

Using Approximation to Relate Computational Classes over the Reals

Manuel L. Campagnolo¹ and Kerry Ojakian²

¹ DM/ISA, Lisbon University of Technology and SQIG/IT Lisbon
mlc@math.isa.utl.pt

² SQIG/IT Lisbon and IST, Portugal
ojakian@math.ist.utl.pt

Abstract. We use our method of approximation to relate various classes of computable functions over the reals. In particular, we compare *Computable Analysis* to the two analog models, the *General Purpose Analog Computer* and *Real Recursive Functions*. There are a number of existing results in the literature showing that the different models correspond exactly. We show how these exact correspondences can be broken down into a two step process of *approximation and completion*. We show that the method of approximation has further application in relating classes of functions, exploiting the transitive nature of the approximation relation. This work builds on our earlier work with our method of approximation, giving more evidence of the breadth of its applicability.

1 Introduction

In short, the goal of this paper is to relate various computational models over the reals, using our notion of *approximation* as a unifying tool and language. *Computable Analysis* (originating with Grzegorzczuk [13]) is a model of computation in which the data consists of the real numbers and the computation proceeds in discrete time steps. From this point of view, a function is considered computable if (*roughly*) from approximations for the inputs we can compute (using, for example, a typical discrete Turing Machine) approximations for the outputs. Various models of *analog computation* also compute on the real numbers, but the computation can instead be argued to proceed in an analog manner. We will consider two analog models, Shannon's *General Purpose Analog Computer* (GPAC) [20] and Moore's *Real Recursive Functions* [16] (with some problems corrected by Costa and Mycka in [17] and [18]). The GPAC is an analog circuit model, which by the work of Graça and Costa [11] is equivalent to certain dynamical systems. The Real Recursive Functions are given by a function algebra in which the typical discrete operations of recursion are replaced by operations which yield the solution of a differential equation. Bournez and Hainry ([1], [2]) have related various classes of Computable Analysis and Real Recursive Functions; we have expanded upon their work in [8] and [7]. Bournez, Campagnolo, Graça, and Hainry [19] have related Computable Analysis to the GPAC. We will

use our method of approximation to provide a unified view of these results and show how this approach can facilitate the technical development.

The motivation for characterizing the functions of Computable Analysis can be seen as twofold. On the one hand this work can be seen as a kind of search for a Church-Turing thesis in the context of real computation. For computation on the natural numbers there is an agreed upon notion of what it means to be *computable*, bolstered by the impressive fact that a number of different models of computation (e.g. Turing Machines, recursive functions, while programs, etc) yield the same set of functions. There are various models of computation over the reals, but many lack that kind of correspondence. In the work surveyed here the authors have developed exact correspondences within real computation by modifying existing models in reasonable ways. Developing these correspondences should be an important step towards some kind of a Church-Turing thesis for real computation. Another motivation for finding these correspondences is the possibility of shedding light on the questions of classic complexity theory. In particular, the complexity class separations (such as the classic P versus NP question) for the naturals correspond to the analogous questions in Computable Analysis. Thus, interesting and useful correspondences to Computable Analysis could allow the use of different tools to attack these questions; for example, perhaps the P versus NP question would be reduced to a question about differential equations, allowing new tools to come into play. This latter line of thought is the motivation behind the work of Costa and Mycka [9], which discusses the equivalence of an analytic condition to $P = NP$.

In section 3 we show how the various results (i.e. relating Computable Analysis, the GPAC, and Real Recursive Functions) all fit into a similar two step pattern we call *approximation and completion*. To get the rough idea, suppose \mathcal{F} is the class of functions arising from one model, say Computable Analysis, and \mathcal{H} is the class of functions of another model, say the GPAC, and the goal is to show that $\mathcal{F} = \mathcal{H}$. The classes happen to be defined in such a way that we can isolate subclasses $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathcal{H}' \subseteq \mathcal{H}$, such that $\mathcal{F} = (\mathcal{F}')^*$ and $\mathcal{H} = (\mathcal{H}')^*$, where we use the star superscript to indicate some kind of “completion” operation. The approximation step involves showing, roughly, that $\mathcal{F}' \approx^{\mathcal{E}} \mathcal{H}'$, which means that for any function in one class, there is a function in the other one that approximates it with an accuracy dictated by \mathcal{E} (\mathcal{E} is some set of functions used to measure the accuracy). The completion step requires showing that from the approximate equality $\mathcal{F}' \approx^{\mathcal{E}} \mathcal{H}'$ we can derive the genuine equality $(\mathcal{F}')^* = (\mathcal{H}')^*$, i.e. $\mathcal{F} = \mathcal{H}$. On the technical side, this way of phrasing the problem allows us to bring in our tools involving the method of approximation. On the more philosophical side, we believe the approximation and completion approach makes important issues concerning these results more explicit. In our discussion on future work, in section 5, we discuss questions this raises.

In section 4, we show how approximations can be used to facilitate the technical work of relating different classes of functions. Supposing we are interested in the claim $\mathcal{F} \approx^{\mathcal{E}} \mathcal{H}$, the transitive property of the approximation relation can facilitate the proof. Rather than attempting to show the claim directly, which

could become cumbersome, we can develop an intermediary class \mathcal{G} and break the task down into two subtasks $\mathcal{F} \approx^\varepsilon \mathcal{G}$ and $\mathcal{G} \approx^\varepsilon \mathcal{H}$. In particular, we will show how this facilitates the elimination of non-analytic functions from models of computation, while maintaining the desired properties.

In section 2 we give a streamlined account of the notion of approximation and related concepts (the details are worked out in our papers [8] and [7], the latter one providing an improved development). In section 5 we discuss ideas for future work.

2 Technical Preliminaries

We provide an outline of the technical development from [8] and [7], where we consider the latter paper to be the improved version, and the one we will generally cite here. We start by discussing the approximation relation and then discuss its connections to function algebras.

2.1 Approximation

To develop formally the definition of approximation we will need to be able to talk about functions and their arguments in a precise way. Unless otherwise stated, a function has a finite number of real arguments and a single real number as an output. We let

$$\mathbf{Var} = \{x_i \mid i \in \mathbb{N}\}$$

be the **set of variables**. A function is always associated with a *finite set* of such variables and the values of the functions result from assigning elements of \mathbb{R} to its variables. For convenience, we will use any lower case letters to refer to variables, and to refer to a finite set of variables we will put a bar over the letter (e.g. x refers to a variable and \bar{x} refers to a *possibly empty* finite subset of \mathbf{Var}); we also use lower case letters to refer to real numbers. When we write a function f as $f(\bar{x})$ we mean that the set of variables associated to f is exactly the set \bar{x} (i.e. no more variables and no less). When we write variables or sets of variables as a list separated by commas, such as \bar{x}, \bar{y} , we intend that the variables are all distinct. If we have lists of variables or numbers \bar{a} and \bar{b} , the same lengths as \bar{x} and \bar{y} , respectively, then by $f(\bar{a}, \bar{b})$ we mean the value of f when the variables \bar{x}, \bar{y} are assigned to \bar{a}, \bar{b} in order (though the elements in a set do not have an order, we can always think of a set of variables as ordered by its indices). In fact, given a function $f(\bar{x})$ with domain X , we often write $\bar{x} \in X$ to mean that the variables of \bar{x} should be assigned arbitrarily in X . We will often speak of a set of functions without explicitly discussing its variables; we always assume that every function in a set of functions exists in all its “instantiations” with variables, e.g. the function “ $x + y$ ” occurring among a set of functions refers to: $x_0 + x_1, x_1 + x_0, x_1 + x_1, x_0 + x_2$, etc. The key point is that given two functions, the notion of variables allows us to associate the arguments of the functions in any way we wish. Any classes of functions we work with will be sufficiently strong that we can freely manipulate variables in typical ways.

We will now define the approximation notion between classes of functions, building on the notion of approximation between two single functions.

Definition 1. Given a function $f(x_1, \dots, x_k)$ we let $\text{domain}(f) \subseteq \mathbb{R}^k$ refer to its domain.

Definition 2. If $X \subseteq \mathbb{R}^k$ and $Y \subseteq \mathbb{R}^{k+r}$, we say $X \subseteq Y$ if for every $\bar{x} \in X$, there is a $\bar{y} \in \mathbb{R}^r$ such that $(\bar{x}, \bar{y}) \in Y$.

Definition 3. Suppose $f(\bar{x})$, $h(\bar{y})$, and $\varepsilon(\bar{u})$ are functions such that $\bar{x}, \bar{u} \subseteq \bar{y}$. Suppose also that $\text{domain}(f) \subseteq \text{domain}(h)$. We say

$$f \preceq^\varepsilon h,$$

if for any assignment to the variables \bar{y} (which induces assignments on its subsets \bar{x}, \bar{u}) such that $\bar{x} \in \text{domain}(f)$, the function ε is defined at \bar{u} and the following holds:

$$|f(\bar{x}) - h(\bar{y})| \leq \varepsilon(\bar{u}).$$

In the above definition we tend to think of the variables \bar{y} as converging to infinity, motivating the following definition.

Definition 4. A function $f(\bar{x}, \bar{y})$ has **unbounded domain in \bar{y}** if for any $\bar{x} \in \mathbb{R}$, the set $\{\bar{y} \mid (\bar{x}, \bar{y}) \in \text{domain}(f), \bar{y} > 0\}$ is either empty or unbounded.

An important point in the relationship of the function and the function approximating it, is the structure of the variables, formalized in the following definition.

Definition 5. Let $\mathcal{P}(\text{Var})$ be the set of finite subsets of Var . We call a function $b : \mathcal{P}(\text{Var}) \rightarrow \mathcal{P}(\text{Var}) \times \mathcal{P}(\text{Var})$ a **structure function**, and let b_1 refer to its first component and b_2 to its second.

Now we define two notions of approximation between classes of functions, say \mathcal{A} and \mathcal{B} , so that roughly, we write $\mathcal{A} \preceq \mathcal{B}$ to mean that for any function in \mathcal{A} , there is a function in \mathcal{B} that “approximates” it. The approximation relation will be defined relative to a class of functions \mathcal{E} that measures the accuracy of the approximation. The definition will also use a set of structure functions \mathcal{S} which are used to enforce certain relationships between the variables of the function being approximated and the function approximating it (for example, sometimes we will want to force the approximating function to have “parameter variables” and at other times we want to forbid this).

Definition 6. Let \mathcal{A} , \mathcal{B} , and \mathcal{E} be non-empty classes of functions. Let \mathcal{S} be a non-empty set of structure functions.

– We write

$$\mathcal{A} \preceq_{\mathcal{S}}^{\mathcal{E}} \mathcal{B}$$

to mean that $\forall f(\bar{x}) \in \mathcal{A}, \forall b \in \mathcal{S}, \forall \varepsilon \in \mathcal{E}$ (with variables $b_1(\bar{x})$), $\exists h \in \mathcal{B}$ (with variables $b_2(\bar{x})$), such that $f \preceq^\varepsilon h$, where ε and h have unbounded domain in their variables other than \bar{x} .

- We write $\mathcal{A} \preceq_{\exists}^{\mathcal{E}} \mathcal{B}$
to mean the same thing as $\mathcal{A} \preceq_{\forall}^{\mathcal{E}} \mathcal{B}$, except that the second and third universal quantifiers (\forall) are replaced by existential quantifiers (\exists).

To obtain “approximations with no error,” we will use the following set of functions:

Zero is the set of all functions $f(x_1, \dots, x_k)$, for any arity $k \in \mathbb{N}$, such that $\text{domain}(f) = \mathbb{R}^k$ and its value is zero everywhere.

We will be interested in the following sets of structure functions.

- We let the minus sign (“–”) refer to the following singleton set of structure functions:
On input \bar{x} it outputs (\bar{x}, \bar{x}) .
Thus to write $\mathcal{A} \preceq_{-}^{\text{Zero}} \mathcal{B}$ means that for any function in \mathcal{A} there is a function in \mathcal{B} , possibly with an extended domain, so that on their common domain they are equal (i.e. the “approximation” must have no error).
- We let the plus sign (“+”) refer to the following set of structure functions given by taking all the functions of the following form:
On input \bar{x} it outputs $(\bar{x} \cup \bar{y}, \bar{x} \cup \bar{y})$, where \bar{y} is a disjoint and possibly empty finite subset of variables.
- We let φ refer to the following set of structure functions given by taking all the functions of the following form:
On input \bar{x} it outputs $(\bar{u} \cup \bar{y}, \bar{x} \cup \bar{y})$, for any *non-empty* \bar{y} disjoint from \bar{x} and any $\bar{u} \subseteq \bar{x}$ (\bar{u} may be empty).

Thus the “–” approximation does not allow “parameter” variables, the φ approximation requires some “parameter” variables, and the + approximation allows “parameter” variables, but does not require them. The following lemma follow immediately from the definitions.

Lemma 1. $\mathcal{A} \preceq_{+}^{\mathcal{E}} \mathcal{B}$ implies $\mathcal{A} \preceq_{\varphi}^{\mathcal{E}} \mathcal{B}$

The stronger definition (i.e. “ \preceq_{+} ”) is useful for the involved technical work with approximating, as done in [7], but ultimately, a weaker kind of approximation (i.e. “ \preceq_{φ} ”) often suffices for our purposes; in particular, a weaker notion satisfies lemma 5. However the stronger notion has other important properties, such as satisfying lemma 3 and satisfying *transitivity* (lemma 2). It will be useful to list a number of conventions regarding the approximation notation.

1. If the quantifier is missing in the superscript we assume it is \forall .
2. If the subscript is missing we assume it is “–”.
3. If \mathcal{E} is missing, we assume $\mathcal{E} = \text{Zero}$.

Thus for example, $\mathcal{A} \preceq \mathcal{B}$ abbreviates $\mathcal{A} \preceq_{-}^{\text{Zero}} \mathcal{B}$.

We recall the definition of “bounding class” (we use our older definition from [8]). A number of technical aspects of the definition are not used in this paper, but are important for some of the results that are referenced.

Definition 7. For a function $f(y, \bar{x})$, we say it **converges uniformly to infinity in y** if for every $n > 0$ there is $m_0 > 0$ such that for any $m, \bar{x} \in \text{domain}(f)$, $m \geq m_0$, we have $f(m, \bar{x}) \geq n$.

Definition 8. A class of functions \mathcal{B} is a **bounding class** if it has the following properties:

1. There is an $f \in \mathcal{B}$ such that $f \geq 1$.
2. $f(\bar{x}) \in \mathcal{B}$ implies the value of $f(\bar{x}) > 0$.
3. For $f(\bar{x}; t) \in \mathcal{B}$, $f(\bar{x}; t) = f(\bar{x}; -t)$, for any variable t .
4. $f \in \mathcal{B}$ implies f is increasing.
5. $f \in \mathcal{B}$ converges uniformly to infinity in any of its variables.
6. If $f(\bar{x}) \in \mathcal{B}$ and \bar{y} are variables disjoint from \bar{x} , then there is $f^*(\bar{x}, \bar{y}) \in \mathcal{B}$ such that $f(\bar{x}) \leq f^*(\bar{x}, \bar{y})$.
7. If $f, g \in \mathcal{B}$, then there are $h_1, h_2, h_3 \in \mathcal{B}$ such that $f + g \leq h_1$, $f \circ g \leq h_2$, and $f * g \leq h_3$.

An example of a bounding class is the following set of functions that grow like a tower of exponentials; it will be useful for our work with the elementary functions.

Definition 9. Let $\exp^{[n]}(z)$ be defined by $\exp^{[0]}(z) = z$ and $\exp^{[n+1]}(z) = \exp^{[n]}(\exp(z))$ for $n \in \mathbb{N}$ and $z \in \mathbb{R}$. Let \mathcal{T} be the bounding class

$$\mathcal{T} = \{\exp^{[n]}(|x_1| + \cdots + |x_k|) \mid k, n \in \mathbb{N}\}.$$

We will form “error classes” by taking the reciprocal of a bounding class, i.e. for a set of functions \mathcal{F} , $1/\mathcal{F} = \{1/f \mid f \in \mathcal{F}\}$. The following lemma indicates that the a form of the approximation relation is transitive.

Lemma 2. Suppose \mathcal{A} , \mathcal{B} , and \mathcal{C} are classes of functions, and \mathcal{D} is a bounding class. If $\mathcal{A} \preceq_+^{1/\mathcal{D}} \mathcal{B} \preceq_+^{1/\mathcal{D}} \mathcal{C}$ then $\mathcal{A} \preceq_+^{1/\mathcal{D}} \mathcal{C}$.

A useful shorthand is the following “approximate equality.”

Definition 10. We write $\mathcal{A} \approx_5^\xi \mathcal{B}$ to mean that both $\mathcal{A} \preceq_5^\xi \mathcal{B}$ and $\mathcal{B} \preceq_5^\xi \mathcal{A}$ hold.

Another important kind of relationship between classes of functions will be that of one class dominating another.

Definition 11. Suppose \mathcal{A} and \mathcal{B} are classes of functions. We write $\mathcal{A} \leq \mathcal{B}$ if for every function $f(\bar{x}) \in \mathcal{A}$ there is a function $h(\bar{x}) \in \mathcal{B}$ such that $\text{domain}(f) \subseteq \text{domain}(h)$, and $|f(\bar{x})| \leq h(\bar{x})$ for all $\bar{x} \in \text{domain}(f)$.

The growth rate of a class of functions turns out to be a significant issue. In fact for two bounding classes \mathcal{B}_1 and \mathcal{B}_2 , if both $\mathcal{B}_1 \leq \mathcal{B}_2$ and $\mathcal{B}_2 \leq \mathcal{B}_1$ (not the same as $\mathcal{B}_1 = \mathcal{B}_2$), then they can typically be interchanged without effecting our results.

2.2 Function Algebras

We will use function algebras to define most of our classes of functions. They are defined by giving some basic functions and closing the class under operations on functions.

Definition 12. *Suppose \mathcal{A} is a class of functions. An **operation with domain \mathcal{A}** is a function which takes as input some functions in \mathcal{A} , and outputs a single function.*

Definition 13. *Suppose \mathcal{B} is a set of functions (called **basic functions**), and \mathcal{O} is a set of operations. Then $FA[\mathcal{B}; \mathcal{O}]$ is called a **function algebra**, and it denotes the smallest set of functions containing \mathcal{B} and closed under the operations in \mathcal{O} . For ease of readability, we often list the elements of \mathcal{B} or \mathcal{O} simply as a list separated by commas.*

One consequence of this definition is that if the set of basic functions have some property which is preserved under the operations, then all functions in the algebra will satisfy it. For $k \in \mathbb{N}$, by \mathcal{C}^k we mean the k -times continuously differentiable functions on \mathbb{R} . We give an example of a function algebra that only contains \mathcal{C}^2 functions. It will use the operation of obtaining a solution to a linear differential equation.

Definition 14. *L1 is the operation which takes as input functions:*

$$g_1(\bar{x}), \dots, g_n(\bar{x}), s_{11}(y, \bar{x}), \dots, s_{nn}(y, \bar{x}),$$

and returns $h_1(y, \bar{x})$ where we have the following defining equations:

$$\begin{aligned} h_1(0, \bar{x}) &= g_1(\bar{x}) \\ &\vdots \\ h_n(0, \bar{x}) &= g_n(\bar{x}) \\ \frac{\partial}{\partial y} h_1(y, \bar{x}) &= s_{11}(y, \bar{x})h_1(y, \bar{x}) + \dots + s_{1n}(y, \bar{x})h_n(y, \bar{x}) \\ &\vdots \\ \frac{\partial}{\partial y} h_n(y, \bar{x}) &= s_{n1}(y, \bar{x})h_1(y, \bar{x}) + \dots + s_{nn}(y, \bar{x})h_n(y, \bar{x}) \end{aligned}$$

Note that technically L1 is not an operation in our sense because it does not have a fixed arity; we can simply view it as a convenient way to refer to a set of operations, each having a fixed arity. Note that an aspect of the operation is to choose the variable y with respect to which we differentiate; we avoid this technical point for this operation and for others. The basic functions will include a function θ_3 , where for any $k \in \mathbb{N}$ ($k > 0$), $\theta_k(x) = \begin{cases} 0, & x < 0; \\ x^k, & x \geq 0. \end{cases}$, a \mathcal{C}^{k-1} version of the discontinuous function which indicates whether a number is to the left or right of zero. We will also include some constants such as π , as well as the set of *projection functions* which we denote by P. By **comp** we mean the operation of composition.

Definition 15. Let \mathcal{L} abbreviate $FA[0, 1, -1, \pi, \theta_3, P; \text{comp}, L]$.

To compare function algebras it will be useful to talk about a class of functions, \mathcal{B} , approximating an operation; intuitively this means that if any functions are approximated by \mathcal{B} then applying the operation maintains this approximation by \mathcal{B} .

Definition 16. Suppose op is an arity k operation with domain \mathcal{A} , and \mathcal{B} is a class of functions. We write $\text{op} \preceq_+^{\mathcal{E}} \mathcal{B}$ to mean:

For any $f_1, \dots, f_k \in \mathcal{A}$, if $f_i \preceq_+^{\mathcal{E}} \mathcal{B}$ ($i = 1 \dots k$) then $\text{op}(f_1, \dots, f_k) \preceq_+^{\mathcal{E}} \mathcal{B}$

The following is an easy but repeatedly used lemma.

Lemma 3. Suppose \mathcal{B}_1 and \mathcal{B}_2 are classes of functions and \mathcal{O}_1 and \mathcal{O}_2 are sets of operations.

If $\mathcal{B}_1 \preceq_+^{\mathcal{E}} FA[\mathcal{B}_2; \mathcal{O}_2]$ and $\text{op} \preceq_+^{\mathcal{E}} FA[\mathcal{B}_2; \mathcal{O}_2]$ holds for every $\text{op} \in \mathcal{O}_1$ then $FA[\mathcal{B}_1; \mathcal{O}_1] \preceq_+^{\mathcal{E}} FA[\mathcal{B}_2; \mathcal{O}_2]$.

We will now recall how composition can be approximated in a general way, using the concept of modulus functions (recalling the definition from [7], for technical reasons, in this paper, we add a few requirements).

Definition 17.

- $|\bar{b} - \bar{a}|$ abbreviates $|b_1 - a_1| + \dots + |b_n - a_n|$.
- Suppose $f(\bar{x})$ and $m(\bar{x}, z)$ are functions such that $\text{domain}(f) \subseteq \text{domain}(m)$. Then m is a **modulus** for f if m is increasing, has unbounded domain in z , and:

For all $\bar{x} \in \text{domain}(f)$ and $z > 0$ such that $\bar{x}, z \in \text{domain}(m)$, we have that $|\bar{x} - \bar{y}| \leq m(\bar{x}, z)$ implies $|f(\bar{x}) - f(\bar{y})| \leq 1/z$, for all $\bar{y} \in \text{domain}(f)$.

- A class of functions \mathcal{M} is a modulus for the class of functions \mathcal{F} if for any $f \in \mathcal{F}$, there is $m \in \mathcal{M}$ such that m is a modulus for f .

It will be useful here and later to define the notion of restricting a function (whether a real function or even in the case where the function is an operation).

Definition 18. If f is any function, say with domain \mathcal{A} , and \mathcal{B} is a set, we write $f|_{\mathcal{B}}$ to indicate the function f with its domain restricted to $\mathcal{A} \cap \mathcal{B}$.

Lemma 4. (see [7]) Let \mathcal{H} and \mathcal{F} be a classes of functions closed under composition and let \mathcal{B} be a bounding class.

If $\mathcal{H} \leq \mathcal{B}$ and \mathcal{H} has a $1/\mathcal{B}$ modulus then $\text{comp}_{|\mathcal{H}} \preceq_+^{1/\mathcal{B}} \mathcal{F}$.

3 Characterizing computable analysis

We will begin by introducing the technical framework and then discuss our general approach to characterizing the classes of computable analysis via the two step pattern of *approximation and completion*. We will use standard notions from Computable Analysis, as described in Ko [14] and Weihrauch [21], though following more closely the former. For the most part Ko restricts his attention to functions defined on a finite interval, while we consider functions defined on all of \mathbb{R} . Thus in this work, a number of notions will depend on both the input value to the function, as well as the usual accuracy parameter (as in, for example, the case of the modulus functions).

By $\mathbf{C}(\mathbb{R})$ we mean the *total* \mathbb{R} -functions $f(\bar{x})$ which can be computed to accuracy $1/n$ ($n \geq 1$). The real input \bar{x} is given by an oracle which gives \bar{x} to any demanded precision as a dyadic rational; the precision $1/n$ is given by putting n on the input tape (we call this the **accuracy input**). Note that we use the approximation of the form $1/n$ rather than $1/2^n$, since for the classes we work with are sufficiently strong that such distinctions have no effect.

We now consider the process of approximation and completion. We will use the approximation as defined in the previous section, as a first step in this process. For the second step, the completion, all the results will use some kind of limit operation, which will be defined relative to a class of suitable functions (the idea of using a limit operation goes back to work from Costa and Mycka, see [17] and [18]).

Definition 19. *We say a class of functions \mathcal{E} **converges to 0** if any function in \mathcal{E} converges to 0 as any of its arguments (which has unbounded definition) converges to $+\infty$.*

A useful class that converges to 0 is $1/\mathcal{ID}$, where \mathcal{ID} is the set of unary identity functions, one for each variable (i.e. $1/\mathcal{ID} = \{\frac{1}{x_0}, \frac{1}{x_1}, \frac{1}{x_2}, \dots\}$). The limit definition follows our older paper [8].

Definition 20. *Suppose \mathcal{E} is a class of functions that converges to 0. \mathcal{E} -LIM is the operation which takes a function $f(t, \bar{x})$ and returns $F(\bar{x}) = \lim_{t \rightarrow \infty} f(t, \bar{x})$ if the limit exists and there is a function $\alpha(t, \bar{x}) \in \mathcal{E}$ such that $F \preceq^\alpha f$, for positive t .*

If we write LIM without a prefix, we mean $1/\mathcal{ID}$ -LIM.

By $\mathcal{F}(\text{op})$, for a class of functions \mathcal{F} and an operation op , we mean the set of functions \mathcal{F} together with those that result from a *single* application of op to a function in \mathcal{F} . The next proposition points out how a sufficiently good approximation leads to a kind of containment when limits are added.

Lemma 5. *Suppose \mathcal{A} and \mathcal{B} are classes of functions and \mathcal{E} is a class of functions that converges to 0. Then $\mathcal{A} \preceq_{\varphi}^{\exists \mathcal{E}} \mathcal{B}$ implies $\mathcal{A} \preceq \mathcal{B}(\mathcal{E}\text{-LIM})$*

The approach of approximation and completion highlights some interesting issues. In making the process of completion more distinct, we raise the question of

considering the range of techniques that might be employed to complete a class. Going even further, this raises the question of eliminating completion from the characterizations of Computable Analysis. Given that Computable Analysis is defined via an implicit completion process (made explicit in proposition 4), it is not surprising that its characterizations can all be shown to employ completion in an explicit manner. Thus, to characterize Computable Analysis without the use of a completion process would be more of a surprise and provide a more distinctly alternative model of computation. We expand upon these thoughts in section 5. In the ensuing subsections we work out the approximation and completion approach for particular cases: The elementary Real Recursive Functions in part 3.1, Computable Analysis in part 3.2, the computable Real Recursive Functions in part 3.3, and the GPAC in part 3.4.

3.1 Elementary computability

In this section we recall our work ([8], [7]) concerning elementary computability, which extended the work of [1]. By $\mathbf{E}(\mathbb{R})$ we mean the same class as $\mathbf{C}(\mathbb{R})$, except that for real input x and accuracy input n , the computation time is restricted to elementary time. A fundamental point in this development is the following approximation theorem (from [7], though stated there with a different, though “equivalent” bounding class).

Proposition 1. $\mathbf{E}(\mathbb{R}) \approx_+^{1/T} \mathcal{L}$

To obtain a class of functions which actually equals $\mathbf{E}(\mathbb{R})$ we will add various kinds of limit operations to \mathcal{L} . The following is a limit operation that resembles LIM_ω (definition 8 from [1]).

Definition 21. *dLIM* is the operation which takes a function $f(t, \bar{x})$ and if $|\frac{\partial}{\partial t} f| \leq 1/2^t$ for $t \geq 1$, it returns $F(\bar{x}) = \lim_{t \rightarrow \infty} f(t, \bar{x})$.

Note that the derivative condition guarantees the existence of the limit. Our main result from [7] is the following.

Proposition 2. $\mathbf{E}(\mathbb{R}) = \mathcal{L}(\text{LIM}) = \mathcal{L}(\text{dLIM})$

Proof. By proposition 1 we know $\mathbf{E}(\mathbb{R}) \approx_+^{1/T} \mathcal{L}$, which implies $\mathbf{E}(\mathbb{R}) \approx_\phi^{\exists} 1/ID \mathcal{L}$. Thus by lemma 5 we can conclude $\mathbf{E}(\mathbb{R})(\text{LIM}) \approx \mathbf{E}(\mathbb{R})$, and since we are dealing with total functions, $\mathbf{E}(\mathbb{R}) = \mathcal{L}(\text{LIM})$. The part on dLIM follows from our work in [7].

Thus the result characterizing $\mathbf{E}(\mathbb{R})$ can be stated as an approximation (proposition 1) and completion (proposition 2), for two different kinds of completion processes.

3.2 Rephrasing computable analysis

We will show how the functions of computable analysis can be defined by an approximation and completion. We use a class of functions defined on the rationals. A \mathbb{Q} -function $f(x)$ is in $\mathbf{C}(\mathbb{Q})$ if there is a computable function on \mathbb{N} that computes it in the following sense: On input $x = (-1)^k(p/q) \in \mathbb{Q}$ (p/q in lowest terms and $k = 0$ or 1) the machine is given the triple (p, q, k) , and it computes a triple (a, b, s) such that $f(x) = (-1)^s(a/b)$; for a sequence of inputs \bar{x} we use a sequence of triples. Note that $\mathbf{C}(\mathbb{R})$ contains only continuous functions, while $\mathbf{C}(\mathbb{Q})$ contains discontinuous functions.

Definition 22. Consider the \mathbb{Q} -functions from $\mathbf{C}(\mathbb{Q})$ that have a computable modulus function (so they are all continuous). Let *ModRec* be the unique continuous extensions of these functions to \mathbb{R} .

Now we make an observation that is similar to corollary 2.14 of Ko [14].

Proposition 3. $\mathbf{C}(\mathbb{R}) \approx_{\varphi}^{1/\mathcal{ID}} \text{ModRec}$

The proof is basically identical to that of [7], lemma 6.7, except that here we consider computable rather than elementary functions, and only care about a $1/\mathcal{ID}$ approximation. From proposition 3 and the fact that $\mathbf{C}(\mathbb{R})$ is closed under LIM, lemma 5 yields the following.

Proposition 4. $\mathbf{C}(\mathbb{R}) = \text{ModRec}(\text{LIM})$

Definition 23. Consider the \mathbb{Q} -functions from $\mathbf{C}(\mathbb{Q})$ that are continuous. Let *CtnRec* be the unique continuous extensions of these functions to \mathbb{R} .

Question 1. Is the following true: $\mathbf{C}(\mathbb{R}) = \text{CtnRec}(\text{LIM})$?

If the answer to the question is *yes*, then we have an interesting characterization of $\mathbf{C}(\mathbb{R})$: Starting with $\mathbf{E}(\mathbb{Q})$, a *discrete* class making no reference to approximation or oracle inputs, we restrict it in a necessary way in order to equal $\mathbf{C}(\mathbb{R})$ (i.e. being continuous) and this turns out to be sufficient. However, a negative answer seems reasonable, in which case there is a continuous function defined via a computable function, whose modulus is *not* computably bounded, which would be an interesting negative result.

3.3 A function algebra for the computable functions over the reals

We recall that Bournez and Hainry [2] find a function algebra characterizing the \mathcal{C}^2 functions of $\mathbf{C}(\mathbb{R})$. For their results they define an operation CLI which is similar to LI except that it receives an extra input function and requires the result of the operation to be bounded by this function. They also define a restricted root-finding operation called UMU (see [2] for details on these operations). Both here and later, we will need to restrict classes of functions in various ways.

Definition 24. Suppose \mathcal{S} is a set which contains finite lists of the form (X_1, \dots, X_k) , where $X_i \subseteq \mathbb{R}$. For a class of functions \mathcal{F} , we let

$$\mathcal{F}_{\mathcal{S}} = \{f(x_1, \dots, x_k)_{|X_1 \times \dots \times X_k} \mid f \in \mathcal{F} \text{ and } (X_1, \dots, X_k) \in \mathcal{S}\}.$$

In this section we will be interested in the following set of lists:

$$\mathcal{R} = \{([a_1, b_1], \dots, [a_k, b_k]) \mid k \in \mathbb{N}, a_i, b_i \in \mathbb{Q} \text{ such that } 0 \in [a_i, b_i]\},$$

They then prove (though restated here) as theorem 7.1 in [2]:

For \mathcal{C}^2 functions

$$\mathbf{C}(\mathbb{R})_{|\mathcal{R}} \approx \mathbf{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}, \text{LIM}_{\omega}]_{|\mathcal{R}}$$

Note that we have used an approximate equality in the above claim; recalling the definition, this means that for a function from either class there is one that *extends* it in the other class. Though the essential aspects of the proofs in [2] seem sound, there are some ambiguities. We restrict our statement of their theorem in two ways. First, we have stated it with reference to the set \mathcal{R} , rather than with reference to all compact domains (which appears to be more in line with their work, namely lemma 7.4 of [2]). Second, by stating it as an approximate equality, rather than an equality, we are not requiring that the domains of corresponding functions be identical. To get an actual equality (requiring equal functions to also have equal domains) would require restricting $\mathbf{C}(\mathbb{R})$ to having exactly the domains that can be achieved with functions from $\mathbf{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}, \text{LIM}_{\omega}]$. This leads to a question of interest here and generally for function algebras containing partial functions.

Question 2. Can we provide a nice characterization of the following set:

$$\{\text{Domain}(f) \mid f \in \mathbf{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}, \text{LIM}_{\omega}]\}.$$

Now we consider formulating their result as an approximation and completion. From proposition 7.1 and an inspection of the proofs of lemmas 7.4 and 7.5 of [2], we can restate the core of their results as follows.

Proposition 5. For \mathcal{C}^2 functions the following holds:

- $\mathbf{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}] \preceq \mathbf{C}(\mathbb{R})$
- $\mathbf{C}(\mathbb{R})_{|\mathcal{R}} \preceq_{\text{ID}}^1 \mathbf{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}]$

Applying lemma 5 and using the fact that $\mathbf{C}(\mathbb{R})$ is closed under LIM, we can conclude the following.

Proposition 6.

- $\mathbf{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}](\text{LIM}) \preceq \mathbf{C}(\mathbb{R})$
- $\mathbf{C}(\mathbb{R})_{|\mathcal{R}} \preceq \mathbf{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}](\text{LIM})$

We can state the above result as an approximate equality, and using our work from [7], we can include dLIM, as an equivalent completion process.

Proposition 7. *For \mathcal{C}^2 functions:*

$$\begin{aligned} \mathbf{C}(\mathbb{R})|_{\mathcal{R}} &\approx \text{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}](\text{LIM})|_{\mathcal{R}} \\ &= \text{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}](\text{dLIM})|_{\mathcal{R}} \end{aligned}$$

Thus their result can be stated as an approximation and completion. Furthermore, following the form of our proof for \mathcal{L} in [7], the following should be true (i.e. without a restriction to \mathcal{C}^2 functions), using our *method of lifting*.

Conjecture 1. *For total functions:*

$$\begin{aligned} \mathbf{C}(\mathbb{R}) &= \text{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}](\text{LIM}) \\ &= \text{FA}[0, 1, \theta_3, P; \text{comp}, \text{CLI}, \text{UMU}](\text{dLIM}) \end{aligned}$$

3.4 GPAC computability

We begin by considering the result from [19], which shows how to characterize $\mathbf{C}(\mathbb{R})$ by a kind of circuit model. We discuss their result and how to phrase it as an approximation and completion. We then discuss some short-comings of the result, which suggests a series of questions. The General Purpose Analog Computer (GPAC) is an analog circuit model of computation, which by the work of Graça and Costa [11] can be characterized as the solution of a system of polynomial differential equations (the solutions of such a system are called the GPAC or GPAC–**generable** functions). In [19], they build upon the definition of GPAC–*generability* to define a notion they call GPAC–**computable**, which adds a mechanism of “converging computation” to the model. We recall the definition from that paper with the modification that we only consider the interval $[0, 1]$ (just for notational convenience, as treating a general interval $[a, b]$ is about the same).

Definition 25. [19] *A function $f : [0, 1] \rightarrow \mathbb{R}$ is GPAC–computable iff there exists some computable polynomials $p : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$, $p_0 : \mathbb{R} \rightarrow \mathbb{R}$, and $n - 1$ computable real values $\alpha_1, \dots, \alpha_{n-1}$ such that:*

1. (y_1, \dots, y_n) is the solution of Cauchy problem $y' = p(y, t)$ with initial condition $(\alpha_1, \dots, \alpha_{n-1}, p_0(x))$ set at time $t_0 = 0$
2. There are $i, j \in \{1, \dots, n\}$ such that $\lim_{t \rightarrow \infty} y_j(t) = 0$ and $|f(x) - y_i(t)| \leq y_j(t)$ for all $x \in [0, 1]$ and all $t \in [0, +\infty)$.³

They then show in [19] (theorem 17) that (recalling definition 24, in this section we use sets of lists which just contain one element and simply list that element):

³ We assume that $y(t)$ is defined for all $t \geq 0$. This condition is not necessarily satisfied for all polynomial ODEs, and we restrict our attention only to ODEs satisfying this condition.

$\mathbf{C}(\mathbb{R})|_{[0,1]} = \text{GPAC-computable}$.

To put the problem into our language, we will define an operation which takes a list of polynomials and a list of initial conditions and returns the first component of the resulting system of polynomial differential equations.

Definition 26. Let PI be the operation which takes as input some polynomials

$$P_0(x), P_1(u_1, \dots, u_n, t), \dots, P_n(u_1, \dots, u_n, t),$$

and some initial conditions: $\alpha_1, \dots, \alpha_{n-1} \in \mathbb{R}$. Consider the following initial value problem:

$$\begin{aligned} \frac{\partial}{\partial t} y_1(t, x) &= P_1(y_1, \dots, y_n, t) \\ &\vdots \\ \frac{\partial}{\partial t} y_n(t, x) &= P_n(y_1, \dots, y_n, t) \\ y_1(0, x) &= \alpha_1 \\ &\vdots \\ y_{n-1}(0, x) &= \alpha_{n-1} \\ y_n(0, x) &= P_0(x) \end{aligned}$$

and return $y_1(t, x)$ if it is defined for all $0 \leq x \leq 1$ and $t \geq 0$.

We define a version of GPAC-generability, relative to a set of real numbers.

Definition 27. For a set $X \subseteq \mathbb{R}$, let GPAC_X be the set of functions that can be created by PI using polynomials with coefficients from X and initial conditions from X .

Definition 28. For $X \subseteq \mathbb{R}$, let \mathcal{GE}_X be the set of functions $y(t, x)$ in GPAC_X such that for any x , $y(t, x) \rightarrow 0$ as $t \rightarrow \infty$.

Now, if we let \mathcal{CR} be the set of computable real numbers, then

$$\text{GPAC-computable} = \text{GPAC}_{\mathcal{CR}}(\mathcal{GE}_{\mathcal{CR}}\text{-LIM}).$$

Note that this uses the observation that for a polynomial, being computable is the same as having computable coefficients. This way of talking about GPAC-computability has the advantage of making the concept of limits a separate and distinct idea, thus allowing us to state the approximation claim (a significant claim, which is basically a restatement of much of the work from [19]).

Proposition 8. $\mathbf{C}(\mathbb{R})|_{[0,1]} \stackrel{\exists \mathcal{GE}_{\mathcal{CR}}}{\preceq_{\emptyset}} \text{GPAC}_{\mathcal{CR}} \subseteq \mathbf{C}(\mathbb{R})|_{[0,1],[0,\infty)}$

In the previous proposition, the inclusion follows immediately from theorem 24 of [19] (a theorem in fact restating a result from [12]). The approximation follows from theorem 17 of [19] and the definitions; if $f(x) \in \mathbf{C}(\mathbb{R})|_{[0,1]}$ then by theorem 17, $f(x)$ is GPAC-computable, which in our terminology implies the approximation (it should be possible to extract a more direct proof from [19], using the proof of theorem 18 and the latter part of section 5).

The following shows essentially that $\mathbf{C}(\mathbb{R})$ is closed under limits.

Lemma 6. $\mathbf{C}(\mathbb{R})_{|[0,1],[0,\infty)}(\mathcal{G}\mathcal{E}_{\mathcal{C}\mathcal{R}}\text{-LIM}) \subseteq \mathbf{C}(\mathbb{R})_{|[0,1]}$

Proof. Suppose $f(x, t) \in \mathbf{C}(\mathbb{R})_{|[0,1],[0,\infty)}$ and $y(x, t) \in \mathcal{G}\mathcal{E}_{\mathcal{C}\mathcal{R}}$, and suppose $g(x) = \lim_{t \rightarrow \infty} f(x, t)$, such that $g \preceq^y f$. We need to show that $g \in \mathbf{C}(\mathbb{R})_{|[0,1]}$, that is on input $x \in [0, 1]$, and accuracy input $n \in \mathbb{N}$, we need to compute $g(x)$ to accuracy $1/n$. We just need to compute $f(x, t)$ for a large enough t , and to find such a t we just follow the algorithm of [19], at the end of section 6.

Proposition 9. $\mathbf{C}(\mathbb{R})_{|[0,1]} = \text{GPAC}_{\mathcal{C}\mathcal{R}}(\mathcal{G}\mathcal{E}_{\mathcal{C}\mathcal{R}}\text{-LIM})$

Proof. Using lemma 6 and proposition 8, together with lemma 5, we can conclude:

$$\begin{aligned} \mathbf{C}(\mathbb{R})_{|[0,1]} &\subseteq \text{GPAC}_{\mathcal{C}\mathcal{R}}(\mathcal{G}\mathcal{E}_{\mathcal{C}\mathcal{R}}\text{-LIM}) \\ &\subseteq \mathbf{C}(\mathbb{R})_{|[0,1],[0,\infty)}(\mathcal{G}\mathcal{E}_{\mathcal{C}\mathcal{R}}\text{-LIM}) \\ &\subseteq \mathbf{C}(\mathbb{R})_{|[0,1]} \end{aligned}$$

Now we consider some short-comings of the result. This result fits into a series of results which characterize computable analysis in a manner that is different and distinct from it. The result of [19] only does this to an extent, because it includes notions from computable analysis in the definition of GPAC-computability. Or stated in terms of the GPAC_X , the approximation and completion results use the fact that $X = \mathcal{C}\mathcal{R}$, the computable reals. Thus a very natural question is to consider if the result of [19] can be modified so that it holds for a notion of GPAC computability that is distinct from computable analysis. In our notation, this amounts to the following question:

Question 3. *Does there exist a nice set $X \subseteq \mathbb{R}$, defined without computable analysis, such that:*

$$\mathbf{C}(\mathbb{R})_{|[0,1]} = \text{GPAC}_X(\mathcal{G}\mathcal{E}_X\text{-LIM})?$$

Phrasing the result as an approximation and completion, emphasizes the exact completion process used, in this case $\mathcal{G}\mathcal{E}_X\text{-LIM}$. Thus a refinement of the above question would be to consider a more natural limit operation $\mathcal{E}\text{-LIM}$, for some nicer \mathcal{E} such as $1/\mathcal{I}\mathcal{D}$. To answer the previous question (say, with a nicer \mathcal{E}), following the approach above, first we would show an approximation:

$$\mathbf{C}(\mathbb{R})_{|[0,1]} \preceq_{\mathcal{P}}^{\exists \mathcal{E}} \text{GPAC}_X \subseteq \mathbf{C}(\mathbb{R})_{|[0,1],[0,\infty)}.$$

Since we are thinking of $X \subseteq \mathcal{C}\mathcal{R}$, the inclusion is immediate, and the work involved is to show the approximation, which, if true, by lemma 5 would answer the question affirmatively.

We hope that we can use our methods to facilitate answering these questions. Note that $\text{GPAC}_X = X(\text{PI})$, which is probably not the same as $\text{FA}[X; \text{PI}]$; recall that the first notation indicates that PI can be applied once with coefficients from X , while the latter function algebra allows repeated use of PI . To bring our methods into play, we would like to be able to discuss the GPAC as a function algebra. This motivates the following question.

Question 4. *Is there an operation PI^* which is similar to PI , but has the property that for $X \subseteq \mathbb{R}$, $X(PI^*) = FA[X; PI^*]$, or at least $X(PI^*) \approx_+^{\mathcal{E}} FA[X; PI^*]$, for a suitable \mathcal{E} ?*

We could refer to $X(PI^*)$ by $GPAC_X^*$. If we found such an operation PI^* , then showing $\mathbf{C}(\mathbb{R}) = GPAC_X^*(\mathcal{E}\text{-LIM})$ could be reduced to showing $\mathbf{C}(\mathbb{R}) \preceq_+^{\mathcal{E}} FA[X; PI^*]$, perhaps with the domains of the functions of $\mathbf{C}(\mathbb{R})$ restricted in some way. Now we could employ a strategy that we will use in the next section. We could consider the intermediary class $FA[\mathcal{CR}; PI^*]$ and then due to transitivity, break down the goal into the two subgoals:

1. $\mathbf{C}(\mathbb{R}) \preceq_+^{\mathcal{E}} FA[\mathcal{CR}; PI^*]$, and
2. $FA[\mathcal{CR}; PI^*] \preceq_+^{\mathcal{E}} FA[X; PI^*]$

We expect the first subgoal to proceed as in [19]. Furthermore, we expect that PI^* has been defined so that $PI^* \preceq_+^{\mathcal{E}} FA[X; PI^*]$, thus by lemma 3, we can reduce the second subgoal to $\mathcal{CR} \preceq_+^{\mathcal{E}} FA[X; PI^*]$, or in words:

Starting with coefficients from X and applying polynomial differential equations, can we approximate all the computable reals?

4 Making classes analytic

In the previous section we discussed some connections between Computable Analysis and analog models like the Real Recursive Functions and the GPAC. These models are claimed to be closely related to classical physics, where initial value problems play a prominent role. As pointed out in [10], models of natural phenomena arising from classical physics typically involve differential equations with analytic solutions. However, the function algebras considered in sections 3.1 and 3.3 do not satisfy this condition since they include the function θ_3 which is not analytic. The role of θ_3 is crucial in the proofs of the results stated in sections 3.1 and 3.3, since it allows us to define continuous “clocks” and simulate the discrete dynamics of Turing Machines. This is done using a technique first applied in [3] and refined in the context of function algebras on the reals in [5] and [6]. In this section we apply the method of approximation to show that θ_3 can be removed from the function algebra \mathcal{L} (we also remove the constant π), obtaining therefore an analytic characterization we denote by \mathcal{L}^a of the elementary computable functions. The basic technical point we use here is the fact that the approximation relation is transitive, which allows us to define an intermediary class of functions for convenience and then dispense with it in the end. For example, in relating $\mathbf{E}(\mathbb{R})$ and \mathcal{L}^a , we use \mathcal{L} as a convenient intermediary class (convenient because it has the non-analytic function θ_3 which makes technical work easier).

Formally, we define the class

$$\mathcal{L}^a = FA[0, 1, -1, P; \text{comp}, LI],$$

which only contains analytic functions. As pointed out in [6], if the input functions to the operation LI are total, then the operation defines a new total function whose bound is exponential in terms of the bound on the input functions, allowing us to conclude the following.

Lemma 7. *All functions in \mathcal{L} and \mathcal{L}^a are total. Furthermore, $\mathcal{L}, \mathcal{L}^a \leq \mathcal{T}$.*

We claim that $\mathcal{L} \approx_+^{1/\mathcal{T}} \mathcal{L}^a$. We just need to show that $\mathcal{L} \preceq_+^{1/\mathcal{T}} \mathcal{L}^a$ since the other direction is trivial. We show how \mathcal{L}^a can approximate θ_3 and π and the operations of \mathcal{L} .

Lemma 8. $\theta_3, \pi \preceq_+^{1/\mathcal{T}} \mathcal{L}^a$.

Proof. Let $\alpha(x, z) \in \mathcal{T}$. Consider the well known function $y(t) = e^{-t^2}$ defined in \mathcal{L}^a by $y' = -2ty$ and $y(0) = 1$ which satisfies $\int_{-\infty}^{+\infty} y(t)dt = \sqrt{\pi}$. Therefore, $c_1(x) = 2 \int_0^x y(t)dt$ gives an approximation of $\sqrt{\pi}$ in \mathcal{L}^a with error smaller than e^{-x^2} . Squaring that function gives an approximation of π . Given c_1 , we can also obtain an approximation $c_2(x)$ of $\frac{1}{\sqrt{\pi}}$ in \mathcal{L}^a since it contains a function that approximates $\frac{1}{x}$ over \mathbb{R}^+ as noticed in Remark 6 of [1]

Now, take the indefinite integral $F(x) = \int_0^x y(t)dt$, which is a strictly increasing function such that $\lim_{x \rightarrow -\infty} F(x) = -\frac{\sqrt{\pi}}{2}$ and $\lim_{x \rightarrow +\infty} F(x) = \frac{\sqrt{\pi}}{2}$. Define the analytic function $H(x, u) = c_2(u)F(xu) + \frac{1}{2}$ which is therefore in \mathcal{L}^a (H stands for an approximation of the Heaviside function). Finally, there is a $u \in \mathcal{L}^a$ such that $\theta^a(x, z) = H(x, u(x, z))x^3$ is the desired approximation of θ_3 , i.e. $|\theta^a(x, z) - \theta_3(x)| < \frac{1}{\alpha(x, z)}$. To see that this is true, notice that u can be chosen sufficiently larger than any $\alpha \in \mathcal{T}$.

To approximate the operations we follow a more general approach than necessary, since we believe it may be more generally applicable (discussed at the end of the section). The goal is to show for any operations op in the algebra that replacing a set of functions f_1, \dots, f_k by their approximations f_1^*, \dots, f_k^* , still defines an approximation $h^* = \text{op}(f_1^*, \dots, f_k^*)$ of $h = \text{op}(f_1, \dots, f_k)$. To achieve this we have to understand how far apart h^* and h are when their arguments vary. Toward this end we define the notion of Lipschitz functions.

Definition 29.

- Let f be a function on n arguments, and L a function on $2n$ arguments. f is L -Lipschitz if $\text{domain}(f) \times \text{domain}(f) \subseteq \text{domain}(L)$, and $|f(\bar{b}) - f(\bar{a})| \leq L(\bar{b}, \bar{a})|\bar{b} - \bar{a}|$ for all \bar{a} and \bar{b} in the domain of f .
- A class of functions \mathcal{F} is \mathcal{B} -Lipschitz if for every f in \mathcal{F} there is an L in \mathcal{B} such that f is L -Lipschitz.

A given f trivially admits the Lipschitz function $|f(\bar{b}) - f(\bar{a})|/|\bar{b} - \bar{a}|$. However, this cannot always be defined (for instance \mathcal{L} is not closed under division). Moreover, we will be interested in *increasing* Lipschitz functions. To show that classes have increasing Lipschitz functions we can use derivative bounds.

Definition 30. Let \mathcal{F} be a class of differentiable functions. Then

$$\mathcal{F}' = \left\{ \frac{\partial}{\partial x_i} f(x_1, \dots, x_k) \mid f \in \mathcal{F} \right\}.$$

Lemma 9. If \mathcal{F} is a class of functions such that $\mathcal{F}' \leq \mathcal{B}$, where \mathcal{B} is a bounding class, then \mathcal{F} is \mathcal{B} -Lipschitz.

Proof. Let $h(x_1, \dots, x_n)$ be in \mathcal{F} . To find a Lipschitz function, consider:

$$\begin{aligned} |h(\bar{b}) - h(\bar{a})| &\leq |h(b_1, b_2, \dots, b_n) - h(a_1, b_2, \dots, b_n)| \\ &\quad + |h(a_1, b_2, \dots, b_n) - h(a_1, a_2, b_3, \dots, b_n)| \\ &\quad \vdots \\ &\quad + |h(a_1, \dots, a_{n-1}, b_n) - h(a_1, \dots, a_n)| \end{aligned}$$

Consider the first term $|h(b_1, b_2, \dots, b_n) - h(a_1, b_2, \dots, b_n)|$. Consider the function $\frac{\partial}{\partial x_1} h(x_1, x_2, \dots, x_n) \in \mathcal{F}'$ and let $\beta(x_1, \dots, x_n) \in \mathcal{B}$ such that β dominates it. Let $L_1(\bar{b}; \bar{a}) = \beta(|a_1| + |b_1|, b_2, \dots, b_n)$, which is dominated by a function in \mathcal{B} (which for convenience we also call L_1). Since β is increasing and $|a_1|, |b_1| \leq |a_1| + |b_1|$, L_1 dominates the derivative $\frac{\partial}{\partial x_1} h(x_1, b_2, \dots, b_n)$ for all x_1 on the interval between a_1 and b_1 , and so we have:

$$\begin{aligned} |h(b_1, b_2, \dots, b_n) - h(a_1, b_2, \dots, b_n)| &\leq \\ |(h(a_1, b_2, \dots, b_n) + |b_1 - a_1| L_1(\bar{b}; \bar{a})) - h(a_1, b_2, \dots, b_n)| &= L_1(\bar{b}; \bar{a}) |b_1 - a_1|. \end{aligned}$$

We obtain $L_2(\bar{b}; \bar{a}), \dots, L_n(\bar{b}; \bar{a})$ for all the terms and we bound the sum by something in \mathcal{B} , yielding our Lipschitz function.

For instance, Proposition 4.3 of [6] gives bounds on the derivatives of functions in \mathcal{L} . The same bounds apply to \mathcal{L}^a . By lemmas 9 and 7 this implies that \mathcal{L} and \mathcal{L}^a are \mathcal{T} -Lipschitz. In our notation this gives the following lemma.

Lemma 10. $\mathcal{L}', (\mathcal{L}^a)' \leq \mathcal{T}$. Moreover, \mathcal{L} and \mathcal{L}^a are \mathcal{T} -Lipschitz.

This permits us to establish a bound on the distance of the solutions of two different initial value problems. In our case we are interested in comparing initial value problems defined with a function and its approximation.

Lemma 11. (Gronwall-type inequality) Let $f(t, y)$ and $f^*(t, y, \alpha)$ be total C^1 functions with domain \mathbb{R} such that $f \preceq_+^{1/\alpha} f^*$. Let $y(t)$ and $y^*(t, \alpha)$ be the solutions of the following initial value problems:

$$y(t_0) = a, \quad \frac{d}{dt} y(t) = f(t, y) \quad \text{and} \quad y^*(t_0) = a^*, \quad \frac{\partial}{\partial t} y^*(t, \alpha) = f^*(t, y^*, \alpha)$$

If f is β -Lipschitz, and we let $\int_{t_0}^x \beta$ abbreviate $\int_{t_0}^x \beta(u, u, y(u), y^*(u, \alpha)) du$, then where defined we have:

$$|y(t) - y^*(t, \alpha)| \leq e^{\int_{t_0}^t \beta} (|a - a^*| + \int_{t_0}^t \frac{e^{-\int_{t_0}^s \beta}}{\alpha} ds)$$

Proof. The hypothesis lead to the inequalities (the first inequality follows from y and y^* being C^1 ; for ease of readability we sometimes drop the arguments to these functions.).

$$\begin{aligned} \frac{\partial}{\partial t}|y(t) - y^*(t, \alpha)| &\leq \left| \frac{d}{dt}y(t) - \frac{\partial}{\partial t}y^*(t, \alpha) \right| \\ &= |f(t, y) - f^*(t, y^*, \alpha)| \\ &\leq |f(t, y) - f(t, y^*)| + |f(t, y^*) - f^*(t, y^*, \alpha)| \\ &\leq \beta(t, t, y, y^*)|y - y^*| + \frac{1}{\alpha}. \end{aligned}$$

The inequality above implies that

$$\frac{\partial}{\partial t}|y(t) - y^*(t, \alpha)| - \beta(t, t, y, y^*)|y(t) - y^*(t, \alpha)| \leq \frac{1}{\alpha}.$$

Multiplication by the integrating factor $e^{-\int^t \beta}$ yields

$$\frac{\partial}{\partial t}(e^{-\int^t \beta}|y(t) - y^*(t, \alpha)|) \leq \frac{e^{-\int^t \beta}}{\alpha}.$$

Integrating this from t_0 to t gives

$$e^{-\int^t \beta}|y(t) - y^*(t, \alpha)| - |a - a^*| \leq \int_{t_0}^t \frac{e^{-\int^s \beta}}{\alpha} ds,$$

which finally leads to the claimed bound.

Lemma 12. $L|_{\mathcal{L}} \preceq_+^{1/\mathcal{T}} \mathcal{L}^a$.

Proof. Let h be the solution of the linear initial value problem $h(\bar{x}, 0) = g(\bar{x})$ and $\frac{\partial}{\partial t}h(\bar{x}, t) = s(\bar{x}, t)h(\bar{x}, t)$, where $g, s \in \mathcal{L}$. Let $g^*, s^* \in \mathcal{L}^a$ be approximations of g and s . We claim that the solution h^* of the linear initial value problem $h^*(\bar{x}, 0, \alpha) = g^*(\bar{x}, \alpha)$ and $\frac{\partial}{\partial t}h^*(\bar{x}, t, \alpha) = s^*(\bar{x}, t, \alpha)h^*(\bar{x}, t, \alpha)$ can be made arbitrarily close to h in \mathcal{L}^a . More precisely, we claim that there is a $H(\bar{x}, t, z) = h^*(\bar{x}, t, \alpha(t, z)) \in \mathcal{L}^a$ such that $|h(\bar{x}, t) - H(\bar{x}, t, z)| \leq 1/z$.

To show this, and since all functions in \mathcal{L} and \mathcal{L}^a are total and C^1 , we apply lemma 11 where f in the lemma is now the product hs and the initial condition in the lemma is now $a = g(\bar{x})$. By lemma 10 there is a $\beta \in \mathcal{T}$ such that f is β -Lipschitz. We want to get rid of the dependence of $\int^x \beta$ on α so we can freely bound $|h - h^*|$ adjusting α . By hypothesis $f \preceq_+^{1/\alpha} f^*$ which means that $f^*(t, y, \alpha) < f(t, y) + 1 \leq \exp(f(t, y))$ for all α . Likewise, the initial conditions in the lemma satisfy $a^* < a + 1$. Consider the initial value problem $\frac{\partial}{\partial t}y^+(t) = \exp(f(t, y^+))$ and $y^+(t_0) = a + 1$. The solution $y^+(t)$ exists in \mathcal{L} , is increasing, and bounds $y^*(t, \alpha)$ for all α . Now, $\int^t \beta$ can be bounded by $\phi(t) = \int_{t_0}^t \beta(u, u, y^+(u)) du$ and this is bounded in \mathcal{T} . Moreover, $\int_{t_0}^t e^{-\int^s \beta} ds$

is bounded by some constant K for β large enough. For the initial condition, we suppose that $|a - a^*| < |a - a^+| \leq 1/\alpha$.

Therefore, the right hand side in the inequality in lemma 11 can be bounded by $\exp(\phi(t))(1 + K)/\alpha$. Therefore, choosing $\alpha(t, z) \in \mathcal{T}$ sufficiently large we can guarantee that $|y(t) - y^*(t, \alpha)| \leq 1/z$. This function $y^*(t, \alpha(t, z))$ corresponds to the desired approximation H mentioned earlier in the proof.

To conclude with our goal, we only have to show that the approximation also holds for composition. To be able to use lemma 4 we have to show that \mathcal{L} has an appropriate modulus. The following lemma shows that this is a direct consequence of \mathcal{L} being \mathcal{T} -Lipschitz.

Lemma 13. *Suppose \mathcal{B} is a bounding class and \mathcal{F} is class of functions that is \mathcal{B} -Lipschitz. Then \mathcal{F} has a $1/\mathcal{B}$ -modulus.*

Proof. Let $f(\bar{x}) \in \mathcal{F}$ and $L(\bar{x}, \bar{y}) \in \mathcal{B}$ be such that $|f(\bar{x}) - f(\bar{y})| \leq L(\bar{x}, \bar{y})|\bar{x} - \bar{y}|$; recall that L is increasing. Let $m(\bar{x}, z) \in \mathcal{B}$ be a function that dominates $L(\bar{x}, \bar{x} + 1)z + 1$, and suppose $|\bar{x} - \bar{y}| \leq 1/m(\bar{x}, z)$. First note that this implies that $\bar{y} \leq \bar{x} + 1$, and thus $|\bar{x} - \bar{y}| \leq 1/m(\bar{x}, z)$ further implies:

$$\begin{aligned} |f(\bar{x}) - f(\bar{y})| &\leq L(\bar{x}, \bar{y}) \frac{1}{m(\bar{x}, z)} \\ &\leq L(\bar{x}, \bar{x} + 1) \frac{1}{L(\bar{x}, \bar{x} + 1)z + 1} \\ &\leq \frac{1}{z} \end{aligned}$$

Proposition 10. $\mathcal{L} \approx_+^{1/\mathcal{T}} \mathcal{L}^a$

Proof. $\mathcal{L}^a \preceq_+^{1/\mathcal{T}} \mathcal{L}$ is immediate, so we consider the approximation $\mathcal{L} \preceq_+^{1/\mathcal{T}} \mathcal{L}^a$. By lemma 8, we can approximate all the basic functions of \mathcal{L} . Since \mathcal{L} is \mathcal{T} -Lipschitz, it has, by lemma 13, a $1/\mathcal{T}$ modulus, and thus we apply lemma 4 to obtain $\text{comp}_{|\mathcal{L}} \preceq_+^{1/\mathcal{T}} \mathcal{L}^a$. By lemma 12, $\text{Ll}_{|\mathcal{L}} \preceq_+^{1/\mathcal{T}} \mathcal{L}^a$. Finally, we apply lemma 3 to conclude the proof.

Proposition 1, together with transitivity obtains the following approximation for $\mathbf{E}(\mathbb{R})$ with analytic functions.

Proposition 11. $\mathbf{E}(\mathbb{R}) \approx_+^{1/\mathcal{T}} \mathcal{L}^a$

As a result we get an improved characterization of the elementary computable functions, applying lemma 5 (and obtaining exact equality because we are dealing with total functions).

Proposition 12. $\mathbf{E}(\mathbb{R}) = \mathcal{L}^a(LIM)$

We consider now the following generalization of Ll where the initial value problem is not required to be linear.

Definition 31. l is the operation which takes input functions $f_1(\bar{x}), \dots, f_k(\bar{x})$ and $g_1(\bar{x}, t, y), \dots, g_k(\bar{x}, t, y)$ of appropriate arities and returns h_1 which is defined by the equations

$$\begin{aligned} h_1(\bar{x}, t_0) &= f_1(\bar{x}) \\ &\dots \\ h_k(\bar{x}, t_0) &= f_k(\bar{x}) \\ \frac{\partial}{\partial t} h_1(\bar{x}, t) &= g_1(\bar{x}, t, \bar{h}) \\ &\dots \\ \frac{\partial}{\partial t} h_k(\bar{x}, t) &= g_k(\bar{x}, t, \bar{h}). \end{aligned}$$

We will write this as $h = l(f, g)$. Replacing ll by l in \mathcal{L} and \mathcal{L}^a lead to the following classes.

Definition 32.

- Let \mathcal{G} be $FA[0, 1, -1, \theta_3, P; \text{comp}, l]$.
- Let \mathcal{G}^a be $FA[0, 1, -1, P; \text{comp}, l]$.

The class \mathcal{G} was investigated in [4] and shown to be closed under an iteration operation, thus allowing Turing Machines to be simulated. It would be interesting to show that the class of analytic functions \mathcal{G}^a could do the same. In fact, related work has already been carried out. In [15] it was shown that analytic maps over unbounded domains can simulate the *transition function* of any Turing Machine, while only recently (by [10]) was it shown that the *iterations* of those transition functions can also be simulated with analytic flows over unbounded domains. These results motivate the search for an approximation between \mathcal{G} and \mathcal{G}^a .

Unlike the case of the elementary functions, there does not appear to be a convenient bounding class like \mathcal{T} . However, there is a generic way to create bounding functions that correspond to a given class of functions, by choosing out the right functions from the class.

Definition 33. Given a class of functions \mathcal{F} , let $\text{bd}\mathcal{F}$ be the functions in \mathcal{F} that satisfy properties 2 through 5 of the definition of bounding class.

Lemma 14. If \mathcal{F} is closed under addition, multiplication, and composition, contains a function $f \geq 1$ and satisfies property 6 of bounding classes, then $\text{bd}\mathcal{F}$ is a bounding class.

Definition 34. We write $\mathcal{A} \preceq_+^{\forall} \mathcal{B}$ to mean $\mathcal{A} \preceq_+^{\forall \frac{1}{\text{bd}\mathcal{B}}} \mathcal{B}$.

Question 5. Does the approximation $\mathcal{G} \approx_+^{\forall} \mathcal{G}^a$ hold?

We could try to prove this using the approach we described for \mathcal{L} . However, some difficulties need to be overcome. Firstly, it is not clear if given approximations f^* and g^* of f and g , $h = l(f, g)$ and $h^* = l(f^*, g^*)$ have the same domains. An additional difficulty is to show that the associated bounding classes are “appropriate,” amounting to the question of whether or not $\text{bd}\mathcal{G} \leq \text{bd}\mathcal{G}^a$.

5 Future work

We make the informal claim that the method of approximation provides a useful way to organize and think about the relationships between various computational classes. It both facilitates the technical work and suggests ideas. In particular we consider the idea of approximation and completion and some issues raised by organizing the work in this manner. We have seen how the various ways to characterize Computable Analysis all fit into this two step process of approximation and completion. One advantage of this approach is to emphasize that the fundamental point of the various characterizations is to find an approximation. It also helps from a technical point of view, separating the proofs into two steps, where the approximation step can be facilitated by the various tools developed for this method. Furthermore, by separating out the notion of completion as a distinct step, we have seen that in some cases, different kinds of completions are equivalent. This raises the question of considering more broadly the ways that a class of functions can be completed. To date, the method of choice has been to use limits, but we could also consider search operators, given the result of Mycka [17], showing that in a certain context with Real Recursive Functions, limits and zero-finding are equivalent.

Question 6. *What other manners of “completion” are interesting and useful in relating computational classes to Computable Analysis?*

A related line of thought, alluded to earlier, is to consider the elimination of the completion step.

Question 7. *Are there characterizations of Computable Analysis, which naturally capture all of its functions, without a completion operation?*

This would be especially interesting in light of the fact that the very definition of Computable Analysis is tied up with a notion of completion. Already this is apparent in the standard definitions of Computable Analysis as a computable process which gets increasingly closer to the result, only finished as the accuracy parameter converges to infinity (proposition 4 points out exactly how to put the limit into the definition). Thus, to find an alternative model with no apparent completion step would provide a more genuinely distinct way of conceiving of Computable Analysis. This would seem to be interesting from two points of view. It would seem to be a very useful kind of result in understanding a Church-Turing thesis for real computation (recalling the discussion of the introduction). Furthermore, recall that in the introduction we considered the project of using the correspondences to Computable Analysis as a way to bring in the methods of analysis to the questions of classic complexity theory. A model of computation which is more different and does not employ completion could be more useful in this vein.

6 Acknowledgements

This work was partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER POCTI/POCI, namely, via CLC, project ConTComp POCTI / MAT / 45978 / 2002, and grant SFRH / BPD / 16936 / 2004.

References

1. O. Bournez and E. Hainry. Elementarily computable functions over the real numbers and \mathbb{R} -sub-recursive functions. *Theoretical Computer Science*, 348(2–3):130–147, 2005.
2. O. Bournez and E. Hainry. Recursive analysis characterized as a class of real recursive functions. *Fundamenta Informaticae*, 74(4):409–433, 2006.
3. M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, 1995.
4. M. L. Campagnolo. *Computational complexity of real valued recursive functions and analog circuits*. PhD thesis, IST, Universidade Técnica de Lisboa, 2001.
5. M. L. Campagnolo, C. Moore, and J. F. Costa. Iteration, inequalities, and differentiability in analog computers. *Journal of Complexity*, 16(4):642–660, 2000.
6. M. L. Campagnolo, C. Moore, and J. F. Costa. An analog characterization of the Grzegorzczuk hierarchy. *Journal of Complexity*, 18(4):977–100, 2002.
7. M. L. Campagnolo and K. Ojakian. The elementary computable functions over the real numbers: Applying two new techniques. Submitted.
8. M. L. Campagnolo and K. Ojakian. The methods of approximation and lifting in real computation. In Douglas Cenzer, Ruth Dillhage, Tanja Grubba, and Klaus Weihrauch, editors, *Proceedings of the Third International Conference on Computability and Complexity in Analysis, CCA 2006, Gainesville, Florida, USA, November 1–5, 2006*, volume 167 of *Electronic Notes in Theoretical Computer Science*, Amsterdam, 2007. Elsevier.
9. J. F. Costa and J. Mycka. The $P \neq NP$ conjecture in the context of real and complex analysis. *Journal of Complexity*, 22(2):287–303, April 2006.
10. D. S. Graça, M. L. Campagnolo, and J. Buescu. Computability with polynomial differential equations. *Advances in Applied Mathematics*, 2007. to appear.
11. D. S. Graça and J. F. Costa. Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5):644–664, 2003.
12. D. S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability and undecidability of maximal intervals of IVPs. *Transactions of the American Mathematical Society*, 2007. to appear.
13. A. Grzegorzczuk. Computable functionals. *Fundamenta Mathematicae*, 42:168–202, 1955.
14. K.-I. Ko. *Complexity Theory of Real Functions*. Birkhäuser, 1991.
15. P. Koiran and C. Moore. Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoretical Computer Science*, 210(1):217–223, 1999.
16. C. Moore. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162:23–44, 1996.
17. J. Mycka. μ -recursion and infinite limits. *Theoretical Computer Science*, 302:123–133, 2003.

18. J. Mycka and J. F. Costa. Real recursive functions and their hierarchy. *Journal of Complexity*, 20(6):835–857, 2004.
19. D. S. Graça O. Bournez, M. L. Campagnolo and E. Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. *Journal of Complexity*, 2007. doi: 10.1016/j.jco.2006.12.005.
20. C.E. Shannon. Mathematical theory of the differential analyzer. *J. Math. Phys. MIT*, 20:337–354, 1941.
21. K. Weihrauch. *Computable Analysis: An Introduction*. Springer-Verlag, 2000.