (12), we have simplified the proofs ahead.

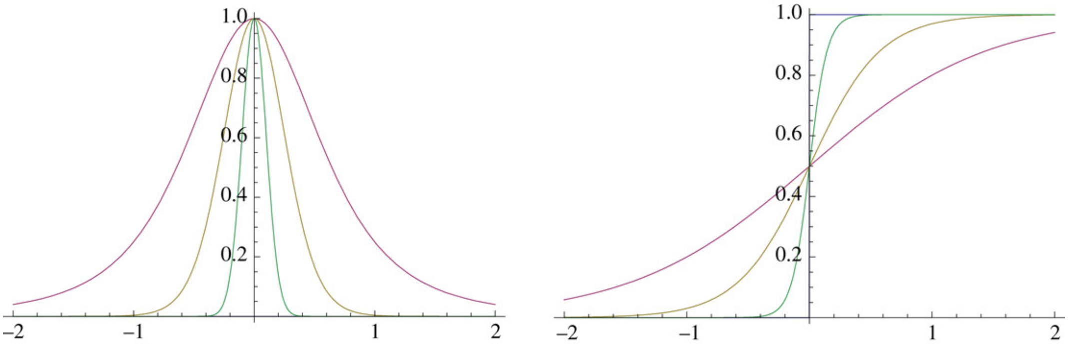


Fig. 1. Convergence of the shown expressions to δ and Θ (minus the $\frac{1}{2}\delta(x)$ part).

Proposition 3.7 *([24,27]). The restrictions to the domain $(0, +\infty)$ of the inverse, division and square root functions are real recursive. The exponential, logarithm, power, sine, cosine and arc-tangent functions are real recursive. The real numbers π and* e *are real recursive constants.*

Proof. The restricted division and logarithm functions are obtained simultaneously, through the differential recursion scheme:

$$\begin{cases} \frac{1}{1} = 1, \\ \log(1) = 1, \end{cases} \quad \text{and} \quad \begin{cases} \partial_x \frac{1}{x} = -1 \times \left(\frac{1}{x}\right) \times \left(\frac{1}{x}\right) = -\frac{1}{x^2}, \\ \partial_x \log(x) = \frac{1}{x}. \end{cases}$$

In the differential recursion scheme of Proposition 3.6, we have $t_0 = 1$, $g_0 = (1, 1)$ and the total, locally Lipschitz function $f(t, z_1, z_2) = (-(z_1)^2, z_1)$. The solution is, therefore, unique, and we obtain the restricted inverse function and the logarithm function, as intended. The following expressions, using the differential recursion scheme (12), give us the remaining functions

(i) $\frac{x}{y} = x \times \frac{1}{y}$ gives us the restricted division;
(ii) $\exp(0) = 1$, $\partial_x \exp(x) = \exp(x)$ solves to the exponential function;
(iii) $x^y = \exp(\log(x)y)$, where $x > 0$, is the power function;
(iv) $\sqrt{x} = x^{\frac{1}{2}}$ gives us the square root from the power function, restricted to positive x;
(v) $(\sin, \cos)(0) = (0, 1)$, $\partial_x(\sin, \cos)(x) = (\cos, -\sin)(x)$;
(vi) $\arctan(0) = 0$, $\partial_x \arctan(x) = \frac{1}{x^2+1}$;
(vii) $\mathrm{e} = \exp(1)$;
(viii) $\pi = 4 \times \arctan(1)$. □

Proposition 3.8 *([24,27]). Kronecker's δ and Heaviside's Θ, given by*

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \Theta(x) = \begin{cases} 1 & \text{if } x \geqslant 0 \\ 0 & \text{otherwise;} \end{cases}$$

are real recursive.

Proof. Set $\delta(x) = \limsup_{y\to\infty} \left(\frac{1}{x^2+1}\right)^y$ and $\Theta(x) = \left(\limsup_{y\to\infty} \frac{1}{1+2^{-xy}}\right) + \frac{1}{2}\delta(x)$ (see Fig. 1). □

Proposition 3.9 *([24,27]). The sawtooth wave function, denoted by r, and the square wave function, denoted by s, are real recursive.*

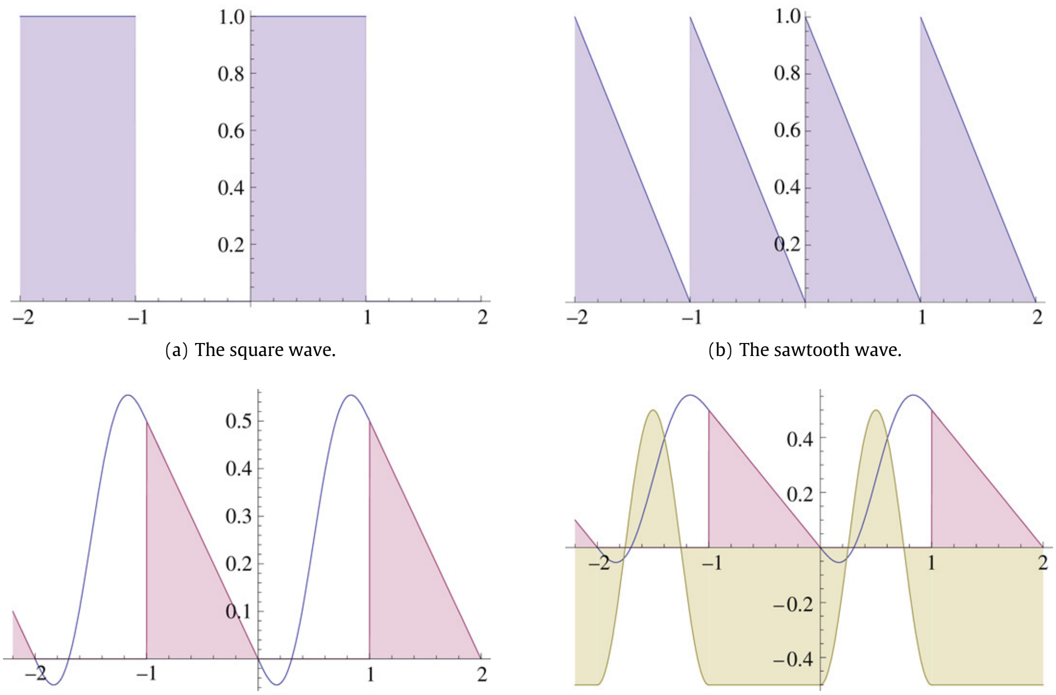
Proof. The square function is given by $s(x) = \Theta(\sin(\pi x))$. We can build the sawtooth using the recursion scheme $\tilde{r}(0) = 0$ and $\partial_x \tilde{r}(x) = 2\sin(\pi x)^2 s(x) - \frac{1}{2}$. We get $r(x) = s(x)\tilde{r}(x+1) + (1 - s(x))\tilde{r}(x)$ (cf. Fig. 2). □

Proposition 3.10. *The characteristics $\chi_=$ of equality, $\chi_\leqslant$ of inequality and $\chi_<$ of proper inequality are real recursive.*

Proof. Take $\chi_=(x, y) = \delta(y - x)$, $\chi_\leqslant(x, y) = \Theta(y - x)$, and $\chi_<(x, y) = \chi_\leqslant(x, y) - \chi_=(x, y)$. □

We will often use the characteristics of equality and inequality to define a function by cases, as in the following proofs. We use the abbreviations $\chi_\neq(x, y) = 1 - \chi_=(x, y)$, $\chi_>(x, y) \equiv \chi_<(y, x)$ and $\chi_\geqslant(x, y) \equiv \chi_\leqslant(y, x)$.

Proposition 3.11. *The functions of unrestricted inverse (with domain $\mathbb{R}\backslash\{0\}$), unrestricted division (with domain $\mathbb{R} \times (\mathbb{R}\backslash\{0\})$) and unrestricted square root (with domain $[0, +\infty)$) are real recursive.*



(a) The square wave.

(b) The sawtooth wave.

(c) $\tilde{r}$.

(d) Plot of $\partial_x \tilde{r}(x) = 2\sin(\pi x)^2 s(x) - \frac{1}{2}$ and $\tilde{r}$.

Fig. 2. Wave functions.

Proof. Take the real recursive function sgn, given by

$$\mathrm{sgn}(x) = (\chi_>(x,0) - \chi_<(x,0)) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0. \end{cases}$$

Now use the restriction of $\frac{1}{x}$ to positive x, and set

$$\frac{1}{x} = \mathrm{sgn}(x)\frac{1}{\mathrm{sgn}(x)x} = \begin{cases} \frac{1}{x} & \text{if } x > 0, \\ -\frac{1}{-x} & \text{if } x < 0 \end{cases}$$

where in the left we mean the unrestricted inverse, and in the right we use the inverse restricted to positive values, which was already defined in Proposition 3.7. Unrestricted division is obtained in the same way as for the restricted case. In an analogous way, we take the restricted square root (to the right), and define an unrestricted square root (to the left):

$$\sqrt{x} = \chi_{\neq}(x,0) \times \sqrt{x + \chi_=(x,0)} = \begin{cases} \sqrt{x} & \text{if } x \geqslant 0, \\ \perp & \text{otherwise.} \end{cases} \quad \square$$

Proposition 3.12. *The floor function, the ceil function, the absolute value function, the Euclidean and supremum norms over* $\mathbb{R}^n$ *are real recursive.*

Proof. We use the following expressions:

(i) $\lfloor x \rfloor = x - r(-x)$;
(ii) $\lceil x \rceil = x + r(x)$;
(iii) $|x| = (2\Theta(x) - 1)x$;
(iv) $\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}$;
(v) $\|(x_1, x_2)\|_\infty = \chi_>(|x_1|, |x_2|)x_1 + \chi_\leqslant(|x_1|, |x_2|)x_2$;
(vi) $\|\boldsymbol{x}\|_\infty = \|(x_1, \|(x_2, \| \ldots \|(x_{n-1}, x_n)\|_\infty) \ldots \|_\infty)\|_\infty$. $\square$

Definition 3.13. The **sigmoidal function**, σ, is given by

$$\sigma(x) = \frac{e^x}{1 + e^x}.$$

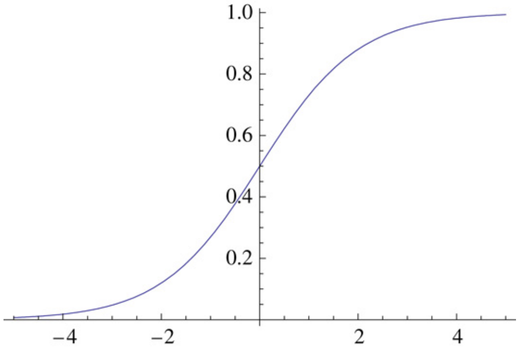


Fig. 3. Plot of $\sigma(x)$.

Proposition 3.14. *The sigmoidal function is real recursive. Furthermore, for any function* $f : \mathbb{R}^{n+1} \to \mathbb{R}$*, we have*

$$\limsup_{y\to\infty} f(\boldsymbol{x}, y) = z \in \mathbb{R} \iff \limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)) = \sigma(z) \in (0, 1),$$
$$\limsup_{y\to\infty} f(\boldsymbol{x}, y) = +\infty \iff \limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)) = 1, \text{ and}$$
$$\limsup_{y\to\infty} f(\boldsymbol{x}, y) = -\infty \iff \limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)) = 0.$$

Proof. The expression given for the sigmoidal function is a composition of functions that we have already shown to be real recursive, and thus σ itself must be real recursive. The first property with respect to the infinite supremum limit is a consequence of σ being a strictly increasing surjection to $(0, 1)$; we find that for any $\boldsymbol{x} \in \mathbb{R}^n$, $y \in \mathbb{R}$,

$$\sigma(\sup_{z\geqslant y} f(\boldsymbol{x}, z)) = \sup_{z\geqslant y} \sigma(f(\boldsymbol{x}, z)).$$

We then make use of the definition of supremum limit; we know that $\limsup_{y\to\infty} f(\boldsymbol{x}, y) \in \mathbb{R}$ if and only if there is a real value r such that $\| \sup_{z\geqslant y} f(\boldsymbol{x}, z)\| < r$ for all sufficiently large y. This is equivalent, as we have seen, to $\sigma(-r) < \sup_{z\geqslant y} \sigma(f(\boldsymbol{x}, z)) < \sigma(r)$, for some r, which is a sufficient and necessary condition for $\limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)) \in (0, 1)$ to hold. So the existence of both supremum limits of f and $\sigma \circ f$ is equivalent. The additional fact that, when these limits are defined,

$$\limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)) = \sigma\left(\limsup_{y\to\infty} f(\boldsymbol{x}, y)\right)$$

can be derived from the continuity of σ and its inverse. The remaining equivalencies are proven in the same fashion. □

Definition 3.15. The infinite infimum limit operator, **Li**, and the infinite limit operator, **L**, are given by

$$\mathbf{Li}(f)(\boldsymbol{x}) = \liminf_{y\to\infty} f(\boldsymbol{x}, y) \qquad \mathbf{L}(f)(\boldsymbol{x}) = \lim_{y\to\infty} f(\boldsymbol{x}, y);$$

where $f : \mathbb{R}^{m+1} \to \mathbb{R}^n$ is in $\mathscr{F}$.

Proposition 3.16 (*[26]*). REC($\mathbb{R}$) *is effectively closed under* **Li** *and* **L**.

Proof. We set $\mathbf{Li}(f)(\boldsymbol{x}) = -\limsup_{y\to\infty} -f(\boldsymbol{x}, y)$. It is known that $\lim_{y\to\infty} f(\boldsymbol{x}, y)$ is defined if and only if $\limsup_{y\to\infty} f(\boldsymbol{x}, y)$ and $\liminf_{y\to\infty} f(\boldsymbol{x}, y)$ are both defined and equal, and in this case $\lim_{y\to\infty} f(\boldsymbol{x}) = \limsup_{y\to\infty} f(\boldsymbol{x}, y)$. So we set

$$\mathbf{L}(f)(\boldsymbol{x}) = \frac{1}{\chi_=(\limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)), \liminf_{y\to\infty} \sigma(f(\boldsymbol{x}, y)))} \limsup_{y\to\infty} f(\boldsymbol{x}, y). \quad \square$$

This ends the most elementary part of real recursive function theory. We have seen that a number of functions are real recursive. In Section 3.3, we will see that this class extends even further. Hopefully, the reader will nurture a growing astonishment at the expressive power of such a simple inductive definition. A full and insightful characterisation will have to wait until Section 5. In the next section, we will stratify REC($\mathbb{R}$) into a hierarchy.

3.2. The η-hierarchy

Remember from Definition 2.26 that we use the capital letter H (eta) to designate the function algebra for REC($\mathbb{R}$).

Definition 3.17. The η**-hierarchy** is the rank hierarchy for the limit operator under the algebra H for REC($\mathbb{R}$). We use H_n to denote the nth level of this hierarchy. In symbols we have:

$$\mathrm{H}_n = H_n^{\mathrm{H},\mathbf{Ls}} = \{f \in \mathrm{REC}(\mathbb{R}) : \mathrm{rk}_{\mathbf{Ls}}^{\mathrm{H}}(f) \leqslant n\}.$$

A clearer picture for this hierarchy may be obtained from the following corollary of Proposition 2.15.

Corollary 3.18. *The η-hierarchy is inductively given by:*

(i) $\mathrm{H}_0 = [-1^n, 0^n, 1^n, U_i^n; \mathbf{C}, \mathbf{R}, \mathbf{V}]$,
(ii) $\tilde{\mathrm{H}}_n = \mathrm{H}_n \cup \{\mathbf{Ls}(f) : f : \mathbb{R}^{m+1} \to \mathbb{R}^k$ *is in* $\mathrm{H}_n\}$, *and*
(iii) $\mathrm{H}_{n+1} = [\tilde{\mathrm{H}}_n; \mathbf{C}, \mathbf{R}, \mathbf{V}]$.

The following corollary comes from the proofs of the previous section.

Corollary 3.19. *The following functions and constants are in* H_0*:*

(i) *The addition, subtraction and multiplication functions.*
(ii) *The inverse, division and square root functions, restricted to a positive argument.*
(iii) *The exponential, logarithm, power, sine, cosine and arc-tangent functions.*
(iv) *The numbers* π *and* e*.*
(v) *The sigmoidal function.*

The following functions are in H_1*:*

(i) *Kronecker's* δ *and Heaviside's* Θ*.*
(ii) *The sawtooth wave function r and the square wave function s.*
(iii) *The characteristics* $\chi_=$, $\chi_\leqslant$ *and* $\chi_<$*.*
(iv) *The unrestricted inverse, unrestricted division and unrestricted square root functions.*
(v) *The floor, ceil, absolute value, supremum norm and Euclidean norm.*

If $t_0 \in \mathbb{R}, g_0 : \mathbb{R}^m \to \mathbb{R}^n, f : \mathbb{R}^{m+n+1} \to \mathbb{R}^n$ *are in* H_i*, then the solution g of* (12) *is also in* H_i*. If* $f : \mathbb{R}^{m+1} \to \mathbb{R}^m$ *is in* H_i*, then* $\mathbf{Li}(f)$, $\mathbf{L}(f)$ *are in* H_{i+1}*.*

Most proofs will now include the position of functions in the η-hierarchy.

3.3. Non-trivial real recursive operators

The operator **Ls** serves as an analogue to the minimalisation operator of classical recursion theory. However, this operator has a distinct feature: there is a real recursive way of telling whether or not the infinite supremum limit exists.

Definition 3.20. For an $(m+1)$-ary function $f \in \mathscr{F}$, the η^s **operator** gives an m-ary function, $\eta^s(f)$, such that:

$$\eta^s(f)(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \limsup\limits_{y\to\infty} f(\boldsymbol{x}, y) \text{ exists,} \\ 0 & \text{otherwise.} \end{cases}$$

The η^i and η operators are defined in the same way, but with respect to lim inf and lim.

Here we will show that we may obtain $\eta(f)$, $\eta^s(f)$ and $\eta^i(f)$ when f is a total function. In fact, REC($\mathbb{R}$) is effectively closed for these operators, i.e., our restriction on f is not required, but we will withhold the (more complicated) proof for now.

Theorem 3.21 (*[27],[22]*). *If f is a total function in* H_i*, then* $\eta(f)$, $\eta^s(f)$ *and* $\eta^i(f)$ *are in* H_{i+1}*.*

Proof. We make the proof for a scalar function $f : \mathbb{R}^m \to \mathbb{R}$; the proof generalises to the vector case, by Proposition 3.2. The function $\eta^s(f)$ is given by the following real recursive expression:

$$\eta^s(f)(\boldsymbol{x}) = 1 - \chi_=\left(\limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)), 1\right) - \chi_=\left(\limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)), 0\right).$$

Then $\limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y))$ exists, because $\sigma \circ f$ is a bounded total function, and $\eta^s(f)(\boldsymbol{x}) = 0$ if and only if $\limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y)) \in \{0, 1\}$, which provides the intended behaviour according to Lemma 3.14. A similar expression gives us $\eta^i(f)$.

We have seen that $\lim_{y\to\infty} f(\boldsymbol{x}, y)$ exists if and only if the supremum and infimum limits of f exist and are equal. If they exist, the supremum and infimum limits of f are equal if and only if the supremum and infimum limits of $\sigma \circ f$ are equal. For this reason $\eta(f)$ can be set as

$$\eta(f)(\boldsymbol{x}) = \eta^s(f)(\boldsymbol{x}) \times \eta^i(f)(\boldsymbol{x}) \times \chi_=\left(\liminf_{y\to\infty} \sigma(f(\boldsymbol{x}, y)), \limsup_{y\to\infty} \sigma(f(\boldsymbol{x}, y))\right). \quad \square$$

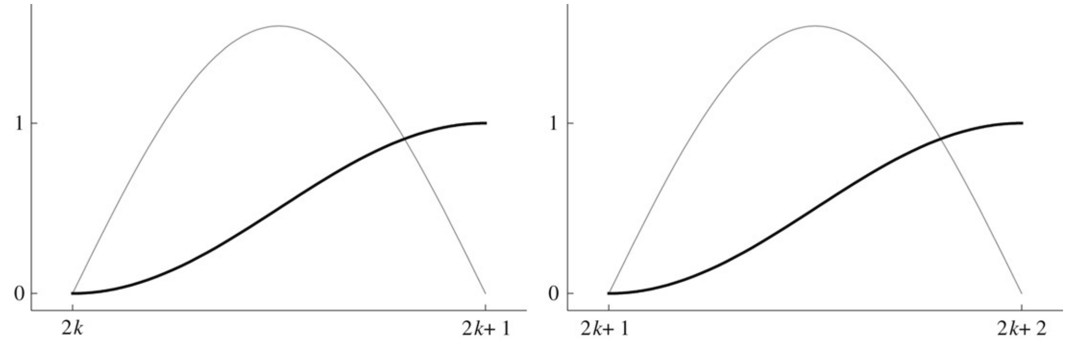


Fig. 4. Plot of $\frac{1-\cos(\pi t)}{2}$ between $2k$ and $2k+1$, and of $\frac{1+\cos(\pi t)}{2}$ between $2k+1$ and $2k+2$, with their respective derivatives drawn in a lighter line.

Definition 3.22. The **restricted iteration operator**, $\bar{\mathbf{I}}$, transforms an n-ary, total, locally Lipschitz function $g \in \mathscr{F}$ with n components, into a total $(n+1)$-ary function $\bar{\mathbf{I}}(g)$ with n components, given by

$$\bar{\mathbf{I}}(g)(\boldsymbol{x}, t) = g^{\lfloor |t| \rfloor}(\boldsymbol{x}) = \underbrace{g \circ g \circ \cdots \circ g}_{\lfloor |t| \rfloor \text{ times}}(\boldsymbol{x}).$$

The following theorem is one of the most fundamental results in the field. The differential recursion scheme that is used goes back to the ideas of Branicky [5] and, more explicitly, Moore [24].

Theorem 3.23 (*[27,22]*). REC($\mathbb{R}$) *is effectively closed under* $\bar{\mathbf{I}}$. *Furthermore, if* $g \in Dom(\bar{\mathbf{I}}) \cap \mathrm{H}_i$, *then* $\bar{\mathbf{I}}(g) \in \mathrm{H}_{\max(i,1)}$.

Proof. Let $g \in \mathrm{H}_i$ be an n-ary, total, locally Lipschitz function with n components. Let f be the $(1+2n)$-ary function with $2n$ components given by:

$$f(t, \boldsymbol{y}, \boldsymbol{z}) = \begin{pmatrix} (g(\boldsymbol{z}) - \boldsymbol{z})\frac{\pi}{2}\sin(\pi t)s(t) \\ \dfrac{(\boldsymbol{y} - \boldsymbol{z})\pi \sin(\pi t)}{\cos(\pi t) - 1 + \delta(\cos(\pi t) - 1)}(1 - s(t)) \end{pmatrix}.$$

The components of f are shown using a column vector; the first line gives the first n components, and we will call these the *first part* of f; the remaining components will be called the *second part* of f. Several observations can be made:

(i) The first part of f will be zero whenever t is in an interval of the form $[2k+1, 2k+2]$, and
(ii) The second part of f will be zero for t in $[2k, 2k+1]$.
(iii) For a fixed $\boldsymbol{z}$, and any $\boldsymbol{y} : [2k, 2k+1] \to \mathbb{R}^n$, $t \in (2k, 2k+1)$,

$$\int_{2k}^{t} f(s, \boldsymbol{y}(s), \boldsymbol{z})\mathrm{d}s = (g(\boldsymbol{z}) - \boldsymbol{z})\frac{\pi}{2}\int_{2k}^{t}\sin(\pi s)\mathrm{d}s = (g(\boldsymbol{z}) - \boldsymbol{z})\frac{1 - \cos(\pi t)}{2}.$$

(iv) If $\boldsymbol{y}$ is fixed, $\boldsymbol{z}(2k+1) = \boldsymbol{z}_0$, then $\boldsymbol{z}(t) = \boldsymbol{z}_0 + (\boldsymbol{y} - \boldsymbol{z}_0)\frac{1+\cos(\pi t)}{2}$ is the unique solution to

$$\partial_t \boldsymbol{z}(t) = \frac{(\boldsymbol{y} - \boldsymbol{z}(t))\pi \sin(\pi t)}{\cos(\pi t) - 1 + \delta(\cos(\pi t) - 1)}$$

in the interval $(2k+1, 2k+2)$.
(v) The function f is total locally Lipschitz, because it is the composition of a total locally Lipschitz function with g.

To understand why (iii) and (iv) are important, we show the plot of the solutions $\frac{1-\cos(\pi t)}{2}$ and $\frac{1+\cos(\pi t)}{2}$, and of its derivatives (see Fig. 4). Notice how these functions go from 0 to 1, and how the derivatives are locally Lipschitz.

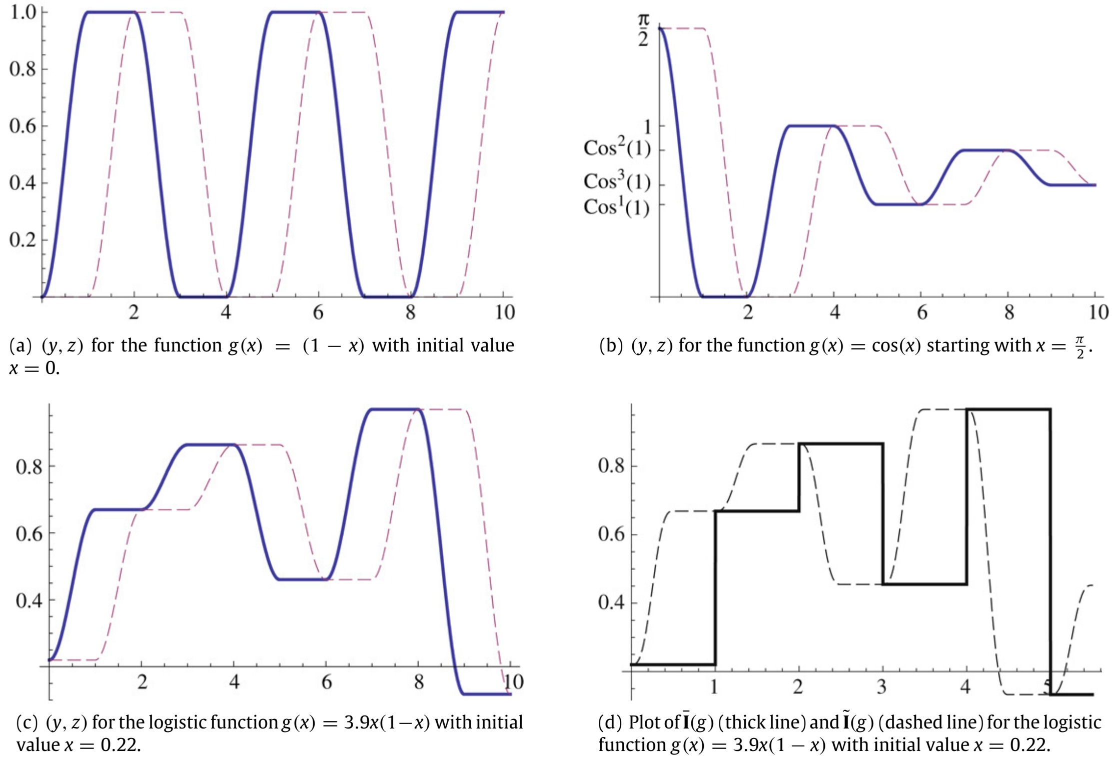
So, should we take the differential recursion scheme

$$(\boldsymbol{y}, \boldsymbol{z})(\boldsymbol{x}, 0) = \boldsymbol{x} \qquad \partial_t(\boldsymbol{y}, \boldsymbol{z})(\boldsymbol{x}, t) = f(t, (\boldsymbol{y}, \boldsymbol{z})(\boldsymbol{x}, t))$$

where $(\boldsymbol{y}, \boldsymbol{z})$ is regarded as a function from $\mathbb{R}^{n+1}$ to $\mathbb{R}^{2n}$, we may set[4]

$$\bar{\mathbf{I}}(g)(\boldsymbol{x}, t) = \boldsymbol{y}(\boldsymbol{x}, 2\lfloor |t| \rfloor).$$

[4] The differential recursion scheme could be changed so that the scaling $2 \times \lfloor |t| \rfloor$ would become unnecessary.



(a) (y, z) for the function $g(x) = (1 - x)$ with initial value $x = 0$.

(b) (y, z) for the function $g(x) = \cos(x)$ starting with $x = \frac{\pi}{2}$.

(c) (y, z) for the logistic function $g(x) = 3.9x(1-x)$ with initial value $x = 0.22$.

(d) Plot of $\bar{\mathbf{I}}(g)$ (thick line) and $\tilde{\mathbf{I}}(g)$ (dashed line) for the logistic function $g(x) = 3.9x(1 - x)$ with initial value $x = 0.22$.

Fig. 5. Plot of $\boldsymbol{y}(t)$ and $\boldsymbol{z}(t)$ for various functions.

These functions can be explained in the following way: as t changes from 0 to 1, $\boldsymbol{z}$ is constant and $\boldsymbol{y}$ goes through the distance from $\boldsymbol{x}$ to $g(\boldsymbol{x})$. For $t \in [1, 2]$, $\boldsymbol{y}$ is constant and $\boldsymbol{z}$ catches up, hence $\boldsymbol{z}(\boldsymbol{x}, 2) = \boldsymbol{y}(\boldsymbol{x}, 2) = g(\boldsymbol{x})$. If $t > 2$, then the same cycle begins again, and, for every $n \in \mathbb{N}$, $\bar{\mathbf{I}}(g)(\boldsymbol{x}, n) = \boldsymbol{y}(2n) = \boldsymbol{z}(2n)$. □

Example 3.24. You may see a plot of (y, z) for various functions in Fig. 5. The function y is shown in a thick line, and z is shown in a thin, dashing line. □

Definition 3.25. The **restricted smooth iteration operator**, $\tilde{\mathbf{I}}$, takes an n-ary, total, locally Lipschitz function $g \in \mathscr{F}$ with n components, and gives a total $(n + 1)$-ary locally Lipschitz function $\bar{\mathbf{I}}(g)$ with n components, such that

$$\bar{\mathbf{I}}(g)(\boldsymbol{x}, t) = \boldsymbol{y}(\boldsymbol{x}, 2t\chi_{\geqslant}(t, 0)),$$

where $\boldsymbol{y}$ is given by the previous proposition.

The restricted iteration operator $\bar{\mathbf{I}}(g)$ is not smooth; in fact, it gives a discontinuous function. The smooth iteration operator $\tilde{\mathbf{I}}$ still verifies $\tilde{\mathbf{I}}(g)(\boldsymbol{x}, n) = \bar{\mathbf{I}}(g)(\boldsymbol{x}, n)$ for all natural n, but has the following advantage, which may be concluded from Theorem 2.27.

Theorem 3.26. *If g is total and locally Lipschitz, then so is* $\tilde{\mathbf{I}}(g)$.

Remark 3.27. $\tilde{\mathbf{I}}(g)$ is also given by:

$$\tilde{\mathbf{I}}(g)(\boldsymbol{x}, t) = \begin{cases} \boldsymbol{x} & \text{if } t \leqslant 0, \\ g^n(\boldsymbol{x}) & \text{if } t \in \left(n + \dfrac{1}{2}, n + 1\right] \text{ for some } n \in \mathbb{N}, \\ g^n(\boldsymbol{x})\xi + g^{n+1}(\boldsymbol{x})(1 - \xi) & \text{if } t \in \left(n, n + \dfrac{1}{2}\right] \text{ for some } n \in \mathbb{N}. \end{cases}$$

Above, ξ is an abbreviation for $\frac{1-\cos(2\pi(t-n))}{2}$. □

Iteration is very useful, and very powerful.

Proposition 3.28 ([31]). *If g is an $(m + 1)$-ary total locally Lipschitz function with n components in* H_i*, then there are two functions S and P in* $\mathrm{H}_{\max(i,1)}$*, such that*

$$S(\boldsymbol{x}, n) = \sum_{i=1}^{\lfloor |n| \rfloor} g(\boldsymbol{x}, i) \qquad P(\boldsymbol{x}, n) = \prod_{i=1}^{\lfloor |n| \rfloor} g(\boldsymbol{x}, i).$$

Proof. We show the proof only for S, since the proof for P is very similar. Begin by setting

$$\tilde{S}(\boldsymbol{x}, \boldsymbol{y}, i) = (\boldsymbol{x}, \boldsymbol{y} + g(\boldsymbol{x}, i), i + 1).$$

Then $\tilde{S}$ is total and locally Lipschitz, and $(\boldsymbol{x}, S(\boldsymbol{x}, n), n) = \bar{\mathbf{I}}(\tilde{S})(\boldsymbol{x}, 0, 1, n)$. Thus S may be obtained by composition, projections and aggregation. □

Very pathological functions of analysis are real recursive too. Take, for instance, the everywhere continuous and nowhere differentiable Weierstraß function w, given by

$$w(x) = \sum_{n=0}^{\infty} a^n cos(b^n \pi x), \quad 0 < a < 1, \; ab > 1 + \frac{1}{3}\pi.$$

As a corollary of the previous proposition, taking the sums of the continuously differentiable $a^n cos(b^n \pi x)$ to the limit, we get:

Corollary 3.29 (*[22]*). *The Weierstraß function is real recursive for any real recursive numbers a and b.*

There are a number of operators which are real recursive, but we may generally say that they come in two flavours. Some of these operators make some calculations with some function, and other operators could be called search operators, because their expressive power arises from searching for some value with certain properties. Differential recursion and iteration are examples of the former. While not entirely obvious, infinite limits and the η operators are good examples of search operators; solving an infinite limit consists in finding the value which is approximated by a function as one argument grows.

Definition 3.30. Let $f : \mathbb{R}^{m+1} \to \mathbb{R}$ be in $\mathscr{F}$. The **Sup** and **Inf** operators are given, component-wise, by

$$\mathbf{Sup}(f)(\boldsymbol{x}) = \sup_{y \in \mathbb{R}} f(\boldsymbol{x}, y) \quad \text{and} \quad \mathbf{Inf}(f)(\boldsymbol{x}) = \inf_{y \in \mathbb{R}} f(\boldsymbol{x}, y).$$

In a similar way to the infinite limits, $\mathbf{Sup}(f)(\boldsymbol{x})$ is undefined if $f(\boldsymbol{x}, y)$ is undefined for any $y \in \mathbb{R}$; **Sup** and **Inf** are typical examples of search operators. In order to show that $\mathrm{REC}(\mathbb{R})$ is effectively closed for **Sup** and **Inf**, we create a periodic function, and take the supremum or infimum limit of that function.

Theorem 3.31 (*[21]*). $\mathrm{REC}(\mathbb{R})$ *is effectively closed for* **Sup** *and* **Inf**. *Furthermore, if* $f \in \mathrm{H}_i$, *then* $\mathbf{Sup}(f), \mathbf{Inf}(f) \in \mathrm{H}_{i+2}$.

Proof. Consider the function $\tilde{f}$, given by $\tilde{f}(\boldsymbol{x}, z, w) = f(\boldsymbol{x}, z\sin(w))$. Because $\sin(y)$ surjectively maps $[w, +\infty)$ into $[-1, 1]$, for any $w \in \mathbb{R}$, we find

$$\limsup_{w \to +\infty} \tilde{f}(\boldsymbol{x}, z, w) = \sup_{y \in [-z, z]} f(\boldsymbol{x}, y).$$

Then we set $\mathbf{Sup}(f)(\boldsymbol{x}) = \lim_{z \to +\infty} \limsup_{w \to \infty} \tilde{f}(\boldsymbol{x}, z, w)$. We proceed in the same way for **Inf**. □

Another search operator is minimalisation over the reals, denoted with a (boldface) $\boldsymbol{\mu}$.

Definition 3.32. Let $f : \mathbb{R}^{m+1} \to \mathbb{R}$ be in $\mathscr{F}$. The $\boldsymbol{\mu}$ operator is given by

$$\boldsymbol{\mu}(f)(\boldsymbol{x}) = \inf\{y \in [0, +\infty) : f(\boldsymbol{x}, y) = 0\}.$$

Theorem 3.33. $\mathrm{REC}(\mathbb{R})$ *is effectively closed for minimalisation. Furthermore, if* $f \in \mathrm{H}_i$, *then* $\boldsymbol{\mu}(f) \in \mathrm{H}_{\max(3,i+2)}$.

Proof. Recall that the sigmoidal function, σ, surjectively maps $[0, +\infty)$ to $[\frac{1}{2}, 1)$, preserving the order (cf. Fig. 3). Its inverse $\sigma^{-1}(y) = \log(y) - \log(1 - y)$ is undefined for $y = 1$. Take

$$\tilde{f}(\boldsymbol{x}, y) = \chi_{\neq}(f(\boldsymbol{x}, |y|), 0) + \chi_{=}(f(\boldsymbol{x}, |y|), 0)\sigma(|y|) = \begin{cases} 1 & \text{if } f(\boldsymbol{x}, |y|) \neq 0, \\ \sigma(|y|) & \text{if } f(\boldsymbol{x}, |y|) = 0 \end{cases}$$

and set $\boldsymbol{\mu}(f)(\boldsymbol{x}) = \sigma^{-1}(\mathbf{Inf}(\tilde{f})(\boldsymbol{x}))$. □

3.4. Real versus classical recursive functions

We will study the relationship between real recursive functions and classical recursive functionals. These are a class of partial, multiple-argument functions from $\mathbb{R}^k \times \mathbb{N}^m$ to $\mathbb{N}^n$. We will denote such a functional by using a semicolon to separate the real-valued arguments, which we will write on the left, from the natural-valued arguments, shown in the right. For instance $F(x; n)$.

Notation 3.34. *We use w, x, y, z to denote variables ranging over $\mathbb{R}$, and a, b, c, i, j to denote variables ranging over $\mathbb{N}$. The corresponding vector forms $\boldsymbol{w}, \boldsymbol{x}, \ldots$ and $\boldsymbol{a}, \boldsymbol{b}, \ldots$ will denote vector-valued variables over tuples of $\mathbb{R}$ and $\mathbb{N}$.* □

Take the following basic functionals as examples.

(i) The zero functionals, where each $\mathcal{Z}^k$ is such that $\mathcal{Z}^k(x_1, \ldots, x_k; a) = 0$;
(ii) The successor functionals, where each $\mathcal{S}^k$, given by $\mathcal{S}^k(x_1, \ldots, x_k; a) = a + 1$;
(iii) The projection functionals, where each $\mathcal{U}_j^{k,m}$ obeys

$$\mathcal{U}_j^{k,m}(x_1, \ldots, x_k; a_1, \ldots, a_m) = a_j;$$

(iv) The oracle functionals, $\mathcal{O}_i^k$, such that[5]

$$\mathcal{O}_i^k(x_1, \ldots, x_k; b) = x_i(b) \text{ (the } b\text{th digit of the binary expansion of } x_i).$$

We write $\mathcal{C}$, $\mathcal{R}$ and μ to stand for the composition, primitive recursion and minimalisation operators. Given $F : \mathbb{R}^k \times \mathbb{N}^{m'} \to \mathbb{N}^n$, $G : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^{m'}$, the functional $\mathcal{C}(F, G) : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^n$ is given by

$$\mathcal{C}(F, G)(\boldsymbol{x}; \boldsymbol{a}) = F(\boldsymbol{x}; G(\boldsymbol{x}; \boldsymbol{a})).$$

Given two functionals $F : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}$ and $G : \mathbb{R}^k \times \mathbb{N}^{m+2} \to \mathbb{N}$, $\mathcal{R}(F, G) : \mathbb{R}^k \times \mathbb{N}^{m+1} \to \mathbb{N}$ is given by

$$\mathcal{R}(F, G)(\boldsymbol{x}; \boldsymbol{a}, 0) = F(\boldsymbol{x}; \boldsymbol{a}),$$

$$\mathcal{R}(F, G)(\boldsymbol{x}; \boldsymbol{a}, b + 1) = G(\boldsymbol{x}; b, \mathcal{R}(F, G)(\boldsymbol{x}; \boldsymbol{a}, b), \boldsymbol{a}).$$

The minimalisation operator μ takes a functional $F : \mathbb{R}^k \times \mathbb{N}^{m+1} \to \mathbb{N}$ and gives $\mu(F) : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}$ such that

$$\mu(F)(\boldsymbol{x}; \boldsymbol{a}) = \min\{b \in \mathbb{N} : F(\boldsymbol{x}; \boldsymbol{a}, b) = 0\}.$$

Finally, $\mathcal{V}$ is the aggregation operator: If $F : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^n$, $F : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^k$ are two functionals, then $\mathcal{V}(F, G) : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^{n+k}$ comes from

$$\mathcal{V}(F, G)(\boldsymbol{x}; \boldsymbol{a}) = (F(\boldsymbol{x}; \boldsymbol{a}), G(\boldsymbol{x}; \boldsymbol{a})).$$

Now take care in the following definition. We begin by defining two classes of *functions*, and only then the relevant class of functionals.

Definition 3.35. The class of **primitive recursive functions**, PRIM, is given by the function algebra

$$\text{PRIM} = [\mathcal{Z}^0, \mathcal{S}^0, \mathcal{U}_j^{0,m}; \mathcal{C}, \mathcal{R}, \mathcal{V}].$$

The class of **partial recursive functions**, PREC, is given by the function algebra

$$\text{PREC} = [\mathcal{Z}^0, \mathcal{S}^0, \mathcal{U}_j^{0,m}; \mathcal{C}, \mathcal{R}, \mathcal{V}, \mu].$$

The class of **partial recursive functionals**, PRECF, is given by the function algebra (k is **not** fixed):

$$\text{PRECF} = [\mathcal{Z}^k, \mathcal{S}^k, \mathcal{U}_j^{k,m}, \mathcal{O}_i^{k,m}; \mathcal{C}, \mathcal{R}, \mathcal{V}, \mu].$$

Note 1. We have described functionals as functions from $\mathbb{R}^k \times \mathbb{N}^m$ to $\mathbb{N}^n$, but in the literature functionals are usually defined as functions from $(\mathbb{N} \to \mathbb{N})^k \times \mathbb{N}^m$ to $\mathbb{N}$. It is important to understand that our approach is only superficially different. We may use any simple bijection from $\mathbb{N} \to \mathbb{N}$ to $\mathbb{R}$, and from $\mathbb{N}$ to $\mathbb{N}^n$, to obtain this result. □

The reason we call PRIM and PREC classes of functions, is because the following may be trivially obtained by induction:

Proposition 3.36. *Every functional in* PRIM *and* PREC *has zero real-valued arguments, i.e., has a signature $\mathbb{R}^0 \times \mathbb{N}^m \to \mathbb{N}^n$.*

We will then omit the $\mathbb{R}^0$ part from the signature of functions in PRIM and PREC. In fact, we could easily take the algebras for PRIM and PREC given, resp., in Examples 2.4 and 2.18, and show that a function(al) $F : \mathbb{R}^0 \times \mathbb{N}^m \to \mathbb{N}^n$ is in the algebras for primitive recursive (or partial recursive) functions of Definition 3.35 if and only if there are n functions $f_1, \ldots, f_n$ in the algebras of Example 2.4 (resp. 2.18) such that $F(; \boldsymbol{a}) = (f_1(\boldsymbol{a}), \ldots, f_n(\boldsymbol{a}))$.

To get a clearer picture of what a partial recursive functional is, we give a computational characterisation.

[5] The definition is ambiguous because a dyadic rational number x has two different binary expansions. In this case, $x(b)$ is the bth digit in the binary expansion ending in an infinite string of 0s.

Theorem 3.37. *A function $f : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^n$ is in* PRECF *if and only if there is a Turing machine with $k + m + n$ tapes with the following behaviour. If we take the binary expansion of $x_1, \ldots, x_k$, and write it in the first k tapes (this expansion might be infinite), write the numbers $a_1, \ldots, a_m$ in each of the following m tapes, and begin the computation; then, if $f(\boldsymbol{x}; \boldsymbol{a})$ is defined, the Turing machine will halt after a finite number of steps, and print $(f(\boldsymbol{x}; \boldsymbol{a}))_1, \ldots, (f(\boldsymbol{x}; \boldsymbol{a}))_n$ in the last n tapes; if $f(\boldsymbol{x}; \boldsymbol{a})$ is undefined, then the machine will not halt.*

The rest of this subsection will be dedicated to:

(I) Showing that every primitive recursive function is *real recursive*, in some sense;
(II) Proving that every partial recursive function is *real recursive*;
(III) Concluding that every partial recursive functional is *real recursive*;

To prove (I), we see that every primitive recursive function has a real extension which is real recursive.

Proposition 3.38. *If $F : \mathbb{N}^m \to \mathbb{N}^n$ is in* PRIM, *then there is a real recursive total locally Lipschitz function $f : \mathbb{R}^m \to \mathbb{R}^n$ in* H_1 *such that*

$$F(; \boldsymbol{a}) = f(\boldsymbol{a}) \quad \text{for all } \boldsymbol{a} \in \mathbb{N}^m.$$

Proof. Our proof is by structural induction on the function algebra for primitive recursive functions. This is clearly true for $\mathcal{Z}$ and $\mathcal{S}$, by taking the real recursive zero and add-one functions, and the inductive step for composition and aggregation is trivial, using the corresponding real recursive operators. Now suppose that $H : \mathbb{N}^{m+1} \to \mathbb{N}$ in PRIM is given by $H = \mathcal{R}(F, G)$, for some $F : \mathbb{N}^m \to \mathbb{N}$, $G : \mathbb{N}^{m+2} \to \mathbb{N}$, i.e.,

$$H(; \boldsymbol{a}, 0) = F(; \boldsymbol{a}) \qquad H(; \boldsymbol{a}, n+1) = G(; n, H(; \boldsymbol{a}, n), \boldsymbol{a}).$$

Then by the induction hypothesis, let f and g be two total locally Lipschitz real recursive functions in H_1 such that $F(; \boldsymbol{a}) = f(\boldsymbol{a})$ and $G(; n, b, \boldsymbol{a}) = g(n, b, \boldsymbol{a})$ for all $\boldsymbol{a} \in \mathbb{N}^m$ and $n, b \in \mathbb{N}$. Form

$$\tilde{h}(n, \boldsymbol{b}, \boldsymbol{a}) = (n+1, g(n, \boldsymbol{b}, \boldsymbol{a}), \boldsymbol{a})$$

by aggregation. Then let h be given by $(n, h(\boldsymbol{a}, n), \boldsymbol{a}) = \tilde{\mathbf{I}}(\tilde{h})(0, f(\boldsymbol{a}), \boldsymbol{a}, n)$, from which we find $h \in \mathrm{H}_1$. This function h will be a real recursive extension of H, and will also be total and locally Lipschitz by Theorem 3.26. □

Now recall the normal form theorem of Kleene.[6]

Theorem 3.39 (*Normal form theorem*). *For every natural $m, n > 0$ are two primitive recursive functions $U : \mathbb{N} \to \mathbb{N}^n$ and $T : \mathbb{N}^{m+2} \to \mathbb{N}$ with the following property. Take any partial recursive function $F : \mathbb{N}^m \to \mathbb{N}^n$, and there will be a number e, called a* ***code*** *of F, such that*

(i) *$F(\boldsymbol{a})$ is defined if and only if $(\exists b \in \mathbb{N})\ T(e, \boldsymbol{a}, b) = 0$, and*
(ii) *$F(\boldsymbol{a}) = U(\mu(T)(e, \boldsymbol{a}))$.*

By Proposition 3.38, there will be a real recursive extension of T and U, but we may furthermore ensure that:

Proposition 3.40. *If $F : \mathbb{N}^{m+1} \to \mathbb{N}$ and $f : \mathbb{R}^{m+1} \to \mathbb{R}$ in H_1 are such that*

$$F(; \boldsymbol{a}, b) = f(\boldsymbol{a}, b) \quad \text{for all } \boldsymbol{a} \in \mathbb{N}^m, b \in \mathbb{N}$$

then there is another real recursive function $\tilde{f} : \mathbb{R}^{m+1} \to \mathbb{R}$ in H_1 such that

$$\mu(F)(; \boldsymbol{a}) = \boldsymbol{\mu}(\tilde{f})(\boldsymbol{a}) \quad \text{for all } \boldsymbol{a} \in \mathbb{N}^m.$$

Proof. Just set $\tilde{f}(\boldsymbol{a}, y) = f(\boldsymbol{a}, \lfloor y \rfloor) r(y) + f(\boldsymbol{a}, \lfloor y+1 \rfloor)(1 - r(y))$. □

The function $\tilde{f}$ is just a linear interpolation of f on the last argument. If $F(b) = 1$ when b is not a prime, and $F(b) = 0$ if b is a prime, then the $\tilde{f}$ we would obtain is shown in Fig. 6.

We now get the following corollary.

Theorem 3.41 (*[27] and See Also [16]*). *If $F : \mathbb{N}^m \to \mathbb{N}^n$ is in* PREC, *then there is a real recursive function $f : \mathbb{R}^m \to \mathbb{R}^n$ in* H_3 *such that*

$$F(; \boldsymbol{a}) \simeq f(\boldsymbol{a}) \quad \text{for all } \boldsymbol{a} \in \mathbb{N}^m.$$

Proof. Let U, T be the primitive recursive functions given by the normal form theorem, u, t be their real recursive extensions given by Proposition 3.38, and let $\tilde{t}$ be given from t by the previous proposition. For any partial recursive function F, let e be one of its codes. Then set $f(\boldsymbol{x}) = u(\boldsymbol{\mu}(\tilde{t})(e, \boldsymbol{x}))$, and conclude that f is a real extension of F. The function f will be in H_3, because $u, \tilde{t}$ are in H_1, and by Proposition 3.33. □

6 Real-recursion theory also admits its own normal form theorems [cf. [18,21]].

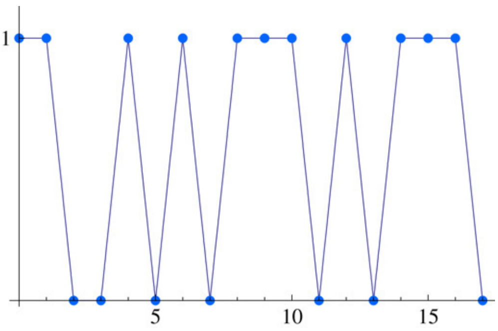


Fig. 6. A plot of $\tilde{f}(y)$, where $F(b)$ is given by the dots.

Thus concluding (II); in order to prove (III), we use the fact that any converging computation will always make use of only a finite part of the oracle.

Definition 3.42. If $\boldsymbol{x} \in \mathbb{R}^k$, then let $\boldsymbol{x}\restriction_n^{\mathbb{N}}$ denote, for some $n \in \mathbb{N}$, the vector in $\mathbb{N}^k$ given by

$$\boldsymbol{x}\restriction_n^{\mathbb{N}} = (x_1\restriction_n \times 2^n, \dots, x_k\restriction_n \times 2^n), \text{ where } x\restriction_n \text{ denotes truncation of } x \text{ to } n \text{ digits in its binary expansion.}$$

Proposition 3.43. *The function given by* $f(\boldsymbol{x}, n) = \boldsymbol{x}\restriction_n^{\mathbb{N}}$ *is real recursive.*

Proof. For $x \in \mathbb{R}$, $x\restriction_n \times 2^n = \lfloor x2^n \rfloor$. The rest comes from aggregation, etc. □

We can now use Theorem II.3.11 from [32], which for our purposes can be formulated as follows.

Proposition 3.44. *A functional* $F : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^n$ *is in* PRECF *if and only if there is a partial recursive function* $\tilde{F} : \mathbb{N}^{k+m} \to \mathbb{N}^n$ *with the property that* $F(\boldsymbol{x}; \boldsymbol{a}) \simeq \boldsymbol{b}$ *if and only if there is an* $n \in \mathbb{N}$ *such that* $\tilde{F}(; \boldsymbol{x}\restriction_n^{\mathbb{N}}, \boldsymbol{a}) \simeq \boldsymbol{b}$.

The following corollary can be understood as saying that for any given input the behaviour of a recursive functional only depends on a finite part of the oracle.

Corollary 3.45. *If a functional* $F : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^n$ *is in* PRECF *then there is a partial recursive function* $\tilde{F} : \mathbb{N}^{k+m} \to \mathbb{N}^n$ *with the property that* $F(\boldsymbol{x}; \boldsymbol{a}) \simeq \boldsymbol{b}$ *if and only if there is an* $m \in \mathbb{N}$ *such that* $\tilde{F}(; \boldsymbol{x}\restriction_n^{\mathbb{N}}, \boldsymbol{a}) \simeq \boldsymbol{b}$ *for all natural* $n \geqslant m$.

And so we conclude this section, with the following result.

Theorem 3.46. *If* $F : \mathbb{R}^k \times \mathbb{N}^m \to \mathbb{N}^n$ *is in* PRECF, *then there is a real recursive function* $f : \mathbb{R}^{k+m} \to \mathbb{R}^n$ *in* H_4 *such that*

$$F(\boldsymbol{x}; \boldsymbol{a}) \simeq f(\boldsymbol{x}, \boldsymbol{a}) \text{ for all } \boldsymbol{x} \in \mathbb{R}^k, \boldsymbol{a} \in \mathbb{N}^m.$$

Proof. Let $\tilde{F}$ be given from the previous corollary, and let $\tilde{f}$ be a real recursive function in H_3 extending $\tilde{F}$. Then take

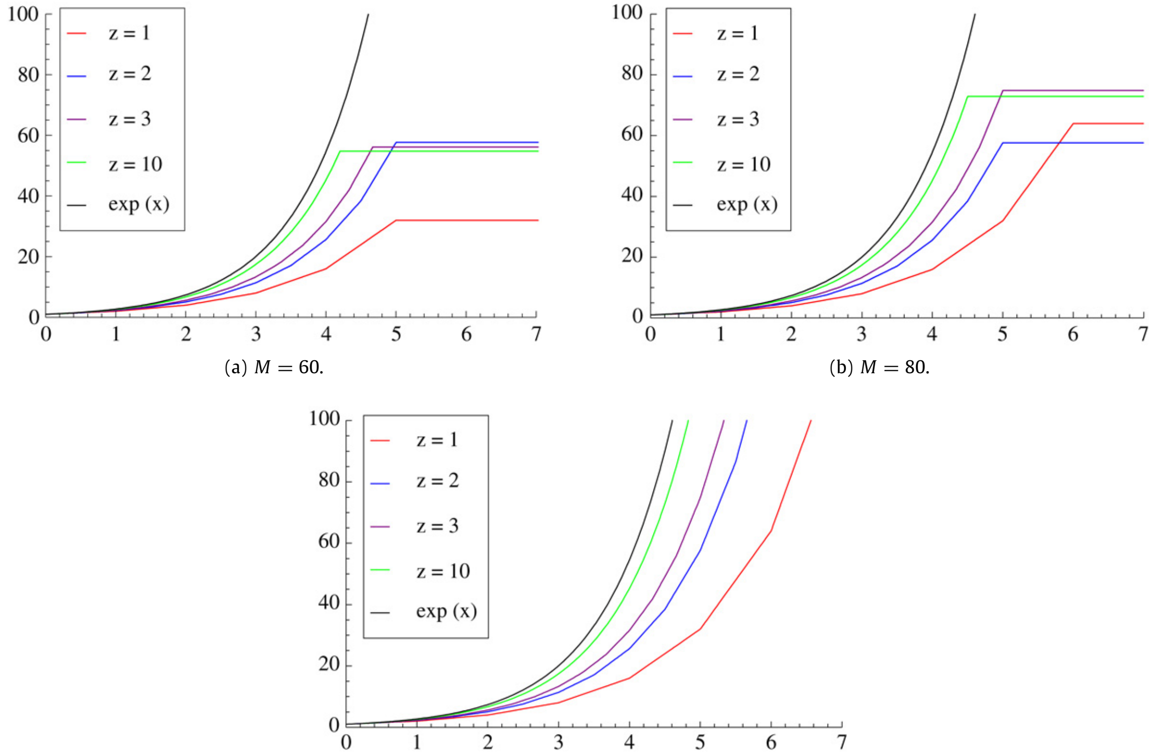
$$f(\boldsymbol{x}, \boldsymbol{a}) = \lim_{n \to +\infty} \tilde{f}(\boldsymbol{x}\restriction_{\lfloor n \rfloor}^{\mathbb{N}}, \boldsymbol{a})$$

and $f \in \mathrm{H}_4$ will extend F. □

We have shown that every partial recursive function or functional has a real recursive extension. However, the class of real recursive functions stretches much further. We will see in Section 5 that any predicate in the analytical hierarchy is real recursive, and as a corollary many non-computable functions, such as the Busy Beaver function, have real recursive extensions.

4. Universality

This section is devoted to solving the problem of universality. We will begin Section 4.1 with a series of considerations on the Euler method to approximate solutions of our simple differential recursion scheme (7). Our first conclusion will be that we may replace differential recursion with the restricted iteration operator of Definition 3.22, by adding a few basic functions to the algebra H. With this, we will form a new algebra I that also gives REC($\mathbb{R}$). In Section 4.2 we show that we may *totalise* any real recursive function, and this will solve the problem of universality, in the negative sense: we will conclude that there is no universal real recursive function.



(a) $M = 60$.

(b) $M = 80$.

(c) $M = 150$.

Fig. 7. The bounded Euler broken line, for $x = 1$ and $f(t, y) = y$, with various values of M and z.

4.1. *The Euler method and differential recursion*

Let $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ be a total locally Lipschitz function; consider again the Cauchy problem

$$g(\boldsymbol{x}, 0) = \boldsymbol{x} \quad \partial_t g(\boldsymbol{x}, t) = f(t, g(\boldsymbol{x}, t)). \tag{13}$$

We study some interesting properties of the Euler method to approximate solutions to (13).

Definition 4.1. The **Euler broken line** for the Cauchy problem (13) is a total function $\hat{g} : \mathbb{R}^{n+2} \to \mathbb{R}^n$ satisfying the following conditions. For any fixed $z > 0$, let $\delta = \frac{1}{z}$, $t_0 = 0$, $t_{i+1} = t_i + \delta$. Abbreviate $\hat{g}(\boldsymbol{x}, t, z) \equiv \hat{g}_z(\boldsymbol{x}, t)$. Then $\hat{g}$ is given by:

$$\hat{g}_z(\boldsymbol{x}, t_0) = \boldsymbol{x};$$

if $t_i \leqslant t < t_{i+1}$, then

$$\hat{g}_z(\boldsymbol{x}, t) = \hat{g}_z(\boldsymbol{x}, t_i) + (t - t_i) f(t_i, \hat{g}_z(\boldsymbol{x}, t_i));$$

and if $-t_{i+1} < t \leqslant -t_i$, then

$$\hat{g}_z(\boldsymbol{x}, t) = \hat{g}_z(\boldsymbol{x}, -t_i) - (t + t_i) f(-t_i, \hat{g}_z(\boldsymbol{x}, -t_i)).$$

The **bounded Euler broken line** for the Cauchy problem (13) is the total function $\tilde{g} : \mathbb{R}^{n+3} \to \mathbb{R}^n$ described below. We set δ, t_i as before, and abbreviate $\tilde{g}(\boldsymbol{x}, t, M, z) \equiv \tilde{g}_{M,z}(\boldsymbol{x}, t)$. Let j, k denote the smallest natural numbers such that $\|\hat{g}_z(\boldsymbol{x}, -t_j)\| \geqslant M$ and $\|\hat{g}_z(\boldsymbol{x}, t_k)\| \geqslant M$. If there is no such k (or j), then t_{k-1} (resp. $-t_{j-1}$) will denote $+\infty$ (resp. $-\infty$). Then $\tilde{g}$ is given by

$$\tilde{g}_{M,z}(\boldsymbol{x}, t) = \begin{cases} \hat{g}_z(\boldsymbol{x}, -t_{j-1}) & \text{if } t \leqslant -t_{j-1} \\ \hat{g}_z(\boldsymbol{x}, t) & \text{if } -t_{j-1} < t < t_{k-1} \\ \hat{g}_z(\boldsymbol{x}, t_{k-1}) & \text{if } t_{k-1} \leqslant t. \end{cases}$$

The bounded Euler broken line is a piece-wise linear approximation of the solution g of (13), using Euler's method, which is bounded in the norm by a value M, and where the number of approximation steps for a segment of length 1 is given by z. Fig. 7 illustrates how the bounded Euler broken line looks for different values of M and z.

Theorem 4.2. *Let g be a solution of (13), and let $\tilde{g}$ be its Euler broken line. Fix an arbitrary $\boldsymbol{x} \in \mathbb{R}^n$, and compact interval $J \subset Dom(g)$. Then, for every large enough $M \in \mathbb{R}$, the function $\tilde{g}_{M,z}(\boldsymbol{x}, \cdot)$ converges to $g(\boldsymbol{x}, \cdot)$ uniformly in J as $z \to +\infty$. In symbols,*

$$(\forall \varepsilon > 0)\exists \tilde{M}(\forall M > \tilde{M})\exists \tilde{z}(\forall z > \tilde{z})(\forall t \in J)\ \|\tilde{g}_{M,z}(\boldsymbol{x}, t) - g(\boldsymbol{x}, t)\| < \varepsilon.$$

Proof. Take any $\varepsilon > 0$. By the compactness of J and continuity of g (Theorem 2.27), set

$$M \geqslant \max_{t\in J} \|g(\boldsymbol{x}, t)\| + |t| + \varepsilon. \tag{14}$$

Let B denote the closed (compact) cylinder $B = J \times \overline{B}(0, M) \subset \mathbb{R}^{n+1}$. Because f and g are locally Lipschitz (Theorem 2.27), then choose two constants K_f and K_g such that

$$\|g(\boldsymbol{x}, \tilde{t}) - g(\boldsymbol{x}, t)\| \leqslant K_g|\tilde{t} - t| \quad \text{for all } t, \tilde{t} \in J; \tag{15}$$

$$\|f(\tilde{t}, \tilde{\boldsymbol{y}}) - f(t, \boldsymbol{y})\| \leqslant K_f|\tilde{t} - t| + K_f\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\| \quad \text{for all } (t, \boldsymbol{y}), (\tilde{t}, \tilde{\boldsymbol{y}}) \in B. \tag{16}$$

Choose $K \in \mathbb{R}$ to be greater than both K_f and $K_g + 1$, and choose z and δ, so that letting d_J denote the diameter of J,

$$z > \frac{1}{\varepsilon}K(\mathrm{e}^{Kd_J} - 1), \qquad \delta = \frac{1}{z}, \qquad t_0 = 0, \qquad t_{i+1} = t_i + \delta.$$

We will show, by induction on i, that setting $\Delta_t = \|\tilde{g}_{M,z}(\boldsymbol{x}, t) - g(\boldsymbol{x}, t)\|$, then

$$\Delta_t < \frac{1}{2}K\delta(\mathrm{e}^{Kt} - 1)$$

for any positive $t \in J$. Because of our choice of z, this ensures that $\Delta_t < \varepsilon$, which is what we intend to prove. The hypothesis is true for $t_i = t_0 = 0$, because $\Delta_0 = 0$. Now suppose it is true for some t_i. Then, choosing any $t \in [t_i, t_{i+1}] \cap J$,

$$\begin{aligned}
\Delta_t &= \|\tilde{g}_{M,z}(\boldsymbol{x}, t) - g(\boldsymbol{x}, t)\| = \left\|\tilde{g}_{M,z}(\boldsymbol{x}, 0) - g(\boldsymbol{x}, 0) + \int_0^t f(t_i, \tilde{g}_{M,z}(\boldsymbol{x}, t_i)) - f(s, g(\boldsymbol{x}, s))\mathrm{d}s\right\| \\
&\leqslant \Delta_{t_i} + \int_{t_i}^t \|f(t_i, \tilde{g}_{M,z}(\boldsymbol{x}, t_i)) - f(s, g(\boldsymbol{x}, s))\|\mathrm{d}s \\
&\leqslant \Delta_{t_i} + \int_{t_i}^t \|f(t_i, \tilde{g}_{M,z}(\boldsymbol{x}, t_i)) - f(t_i, g(\boldsymbol{x}, t_i))\|\mathrm{d}s + \int_{t_i}^t \|f(t_i, g(\boldsymbol{x}, t_i)) - f(s, g(\boldsymbol{x}, s))\|\mathrm{d}s.
\end{aligned}$$

By induction hypothesis, $\Delta_{t_i} = \|\tilde{g}_{M,z}(\boldsymbol{x}, t_i) - g(\boldsymbol{x}, t_i)\| < \varepsilon$, and so $(t_i, \tilde{g}_{M,z}(\boldsymbol{x}, t_i)) \in B$. Then we may apply the Lipschitz properties (15) and (16) to the integrals, and find

$$\Delta_t \leqslant \Delta_{t_i} + K_f(t - t_i)\Delta_{t_i} + K_f(K_g + 1)\left|\int_{t_i}^t |s - t_i|\mathrm{d}s\right|$$

which implies $\Delta_t \leqslant (1 + K(t - t_i))\Delta_{t_i} + \frac{1}{2}K^2(t - t_i)^2$. Using again the induction hypothesis, and $(t - t_i) \leqslant \delta$,

$$\Delta_t \leqslant \frac{1}{2}K\delta\mathrm{e}^{Kt_i}(1 + K(t - t_i)) - \frac{1}{2}K\delta(1 + K(t - t_i)) + \frac{1}{2}K^2\delta(t - t_i).$$

Now, because $1 + x \leqslant \mathrm{e}^x$ and $K^2 > K$, we arrive at

$$\Delta_t \leqslant \frac{1}{2}K\delta(\mathrm{e}^{Kt} - 1) < \varepsilon,$$

as intended. The proof is symmetrical for a negative t. □

Corollary 4.3. *If g is a solution of (13), and $\tilde{g}$ is its Euler broken line, then*

$$\limsup_{M\to+\infty} \limsup_{z\to+\infty} \tilde{g}_{M,z}(\boldsymbol{x}, t) = g(\boldsymbol{x}, t) \quad \textit{for } (\boldsymbol{x}, t) \in Dom(g).$$

We may, in fact, obtain the following.

Proposition 4.4. *If g is a solution of (13), and $\tilde{g}$ is its Euler broken line, then*

$$\limsup_{M\to+\infty} \limsup_{z\to+\infty} \tilde{g}_{M,z}(\boldsymbol{x}, t) \simeq g(\boldsymbol{x}, t).$$

Proof. We already know that g will equal $\mathbf{Ls}(\mathbf{Ls}(\tilde{g}))$ where it is defined, and so we only need to show that $\mathbf{Ls}(\mathbf{Ls}(\tilde{g}))$ will be undefined where g is undefined. This is an easy conclusion derived from Theorem 2.25, and from the definition of the bounded Euler broken line. For any fixed $\boldsymbol{x}$, and any compact interval J, we have two cases. Either that $g(\boldsymbol{x}, \cdot)$ is bounded in J, and so $\limsup_{z\to+\infty} \tilde{g}_{M,z}(\boldsymbol{x}, t) = g(\boldsymbol{x}, t)$ for every $t \in J$ and every large enough M; or otherwise $\limsup_{t\to B^-} \|g(\boldsymbol{x}, t)\| = +\infty$ for some positive B in J (the case is symmetrical for a negative A). In this case, g will be continuous and defined for every $0 \leqslant t < B$. So let $t_M \in J$ be the smallest value for which $\|g(\boldsymbol{x}, t_M)\| \geqslant M$, i.e,

$$t_M = \min\{t \in J : \|g(\boldsymbol{x}, t_M)\| \geqslant M\}$$

(which is well-defined because g is continuous). Then by the definition of $\tilde{g}$

$$\limsup_{z\to+\infty} \tilde{g}_{M,z}(\boldsymbol{x}, t) = \begin{cases} g(\boldsymbol{x}, t) & \text{if } t < t_M, \\ g(\boldsymbol{x}, t_M) & \text{if } t \geqslant t_M. \end{cases}$$

By the continuity of g, $\|g(\boldsymbol{x}, t)\|$ is always bounded for any compact interval $[0, T] \subset [0, B)$, and so $t_M > T$ if $M \geqslant \max_{t\in[0,T]} \|g(\boldsymbol{x}, t)\|$. So $t_M \to B$ as $M \to +\infty$. We may then conclude that

$$\limsup_{M\to+\infty} \limsup_{z\to+\infty} \|\tilde{g}_{M,z}(\boldsymbol{x}, t)\| = +\infty$$

for any $t \in [B, +\infty)$, and so $[B, +\infty)$ is disjoint from $Dom(\mathbf{Ls}(\mathbf{Ls}(\tilde{g})))$. □

The Euler broken line can be obtained in a real recursive way.

Proposition 4.5 ([9,22]). *For any real recursive f in $Dom(\mathbf{R})$, the Euler broken line $\tilde{g}$ of $g = \mathbf{R}(f)$ is real recursive.*

Proof. Let b be given by

$$b(t, \boldsymbol{y}, \delta, M) = \chi_<(\|\boldsymbol{y} + \delta \times f(t, \boldsymbol{y})\|, M) = \begin{cases} 1 & \text{if } \|\boldsymbol{y} + \delta \times f(t, \boldsymbol{y})\| < M; \\ 0 & \text{otherwise.} \end{cases}$$

Now take the auxiliary function $\hat{g}$, given by

$$\hat{g}(t, \boldsymbol{y}, \delta, M) = (t + \delta \times b(t, \boldsymbol{y}, \delta, M), \boldsymbol{y} + \delta \times f(t, \boldsymbol{y}) \times b(t, \boldsymbol{y}, \delta, M), \delta, M).$$

It is straightforward to see that if f is real recursive, then so is $\hat{g}$. The function $\hat{g}$ calculates the point $(t_{i+1}, \tilde{g}_{M,z}(\boldsymbol{x}, t_{i+1}), \delta, M)$ in the bounded Euler broken line, when given the current $t = t_i, \boldsymbol{y} = \tilde{g}_z(\boldsymbol{x}, t_i), \delta = \frac{1}{z}$ and M. If $\delta = -\frac{1}{z}$, then $\hat{g}$ will calculate $(-t_{i+1}, \tilde{g}_z(\boldsymbol{x}, -t_{i+1}), \delta, M)$ instead. We may then use another function, $\bar{g}$, with its ith component given by:

$$(\bar{g}(\boldsymbol{x}, t, M, z))_i = \left(\hat{g}^{\lfloor tz \rfloor}\left(0, \boldsymbol{x}, \frac{t}{|t|z}, M\right)\right)_{i+1}.$$

The function $\bar{g}$, given t, z, M, calculates $\tilde{g}(\boldsymbol{x}, t_i)$ for the largest $t_i \leqslant t$. Then $\tilde{g}$ may be given by a linear interpolation:

$$\tilde{g}_{M,z}(\boldsymbol{x}, t) = r(zt)\bar{g}(\boldsymbol{x}, t, M, z) + (1 - r(zt))\bar{g}\left(\boldsymbol{x}, t + \frac{1}{z}, M, z\right). \quad \square$$

Supported by this proposition, we define the following new operator.

Definition 4.6. The **Euler operator E** : $\mathscr{F} \to \mathscr{F}$, takes a total locally Lipschitz function $f : \mathbb{R}^{m+1} \to \mathbb{R}^n$, and maps $f \mapsto \tilde{g}$, where $\tilde{g}$ is the Euler broken line of Definition 4.1.

Then Proposition 4.5 is equivalent to saying that $\mathbf{E}$ is a real recursive operator. We may, however, take the following stronger result.

Theorem 4.7. *The class of real recursive functions is also given by the function algebra*

$$[-1^n, 0^n, 1^n, U_i^n; \mathbf{C}, \mathbf{E}, \mathbf{Ls}, \mathbf{V}].$$

Proof. Clearly this function algebra only gives real recursive functions, because $\mathrm{REC}(\mathbb{R})$ contains every basic function and is closed for every operator by Proposition 4.5. However, because $\mathbf{R} = \mathbf{Ls} \circ \mathbf{Ls} \circ \mathbf{E}$ (Proposition 4.4), this algebra will also give us every real recursive function. □

We will, however, use the following.

Theorem 4.8 ([22]). *The class of real recursive functions is given by the function algebra*

$$\mathrm{REC}(\mathbb{R}) = \mathrm{I} = [-1^n, 0^n, 1^n, U_i^n, +, \times, x^y; \mathbf{C}, \bar{\mathbf{I}}, \mathbf{Ls}, \mathbf{V}].$$

Above, $+$, $\times$ and x^y denote the binary addition, multiplication and exponentiation (the later being undefined for negative bases).

Proof. We only need to show that we may obtain **E** using only restricted iteration, $+$, $\times$, x^y and the remaining operators. But looking at the proof of Proposition 4.5, we see that this is the case, but we additionally use restricted inverse $1/\cdot$, the characteristic of strict inequality $\chi_<$, the absolute value function $|\cdot|$, the Euclidean norm $\|\cdot\|$, and the floor function $\lfloor\cdot\rfloor$. However, all of these may be obtained from the basic functions: $\frac{1}{x} = x^{-1}$; the expressions given in Proposition 3.8 may still be used to give Kronecker's δ and Heaviside's Θ — and thus $\chi_<$ and $|\cdot|$; the Euclidean norm comes from $\sqrt{\cdot}$ and $\times$, and $\sqrt{x} = x^{\frac{1}{2}}$; finally, iterating the successor function $s(x) = x + 1$ we get $\lfloor x\rfloor = \bar{\mathbf{I}}(s)(0, x)$. In this way, we may obtain **E** using the shown function algebra. □

This specific function algebra was named, using the capital Greek letter I (iota), because it will be fundamental in most of the following results.

4.2. Totalisation operators and universality

We will solve the problem of universality in two steps. First, we show that any real recursive function can be extended to a total function. Then, we show that if there were a universal real recursive function, this would be impossible.

Definition 4.9. Given a function $f : \mathbb{R}^m \to \mathbb{R}^n$ in $\mathscr{F}$, its **totalisation** is a function $(\chi_f, \tau_f) : \mathbb{R}^m \to \mathbb{R}^{n+1}$, such that

$$\chi_f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in Dom(f), \\ 0 & \text{otherwise;} \end{cases} \qquad \tau_f(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in Dom(f), \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 4.10 ([22]). *If f is a real recursive function, then so is its totalisation.*

Proof. The proof is by structural induction on the algebra I. Because the basic functions are all total, the result is trivial for the basic case. If f is $\mathbf{C}(g, h)$, then the totalisation of g and h is real recursive, by the induction hypothesis, and then $\chi_f(\boldsymbol{x}) = \chi_g(\tau_h(\boldsymbol{x})) \times \chi_h(\boldsymbol{x})$, $\tau_f(\boldsymbol{x}) = \tau_g(\tau_h(\boldsymbol{x})) \times \chi_h(\boldsymbol{x})$ gives the result for composition. The result is also trivial for restricted iteration, because it maps total functions to total functions. Now, if f is $\mathbf{Ls}(g)$, then the totalisation of g must be real recursive, by our induction hypothesis. Then, for any $\boldsymbol{x}$,

$$f(\boldsymbol{x}) = \limsup_{y\to\infty} g(\boldsymbol{x}, y),$$

and we must have one of the two cases:

(i) $g(\boldsymbol{x}, y)$ is defined for every large enough y; in symbols,

$$\exists\tilde{y}(\forall y > \tilde{y})\ (\boldsymbol{x}, y) \in Dom(g);$$

(ii) $g(\boldsymbol{x}, y)$ is undefined for arbitrarily large y; in symbols,

$$\forall\tilde{y}(\exists y > \tilde{y})\ (\boldsymbol{x}, y) \notin Dom(g);$$

In the first case, then clearly $f(\boldsymbol{x}) = \limsup_{y\to\infty} \tau_g(\boldsymbol{x}, y)$, but in the second case, this might not be so. However, we have that

$$\limsup_{y\to\infty} \left[\tau_g(\boldsymbol{x}, y) + (1 - \chi_g(\boldsymbol{x}, y))y\right]$$

will be defined if and only if $f(\boldsymbol{x})$ is defined,[7] and will be equal to $f(\boldsymbol{x})$ if it is indeed defined. So we may set

$$\chi_f(\boldsymbol{x}) = \eta^s(\tau_g + (1 - \chi_g) \times U^{n+1}_{n+1})(\boldsymbol{x})$$
$$\tau_f(\boldsymbol{x}) = \chi_f(\boldsymbol{x}) \times \limsup_{y\to\infty} \left(\chi_f(\boldsymbol{x}) \times \tau_g(\boldsymbol{x}, y)\right).$$

The induction step for aggregation is very simple: if f is $\mathbf{V}(g, h)$, we may use the induction hypothesis and set $\chi_f(\boldsymbol{x}) = \chi_g(\boldsymbol{x}) \times \chi_h(\boldsymbol{x})$ and $\tau_f = (\tau_g, \tau_h)$. □

Now we may solve the problem of universality. We will simplify our result by making the following assumption.

Assumption on Gödelisation. There is an effective enumeration d_e of all the descriptions in D_{I}. □

This assumption can be made into a theorem by any standard method of encoding. Whenever d_e is a good description (Definition 2.5), ϕ_e will denote the described function, and e is said to be a **code** of ϕ_e. We may then specialise Definition 2.16 into the following form.

7 Recall that we treat $+\infty$, $-\infty$ and $\perp$ in the same way. If this is somehow confusing, remember that we may use the sigmoidal function, as we did in order to prove Propositions 3.16 and 3.21.

Definition 4.11. A real recursive function $\Psi : \mathbb{R}^{m+1} \to \mathbb{R}^n$ is called **universal** if for every $e \in \mathbb{N}$, $\boldsymbol{x} \in \mathbb{R}^m$, we have

$$\Psi(e, \boldsymbol{x}) \simeq \phi_e(\boldsymbol{x})$$

whenever d_e is a good description of an m-ary function with n components (decidability of this condition allows us distinguish such cases).

Theorem 4.12 ([22,30]). *There is no universal real recursive function.*

Proof. A diagonal argument, very similar in nature to its counterparts in classical recursion theory, will give us *reductio ad absurdum*. For clarity we present only the case when $m = n = 1$, but the argument may easily be extended. Suppose that there was a universal real recursive function $\Psi : \mathbb{R}^2 \to \mathbb{R}$. Then by the previous theorem we could find its real recursive totalisation χ_Ψ and τ_Ψ, and the function given by

$$g(x) = \log(1 - \chi_\Psi(x, x)) = \log(1 - \chi_{\phi_x}(x)) = \begin{cases} 1 & \text{if } x \notin Dom(\phi_x), \\ \perp & \text{otherwise;} \end{cases}$$

would be a real recursive function of arity 1. So let e be a code of g. We have that $e \in Dom(g)$ if and only if $e \notin Dom(\phi_e)$, which is the contradiction we sought. □

5. Understanding REC($\mathbb{R}$)

We have shown in Section 3.3 that REC($\mathbb{R}$) is effectively closed for the **Sup** and **Inf** search operators. We may strengthen this result in the following way.

Theorem 5.1 ([21]). *The class of real recursive functions is given by the function algebra*

$$\mathrm{REC}(\mathbb{R}) = [-1^n, 0^n, 1^n, \mathrm{U}_i^n, +, \times, x^y; \mathbf{C}, \bar{\mathbf{I}}, \mathbf{Sup}, \mathbf{V}].$$

Proof. Let $\mathcal{A}$ denote the given algebra. All we have done was to replace the infinite supremum limit by the supremum in the algebra I. We know that I is closed for the supremum operator, and so all we need to show is that we may obtain the infinite supremum limit in the algebra $\mathcal{A}$. This algebra is also closed for the infimum, because $\mathbf{Inf}(f) = -\mathbf{Sup}(-f)$. But by definition,

$$\limsup_{y\to\infty} f(\boldsymbol{x}, y) = \inf_{y\in\mathbb{R}} \sup_{z>y} f(\boldsymbol{x}, z) = \inf_{y\in\mathbb{R}} \sup_{z\in\mathbb{R}} f(\boldsymbol{x}, z^2 + y),$$

and so $\mathcal{A}$ is also closed for **Ls**. □

We therefore reduce our original function algebra to a fairly trimmed down inductive definition. However, in this section we will show that the expressive power of this function algebra is much greater than what was anticipated. In Section 5.1 we will introduce the analytical hierarchy of predicates, and show that the graph of any real recursive function is in this hierarchy. In 5.2 we will show the converse, that any function with a graph in the analytical hierarchy must be real recursive. The expressive power of the analytical hierarchy is evidently great, and so this result explains why it seems to be so hard to find a function which is not real recursive.

5.1. The analytical hierarchy

The analytical hierarchy is a hierarchy of predicates of second-order arithmetic, and is studied in a variety of contexts. It was originally devised by Lusin (1925) for the then-incipient field of descriptive set theory and discovered independently by Kleene (1955) in the study of recursion on higher types. The name 'analytical' is used because second-order arithmetic allows for the formalisation of elementary analysis.

We present the analytical hierarchy of predicates, and relate it with the η-hierarchy.

Definition 5.2. A predicate P over real and natural numbers is called **recursive** if there is a partial recursive functional F such that

$$F(\boldsymbol{x}; \boldsymbol{a}) = \begin{cases} 1 & \text{if } P(\boldsymbol{x}, \boldsymbol{a}) \text{ holds,} \\ 0 & \text{otherwise.} \end{cases}$$

A predicate Q over real and natural numbers is called **arithmetical** if it is given using natural number quantifiers over a recursive predicate, i.e., if for some recursive predicate P,

$$Q(\boldsymbol{x}, \boldsymbol{a}) \iff (\forall b_1)(\exists b_2) \ldots (\forall b_{n-1})(\exists b_n) P(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}).$$

Definition 5.3. The **analytical hierarchy of predicates** consists of three $\mathbb{N}$-indexed families of predicates over real and natural numbers:

(i) Σ_0^1 is the class of arithmetical predicates, and $\Pi_0^1 = \Sigma_0^1$.
(ii) Σ_{n+1}^1 is the class of predicates given by $\exists y\, P(\boldsymbol{x}, y, \boldsymbol{a})$, with P in Π_n^1.

(iii) Π^1_{n+1} is the class of predicates given by $\forall y\ P(\boldsymbol{x}, y, \boldsymbol{a})$, with P in Σ^1_n.
(iv) $\Delta^1_n = \Sigma^1_n \cap \Pi^1_n$.

We will call **analytical** to the predicates in the analytical hierarchy. We write Δ^1_ω to stand for $\cup_{n\in\mathbb{N}}\Delta^1_n$, which is exactly the set of all analytical predicates. We will make abundant use of the following result [cf. [32, p. 377]].

Proposition 5.4. (a) *Σ^1_{n+1} is closed for existential quantification over $\mathbb{R}$.*
(b) *Π^1_{n+1} is closed for universal quantification over $\mathbb{R}$.*
(c) *Π^1_{n+1} and Σ^1_{n+1} are closed for existential and universal quantification over $\mathbb{N}$.*
(d) *We may exchange quantifiers over $\mathbb{N}$ with quantifiers over $\mathbb{R}$, i.e.,*
 (di) *If $P \in \Sigma^1_n$ then some $\tilde{P}$ also in Σ^1_n is such that $\forall a\ P \iff \forall x\ \tilde{P}$.*
 (dii) *If $P \in \Pi^1_n$ then some $\tilde{P}$ also in Π^1_n is such that $\exists a\ P \iff \exists x\ \tilde{P}$.*

Recall the following definition from Section 2.5.

Definition 5.5. The **graph** of a function $f : \mathbb{R}^m \to \mathbb{R}^n$, denoted G_f, is the $(n+m)$-ary predicate given by

$$G_f(\boldsymbol{z}, \boldsymbol{x}) \iff \boldsymbol{x} \in Dom(f) \wedge \boldsymbol{z} = f(\boldsymbol{x}).$$

Definition 5.6. We say that a function $f : \mathbb{R}^m \to \mathbb{R}^n$ is in Σ^1_k if its graph is in Σ^1_k. Similarly for Π^1_k and Δ^1_k.

We know that quantifiers may be used to express a rich variety of mathematical ideas, and so we expect that there are many functions in the analytical hierarchy.

Proposition 5.7. *The functions 1^n, $\bar{1}^n$, 0^n, U^n_i, $+$, $\times$, x^y, $|\cdot|$ and $\lfloor\cdot\rfloor$, as well as the predicates of equality and inequality over the reals, are in Δ^1_0.*

Proof. We begin by showing that there is a recursive way to decide the predicate over the reals given by the expression 'x and y are not different up to the nth digit', written $x =_n y$. An algorithm to decide this predicate needs to solve the ambiguity of the representation of a real number by binary expansion, and we can make it work in the following way: given two real numbers x, y and a natural number n, we obtain the first n digits of the two reals and verify if they are the same. If they are, then we decide that $x =_n y$. If the digits are not equal we consider the first different digit — one is 0 and the other 1 — and check if the digits after the 0 are all 1's and the digits after the 1 are all 0's.[8] If so, then we decide that $x =_n y$, and we decide that $x \neq_n y$ otherwise. The predicate of real number equality is then given by: $\forall n\ \ x =_n y$, which is in Δ^1_0. For the function $+$, we define a predicate, of the expression $z =_n x + y$, that decides if $z = x + y$ for the first n digits of z, x and y. This function computes the sum of the truncations of x and y to the nth fractionary digit and checks if resulting rational number coincides with z to the nth digit using the method shown above. If so, the function is valued 1, and 0 otherwise. Now we have that $z = x + y$ if and only if $\forall n\ z =_n x + y$, which is Δ^1_0. The proof is similar for the remaining operations. □

A single real number can code any finite tuple of real numbers by alternating the digits of the real numbers in the tuple (we will make use of this fact in the next section). In this sense, we write $y_{n,i}$ to stand for the ith real number in the n-ary tuple coded by y. For an m-ary tuple $\boldsymbol{y}$, we write $\boldsymbol{y}_{n,i}$ to stand for the tuple $((y_1)_{n,i}, \ldots, (y_m)_{n,i})$. Then it is not hard to see that if some n-ary predicate P is in Δ^1_n (or Σ^1_n, or Π^1_n), then the $(n+1)$-ary predicate $\tilde{P}$ given by

$$\tilde{P}(\boldsymbol{y}, n) \iff (\forall i \leqslant n) P(\boldsymbol{y}_{n,i})$$

is also in Δ^1_n (resp. Σ^1_n, Π^1_n).

Proposition 5.8 ([21]). *All real recursive functions belong to the analytical hierarchy, in the sense of Definition 5.6.*

Proof. The result is proved by induction on the structure of REC($\mathbb{R}$) presented in Proposition 5.1. Proposition 5.7 gives us the result for the atomic functions. Proposition 5.4 will suffice to show closure under the operators. If f and g are in Σ^1_n, then $\mathbf{C}(f, g)$ is in Σ^1_n, since:

$$\boldsymbol{z} = \mathbf{C}(f, g)(\boldsymbol{x}) \iff \exists \boldsymbol{y}\ \boldsymbol{z} = f(\boldsymbol{y}) \wedge \boldsymbol{y} = g(\boldsymbol{x}).$$

Let f be an m-ary total locally Lipschitz function with m components in Σ^1_n. Then $\bar{\mathbf{I}}(f)$ is in Σ^1_n, since $\boldsymbol{z} = \bar{\mathbf{I}}(f)(\boldsymbol{x}, y)$ if and only if

$$\exists \boldsymbol{w} \exists k \left[k = \lfloor|y|\rfloor \wedge \boldsymbol{w}_{k,1} = f(\boldsymbol{x}) \wedge ((\forall i \leqslant k))[\boldsymbol{w}_{k,i+1} = f(\boldsymbol{w}_{k,i})] \wedge \boldsymbol{z} = \boldsymbol{w}_{k,k}\right].$$

If f is an $(m+1)$-ary function in Σ^1_n, then $\mathbf{Sup}(f) \in \Pi^1_{n+1} \subseteq \Sigma^1_{n+2}$, since

$$\boldsymbol{z} = \mathbf{Sup}(f)(\boldsymbol{x}) \iff (\forall i \leqslant m)\forall y\ z_i \geqslant (f(\boldsymbol{x}, y))_i \wedge (\forall \tilde{z} < z_i)\exists \tilde{y}\ \tilde{z} < (f(\boldsymbol{x}, \tilde{y}))_i.$$

Furthermore, if f, g are in Σ^1_n then $\mathbf{V}(f, g)$ is trivially also in Σ^1_n. □

---

[8] E.g. $x = 101.1\underline{1}0000$ and $y = 101.1\underline{0}1111$, where the first different digit is underlined.

5.2. Real recursive functions and the analytical hierarchy

In this section, we show one of the most important results of this text:

Theorem 5.9 ([21]). REC($\mathbb{R}$) *is the class of functions with a graph in the analytical hierarchy, i.e.,*

$$\mathrm{REC}(\mathbb{R}) = \{f : \text{the predicate given by } \boldsymbol{z} = f(\boldsymbol{x}) \text{ is in } \Delta^1_\omega\}.$$

This will be carried out in a few steps. We have already shown in the previous subsection that every real recursive function has a graph in the analytical hierarchy. We will now prove (I) that every predicate in the analytical hierarchy has a real recursive characteristic, and (II) that if the graph of a function has a real recursive characteristic then the function itself is real recursive.

Proposition 5.10 ([27]). *The characteristic of every predicate $P \in \Pi^1_1$ has a real recursive extension.*

Proof. We use the normal form theorem [32, p. 380] for Π^1_1 predicates, which states that $P \in \Pi^1_1$ if and only if some recursive predicate R verifies

$$P(\boldsymbol{x}, \boldsymbol{a}) \iff \forall y \exists b\, R(\boldsymbol{x}, y; \boldsymbol{a}, b).$$

But then, setting $Q(\boldsymbol{x}, y; \boldsymbol{a}) \iff \exists b\, R(\boldsymbol{x}, y; \boldsymbol{a}, b)$, we get a predicate $Q \in \Sigma^0_1 \subset \Delta^0_2$.[9] From Shoenfield's limit lemma [32, p. 373], there must then be a recursive functional G such that

$$\lim_{b \to +\infty} G(\boldsymbol{x}, y; \boldsymbol{a}, b) = \chi_Q(\boldsymbol{x}, y; \boldsymbol{a}) = \begin{cases} 1 & \text{if } \exists b\, R(\boldsymbol{x}, y; \boldsymbol{a}, b) \text{ holds} \\ 0 & \text{otherwise.} \end{cases}$$

Above, the variable b ranges over the natural numbers. Therefore, by Proposition 3.46, G must have a real recursive extension g, and so the characteristic of Q:

$$\lim_{z \to +\infty} g(\boldsymbol{x}, y, \boldsymbol{a}, \lfloor z \rfloor) = \chi_Q(\boldsymbol{x}, y; a)$$

must also have a real recursive extension. We then set

$$\chi_P(\boldsymbol{x}, \boldsymbol{a}) = \inf_{y \in \mathbb{R}} \chi_Q(\boldsymbol{x}, y, \boldsymbol{a}) = \begin{cases} 1 & \text{if } P(\boldsymbol{x}, \boldsymbol{a}) \text{ holds} \\ 0 & \text{otherwise.} \quad \square \end{cases}$$

Proposition 5.11 ([21]). *The characteristic of every predicate P in the analytical hierarchy is real recursive.*

Proof. All predicates in $\Delta^1_0 \subset \Pi^1_1$ have real recursive characteristics, by the previous proposition. We now show that if P is an $(n + 1)$-ary predicate with a real recursive characteristic χ_P, then there are real recursive characteristics of the predicates given by $\forall y P(\boldsymbol{x}, y)$ and $\exists y P(\boldsymbol{x}, y)$. We have shown in Proposition 3.31 that if a function is real recursive, then so is its supremum and infimum over $\mathbb{R}$. So we have that $\forall y P(\boldsymbol{x}, y)$ if and only if $\mathbf{Inf}(\chi_P)(\boldsymbol{x}) = 1$ and that $\exists y P(\boldsymbol{x}, y)$ if and only if $\mathbf{Sup}(\chi_P)(\boldsymbol{x}) = 1$. In this way we conclude that all analytical predicates have real recursive characteristics. $\square$

The proof of (II) is easy for scalar functions.

Proposition 5.12 ([21]). *Let χ_f denote the characteristic function of the graph of $f : \mathbb{R}^m \to \mathbb{R}$, i.e.,*

$$\chi_f(z, \boldsymbol{x}) = \begin{cases} 1 & \text{if } z = f(\boldsymbol{x}) \\ 0 & \text{otherwise.} \end{cases}$$

If χ_f is real recursive, then so is f.

Proof. We construct a search operator, somewhat like minimalisation, but with the whole $\mathbb{R}$ as search domain. Consider again the function $\sigma(x) = \frac{e^x}{1+e^x}$ and its inverse $\sigma^{-1}(y) = \log(y) - \log(1 - y)$. The function σ surjectively maps $\mathbb{R}$ into (0, 1). So let

$$F(\boldsymbol{x}, z) = (1 - \chi_f(z, \boldsymbol{x})) + \chi_f(z, \boldsymbol{x})\sigma(z) = \begin{cases} \sigma(z) & \text{if } z = f(\boldsymbol{x}), \\ 1 & \text{otherwise.} \end{cases}$$

We may then set

$$f(\boldsymbol{x}) = \sigma^{-1}(\mathbf{Inf}(F)(\boldsymbol{x})). \quad \square$$

Because every graph of every function in the analytical hierarchy must be real recursive (Proposition 5.11), and if the graph of such a scalar function is real recursive, then so is the function itself (Proposition 5.12), we get the following.

Corollary 5.13. *Every scalar function in the analytical hierarchy is real recursive.*

[9] These are levels in the arithmetical hierarchy, which is defined in a similar way to the analytical hierarchy; cf. [32, p. 367].

Notice that the class of functions with a graph in the analytical hierarchy is closed for component selection, and this immediately gives us Theorem 5.9. But we may make a more explicit proof.

Definition 5.14. The function $\tilde{\gamma}$ is the injection from $(0, 1)^2$ to $(0, 1)$ given by

$$\tilde{\gamma}(x, y) = \sum_{i=1}^{\infty} D(x, i)2^{-2i+1} + D(y, i)2^{-2i},$$

where $D(x, i)$ denotes the ith digit of the binary expansion of x; $\tilde{\gamma}_1$ and $\tilde{\gamma}_2$ denote each component of the inverse of $\tilde{\gamma}$, given by

$$\tilde{\gamma}_1(z) = \sum_{i=1}^{\infty} D(z, 2i-1)2^{-i}, \qquad \tilde{\gamma}_2(z) = \sum_{i=1}^{\infty} D(z, 2i)2^{-i}.$$

Then we set $\gamma, \gamma_1, \gamma_2$ to be given by

$$\gamma(x, y) = \tilde{\gamma}(\sigma(x), \sigma(y)), \qquad \gamma_1(z) = \sigma^{-1}(\tilde{\gamma}_1(z)), \qquad \gamma_2(z) = \sigma^{-1}(\tilde{\gamma}_2(z)).$$

We may easily see that γ, γ_1 and γ_2 are scalar functions in the analytical hierarchy, forming an injection from $\mathbb{R}^2$ to $(0, 1)$. So by Corollary 5.13, with the use of composition and aggregation, we conclude the following.

Proposition 5.15 ([26]). *The functions γ, γ_1 and γ_2 are real recursive. Furthermore, for every n there are two real recursive functions $\boldsymbol{\gamma}_n : \mathbb{R}^n \to (0, 1)$ and its inverse $\boldsymbol{\gamma}_n^{-1} : (0, 1) \to \mathbb{R}^n$ forming an injection from $\mathbb{R}^n$ to $(0, 1)$.*

We could construct such functions H_3 by using sums and infinite limits. The following final corollary implies Theorem 5.9.

Corollary 5.16. *Let χ_f denote the characteristic function of the graph of $f : \mathbb{R}^m \to \mathbb{R}^n$. If χ_f is real recursive, then so is f.*

Proof. Consider the function $\tilde{f}$, given by

$$\tilde{f}(\boldsymbol{x}) = \boldsymbol{\gamma}_n(f(\boldsymbol{x})).$$

Then $\tilde{f}$ is scalar, and its characteristic function is given by

$$\chi_{\tilde{f}}(z, \boldsymbol{x}) = \begin{cases} 1 & \text{if } z \in (0, 1) \text{ and } \chi_f(\boldsymbol{\gamma}_n^{-1}(z), \boldsymbol{x}) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

This expression gives a real recursive function, and so $\tilde{f}$ is real recursive by Proposition 5.12. But then $f(\boldsymbol{x}) = \boldsymbol{\gamma}_n^{-1}(\tilde{f}(\boldsymbol{x}))$, and therefore f is also real recursive. □

6. Towards solving the problem of collapse

Consider the rank hierarchy for the infinite limit operator under the algebra I:

Definition 6.1. The **ι-hierarchy** is the rank hierarchy for the limit operator under the algebra I for REC($\mathbb{R}$). We use I_k to denote the nth level of this hierarchy. In symbols,

$$\mathrm{I}_k = H_k^{\mathrm{I},\mathbf{Ls}} = \{f \in \mathrm{REC}(\mathbb{R}) : \mathrm{rk}^{\mathrm{I}}_{\mathbf{Ls}}(f) \leqslant k\}.$$

A clearer picture for this hierarchy may be obtained from the following corollary of Proposition 2.15.

Corollary 6.2. *The ι-hierarchy is inductively given by:*

(i) $\mathrm{I}_0 = [-1^n, 0^n, 1^n, U_i^n; \mathbf{C}, \bar{\mathbf{I}}, \mathbf{V}]$,
(ii) $\tilde{\mathrm{I}}_k = \mathrm{I}_k \cup \{\mathbf{Ls}(f) : f : \mathbb{R}^{m+1} \to \mathbb{R}^n$ *is in* $\mathrm{I}_k\}$, *and*
(iii) $\mathrm{I}_{k+1} = [\tilde{\mathrm{I}}_k; \mathbf{C}, \bar{\mathbf{I}}, \mathbf{V}]$.

The purpose of this section is to show that the ι-hierarchy does not collapse. In Section 6.1 we show that REC($\mathbb{R}$) is closed for unrestricted iteration, and that there is a real recursive way to manipulate stacks of functions. In Section 6.2 we construct restrictions of universal real recursive functions for bounded levels in the ι-hierarchy. In Section 6.3 we will conclude that the ι-hierarchy does not collapse, and explain why this result does not immediately imply a similar result for the η-hierarchy.

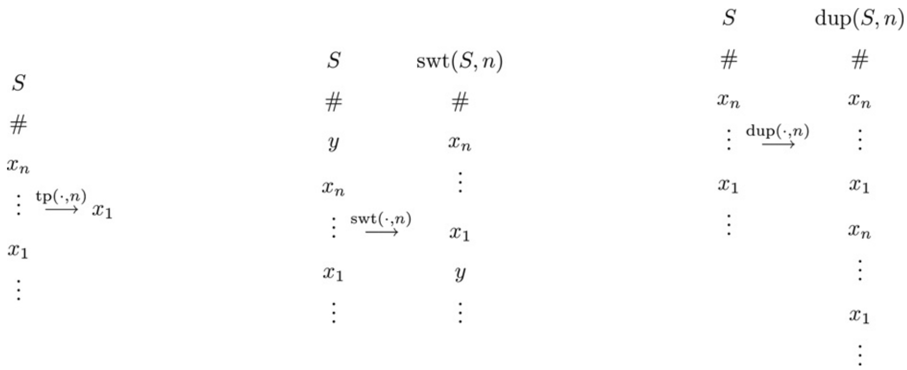


Fig. 8. Applying tp, swt and dup to a stack S.

6.1. Unrestricted iteration and stacks

From the previous section we immediately see that REC($\mathbb{R}$) is closed for unrestricted iteration.

Definition 6.3. The **iteration operator I** maps any n-ary function f with n components into an $(n+1)$-ary function with n components $\mathbf{I}(f)$, given by

$$\mathbf{I}(f)(\boldsymbol{x}, t) = f^{\lfloor|t|\rfloor}(\boldsymbol{x}) = \underbrace{f \circ f \circ \cdots \circ f}_{\lfloor|t|\rfloor \text{ times}}(\boldsymbol{x}).$$

Notation 6.4. *When the given number of iterations is expected to be a natural number, we omit the flooring and absolute value symbols in the number of iterations. Then $f^n(\boldsymbol{x})$ is an abbreviation to $f^{\lfloor|n|\rfloor}(\boldsymbol{x})$.*

Proposition 6.5. REC($\mathbb{R}$) *is effectively closed under* **I**.

Proof. The analytical hierarchy is easily closed under **I**, by using the same expression we have used in the proof of Proposition 5.8 to obtain unrestricted iteration. □

Using this unrestricted iteration operator, we may find a way to manipulate stacks of real values by using the pairing function γ of the previous section. We denote a stack of real numbers by $\#x_n \ldots x_1$, where # marks the top of the stack. We represent an empty stack by the number 0. The stack $\#x_n \ldots x_1$ is represented with the number $\gamma(x_n, \ldots \gamma(x_1, 0) \ldots)$. Recall that the range of γ is $(0, 1)$, and so there is no risk of confusing an empty stack with a non-empty stack.

We can then define four *basic stack manipulation functions*. The psh function, which pushes a value on top of the stack, is given by $\text{psh}(S, x) = \gamma(x, S)$. The pop function removes the top of the stack: $\text{pop}(S) = \gamma_2(S)$. The top function gives the value on the top of the stack, and 0 if the stack is empty: $\text{top}(S) = (1 - \delta(S))\gamma_1(S + \frac{1}{2}\delta(S))$. The emp function gives 1 if the stack is empty and 0 otherwise: $\text{emp}(S) = \delta(S)$. We abbreviate $\text{top}(\text{pop}^{n-1}(S)) \equiv \text{tp}(S, n)$. More complex stack manipulation functions can be defined using the four basic functions. The function swt, for instance, pushes the top of the stack into the $(n+1)$th position: $\text{swt}(S, n) = U_1^4(\mathbf{I}(f)(S, 0, n, 0, 2n+2))$, where

$$f(S_1, S_2, n, r) = \begin{cases} (\text{pop}(S_1), 0, n, \text{top}(S_1)) & \text{if } r = 0 \text{ and } n > 0, \\ (\text{pop}(S_1), \text{psh}(S_2, \text{top}(S_1)), n-1, r) & \text{if } r \neq 0 \text{ and } n > 0, \\ (\text{psh}(S_1, r), S_2, 0, 0) & \text{if } r \neq 0 \text{ and } n = 0, \\ (\text{psh}(S_1, \text{top}(S_2)), \text{pop}(S_2), 0, 0) & \text{if } r = 0 \text{ and } n = 0. \end{cases}$$

Notice that the above definition by cases can be *implemented* using the characteristics of equality and inequality, along with products and sums. The function dup duplicates the top n elements of the stack: $\text{dup}(S, n) = U_1^4(\mathbf{I}(g)(S, 0, 0, n, 3n+1))$, where

$$g(S_1, S_2, S_3, n) = \begin{cases} (\text{pop}(S_1), \text{psh}(S_2, \text{top}(S_1)), 0, n-1) & \text{if } n > 0, \\ (S_1, S_2, S_2, 0) & \text{if } n = 0 \text{ and emp}(S_3), \\ (\text{psh}(S_1, \text{top}(S_2)), \text{pop}(S_2), S_3, 0) & \text{if } n = 0 \text{ and not emp}(S_2), \\ (\text{psh}(S_1, \text{top}(S_3)), S_2, \text{pop}(S_3), 0) & \text{if } n = 0 \text{ and not emp}(S_3). \end{cases}$$

The effect of $\text{tp}(S, n)$, $\text{swt}(S, n)$ and $\text{dup}(S, n)$ on the stack S are illustrated on Fig. 8.

In the next section, we will construct the universal real recursive functions $\Psi_k^{m,n}$ for m-ary, n-component functions in I_k, by manipulating an analogue of an *execution stack*.

6.2. Universal level-bounded functions

We show that the ι-hierarchy does not collapse by constructing functions $\Psi_k^{m,n}$ which are universal for I_k.

Definition 6.6. A function $\Psi_k^{m,n} : \mathbb{R}^{m+1} \to \mathbb{R}^n$ is called a **universal function for** I_k if for every good description $d_e^{m,n}$ with $\mathrm{rk}(d_e^{m,n}) \leqslant k$, and every $\boldsymbol{x} \in \mathbb{R}^m$,

$$\Psi_k^{m,n}(e, \boldsymbol{x}) \simeq \phi_e^{m,n}(\boldsymbol{x}).$$

Theorem 6.7 ([22]). *For any m, n, k, there is a universal function $\Psi_k^{m,n}$ for I_k.*

Proof. We construct functions that simulate any real recursive function step-by-step, given its code, by maintaining two stacks. On the first stack we keep real values and on the second stack we maintain codes of descriptions of real recursive functions or of stack manipulation instructions. A switch instruction $\mathrm{swt}(\cdot, n)$ is represented by the number $3n + 1$, and a duplicate instruction $\mathrm{dup}(\cdot, n)$ is represented by $3n+2$. The description d_e is represented by $3e$. If the description in the top of the second stack describes an n-ary function, it is expected that n real values are in the first stack, each corresponding to one argument, with the last argument on top. To implement the aggregation operator, it will be necessary to duplicate and switch the contents of the stack, and that is why we encode the swt and dup instructions. Let $\tilde{\Psi}_0$ be given by:

$$\tilde{\Psi}_0(S_1, S_2) = \begin{cases} (\mathrm{psh}(\mathrm{pop}^n(S_1), 1), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{fun}_{1^n} \rangle, \\ (\mathrm{psh}(\mathrm{pop}^n(S_1), -1), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{fun}_{-1^n} \rangle, \\ (\mathrm{psh}(\mathrm{pop}^n(S_1), 0), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{fun}_{0^n} \rangle, \\ (\mathrm{psh}(\mathrm{pop}^n(S_1), \mathrm{tp}(S_1, n-i+1)), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{fun}_{U_i^n} \rangle, \\ (\mathrm{psh}(\mathrm{pop}^2(S_1), \mathrm{tp}(S_1, 2) + \mathrm{top}(S_1)), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{fun}_{+} \rangle, \\ (\mathrm{psh}(\mathrm{pop}^2(S_1), \mathrm{tp}(S_1, 2) \times \mathrm{top}(S_1)), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{fun}_{\times} \rangle, \\ (\mathrm{psh}(\mathrm{pop}^2(S_1), \mathrm{tp}(S_1, 2)^{\mathrm{top}(S_1)}), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{fun}_{x^y} \rangle, \\ (S_1, \mathrm{psh}(\mathrm{psh}(\mathrm{pop}(S_2), 3e_1), 3e_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{Op}_{\mathbf{C}}, d_{e_1}, d_{e_2} \rangle, \\ (\mathrm{pop}(S_1), \mathrm{psh}^{\lfloor |\mathrm{top}(S_1)| \rfloor}(\mathrm{pop}(S_2), 3e)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{Op}_{\bar{\mathbf{I}}}, d_e \rangle, \\ (S_1, \mathrm{aggr}(\mathrm{pop}(S_2), 3e_1, 3e_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{Op}_{\mathbf{V}}, d_{e_1}, d_{e_2} \rangle, \\ (\mathrm{swt}(S_1, n), \mathrm{pop}(S_2)) & \text{if } \mathrm{top}(S_2) \text{ is } 3n+1, \\ (\mathrm{dup}(S_1, n), \mathrm{pop}(S_2)) & \text{if } \mathrm{top}(S_2) \text{ is } 3n+2, \\ (S_1, S_2) & \text{if } \mathrm{emp}(S_2) \end{cases}$$

where for every m-ary descriptions d_{e_1} with n components and d_{e_2} with k components, $\mathrm{aggr}(S, 3e_1, 3e_2)$ carries out the following pushes to the stack S (the corresponding instructions are shown in parenthesis):

(i) push $3m + 2$ (duplicate the top m elements);
(ii) push $3e_1$ (apply the function described by e_1);
(iii) push $3(m + n - 1) + 1$ a total of n times (move the result of applying this function below the previously duplicated values);
(iv) push $3e_2$ (apply the function described by e_2).

By induction on the structure of I we conclude that if S_1 encodes a stack with the real numbers $x_n, \ldots, x_1, y_j, \ldots, y_1$, and S_2 encodes a stack with the numbers $3e, e_1, \ldots e_k$, where d_e describes an n-ary function with m components and $\mathrm{rk}(d_e) = 0$, then, by iterating $\tilde{\Psi}_0$, S_2 will eventually contain only $e_1, \ldots, e_k$ and then we will have m real numbers in the top of S_1, given by each component of $\phi_e(x_1, \ldots, x_n)$, followed by $y_j, \ldots, y_1$. This is illustrated by Fig. 9.

We may set

$$\Psi_0(S, e) = U_1^2(\lim_{z \to +\infty} \mathbf{I}(\tilde{\Psi}_0)(S, \mathrm{psh}(0, e)),$$

and then Ψ_0 is a universal function when the input and output is given using a stack. Now we set

$$(\Psi_0^{m,n}(e, \boldsymbol{x}))_i = \mathrm{tp}(\Psi_0(\mathrm{psh}(0, \boldsymbol{x}), 3e), i);$$

$$\begin{array}{cc} S_1 & S_2 \\ \# & \# \\ x_n & 3e \\ \vdots & e_1 \\ x_1 & \vdots \\ y_j & e_k \\ \vdots & \\ y_1 & \end{array} \xrightarrow{\tilde{\Psi}_0} \dots \xrightarrow{\tilde{\Psi}_0} \begin{array}{cc} S_1 & S_2 \\ \# & \# \\ (\phi_e(x_1,\dots,x_n))_m & e_1 \\ \vdots & \vdots \\ (\phi_e(x_1,\dots,x_n))_1 & e_k \\ y_j & \\ \vdots & \\ y_1 & \end{array}$$

Fig. 9. Stack manipulation by $\tilde{\Psi}_0$.

obtaining a universal function for I_0; above, $\mathrm{psh}(0, \boldsymbol{x})$ abbreviates $\mathrm{psh}(\dots \mathrm{psh}(0, x_1), \dots x_m)$.

Now, for every $k \geqslant 1$, we define $\tilde{\Psi}_k$ as

$$\tilde{\Psi}_k(S_1, S_2) = \begin{cases} (\limsup_{y\to\infty} \Psi_{k-1}(\mathrm{psh}(S_1, y), 3e), \mathrm{pop}(S_2)) & \text{if } \frac{1}{3}\mathrm{top}(S_2) \text{ is } \langle \mathtt{Op}_{\mathbf{Ls}}, d_e\rangle, \\ \tilde{\Psi}_0(S_1, S_2) & \text{otherwise;} \end{cases}$$

$$\Psi_k(S, e) = U_1^2(\lim_{z\to+\infty} \mathbf{I}(\tilde{\Psi}_k)(S, \mathrm{psh}(0, e));$$

and

$$(\Psi_k^{m,n}(e, \boldsymbol{x}))_i = \mathrm{tp}(\Psi_k(\mathrm{psh}(0, \boldsymbol{x}), 3e), i).$$

In this way, $\Psi_k^{m,n}$ is in some bounded level of the I-hierarchy (not necessarily I_k), and simulates any real recursive function in I_k when given one of its codes with rank bounded by k. □

6.3. Conclusions for the problem of collapse

Notice that the existence of our universal functions for I_k will not give us the result of non-collapse. We will need the following concept.

Definition 6.8. A number $e \in \mathbb{N}$ is called a **low-rank code** of ϕ_e if $\mathrm{rk}(d_e) = \mathrm{rk}(\phi_e)$.

And now we show:

Proposition 6.9 ([22]). *There is no real recursive function which restricts a universal function to low-rank codes, i.e., there is no real recursive function $\psi^{m,n}$ such that if e is a low-rank code then*

$$\psi^{m,n}(e, \boldsymbol{x}) \simeq \phi_e(\boldsymbol{x}) \quad \textit{for all } \boldsymbol{x} \in \mathbb{R}^m,$$

whenever d_e is a good description of an m-ary function ϕ_e with n components.

Proof. We take the same expression for g of the proof of Proposition 4.12, now using a totalisation of $\psi^{m,n}$. We then choose a low-rank code e of g, rather than any code of g. □

Now we may conclude:

Theorem 6.10 ([22]). *The ι-hierarchy does not collapse.*

Proof. Suppose, by contradiction, that it would collapse. Then for some number k we would have $\mathrm{REC}(\mathbb{R}) \subseteq \mathrm{I}_k$. But then every low-rank code would have a rank of at most k, and we could set $\psi^{m,n} = \Psi_k^{m,n}$ and obtain a universal function restricted to low-rank codes. This contradicts the previous proposition, and so we are forced to conclude that the ι-hierarchy does not collapse. □

But how about the η-hierarchy? The non-collapsing character of the ι-hierarchy does not imply the non-collapse of the η-hierarchy, because we have not shown that every bounded level in the η-hierarchy is fully contained on some bounded level in the ι-hierarchy. In fact, we use two infinite limits for each differential recursion in our proof that $\mathrm{REC}(\mathbb{R}) \subseteq \mathrm{I}$, and so by nesting differential recursions in a description $d \in \mathcal{D}_{\mathrm{H}}$, our proof would give a description $\tilde{d} \in \mathcal{D}_{\mathrm{I}}$ with twice the rank as the number of nested differential recursions. We believe that this does not have to be the case, i.e., the η-hierarchy is collapse-equivalent to the ι-hierarchy, in the following sense:

Definition 6.11. Two hierarchies $\mathcal{A}_\omega$ and $\mathcal{B}_\omega$ are called **collapse-equivalent** if for all n there is a number m such that $\mathcal{A}_n \subseteq \mathcal{B}_m$ and $\mathcal{B}_n \subseteq \mathcal{A}_m$.

We could easily prove the following proposition.

Proposition 6.12. *The ι-hierarchy and the analytical hierarchy are collapse-equivalent.*

We conclude this section with an open conjecture, once believed to have been proven true [22]. Now we know that the proof was flawed. The conjecture implies the non-collapse of the η-hierarchy, and this is the justification for this section's name.

Conjecture 6.13. *The η-hierarchy and the ι-hierarchy are collapse-equivalent.*

7. Conclusion

This paper gives a solid foundation for the theory of real recursive functions.

This text is entirely self-contained, and may be used for teaching.

Research in this field was sometimes troubled by unclear assumptions or imperfect proofs. Our article builds a robust theory by defining differential recursion supported on the concept of local Lipschitz continuity (Section 2.2), and by presenting a specific, quantifier based definition of the supremum limit (Section 2.5). Note that the results obtained previously [27,31] where completely sound: some freedom may be allowed with respect to the definition of the differential recursion operator, as well as the definition of the infinite limit operator, and the same class of functions will be obtained. However, these small differences in the definitions may or may not result in different limit hierarchies. We still do not know whether the differential recursion schemes presented in [24,27,22,19] result, or do not result, in collapse-equivalent limit hierarchies.

Our form of the theory is smooth enough to be taught for a wider group of the scientific community. Moreover, it creates the possibility to restart research of those problems involving real recursive functions which did not find satisfactory solutions in the past, namely the study of complexity classes over real-valued functions.

The following topics remain relevant to our research. Is the η-hierarchy (Section 3.2) collapse-equivalent to the ι-hierarchy (Section 6)? In any case, does the former collapse? What is the proper presentation of non-determinism in the setting of real recursive functions? How can we describe the classical complexity classes in the language of the real recursive functions, as in previous work [9,29,3]?

As is now, the theory seems to be promising subject of computation with continuous variables, and our results secure enough to develop a new sub-field of computability theory.

Acknowledgement

José Félix Costa and Bruno Loff were supported by Feder and FCT Plurianual 2007.

References

[1] Stephen Bellantoni, Predicative Recursion and Computational Complexity, Ph.D. Thesis, University of Toronto, 1992.
[2] Olivier Bournez, Emmanuel Hainry, Real recursive functions and real extensions of recursive functions, in: Maurice Margenstern (Ed.), Machines, Computations and Universality, MCU 2004, in: Lecture Notes in Computer Science, vol. 3354, Springer-Verlag, 2004, pp. 116–127.
[3] Olivier Bournez, Emmanuel Hainry, Elementarily computable functions over the real numbers and $\mathbb{R}$-sub-recursive functions, Theoretical Computer Science 348 (2–3) (2005) 130–147.
[4] Olivier Bournez, Emmanuel Hainry, Recursive analysis characterized as a class of real recursive functions, Fundamenta Informaticae 74 (4) (2006) 409–433.
[5] Michael Branicky, Universal computation and other capabilities of hybrid and continuous dynamical systems, Theoretical Computer Science 138 (1) (1995) 67–100.
[6] Vannevar Bush, Harold Hazen, The differential analyser, Journal of the Franklin Institute 212 (4) (1931) 447–488.
[7] Manuel Campagnolo, The complexity of real recursive functions, in: Cristian Calude, Michael Dinneen, Ferdinand Peper (Eds.), Unconventional Models of Computation, UMC 2002, in: Lecture Notes in Computer Science, vol. 2509, Springer-Verlag, 2002, pp. 1–14.
[8] Manuel Campagnolo, Cris Moore, José Félix Costa, Iteration, inequalities, and differentiability in analog computers, Journal of Complexity 16 (4) (2000) 642–660.
[9] Manuel Campagnolo, Cris Moore, José Félix Costa, An analog characterization of the Grzegorczyk hierarchy, Journal of Complexity 18 (4) (2002) 977–1000.
[10] Manuel Campagnolo, Kerry Ojakian, The elementary computable functions over the real numbers: Applying two new techniques, Archive for Mathematical Logic 46 (7–8) (2008) 593–627.
[11] Peter Clote, Computation models and function algebras, in: Handbook of Computability Theory, in: Studies in Logic and the Foundations of Mathematics, vol. 140, Elsevier, 1999, pp. 589–681 (Chapter 17).
[12] Rodney Driver, Introduction to Ordinary Differential Equations, Harper and Row, 1978.
[13] M.D. Gladstone, Simplifications of the recursion scheme, Journal of Symbolic Logic 36 (4) (1971) 653–665.
[14] Luís Mendes Gomes, Applications of real recursive infinite limits, Ph.D. Thesis, University of Azores, 2006.
[15] Luís Mendes Gomes, José Félix Costa, Hybrid finite computation, in: Barry Cooper, Benedikt Löwe, Andrea Sorbi (Eds.), Computation and Logic in the Real World, CiE 2007, 2007, pp. 178–185.
[16] Daniel Graça, Manuel Campagnolo, Jorge Buescu, Robust simulations of Turing machines with analytic maps and flows, in: New Computational Paradigms, CiE 2005, in: Lecture Notes in Computer Science, vol. 3526, Springer-Verlag, 2005, pp. 169–179.
[17] Daniel Graça, José Félix Costa, Analog computers and recursive functions over the reals, Journal of Complexity 19 (5) (2003) 644–664.
[18] Emmanuel Hainry, Modèles de calcul sur les réels résultats de comparaisons, Ph.D. Thesis, Institut National Polytechnique de Lorraine, 2006.
[19] Akitoshi Kawamura, Differential recursion and differentially algebraic functions, Archive preprint http://arxiv.org/abs/0704.0301, Revised version of Real primitive recursive functions and differential algebraicity, presented at Computability in Europe 2006, Logical approaches to computational barriers, 2007.

[20] Erwin Kreyszig, Introductory Functional Analysis With Applications, John Wiley & Sons, 1989.
[21] Bruno Loff, A functional characterisation of the analytical hierarchy, in: Barry Cooper, Benedikt Löwe, Andrea Sorbi (Eds.), Computability in Europe 2007: Computation and Logic in the Real World, 2007, pp. 247–256.
[22] Bruno Loff, José Félix Costa, Jerzy Mycka, Computability on reals, infinite limits and differential equations, Applied Mathematics and Computation 191 (2007) 353–371.
[23] Albert Meyer, Dennis Ritchie, The complexity of loop programs, in: Proceedings of the 22nd National Conference, Thompson Book Company, 1967, pp. 465–470.
[24] Cris Moore, Recursion theory on the reals and continuous-time computation, Theoretical Computer Science 162 (1) (1996) 23–44.
[25] Jerzy Mycka, Infinite limits and R-recursive functions, Acta Cybernetica 16 (1) (2003) 83–91.
[26] Jerzy Mycka, μ-recursion and infinite limits, Theoretical Computer Science 302 (2003) 123–133.
[27] Jerzy Mycka, José Félix Costa, Real recursive functions and their hierarchy, Journal of Complexity 20 (6) (2004) 835–857.
[28] Jerzy Mycka, José Félix Costa, The conjecture $P \neq NP$ presented by means of some classes of real functions, in: Arnold Beckmann, Ulrich Berger, Benedikt Löwe, John Tucker (Eds.), Logical Approaches to Computational Barriers, CiE 2006, in: Lecture Notes in Computer Science, vol. 3988, Springer-Verlag, 2006, pp. 47–57.
[29] Jerzy Mycka, José Félix Costa, The $P \neq NP$ conjecture in the context of real and complex analysis, Journal of Complexity 22 (2) (2006) 287–303.
[30] Jerzy Mycka, José Félix Costa, Undecidability over continuous-time, Logic Journal of the IGPL 14 (2006) 649–658.
[31] Jerzy Mycka, José Félix Costa, A new conceptual framework for analog computation, Theoretical Computer Science 374 (1–3) (2007) 277–290.
[32] Piergiorgio Odifreddi, Classical Recursion Theory, in: Studies in Logic and the Foundations of Mathematics, vol. 125, North Holland, 1989.
[33] Gerald Teschl, Ordinary differential equations and dynamical systems. Available at: http://www.mat.univie.ac.at/~gerald/ftp/book-ode/ode.pdf, 2007.
[34] Wolfgang Walter, Ordinary Differential Equations, Springer-Verlag, 1998.
[35] António Weinholtz, Equações Diferenciais — Uma Introdução, Universidade de Lisboa, Faculdade de Ciências, Departamento de Matemática, 2000.
[36] Michael Williams, A History of Computing Technology, 2nd ed., IEEE Computer Society Press, 1997.