Characterizing time computational complexity classes with polynomial differential equations

Riccardo Gozzi
Instituto Superior Técnico, Universidade de Lisboa, Portugal
Instituto de Telecomunicações, Portugal

Daniel S. Graça
Universidade do Algarve, Portugal
Instituto de Telecomunicações, Portugal

Abstract

In this paper we show that several classes of languages from computational complexity theory, such as EXPTIME, can be characterized in a continuous manner by using only polynomial differential equations. This characterization applies not only to languages, but also to classes of functions, such as the classes defining the Grzegorczyk hierarchy, which implies an analog characterization of the class of elementary computable functions and the class of primitive recursive functions.

1 Introduction

In the papers [BGP16b], [BGP17b] the authors presented a characterization of the computational complexity class P by using polynomial ordinary differential equations (ODEs). This characterization is purely continuous and hence establishes a continuous characterization of the class P typically associated to discrete models of computation such as the standard Turing machine. This result does not use any reference to a (discrete) machine and hence provides an implicit characterization of P.

Interestingly, this result also provides a form of equivalence between analog and digital models of computations, both at a computability level and at a complexity level. This stems from the fact that the class of functions which can be computed with these polynomial ODEs corresponds to an analog model of computation, Shannon's General Purpose Analog Computer (GPAC) [Sha41], [GC03], which is intended to model differential analyzers, which were the analog computers in use before the advent of the digital computer [Bus31]. It was shown that the GPAC and Turing based models of computation, including computable analysis, are computationally equivalent at a computability level [GCB08], [BCGH06], [BCGH07] and at a complexity level, when we consider functions computable in polynomial time [BGP16b], [BGP17b].

A natural question to ask is whether the above continuous characterization of polynomial time computability via ODEs extends to other computational complexity classes. This is not obvious from [BGP16b], [BGP17b], since these papers are technically dense, and rely on other papers such as [BGP16a] which are tailored to the polynomial case. Polynomials have many desirable properties which are unfortunately not shared by some other types of functions. For example, the class of polynomials is closed under composition, while the class of exponential functions is not, which may (and will, as we will see later) pose problems when trying to characterize the class EXPTIME with polynomial ODEs.

The purpose of the present paper is to understand how the results of [BGP16b], [BGP17b] can be extended to other complexity classes such as EXPTIME. For this purpose we will first focus on classes of functions, such as the class FEXPTIME of functions computable in exponential time and we will show how this class can be characterized using ODEs. Other classes of functions that we will characterize include all the classes defining the Grzegorczyk hierarchy and, in particular, the elementary computable functions and the primitive recursive functions. We note that some similar results were proved in [CMC00], [CM01], [CMC02], [BH04], [BH05]. However the approach used in these papers differs from the current one in the sense that they use real recursive functions which use partial differential equations, among other differences.

As we will see in the following sections, we will be able to reuse some results from previous papers such as [BGP16b], [BGP17b], [BGP16a], while other results will have to be adapted. As we mentioned, the papers [BGP16b], [BGP17b] are technically dense and often not easy to follow. In this paper we also intend to provide a "road map" to the results of [BGP17b], which might help simplify their understanding, by clearly separating the "core" results of the papers [BGP16b], [BGP17b], from their "non-core" counterparts.

To obtain our results, first we characterize the class FEXPTIME of discrete functions computable in exponential time using ODEs (Theorem 29). The idea behind this result is to use the encoding provided by [BGP17b] of the transition function of a given Turing machine as a solution of a polynomial ODE and iterate it with polynomial ODEs. The challenge will be in making all this work using resources which are exponentially bounded on the input size, and not double exponentially bounded as a more direct approach similar to that of [BGP17b] would yield. This will provide the elements necessary to understand how to extend this result to other discrete complexity classes of functions (Theorem 35). As an example of this extension we will show how it is possible to apply Theorem 35 to the whole Grzegorczyk hierarchy, obtaining in this way a completely analog characterization of the Grzegorczyk hierarchy by using ODEs and, in particular, of the class of elementary functions and the class of primitive recursive functions. In the case of the Grzegorczyk hierarchy it will not be enough to apply Theorem 35, since for the complexity time bound we need to extend discrete time bounds (i.e. defined over $\mathbb{N}$) classes to real functions. While this is trivially done for polynomial or exponential functions, this is not straightforward for classes such as the Grzegorczyk hierarchy. We present a solution to this problem by constructing appropriate bounds using ODEs which perform several operations like iteration, etc.

Finally, combining our approach with that of [BGP17b] we show that we can characterize classes of languages instead of classes of functions, and this will be concretely implemented for describing the complexity class EXPTIME using differential equations.

The outline of the paper is as follows. On section 2 we review some basic notions about computability with ODEs. In section 3 we briefly review the results of [BGP17b] which show how one can characterize the class FP of discrete functions computable in polynomial time in terms of differential equations. In section 4 we discuss the problems of extending the results from [BGP17b] to the exponential case. Then we present a solution to solve these problems and we use it to characterize the class FEXPTIME using differential equations. In section 5 we discuss the conditions that allow us to repeat what was done with the exponential case for higher complexity classes. In section 6 we show that, analogously to what was done in [BGP17b] for the polynomial case, the analog characterization of the class FEXPTIME and of higher complexity classes via ODEs can be done by considering just one complexity parameter, the length of the solution curve of the ODE. In section 7 we apply the generalization to the case of the Grzegorczyk hierarchy. To do so, we develop a technique to obtain by means of system of differential equations the correct set of boundaries for each level of the hierarchy. In section 8 we show how the class EXPTIME can be characterized using differential equations. Finally, in section 9 we present the conclusions and discuss some open problems.

2 Computing with polynomial differential equations

As we have mentioned, polynomial ODEs correspond to Shannon's GPAC. More concretely a function is *generable* if it is the solution of some polynomial initial-value problem (PIVP) defined with a polynomial ODE. We call such functions PIVP functions. Note that in this sense a generable function is a one-variable function. The notion of generable function can be extended to multivariate functions as in [BGP17b]. Moreover, by restricting the coefficients of the polynomials used to define the ODE, we can also restrict the class of functions computed by this model. This is usually assumed to avoid potential problems where the ODE gains unreasonable computational power by using the coefficients of the polynomials as oracles. More concretely, let $\mathbb{K}$ be some field with the property that $\mathbb{Q} \subseteq \mathbb{K}$ and recall that $J_f(x)$ denotes the Jacobian of f at point x.

Definition 1 *Let $D \subseteq \mathbb{R}^k$ be a domain (i.e. an open and connected set) and $f : D \to \mathbb{R}^l$. We say that $f \in \mathrm{GVAL}_{\mathbb{K}}$ if and only if there is an initial-value* *problem (IVP)*

$$J_y(x) = p(y),\ y(x_0) = y_0 \tag{1}$$

with solution y such that $f(x) = (y_1(x), \ldots, y_l(x))$, where p is some $n \times k$ matrix, $n \geq l$, with the property that each entry of p is a polynomial with coefficients in $\mathbb{K}$, and $x_0 \in \mathbb{K}^k \cap D, y_0 \in \mathbb{K}^l$.

We remark that, for the class $\mathrm{GVAL}_{\mathbb{K}}$, a function $f \in \mathrm{GVAL}_{\mathbb{K}}$ corresponds to (components of) a solution y of a differential equation. Note also that it can be shown that each IVP of the form of (1) has one and only one solution [BGP17a]. Moreover the one-variable functions of $\mathrm{GVAL}_{\mathbb{R}}$ are exactly the PIVP functions, which are formed by components of some polynomial initial value-problem $y' = p(y)$, $y(t_0) = y_0$, where $y : \mathbb{R} \to \mathbb{R}^n$ for some $n \geq 1$. We can also establish complexity measures for functions in $\mathrm{GVAL}_{\mathbb{K}}$ by putting a bound on the growth of functions. Throughout this paper, unless other specified, $\|\cdot\|$ denotes the sup norm in $\mathbb{R}^k$, i.e. if $x \in \mathbb{R}^k$, then $\|x\| = \max_{1 \leq i \leq k} |x_i|$, where $x = (x_1, \ldots, x_k)$.

Definition 2 *We say that $f \in \mathrm{GVAL}_{\mathbb{K}}(sp)$, where $sp : \mathbb{R} \to \mathbb{R}$, if $f \in \mathrm{GVAL}_{\mathbb{K}}$ and there is some solution y of (1) for which one has: (i) $f(x) = (y_1(x), \ldots, y_l(x))$ and (ii) $\|y(x)\| \leq sp(\|x\|)$.*

If $\mathbb{K} \subsetneq \mathbb{R}$, it is often convenient to ensure that $\mathbb{K}$ is closed under the application of functions in $\mathrm{GVAL}_{\mathbb{K}}$. This is the case for *generable fields*. A detailed treatment of generable fields and their properties can be found in [BGP17a]. As an example, $\mathbb{R}_P$, the field of real numbers that are polynomial-time computable in the sense of computable analysis, is a generable field. More specifically, the only property of these fields that is relevant for this paper is the following: a field $\mathbb{K}$ is generable if $\mathbb{Q} \subseteq \mathbb{K}$ and $f(\alpha) \in \mathbb{K}^l$ for any $(f : \mathbb{R}^k \to \mathbb{R}^l) \in \mathrm{GVAL}_{\mathbb{K}}$ and any $\alpha \in \mathbb{K}^l$. It can be shown (see [BGP17a, Section 6]) that there exists a minimal generable field, denoted by $\mathbb{R}_G$ (the field is minimal in the sense that if $\mathbb{K}$ is another generable field, then $\mathbb{R}_G \subseteq \mathbb{K}$), and that any element of $\mathbb{R}_G$ is computable in polynomial time. From now on we will always assume that $\mathbb{K} = \mathbb{R}_G$ and denote $\mathrm{GVAL}_{\mathbb{R}_G}$ by GVAL for simplicity.

We note that several standard functions from analysis such as the trigonometric functions, polynomials, the exponential and logarithmic functions all belong to GVAL (i.e. they are generable). This class also has important stability properties, since it is closed under addition, subtraction, product, composition, and ODE solving [BGP17a]. Specifically, the fact that trigonometric functions belong to GVAL and that this class is closed under composition will be essential for proving some of our results in later sections.

In what follows, it will be convenient to consider generable functions which are bounded by a polynomial.

Definition 3 *We say that $f \in \mathrm{GPVAL}$ if and only if $f \in \mathrm{GVAL}(p)$ for some polynomial p.*

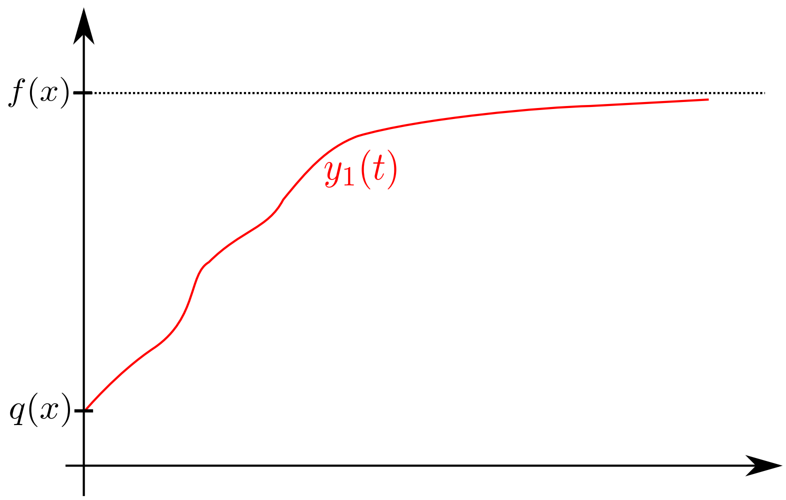

Figure 1: Computing the function f with an ODE.

However, functions such as the Gamma function or the Riemann Zeta function are not generable. The problem does not arise from the model of computation but from the notion of computation used, where real-time computation is used e.g. in the one-variable case. For example, when computing sin, we immediately get $\sin(t)$ as soon as we feed the input t to the GPAC computing sin. This is in contrast to other models of computation (e.g. Turing machines), where some time is allowed for the computation since the moment the input is provided to the model up to the moment in which an answer is provided. For this reason, the GPAC model was extended in [Gra04] to more closely match this notion of computation. The idea is to use a kind of limit computation (see Fig. 1), where at each moment the computation provides an approximation of the correct answer with an error which goes to 0 as computation time goes to infinity, matching what is done in computable analysis [Wei00], [KF82]. Contrarily to what happens for $\text{GVAL}_{\mathbb{K}}$, in this setting a function f is not longer (formed by components of) a solution y of a differential equation. Instead $f(x)$ is obtained as the *limit* $\lim_{t\to+\infty y(t)}$ of a solution y of an ODE, with initial condition $y(0)$ dependent in a "simple manner" of the value x.

Definition 4 (ATS) *Let $f \subseteq \mathbb{R}^n \to \mathbb{R}^m$. We say that $f \in \text{ATS}(\Pi, \Upsilon)$, where $\Pi, \Upsilon : \mathbb{R}^2 \to \mathbb{R}$ if and only if there exist multivariate polynomials p, q with coefficients in $\mathbb{R}_G$ such that for any $x \in \text{dom}(f)$, there exists (a unique) $y : \mathbb{R} \to \mathbb{R}^d$ satisfying for all $t \geq 0$:*

- $y(0) = q(x)$ *and* $y'(t) = p(y(t))$*;*
- $\forall \mu > 0$ *if* $t \geq \Pi(\|x\|, \mu)$ *then* $\|(y_1(t), \ldots, y_m(t)) - f(x)\| \leq e^{-\mu}$*;*
- $\|y(t)\| \leq \Upsilon(\|x\|, t)$.

The function Π is usually referred to as the time bound, and the function Υ as the space bound (the notation ATS is an abbreviation for "analog time space"). Note that, differently to what happens with Turing machines in discrete complexity theory, here space and time boundaries are independent and unrelated one to another. In the paper [BGP16a] the focus was on the class ATSP of functions computed with a polynomial time and space bound. This class will also be important in the present paper.

Definition 5 *Let $f \subseteq \mathbb{R}^n \to \mathbb{R}^m$. We say that $f \in \mathrm{ATSP}$ if $f \in \mathrm{ATS}(\Pi, \Upsilon)$ for some polynomials Π and Υ.*

When using the class ATSP in proofs, it is often useful to use one of the several equivalent definitions of ATSP, which were proved in [BGP16a]. In particular, the following definition will be helpful (where the notation $C^0(\mathbb{R}, \mathbb{R}^n)$ stands for the set of continuous functions from $\mathbb{R}$ to $\mathbb{R}^n$), where $\mathbb{R}_+ = [0, +\infty[$:

Proposition 6 *Let $f \subseteq \mathbb{R}^n \to \mathbb{R}^m$. Then $f \in \mathrm{ATSP}$ if and only if there exist $\delta \geq 0$, $d \in \mathbb{N}$, multivariate polynomials p with coefficients in $\mathbb{R}_G$, $y_0 \in \mathbb{R}_G^d$, and polynomials $\Pi, \Upsilon, \Lambda : \mathbb{R}_+^2 \to \mathbb{R}_+$ such that for any $x \in C^0(\mathbb{R}, \mathbb{R}^n)$, there exists (a unique) $y : \mathbb{R} \to \mathbb{R}^d$ satisfying for all $t \geq 0$:*

- *$y(0) = y_0$ and $y'(t) = p(y(t), x(t))$;*
- *$\|y(t)\| \leq \Upsilon(\sup_\delta \|x\|(t), t)$, where $\sup_\delta f(t) = \sup_{u \in [t-\delta, t] \cap [0, +\infty[} f(u)$;*
- *For any $I = [a, b] \subseteq \mathbb{R}$ with $a \geq 0$, if there exist $\bar{x} \in \mathrm{dom}(f)$ and $\mu \geq 0$ such that for all $t \in I$ we have $\|x(t) - \bar{x}\| \leq e^{-\Lambda(\|x\|, \mu)}$, then $\|(y_1(t), \ldots, y_m(t)) - f(\bar{x})\| \leq e^{-\mu}$ whenever $a + \Pi(\|\bar{x}\|, \mu) \leq u \leq b$.*

The first condition of proposition 6 defines the ODEs generating the function f, the second condition ensures that the norm of the solution of this system satisfies a polynomial bound, while the third condition takes care of the convergence of such solution: specifically, this solution y converges to the correct value of the function $f(\bar{x})$ only if there exists an interval where the value of a continuous function x is near the point $\bar{x}$. If this happens, then, after a delay given by a polynomial boundary Π, we get a convergence similar to the one of the original definition of ATSP. The proposition above is particularly important because, allowing possible fluctuations for the input x, it provides a strong robustness property to the functions in ATSP that plays a determinant role on ensuring closure by composition for the class.

The class ATSP has several nice properties, which were proved in [BGP17b]. Next we provide a summary of the most relevant results.

Proposition 7 (ATSP closure by arithmetic operations) *If $g, f \in \mathrm{ATSP}$, then $g + f, g - f, g \cdot f \in \mathrm{ATSP}$, with the obvious restrictions on the domains of definition.*

Proposition 8 (ATSP modulus of continuity) *If $f \in \text{ATSP}$, then f admits a polynomial modulus of continuity: there exists a polynomial $\Omega : \mathbb{R}^2 \to \mathbb{R}$ such that for all $x, y \in \text{dom}(f)$ and $\mu \geq 0$,*

$$\|x - y\| \leq e^{-\Omega(\|x\|,\mu)} \quad \Rightarrow \quad \|f(x) - f(y)\| \leq e^{-\mu}.$$

In particular, f is continuous.

One the most important properties of ATSP is closure under composition.

Theorem 9 (ATSP closure by composition) *If $f, g \in \text{ATSP}$, and $f(\text{dom}(f)) \subseteq \text{dom}(g)$, then $g \circ f \in \text{ATSP}$.*

Since the class ATSP is, in a sense, an extension of the class GPVAL, one should expect that the latter is included in the former. This is true in a certain type of domain, which is enough for our purposes:

Definition 10 *A set X is called a star domain if there exists $x_0 \in X$ such that for all $x \in X$ the line segment from x_0 to x is in X. Such an x_0 is called a vantage point.*

The following theorem is from [BGP17b].

Theorem 11 *If $f \in \text{GPVAL}$ has a star domain with a vantage point in $\mathbb{R}_G$ then $f \in \text{ATSP}$.*

One key element to extend the polynomial characterization of P and FP given in [BGP16a] to higher complexity classes was to understand which ones of the above properties could be maintained for the general case of Definition 4 independently from the functions chosen to play the role of space and time boundaries. Indeed, some proofs of the properties listed above explicitly make use of properties of polynomials, and exploit the equivalence between ATSP and other analog classes such as the one of Definition 6, that does not apply for the class with generic boundaries. More importantly, it is essential to identify which of the above properties are necessary for the whole construction to hold and lead to the desired equivalence. As we will show, the fundamental properties in this sense are closure under arithmetic operations and closure under composition. Therefore, our modification of the polynomial construction is designed with the intent of enabling these two closure properties for classes of the type of Definition 4, while at the same time allowing some freedom on the choice of the time and space boundaries.

3 Computing discrete functions: the polynomial time case

In this section we review several useful results from [BGP17b]. In particular we will see how we can code the transition function of a Turing machine (TM for

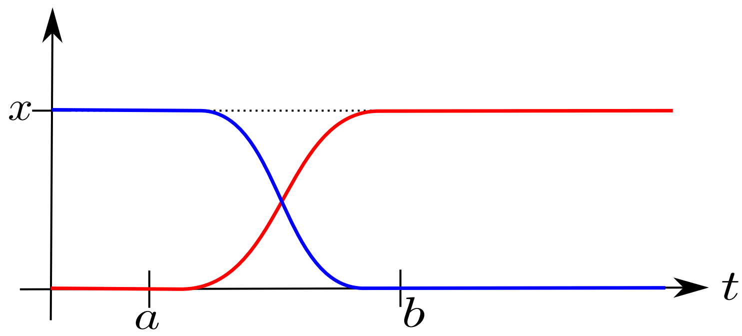

Figure 2: The graphs of functions $\text{hxl}_{[a,b]}$ (in blue) and $\text{lxh}_{[a,b]}$ (in red) having last argument x.

short) into a function in ATSP and how we can iterate it a polynomial number of times, again using a function in ATSP. This will imply that any function $f : \Gamma^* \to \Gamma^*$ computable in polynomial time, where Γ is a finite alphabet, will belong to ATSP in a certain sense. Namely, the notion of emulable function will have to be introduced to be able to go from Γ^* to $\mathbb{R}^k$, in a way which is similar to what is done when using the representation approach [BHW08], [Wei00] in computable analysis. All these notions and ideas will be helpful later when tackling the exponential time case.

3.1 Simulating Turing machines

In this section we will encode the configuration of a Turing machine M as an element of $\mathbb{R}^4$ and we will build a corresponding transition function $f_M : \mathbb{R}^4 \to \mathbb{R}^4$. This transition function is built by composing some basic functions, which can be seen as "bricks" to build more complex functions with a desired behavior. Here we present a selection of those function which will be useful on what follows.

Proposition 12 *Let $G \subseteq \mathbb{R}_G^d \subseteq \mathbb{R}^d$ be a finite subset and $f : G \to \mathbb{R}_G \subseteq \mathbb{R}$ be a function. Then there exists a function $L_f \in$ ATSP (Lagrange polynomial), $L_f : \mathbb{R}^d \to \mathbb{R}$, with the property that $L_{f\restriction G} = f$, where $L_{f\restriction G}$ denotes the restriction of L_f to G. In other words, there exists a function $L_f \in$ ATSP which extends G to the whole real domain.*

Proposition 13 *The real functions $x \to |x|, \max, \min$ all belong to* ATSP.

Other two crucial functions are lxh and hxl that allow to smoothly approximate a *step* function in a continuous manner. Their graph is depicted in Fig. 2.

Proposition 14 (Low-X-High and High-X-Low) *For every $I = [a, b]$, $a, b \in \mathbb{R}_G$ there exist two real functions $\text{lxh}_I, \text{hxl}_I \in$ ATSP such that for any $\mu > 0$ and $x, t \in \mathbb{R}$ one has:*

- lxh_I *is of the form* $\text{lxh}_I(t,\mu,x) = \phi_1(t,\mu,x)x$*, where* $\phi_1(t,\mu,x) \in \text{ATSP}$ *and* $0 < \phi_1(t,\mu,x) < 1$*;*
- hxl_I *is of the form* $\text{hxl}_I(t,\mu,x) = \phi_2(t,\mu,x)x$*, where* $\phi_2(t,\mu,x) \in \text{ATSP}$ *and* $0 < \phi_2(t,\mu,x) < 1$*;*
- *If* $t \leq a$*, then* $|\text{lxh}_I(t,\mu,x)| \leq e^{-\mu}$ *and* $|x - \text{hxl}_I(t,\mu,x)| \leq e^{-\mu}$*;*
- *If* $t \geq b$*, then* $|x - \text{lxh}_I(t,\mu,x)| \leq e^{-\mu}$ *and* $|\text{hxl}_I(t,\mu,x)| \leq e^{-\mu}$*;*
- *In all cases,* $|\text{lxh}_I(t,\mu,x)| \leq |x|$ *and* $|\text{hxl}_I(t,\mu,x)| \leq |x|$.

Before showing how a configuration can be encoded into an element of $\mathbb{R}^4$ and how the transition function of a Turing machine M can be encoded as a function $f_M : \mathbb{R}^4 \to \mathbb{R}^4$, we present the notion of Turing machine that we will use on what follows.

Definition 15 (Turing Machine) *A Turing Machine is a tuple* $M = (Q, \Sigma, \Gamma, b, \delta, q_0, q_\infty)$*, where* $Q = \{0, ..., m\}$ *are the states of the machine,* $\Sigma = \{s_0, ..., s_{k-2}\}$ *is the tape alphabet and* $b = s_0$ *is the blank symbol,* $\Gamma \subseteq \Sigma - b$ *is the input alphabet,* $q_0 \in Q$ *is the initial state,* $q_\infty \in Q$ *is the halting state, and* $\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, S, R\}$ *is the transition function with* $L = -1, S = 0, R = 1$*. We write* $\delta_1, \delta_2, \delta_3$ *as the components of* δ*. That is* $\delta(q,\sigma) = (\delta_1(q,\sigma), \delta_2(q,\sigma), \delta_3(q,\sigma))$*, where* δ_1 *is the new state,* δ_2 *the new symbol, and* δ_3 *the head move direction. We require that* $\delta(q_\infty, \sigma) = (q_\infty, \sigma, S)$*.*

Note that each symbol s_i from Σ can be associated uniquely to the natural number $i \in \mathbb{N}$ without any danger of confusion. Hence, in what follows, we will assume that $\Sigma = \{0, ..., k-2\}$, without making an explicit reference to the correspondence $s_i \mapsto i$, in order the simplify the notation. We also note that we assume that Σ has $k-1$ symbols and not k symbols. As indicated in [BGP17b, Remark 36] this assumption is important to obtain the continuous (real) map of [BGP17b] which simulates the transition function of a Turing using the encoding of configurations of Definition 17. Other continuous transition functions exist (see e.g. [GCB08]) where this requirement is not needed, but they are "less efficient" and not suitable enough for complexity results.

Definition 16 (Configuration) *A configuration for a Turing Machine* M *is a tuple* $c = (x, \sigma, y, q)$*, where* $x \in \Sigma^*$ *is the part of the tape at the left of the head,* $y \in \Sigma^*$ *is the part at the right,* $\sigma \in \Sigma$ *is the symbol under the head, and* $q \in Q$ *is the current state. More precisely,* x_1 *is the symbol immediately at the left of the head and* y_1 *is the symbol immediately at the right.*

$$...0\,0\,0\,x_n\,x_{n-1}\ldots\,x_2\,x_1\,\sigma\,y_1\,y_2\ldots\,y_l\,0\,0\,0...$$

The set of configurations of M *is denoted as* C_M*. For an input word* $w \in \Sigma^*$ *the initial configuration is defined by* $c_0(w) = (\lambda, b, w, q_0)$*. Instead, if at the end of the computation the tape contain the word* $w \in \Sigma^*$*, then the final configuration is defined by* $c_\infty(w) = (\lambda, b, w, q_\infty)$*, where* λ *is the empty word.*

We now show how a configuration can be encoded as an element of $\mathbb{R}^4$.

Definition 17 *Let $c = (x, \sigma, y, q)$ be a configuration of a Turing machine M with an alphabet $\Sigma = \{0, ..., k-2\}$. Then the real encoding of c is defined as $\langle c \rangle = (0.x, \sigma, 0.y, q) \in \mathbb{Q} \times \Sigma \times \mathbb{Q} \times Q$ where $0.x = x_1 k^{-1} + x_2 k^{-2} + ... + x_{|x|} k^{|x|} \in \mathbb{Q}$ and similarly for $0.y \in \mathbb{Q}$.*

Let M also denote the transition function of the Turing machine M, i.e. if c is a configuration, then $M(c)$ denotes the configuration which follows from c after one step of the computation of the TM M.

Theorem 18 ([BGP17b]) *Let M be a Turing machine with an alphabet $\Sigma = \{0, ..., k-2\}$. Then there is a function $\overline{M} : \mathbb{R}^5 \to \mathbb{R}^4 \in \mathrm{ATSP}$ with the property that if c is a configuration of M and $\mu > 0$, then for any $\tilde{c} \in \mathbb{R}^4$ one has*

$$\|\langle c \rangle - \tilde{c}\| \leq \frac{1}{2k^2} - e^{-\mu} \quad \Longrightarrow \quad \left\|\overline{M}(\tilde{c}, \mu) - \langle M(c) \rangle\right\| \leq k \left\|\langle c \rangle - \tilde{c}\right\|.$$

To simulate the computation of a Turing machine M over the real numbers, we need not only to be able to code the transition function of M as a function over the reals, but also to iterate this function over the set of (the coding of) its configurations $\mathcal{C}_M$. Since we are not using exact quantities, to be useful, the iteration also has to work on the set of elements of $\mathbb{R}^4$ "close enough" to $\mathcal{C}_M$. This type of iteration can be performed for an arbitrary number of times (see e.g. [GCB08]), but at a too high cost in terms of complexity (resources). Since here we want an "efficient" simulation of the Turing machine (for example we want to simulate polynomially many steps of a TM with a function in ATSP, which uses polynomial time and space, and not with a function which uses e.g. exponential space as in [GCB08]), we need a more refined approach.

The following theorem is from [BGP17b] and shows that a function f in ATSP which satisfies certain conditions (the transition function $\overline{M}$ of Theorem 18 satisfies those conditions) can be iterated a polynomial number of times with a function also in ASTP.

Theorem 19 ([BGP17b], Polynomial iteration of a function in ATSP) *Let $f : \mathbb{R}^m \to \mathbb{R}^m \in \mathrm{ATSP}$, $\eta \in [0, \frac{1}{2}[$ and assume that there exists a family of subsets $I_n \subseteq I$, $n \in \mathbb{N}$, and polynomials $\Omega : \mathbb{R} \to \mathbb{R}$, $\Pi : \mathbb{R}^2 \to \mathbb{R}$ such that for all $n \in \mathbb{N}$:*

- *$I_{n+1} \subseteq I_n$ and $f(I_{n+1}) \subseteq I_n$;*
- *For all $x \in I_n$, $\left\|f^{[n]}(x)\right\| \leq \Pi(\|x\|, n)$;*
- *For all $x \in I_n, y \in \mathbb{R}^m, \mu \geq 0$, if $\|x - y\| \leq e^{-\Omega(\|x\|)-\mu}$ then $y \in I$ and $\|f(x) - f(y)\| \leq e^{-\mu}$.*

Let $f^*_\eta(x, u) = f^{[n]}(x)$ for $x \in I_n$ and $u \in [n - \eta, n + \eta]$, where $n \in \mathbb{N}$. Then, $f^*_\eta \in \mathrm{ATSP}$.

3.2 Equivalence relation between FP and ATSP

Using the results of the previous section it can be shown [BGP17b] that FP (the class of functions $f : \Gamma^* \to \Gamma^*$ computable in polynomial time, where Γ is a finite alphabet) and ATSP are equivalent using a notion of emulation that we now present. From now on, we fix an alphabet Γ and we assume that Γ comes with an injective mapping $\gamma : \Gamma \to \mathbb{N}\backslash\{0\}$, which means that every non-blank symbol of the alphabet has a unique assigned positive number (the blank is assigned to 0). By extension, γ applies letter wise over words.

Definition 20 (Discrete emulation) *Let G be a set of functions from $\mathbb{R}^2$ to $\mathbb{R}^2$. The function $f : \Gamma^* \to \Gamma^*$ is called emulable under G if there exists $g \in G$ and $k = 2 + max(\gamma(\Gamma))$ such that for any word $w \in \Gamma^*$ one has*

$$g(\Psi_k(w)) = \Psi_k(f(w)) \tag{2}$$

where $\Psi_k(w) = \left(\sum_{i=1}^{|w|} \gamma(w_i)k^{-i}, |w|\right)$. In these circumstances we also say that g emulates f (see diagram below).

$$\begin{array}{ccc} \Gamma^* & \xrightarrow{f} & \Gamma^* \\ {\scriptstyle\Psi_k}\big\downarrow & & \big\downarrow{\scriptstyle\Psi_k} \\ \mathbb{R}^2 & \xrightarrow{g} & \mathbb{R}^2 \end{array}$$

Note that the above definition could be generalized to consider other encodings from Γ^* to $\mathbb{R}^n$ instead of Ψ_k. However, since the encoding Ψ_k as defined above was used in [BGP17b], for simplicity and for coherence of the presentation with what was done in [BGP17b], we will continue to use the encoding Ψ_k here.

As an intuition behind the above definition, a function g defined over a continuous domain emulates a function f defined over a discrete domain if the operation of encoding commutes with the applications of the functions. This means that, given a fixed input word from a discrete alphabet, encoding from the discrete to the continuous can be done either directly to the input before the application of function g, either after the application of function f to the input, yielding the same result.

For simplicity, later on we will use the notation $0.\gamma(w)$ to denote the quantity $\sum_{i=1}^{|w|} \gamma(w_i)k^{-i}$. In particular, $\Psi_k(w) = (0.\gamma(w), |w|)$.

Theorem 21 ([BGP17b], FP equivalence) *Let $f : \Gamma^* \to \Gamma^*$. Then $f \in$ FP if and only if f is emulable under* ATSP.

To show that if $f \in$ FP, then f is emulable under ATSP, recall that if $f \in$ FP, then there is some TM M and some polynomial p such that $f(x)$ can be computed by the TM M in time $p(|x|)$. By Theorem 18, there is a function $\overline{M} \in$ ATSP such that $\overline{M}$ simulates M on $\mathbb{R}^4$ using the encoding given by Definition 17 (and a suitable μ. See [BGP17b] for more details). This function can then be iterated using Theorem 19 to obtain a corresponding function $\overline{M}_\eta^* \in$ ATSP. Let

$w \in \Gamma^*$ be an input for f. Then $0.\gamma(f(w))$ is given as $\pi_3(\overline{M}^*_\eta(c_0(w), p(|w|))) = g_1(\Psi_k(w))$, with $g_1 \in$ ATSP, where $\pi_i : \mathbb{R}^n \to \mathbb{R}$ is defined as $\pi_i(x_1, \ldots, x_n) = x_i$. To show (2), i.e. that f is emulable under ATSP, it just remains to show that $|f(w)|$ can be computed from w and $|w|$ by a function g_2.

In [BGP17b], it was shown that there is a function $tlength_M : \mathbb{R} \times \mathbb{N} \to \mathbb{N}$ in ATSP such that $tlength_M(0.\gamma(w), n) = |w|$ when $n \geq |w|$. Furthermore, since a TM can only add, at most, one symbol per computation step, we conclude that $|f(w)| \leq |w| + p(|x|)$. Then $|f(w)| = tlength_M(0.\gamma(f(w)), |w| + p(|x|)) = tlength_M(g_1(\Psi_k(w)), |w| + p(|x|)) = g_2(\Psi_k(w))$, where $g_2 \in$ ATSP. Taking $g = (g_1, g_2) \in$ ATSP, we conclude that f is emulable under ATSP. This yields one direction of the proof of the above result. The reverse direction relies on numerically simulating the ODE used in ATSP using the efficient simulation of [PG16]. More details will be given in the following section.

4 Computing discrete functions: the exponential case

One might think that the class FEXPTIME of functions computable in exponential time by Turing machines can be obtained similarly as section 3 by considering the class ATSE as the class formed by all functions in ATS(Π, Υ) where Π, Υ are exponentials functions. Unfortunately, a detailed analysis of the arguments used in [BGP17b] shows that to obtain the equivalence of FP with ATSP, we need to use two closure properties of the ATSP class, the closure under arithmetic operations and the closure under composition. As discussed in section 3 the fact that these properties hold for the exponential ATSE class is not obvious. Specifically, it can be easily shown that the closure by composition of the analog class is only possible if the class of functions used as complexity bounds in Definition 4 satisfies closure properties under composition. This is not a problem when the bounds are polynomial, as in ATSP, since the composition of two polynomials is again another polynomial, but this poses problems when composing exponential bounds, since the composition of two exponentials is not necessarily exponential. Therefore ATSE as defined above in not closed under composition.

In the present paper we solve the above problem and define a class ATSE to characterize the exponential case in a manner that allows compositions compatible with the argument used in [BGP17b] to show that FP with ATSP are equivalent. In this manner we will be able to show that ATSE is equivalent to FEXPTIME. Although we use FEXPTIME in our arguments and definitions, they can be applied to a broader class of (super-polynomial) functions as we will later show in section 5 (see Theorem 35). In the following we are going to define the class of boundaries that we are going to consider for the exponential case, that we call *exponential boundary functions*.

Definition 22 *We say that a function $f : \mathbb{R} \to \mathbb{R}$ is an* exponential boundary *if $f(x) = q(x)c^{p(x)}$ for some polynomials p, q and $c \in \mathbb{R}_G$.*

We first define the class ATSE which will emulate FEXPTIME in the sense of Definition 20. Following Definition 5, we might be tempted to define ATSE similarly as ATSP, in the sense that $f \in$ ATSE if $f \in \mathrm{ATS}(\Pi, \Upsilon)$ for some exponential functions Π and Υ. While that idea worked well for the polynomial case, it no longer works for exponential functions. To see this with an example, assume that we characterized ATSE as just described. Then given some $f \in$ ATSE, to compute $f(x)$ with accuracy $e^{-\mu}$, we would need to wait a time $t^* = \Pi(\|x\|, \mu)$ exponential in $\|x\|$ and μ to get an approximation of $f(x)$ with accuracy $e^{-\mu}$, due to the second condition of Definition 4. Moreover, at time t^*, we would have that the norm of the solution y of the ODE computing $f(x)$ is bounded by $\Upsilon(\|x\|, t^*) = \Upsilon(\|x\|, \Pi(\|x\|, \mu))$, which is a double exponential in both $\|x\|$ and μ, while what would be natural is that $\|y(t^*)\|$ is bounded by an exponential in $\|x\|$ and μ, and not a double exponential in these parameters. Note that this problem does not happen when both Π and Υ are polynomials: since the composition of two polynomials is again a polynomial, when computing $f(x)$ with accuracy $e^{-\mu}$ we will be able to use an approximation $y(t^*)$ which norm is bounded by a polynomial in $\|x\|$ and μ.

Our main insight to solve the above problem is that, contrarily to what was done for the case of ATSP, the complexity bounds for ATSE will depend differently on each of the parameters $\|x\|, \mu$, and t. Namely, the dependence will be exponential (or belong to whatever class we are considering) on $\|x\|$, but only polynomial on the remaining parameters.

Another problem that we face is that we expect ATSE to not be closed under composition. This is because it is natural that the exponential function e^x belongs to ATSE. Thus if ATSE would be closed under composition, then the double exponential function e^{e^x} would also belong to it. However, this is not natural since we expect that the double exponential function grows "too quickly" to belong to ATSE. This poses a problem since in [BGP17b] one starts from several "simple functions" in ATSP to build more complex functions, via composition and other operations, which still belong to ATSP and perform useful tasks such as simulating the transition function of a given Turing machine, etc. Therefore we encounter an apparent contradiction, reaching a point where closure by composition is required by the structure of the simulation, but at the same time it is not natural and unfeasible at an exponential level. Here we solve this problem by noting that to prove our main equivalence results it will be enough to compose functions in ATSP with functions in ATSE and that composing functions in ATSP with functions in ATSE still yields a function in ATSE. Hence a pure composition at an exponential level (for the ATSE class) is never needed. Nevertheless, extra care is still required in the analysis, since a similar result has not been obtained for the other direction of the composition between ATSE and ATSP functions (i.e. $f \circ g$ might not belong to ATSE if $f \in$ ATSE and $g \in$ ATSP).

4.1 Definitions of the exponential analog classes

Definition 23 (ATSE) *Let $f \subseteq \mathbb{R}^n \to \mathbb{R}^m$. Then $f \in \text{ATSE}$ if and only if $f \in \text{ATS}(\Pi, \Upsilon)$ for some $\Pi, \Upsilon : \mathbb{R}^2 \to \mathbb{R}$ with the following properties:*

- *$\Pi(\|x\|, \mu) = \Pi_1(\|x\|)\Pi_2(\mu)$ for an exponential boundary function Π_1 and a polynomial function Π_2 (time bound);*
- *$\Upsilon(\|x\|, t) = \Upsilon_1(\|x\|)\Upsilon_2(t)$ for an exponential boundary function Υ_1 and a polynomial function Υ_2 (space bound).*

In other words, a function belongs to ATSE if and only if the complexity bounds depend exponentially on $\|x\|$ (x is the argument of f) and polynomially on the remaining parameters. Quite naturally, this exponential class is closely related (by means of Theorem 25) to a class of *Exponentially-Bounded-Generable-Functions*, or GEVAL, whose definition is a straightforward extension of the polynomial version GPVAL described previously.

Definition 24 (GEVAL) *Let D be a domain in $\mathbb{R}^k$ and let $f : D \to \mathbb{R}^m$. We say that $f \in \text{GEVAL}$ if and only if there exists an exponential boundary function $sp : \mathbb{R} \to \mathbb{R}$, $n \geq m$, a $n \times k$ matrix p consisting of polynomials with coefficients in $\mathbb{R}_G$, $x_0 \in \mathbb{R}_G^d \cap D$, $y_0 \in \mathbb{K}^n$ and $y : D \to \mathbb{R}^n$ satisfying for all $x \in D$:*

- *$y(x_0) = y_0$ and $J_y(x) = p(y(x))$;*
- *$f(x) = (y_1(x), \ldots, y_m(x))$;*
- *$\|y(x)\| \leq sp(\|x\|)$.*

Note that, since exponential functions can be easily expressed as solutions of polynomial ODEs (for example $f(x) = e^x$ is the solution of $y' = y, y(0) = 1$) and are always bounded by some exponential boundary function, they trivially belong to the class GEVAL defined above.

4.2 Properties of the exponential analog classes

Using straightforward adaptations of the proofs presented for their polynomial counterparts in [BGP17b], we get the following properties:

Theorem 25 *If $f \in \text{GEVAL}$ has a star domain with a vantage point in $\mathbb{R}_G$, then $f \in \text{ATSE}$.*

As a consequence of the above theorem together with our previous observation, exponential functions defined over star domains belong to ATSE.

Theorem 26 (Exponential bound of ATSE) *Let $f \in \text{ATSE}$. Then there exists an exponential boundary function E such that $\|f(x)\| \leq E(\|x\|)$.*

As mentioned earlier, the closure of the ATSE class by composition and arithmetic operations is not straightforward, but can be obtained (in a limited form for composition) under certain circumstances.

Theorem 27 (ATSE closure by arithmetic operations) *If $g, f \in$ ATSE, then $g+f, g-f, g \cdot f \in$ ATSE, with the obvious restrictions on the domains of definition.*

Proof. We present the proof only for the closure by product, since the other cases are similar. We prove the theorem for dimension one, but the proof can be easily repeated for higher dimensions. Let us consider a function $f \in \text{ATSE}(\Pi_1\Pi_2, \Upsilon_1\Upsilon_2)$ and a function $g \in \text{ATSE}(\Pi_1^*\Pi_2^*, \Upsilon_1^*\Upsilon_2^*)$ with parameters d, p, q and d^*, p^*, q^* respectively, where d, d^* represent the dimensions of the two dynamical systems, p, p^* the polynomials defining the right hand terms of the differential equations and q, q^* the exponential boundary functions defining the initial conditions (recall that Definition 23 depends on Definition 4 which defines ATS). Recall that, by definition of the class, we have that $\Pi_1, \Upsilon_1, \Pi_1^*, \Upsilon_1^*$ are exponential boundary functions and that $\Pi_2, \Upsilon_2, \Pi_2^*, \Upsilon_2^*$ are polynomials. We also assume, without loss of generality, that these functions are non-decreasing. Let $x \in \text{dom}\, f \cap \text{dom}\, g$ and consider the following system:

$$\begin{cases} y(0) = q(x) \\ y'(t) = p(y(t)) \end{cases} \quad \begin{cases} z(0) = q^*(x) \\ z'(t) = p^*(z(t)) \end{cases} \quad \begin{cases} w(0) = y_1(0)z_1(0) \\ w'(t) = p_1(y(t))z_1(t) + y_1(t)p_1^*(z(t)) \end{cases} \tag{3}$$

where we are using the notation p_1, z_1 and p_1^* to indicate the first component of the vectors. Note that the solution of (3) for the variable w is: $w(t) = y_1(t)z_1(t)$ and that this system has polynomial right hand terms. We can now proceed with the analysis of the boundaries. Since $f, g \in$ ATSE, from Definition 23, we conclude (taking $\mu = 0$) that for $t \geq \Pi_1(\|x\|)\Pi_2(0)$ we have $\|(y_1(t), \ldots, y_m(t)) - f(x)\| \leq e^{-0} = 1$ and $\|y(t)\| \leq \Upsilon_1(\|x\|)\Upsilon_2(t)$. This implies that $\|f(x)\| \leq 1 + \|y(t)\| \leq 1 + \Upsilon_1(\|x\|)\Upsilon_2(t)$. In particular, taking $t = \Pi_1(\|x\|)\Pi_2(0)$ gives $\|f(x)\| \leq 1 + \Upsilon_1(\|x\|)\Upsilon_2(\Pi_1(\|x\|)\Pi_2(0))$. A similar argument yields $\|g(x)\| \leq 1 + \Upsilon_1^*(\|x\|)\Upsilon_2^*(\Pi_1^*(\|x\|)\Pi_2^*(0))$. Denote by $l(\|x\|)$ and $l^*(\|x\|)$ these two bounds for $\|f(x)\|$ and $\|g(x)\|$, respectively (i.e $l(\|x\|) = 1 + \Upsilon_1(\|x\|)\Upsilon_2(\Pi_1(\|x\|)\Pi_2(0))$ and similarly for $l^*(\|x\|)$). Note that we get that $l(\|x\|)$ and $l^*(\|x\|)$ are exponential boundaries functions, and this would not have been true if $\Pi_2, \Upsilon_2, \Pi_2^*, \Upsilon_2^*$ were not polynomials. Now consider $t \geq \Pi_1(\|x\|)\Pi_2(\mu + \ln 2l^*(\|x\|))$; then $\|f(x) - y_1(t)\| \leq e^{-(\mu + \ln 2\|g(x)\|)}$ and in the same way if $t \geq \Pi_1^*(\|x\|)\Pi_2^*(\mu + \ln(2 + 2l(\|x\|)))$ then $\|g(x) - z_1(t)\| \leq e^{-(\mu + \ln(2+2\|f(x)\|))}$. Therefore if we consider times greater than the maximum of these two bounds we have:

$$\|y_1(t)z_1(t) - f(x)g(x)\| \leq \|(y_1(t) - f(x))g(x)\| + \|y_1(t)(z_1(t) - g(x))\| \leq e^{-\mu}$$

this proves the time bound of Definition 23. The space bound is verified by observing that $\|y_1(t)\| \leq l(\|x\|)$ and $\|z_1(t)\| \leq l^*(\|x\|)$ and this concludes the proof. ■

Theorem 28 (Composition of ATSE and ATSP) *Let f be a function, $f \in$ ATSE and g be a function, $g \in$ ATSP. Then $g \circ f \in$ ATSE.*

Proof. Let $f : \mathbb{R}^n \to \mathbb{R}^m$ and let $g : \mathbb{R}^m \to \mathbb{R}^l$. We will show that $g \circ f \in$ ATSE by using proposition 6 for g. Indeed, if $g \in$ ATSP, by proposition 6, then there are polynomials Π, Υ, Λ such that proposition 6 holds with corresponding r, δ, z_0, where r is the polynomial defining the differential equation, $\delta \in \mathbb{R}$ is the same parameter as in proposition 6, and z_0 is the initial condition of the system. In the same way we obtain that there exist two exponential boundary functions Π'_1, Υ'_1, and two polynomials Π'_2, Υ'_2 such that $f \in \text{ATSE}(\Pi'_1\Pi'_2, \Upsilon'_1\Upsilon'_2)$ with corresponding parameters d, p, q for the dimension and polynomials defining the system. Assume, without loss of generality, that all the functions used as boundaries for definitions of f and g are increasing functions. Let $x \in \mathbb{R}^n$ and consider the following system:

$$\begin{cases} y(0) = q(x) \\ y'(t) = p(y(t)) \end{cases} \quad \begin{cases} z(0) = z_0 \\ z'(t) = r(z(t), y_{1..m}(t)) \end{cases} \quad \begin{cases} x(0) = x \\ x'(t) = 0 \end{cases}$$

where $y_{1..m}(t) = (y_1(t), \ldots, y_m(t))$. Define $v(t) = (x(t), y(t), z(t))$. Then it immediately follows that v satisfies a PIVP of the form:

$$\begin{cases} v(0) = poly(x) \\ v'(t) = poly(v(t)). \end{cases} \tag{4}$$

We will show that the polynomial IVP (4) computes $g \circ f$ according to Definition 23. First we check the space bound condition of that definition. By definition of v we have

$$\begin{aligned} \|v(t)\| &= max(\|x(t)\|, \|y(t)\|, \|z(t)\|) \\ &\leq max(\|x\|, \|y(t)\|, \Upsilon(sup_{u\in[t,t-\delta]}\|y_{1..m}(u)\|, t)) \\ &\leq exp(\|x\|)\ poly(sup_{u\in[t,t-\delta]}\|y(u)\|, t) \\ &\leq exp(\|x\|)\ poly(sup_{u\in[t,t-\delta]}\Upsilon'_1(\|x\|)\Upsilon'_2(u), t) \\ &\leq exp(\|x\|)\ poly(t) \end{aligned}$$

where the notation *poly* and *exp* indicate an unspecified polynomial and an unspecified exponential boundary function, respectively. This proves the space bound.

Let us now tackle the time bound of Definition 23. Define $\bar{x} = f(x)$, $\Upsilon^*(\alpha) = \Upsilon'_1(\alpha)\Upsilon'_2(\Pi'_1(\alpha)\Pi'_2(0)) + 1$ and $\Pi''(\alpha, \mu) = \Pi'_1(\alpha)\Pi'_2(\Lambda(\Upsilon^*(\alpha), \mu)) + \Pi(\Upsilon^*(\alpha), \mu)$. Note that Υ^* is an exponential boundary function. Furthermore, we can easily find two functions, Π^*_1 and Π^*_2, such that $0 \leq \Pi''(\alpha, \mu) \leq \Pi^*_1(\alpha)\Pi^*_2(\mu)$ where Π^*_1 is an exponential boundary function and Π^*_2 is a polynomial. From the fact that $f \in \text{ATSE}(\Pi'_1\Pi'_2, \Upsilon'_1\Upsilon'_2)$, by using time $t^* = \Pi'_1(\|x\|)\Pi'_2(0)$, we conclude that $\|y_{1..m}(t^*) - f(x)\| \leq e^{-0}$, which implies that

$$\begin{aligned}\|\bar{x}\| &= \|f(x)\| \\ &\leq \|y(t^*)\| + 1 \\ &\leq \Upsilon_1'(\|x\|)\Upsilon_2'(t^*) + 1 \\ &= \Upsilon_1'(\|x\|)\Upsilon_2'(\Pi_1'(\|x\|)\Pi_2'(0)) + 1 \\ &= \Upsilon^*(\|x\|).\end{aligned}$$

Let $\mu > 0$. By definition of ATSE, if $t \geq \Pi_1'(\|x\|)\Pi_2'(\Lambda(\Upsilon^*(\|x\|), \mu))$ then $\|y_{1..m}(t) - \bar{x}\| \leq e^{-\Lambda(\Upsilon^*(\|x\|),\mu))} \leq e^{-\Lambda(\|x\|,\mu)}$. Therefore, due to proposition 6, we have

$$t \geq \Pi_1'(\|x\|)\Pi_2'(\Lambda(\Upsilon^*(\|x\|), \mu)) + \Pi(\|\bar{x}\|, \mu) \quad \Rightarrow \quad \|v_{1..l}(t) - g(f(x))\| \leq e^{-\mu}.$$

Note that $\Pi_1'(\|x\|)\Pi_2'(\Lambda(\Upsilon^*(\|x\|), \mu)) + \Pi(\|\bar{x}\|, \mu)$ depends exponentially on $\|x\|$, but only polynomially on μ. This shows the time bound for ATSE. ■

4.3 Analog characterization of FEXPTIME

Finally, we can state one of the main results of this paper. By adapting the construction already developed in [BGP17b], we can show the equivalence between the class ATSE and the class of FEXPTIME of functions computable in exponential time. This equivalence is described by the following theorem:

Theorem 29 (FEXPTIME equivalence) *Let* $f : \Gamma^* \to \Gamma^*$. *Then* $f \in$ FEXPTIME *if and only if* f *is emulable under* ATSE.

Next we proceed with the proof of Theorem 29. We will prove the direct and reverse direction of the theorem separately.

For the direct direction of Theorem 29, we have to show that if $f \in$ FEXPTIME, then f is emulable under ATSE. Suppose that f is computable by a Turing machine M. To achieve our purpose, we will need once again to be able to iterate the transition function of M, which belongs to ATSP, due to Theorem 18, with the help of Theorem 19. The main difference to the polynomial case will be the number of iterations of the transition function required to simulate the functioning of the Turing machine M until it halts. Since $f \in$ FEXPTIME, the number of steps which will have to be simulated is exponential on the size of the input and not polynomial as in the original case analyzed in [BGP17b]. Nevertheless, an exponential version of Theorem 19 is not necessary, and the original polynomial version is enough for our goal. More concretely, let us show in more detail how the construction of the simulation is made.

Let us assume that $M = (Q, \Sigma, b, \delta, q_0, F)$, where $\Sigma = \{0, 1, \ldots, k-2\}$ with $b = 0$, and $\gamma(\Gamma) \subset \Sigma \backslash \{b\}$, and consider an exponential boundary function $e_M(|w|) \equiv K^{|w|} \in \mathbb{N}$ for some constant $K \in \mathbb{N}$ such that for any word $w \in \Gamma^*$ the TM M halts in at most $e_M(|w|)$ steps, that is $M^{[e_M(|w|)]}(c_0(\gamma(w))) = c_\infty(\gamma(f(w)))$. We assume that $\Psi_k(w) = (0.\gamma(w), |w|)$ for any word $w \in \Gamma^*$ as before. Define $\mu = \ln(4k^2)$ and $h(c) = \overline{M}(c, \mu)$ for all $c \in \mathbb{R}^4$. Define $I_\infty = \langle C_M \rangle$, where C_M is the set of all configurations of M, and $I_n = I_\infty + [\epsilon_n, \epsilon_n]^4$ where $\epsilon_n = \frac{1}{4k^{2+n}}$ for all $n \in \mathbb{N}$. Note that $\epsilon_{n+1} \leq \frac{\epsilon_n}{k}$ and that $\epsilon_0 \leq \frac{1}{2k^2} - e^{-\mu}$. Due to Theorem 18, we have that the transition function of this machine, h, satisfies $h \in$ ATSP and $h(I_{n+1}) \subseteq I_n$. In particular $\|h^{[n]}(\bar{c}) - h^{[n]}(c)\| \leq k^n \|c - \bar{c}\|$ for all $c \in I_\infty$, $\bar{c} \in I_n$, and $n \in \mathbb{N}$. Let $\delta \in [0, \frac{1}{2}[$ and define $J = \bigcup_{n \in \mathbb{N}} I_n \times [n - \delta, n + \delta]$. Then apply Theorem 19 to the function h to get $h^*_\delta : J \to I_0 \in$ ATSP such that for all $c \in I_\infty$ and $n \in \mathbb{N}$ we have $h^*_\delta(c, n) = h^{[n]}(c)$.

Let π_i denote the ith projection, that is, $\pi_i(x) = x_i$. Then $\pi_i \in$ ATSP. Take $g(y, l) = \pi_3(h^*_\delta(0, b, y, q_0, e_M(l)))$. Note that g is well defined. Indeed, if $(y, l) = \Psi_k(w)$ for some $w \in \Gamma^*$, then $y = 0.\gamma(w)$, $l = |w|$, and $(0, b, y, q_0) = \langle(\lambda, b, \gamma(w), q_0)\rangle = \langle c_0(\gamma(w))\rangle \in I_\infty$. Therefore, by construction, for any word $w \in \Gamma^*$ we have

$$\begin{aligned}
g(\Psi_k(w)) &= \pi_3(h^*_\delta(\langle c_0(\gamma(w))\rangle, e_M(|w|))) \\
&= \pi_3(h^{[e_M(|w|)]}\langle c_0(\gamma(w))\rangle) \\
&= \pi_3(\langle C_M^{[e_M(|w|)]}(c_0(\gamma(w)))\rangle) \\
&= \pi_3(\langle c_\infty(\gamma(f(w)))\rangle) \\
&= 0.\gamma(f(w)) = \pi_1(\Psi_k(f(w))).
\end{aligned}$$

Recall that to show the validity of the emulation we need to compute $\Psi_k(f(w))$ and so far we only have the first component, the output of the tape encoding, but we miss the second component: its length. To complete the task, we can proceed similarly as in section 3 and apply the function $tlength_M$ from [BGP17b] presented there.

In particular, we get that $tlength_M(g(\Psi_k(w)), |w| + e_M(|w|)) = |0.\gamma(f(w))| = |f(w)|$, since $0.\gamma(f(w))$ does not contain any blank character by definition of γ and $|f(w)| \leq |w| + e_M(|w|)$. What we have just showed above proves that

$$\bar{g}(\Psi_k(w)) \equiv (g(\Psi_k(w)), tlength_M(g(\Psi_k(w)), |w| + e_M(|w|))) \tag{5}$$

emulates f according with Definition 20. To conclude the direct direction of the theorem, the last result to prove is that $\bar{g} \in$ ATSE.

Recalling that $g(\Psi_k(w)) = \pi_3(h^*_\delta(\langle c_0(\gamma(w))\rangle, e_M(|w|)))$, we conclude from Theorem 19 that $h^*_\delta \in$ ATSP. It is trivial to show that $e_M \in$ ATSE. Then, due to the composition theorem, Theorem 28, we obtain that $h^*_\delta(\langle c_0(\gamma(w))\rangle, e_M(|w|)) \in$ ATSE. Due to (5), where $g(\Psi_k(w)) = \pi_3(h^*_\delta(\langle c_0(\gamma(w))\rangle, e_M(|w|)))$, and since $tlength, \pi_3 \in$ ATSP and $h^*_\delta(\langle c_0(\gamma(w))\rangle, e_M(|w|)), e_M \in$ ATSE, we can apply again Theorem 28 to conclude that $\bar{g} \in$ ATSE and thus prove the direct direction of Theorem 29.

We now proceed with the reverse direction of the proof. In other words, we have to show that if f is emulable under ATSE, then $f \in$ FEXPTIME. The argument presented here is similar to the one already provided in [BGP17b] for the ATSP equivalence. The main difference is the nature of the boundaries. To be able to prove this result, it is necessary to introduce a theorem about the complexity of solving polynomial differential equations. The proof and more details about the theorem can be found in [PG16]. Before we state this theorem, we need to introduce some notation. We use the multi-index notation for multivariate polynomials as follows: a monomial $x_1^{\alpha_1} \dots x_k^{\alpha_k}$ is represented by x^α, where $x = (x_1, \dots, x_k)$, $\alpha = (\alpha_1, \dots, \alpha_k)$, $|a| = \alpha_1 + \dots + \alpha_k$ is the degree of the monomial. For any multivariate polynomial $p(x) = \sum_{|\alpha| \le k} a_\alpha x^\alpha$ we call k the degree of p, $k = \deg(p)$, if k is the minimal integer for which the condition $p(x) = \sum_{|\alpha| \le k} a_\alpha x^\alpha$ holds, and we denote the sum of the norm of the coefficients by $\Sigma p = \sum_{|\alpha| \le k} |a_\alpha|$. For a vector of polynomials, we define the degree and Σp as the maximum over all components. For any continuous y and polynomial p, define the *pseudo-length* as:

$$PsLen_{y,p}(a, b) = \int_a^b \Sigma p \, \max(1, \|y(u)\|)^{\deg(p)} du$$

We are now in condition to state the above mentioned theorem.

Theorem 30 ([PG16]) *Let $I = [a, b]$ be an interval, $p : \mathbb{R}^n \to \mathbb{R}^n$ be such that each of its one-dimensional components is a polynomial of degree at most k, and let $y_0 \in \mathbb{R}^n$. Assume that $y : I \to \mathbb{R}^n$ is a solution of the IVP $y(a) = y_0$, $y'(t) = p(y(t))$. Then $y(b)$ can be computed with precision $2^{-\mu}$ in time bounded by poly$(\deg(p), PsLen_{y,p}(a, b), \log \|y_0\|, \Sigma p, \mu)^n$.*

More precisely, the theorem states that there is a Turing machine M such that for any oracle O representing (a, y_0, p, b) and any input $\mu \in \mathbb{N}$, M outputs a value $M^O(\mu)$ satisfying $\|M^O(\mu) - y(b)\| \le 2^{-\mu}$, where y is the solution of the previous differential equation and the number of steps of the machine is bounded by the above expression.

Now we can continue with the proof of the reverse direction of Theorem 29. Assume that f is emulable under ATSE. Let g be the function used in the emulation which satisfies (2). Then $g \in \text{ATSE}(\Pi_1\Pi_2, \Upsilon_1\Upsilon_2)$, where d is the dimension and p, q are the polynomials generating the ATSE dynamical system associated to g. Recall that, by definition of the class ATSE, Υ_1, Π_1 are exponential boundary functions and Υ_2, Π_2 are polynomials. Let $w \in \Gamma^*$. We will now describe an FEXPTIME algorithm to compute $f(w)$. Consider the following system

$$y'(t) = p(y(t)), \; y(0) = q(\Psi_k(w)).$$

Note that the coefficients of p, q and $q(\Psi_k(w))$ are polynomially computable in the sense of computable analysis. To show that $f \in$ FEXPTIME, we have to present an algorithm which, on input x computes $f(x)$ in exponential time. The algorithm works in two steps: first, we compute a rough approximation of the output to be able to guess the length of the output. Then we rerun the system with enough precision to get the full output.

Let $t_w = \Pi_1(|w|)\Pi_2(2)$ for any $w \in \Sigma^*$. Note that t_w is computable and that it is exponentially bounded in $|w|$ because Π_1 is an exponential boundary function. Now apply Theorem 30 to compute $\bar{y} = (\bar{y}_1, \bar{y}_2) \in \mathbb{R}^2$ such that $\|\bar{y} -$

$y(t_w)\| \leq e^{-2}$. This operation takes a computational time that is exponential in $|w|$ because t_w is exponentially bounded and because

$$\begin{aligned} PsLen_{y,p}(0,t_w) &= \int_0^{t_w} \Sigma p \ \max(1, \|y(u)\|)^{\deg(p)} du \\ &\leq \int_0^{t_w} \Sigma p \ \max\left(1, \sup_{t\in[0,t_w]} \|y(t)\|\right)^{\deg(p)} du \\ &= \Sigma p \ \max\left(1, \sup_{t\in[0,t_w]} \|y(t)\|\right)^{\deg(p)} t_w \\ &= poly(t_w, \sup_{[0,t_w]} \|y(t)\|) \end{aligned}$$

and, by construction, $\|y(t)\| \leq \Upsilon_1(\|\Psi_k(w)\|)\Upsilon_2(t_w)$ for $t \in [0, t_w]$ where Υ_1 is an exponential boundary function and Υ_2 is a polynomial (we assume, without loss of generality, that Υ_2 is an increasing function), which implies that $PsLen_{y,p}(0,t_w) \leq poly(t_w, \Upsilon_1(\|\Psi_k(w)\|)\Upsilon_2(t_w)) = \exp(|w|)$. Furthermore, by definition of t_w, we have $\|y(t_w) - g(\Psi_k(w))\| \leq e^{-2}$ and thus $\|\bar{y} - \Psi_k(f(w))\| \leq 2e^{-2} \leq \frac{1}{3}$. But, since $\Psi_k(f(w)) = (0.\gamma(f(w)), |f(w)|)$, from $\bar{y}_2$ we can find $|f(w)|$ by rounding to the closest integer. In other words we can compute $|f(w)|$ in exponential time in $|w|$. Note that this implies that $|f(w)|$ is at most exponential in $|w|$.

Let $t'_w = \Pi_1(|w|)\Pi_2(2 + |f(w)| \ln k)$. Note that there is always an exponential boundary function on $|w|$ such that it is greater than t'_w. Indeed, Π_1 is an exponential boundary function, Π_2 is a polynomial, and $|f(w)|$ is at most exponential in $|w|$. We can then apply the same reasoning and use again Theorem 30 to get an $\tilde{y}$ such that $\|\tilde{y} - y(t'_w)\| \leq e^{-2-|f(w)| \ln k}$. Once again this operation takes a time exponential in $|w|$. Furthermore, $\|\tilde{y}_1 - 0.\gamma(f(w))\| \leq 2e^{-2-|f(w)| \ln k} \leq \frac{1}{3}k^{-|f(w)|}$. We claim that this allows us to recover $f(w)$ unambiguously in exponential time in $|w|$. Indeed, it implies that $\|k^{|f(w)|}\tilde{y}_1 - k^{|f(w)|}0.\gamma(f(w))\| \leq \frac{1}{3}$. Unfolding the definition shows that $k^{|f(w)|}0.\gamma(f(w)) = \sum_{i=1}^{|f(w)|} \gamma(f(w)_i)k^{|f(w)|-i} \in \mathbb{N}$, thus by rounding $k^{|f(w)|}\tilde{y}_1$ to the closest integer we recover $\gamma(f(w))$ and then $f(w)$. This is all done in polynomial time in $|f(w)|$, and so in exponential time in $|w|$. This completes the proof of Theorem 29.

5 Going beyond the exponential case

As we have showed previously in this paper, the key intuition of splitting the dependence of the time and space boundaries from one single term, Υ, into the product of two separate components $\Upsilon_1\Upsilon_2$ with different behaviors has allowed us to capture the full power of exponential time computation with the suitable dynamical systems of polynomial differential equations. Since this procedure has successfully divided the part of the construction that may continue to depend polynomially on the input from the part that has to depend exponentially

from the input in order to obtain the equivalence, and since this division process does not seem to explicitly depend on properties possessed only by exponential type of boundaries, it is natural to wonder if an equivalence can be obtained in the same way for other standard complexity classes in a straightforward manner. More precisely, if the second part of the time and space boundaries (Π_2 and Υ_2 in the original ATSE definition) is kept polynomial, it is natural to wonder which classes of functions can be used as first term (Π_1 and Υ_1) in order to characterize other classes from standard complexity theory.
Following every step of the proofs, starting from the basic properties and definitions of the GEVAL and ATSE classes, it is possible to see that four conditions are sufficient for the construction to hold.

In this section we extend the result of Theorem 21 to other complexity classes. More precisely, we list which conditions the analog classes involved have to satisfy in order to repeat the simulation process already obtained for the polynomial case. First, let us define a version of the ATS class in which the time and space boundaries are defined using functions from a set A of functions over the reals. More concretely, and in a similar manner to Definition 23, we introduce the following definition.

Definition 31 *Let $f \subseteq \mathbb{R}^n \to \mathbb{R}^m$ and let A be a class of functions from $\mathbb{R}_+ = [0, +\infty[$ to $\mathbb{R}_+$. Then $f \in \mathrm{ATS}_\mathrm{p}(A)$ if and only if $f \in \mathrm{ATS}(\Pi, \Upsilon)$ for some $\Pi, \Upsilon : \mathbb{R}^2 \to \mathbb{R}$ with the following properties:*

- *$\Pi(\|x\|, \mu) = \Pi_1(\|x\|)\Pi_2(\mu)$ for some function $\Pi_1 \in A$ and a polynomial function Π_2 (time bound);*
- *$\Upsilon(\|x\|, t) = \Upsilon_1(\|x\|)\Upsilon_2(t)$ for some function $\Upsilon_1 \in A$ and a polynomial function Υ_2 (space bound).*

As we already discussed in the previous section, we need to be able to enforce closure by composition and by arithmetic operations for the class $\mathrm{ATS}_\mathrm{p}(A)$ to extend the result of Theorem 21 to other complexity classes. Nevertheless, a more careful analysis of the details of the construction shows that the functions in A should satisfy other additional properties that are trivially shared by polynomials and exponentials, but that are not obvious for a generic class A. Before stating sufficient conditions that ensure closure by composition and by arithmetic operations for the class $\mathrm{ATS}_\mathrm{p}(A)$, we recall the notion of time-constructible functions [Gol08].

Definition 32 (Time-constructible function) *Let $f : \mathbb{N} \to \mathbb{N}$ be a function. We call f time-constructible if there exists a Turing machine M which, given as an input a string 1^n, outputs the binary representation of $f(n)$ in time $O(f(n))$.*

We now present conditions that ensure that the result of Theorem 21 can be extended to other complexity classes.

Definition 33 (Sufficient conditions) *Let A be a class of functions from $\mathbb{R}_+$ to $\mathbb{R}_+$ such that:*

(1) *If* $f, g \in A$ *then there exists* $h \in A$ *such that* $f \star g(x) \leq h(x)$ *for every* $x \in \mathbb{R}_+$*, where* $\star$ *denotes any operator in the list of arithmetical operations:* $(+, -, \times)$

(2) *If* p *is polynomial and* $f \in A$ *then there exists* $g \in A$ *such that* $p \circ f(x) \leq g(x)$ *and* $f \circ p(x) \leq g(x)$ *for every* $x \in \mathbb{R}_+$*. Moreover, the identity operator belongs to* A

(3) *If* $f \in A$*, then there exists* $g \in A$ *such that* $f(n) \leq g(n)$ *for every* $n \in \mathbb{N}$ *and* $g \in \mathrm{ATS_p}(A)$

(4) *If* $f \in A$ *then there exists* $g : \mathbb{N} \to \mathbb{N}$ *and* $h \in A$ *and such that* $f(n) \leq g(n) \leq h(n)$ *for every* $n \in \mathbb{N}$ *and* g *is a time-constructible function.*

The first condition enforces a form of closure for the main arithmetical operations which are of interest to us. The second condition provides enough elements so that we can have a variant of Theorem 28 for the class $\mathrm{ATS_p}(A)$ as well as allowing us to replace a polynomial in the proofs by a member of A when needed (polynomials may not belong to A). The third condition is sufficient to show that if a function is computed by a Turing machine in time $f|_{\mathbb{N}}$, where $f \in A$ [1], then we may replace f by a function $g \in \mathrm{ATS_p}(A)$ which will play a role similar to the time bound e_M in the proof of Theorem 29. The fourth and final condition is due to the fact that we do not have any assurances about the computability or complexity needed to compute elements of A. If some bound f of $\mathrm{ATS_p}(A)$ has these problems, we need to be sure that we can replace it by a *well-behaved* bound g, which can be used to prove the reverse direction of 29, e.g. when computing the value t_w. Of course, the function g should not grow quicker than any element of A, hence there is the need to ensure that there is a function $h \in A$ which grows at least as quickly as g over the naturals.

We recall that if $f \in A$ is a function such that $f(\mathbb{N}) \subseteq [0, +\infty[$, then we say that a Turing machine M computes a set of functions $\mathcal{F}$ in time $O(f(n))$ if there are some $c, n_0 \in \mathbb{N}$ such that if w is a word of length $n \geq n_0$, then M computes $g(w) \in \mathcal{F}$ in time $\leq cf(n)$. Now we can define $\mathrm{FTIME}(A) = \{g | g : \mathbb{N} \to \mathbb{N}$ is a function computable in time $O(f(n))$ for some $f \in A\}$. Another important remark is the following:

Remark 34 *We note that, similarly to what happens to* GPVAL *and* GEVAL*, we can more generally define a class* $\mathrm{GVAL}(A)$*. If* A *satisfies the conditions of Definition 33, then it follows from lemma 24 and corollary 26 of [BGP17a] that* $\mathrm{GVAL}(A)$ *is closed under addition, difference, product, and ODE solving (i. e., if* $f \in \mathrm{GVAL}(A)$*, then a solution of* $y' = f(y)$ *also belongs to* $\mathrm{GVAL}(A)$*).*

At this point we can finally state the following generalized equivalence theorem.

[1] Note that $f(\mathbb{N}) \subseteq [0, +\infty[$. Hence, although $f(n)$ might not belong to $\mathbb{N}$ when $n \in \mathbb{N}$, we can still say that a Turing machine computes in time $\leq f(n)$

Theorem 35 (Generalized equivalence) *Let A be a class of functions that satisfies the conditions of Definition 33. Then given a function $f : \Gamma^* \to \Gamma^*$, we have that $f \in \mathrm{FTIME}(A)$ if and only if f is emulable under $\mathrm{ATS_p}(A)$.*

We will use this theorem to characterize the Grzegorczyk hierarchy with ODEs. In particular, we will also be able to characterize the class of elementary computable functions and the class of primitive recursive functions with ODEs. Nonetheless, before proceeding with the application of Theorem 35 above to the case of the Grzegorczyk hierarchy, in the next section we show that, similarly to the case of the polynomial class ATSP, also the generalized class $\mathrm{ATS_p}(A)$ can be interpreted just by means of the length of the solutions of the ODEs involved. This possibility is particularly relevant because it reduces the number of parameters necessary to describe the complexity of the system. Indeed, while the $\mathrm{ATS_p}(A)$ class needed to make use of both a space boundary (on the norm of the solution) and a time boundary (on the convergence rate), this interpretation makes use of just the length of the solution curve. This perspective was illustrated at a polynomial level in [BGP17b] introducing another analog class called ALP, *Analog-Length-Polynomial*, defined using the length of the solution $y(t)$, and then proving the equivalence of this class with the class ATSP. In the next section we follow a similar path and generalize this result to the case of analog classes constructed over boundaries taken from a generic set of functions A.

6 Generalization of the Analog Length class

We start this section by introducing the definitions required to understand the meaning of the Analog Length class.

Definition 36 *Let $f \subseteq \mathbb{R}^n \to \mathbb{R}^m$. We say that $f \in \mathrm{AL}(\Pi, \Upsilon)$, where $\Pi : \mathbb{R}^2 \to \mathbb{R}$ if and only if there exist a multivariate polynomial p with coefficients in $\mathbb{R}_G$ and a function $q \in \mathrm{GPVAL}$ such that for any $x \in \mathrm{dom}(f)$, there exists (a unique) $y : \mathbb{R} \to \mathbb{R}^d$ satisfying for all $t \geq 0$:*

- *$y(0) = q(x)$ and $y'(t) = p(y(t))$;*
- *$\forall \mu > 0$ if $\mathrm{len}_y(0, t) \geq \Pi(\|x\|, \mu)$ then $\|(y_1(t), \ldots, y_m(t)) - f(x)\| \leq e^{-\mu}$;*
- *$\|y'(t)\| \geq 1$.*

In Definition 36 the first item has the same role of the first condition of the definition of $\mathrm{ATS_p}(A)$, which is to describe the evolution of the dynamical system. The second item is related to the length of the solution $y(t)$. Specifically, we are requiring that whenever a certain length of the solution is reached by the system, where the exact amount is dictated by the boundary Π, then it starts the convergence of the system to the correct value of the function f. The way this convergence is obtained is similar to one of $\mathrm{ATS_p}(A)$. Finally, the third item is necessary to exclude pathological cases in which the evolution of the dynamical system is too slow and where it might happen that there is some $M > 0$ such that $\mathrm{len}_y(0,t) \leq M$ for all $t \geq 0$, thus making the second condition meaningless. A example of such situation is provided by the ODE $y' = -y$ with initial condition $y(0) = 1$. It is not difficult to see that its solution is $y(t) = e^{-t}$ and that as $t \to +\infty$, $y(t)$ moves monotonically from the value 1 to 0. Thus the length of the solution curve of this ODE is bounded even when $t \to +\infty$. We also note that we can avoid this situation, using different requirements. For example, instead of requiring that $\|y'(t)\| \geq 1$ for the last condition, we could have required that there is an component y_i of y such that $|y_i(t)| \geq 1$ for all $t \geq 0$.

Definition 37 *Let $f \subseteq \mathbb{R}^n \to \mathbb{R}^m$ and let A be a class of functions from $\mathbb{R}_+ = [0, +\infty[$ to $\mathbb{R}_+$. Then $f \in \mathrm{AL}(A)$ if and only if $f \in \mathrm{AL}(\Pi)$ for some $\Pi : \mathbb{R}^2 \to \mathbb{R}$ with the property that $\Pi(\|x\|, \mu) = \Pi_1(\|x\|)\Pi_2(\mu)$ for some function $\Pi_1 \in A$ and a polynomial function Π_2.*

Notice that Definition 37 is just the generalization of 36 when the boundary Π is taken from a set A, exactly as definition of $\mathrm{ATS_p}(A)$ was the generalization of the class ATS.

To proceed with the arguments of this section we now need to introduce a lemma showing some useful boundaries over multivariate polynomials.

Lemma 38 *Let $p : \mathbb{R}^k \to \mathbb{R}$ be a multivariate polynomial of degree n, with $p(x) = \Sigma_{|\alpha| \leq n} a_\alpha x^\alpha$, where $\alpha = (\alpha_1, \ldots, \alpha_k) \in \mathbb{N}^k$, $|\alpha| = \alpha_1 + \ldots + \alpha_k$, and $x^\alpha = x_1^{\alpha_1} \cdots x_k^{\alpha_k}$. Then there are polynomials $P_1, \ldots, P_k$ of degree at most n, such that*

$$\|p(x)\| \leq P_1(|x_1|) \cdots P_k(|x_k|).$$

Proof. We have

$$\|p(x)\| \leq \sum_{|\alpha| \leq n} |a_\alpha| \, \|x_1^{\alpha_1} \cdots x_k^{\alpha_k}\| \leq \sum_{|\alpha| \leq n} |a_\alpha| \, |x_1|^{\alpha_1} \cdots |x_k|^{\alpha_k}$$

Now notice that $|x_i|^{|\alpha|} \leq \max(1, |x_i|)^n$ since: (i) if $|x_i| \leq 1$, then $|x_i|^{|\alpha|} \leq 1 = \max(1, |x_i|)^n$ and (ii) if $|x_i| \geq 1$, then $|x_i|^{|\alpha|} \leq |x_i|^n = \max(1, |x_i|)^n$. This implies that

$$\begin{aligned}\|p(x)\| &\leq \sum_{|\alpha| \leq n} |a_\alpha| \max(1, |x_1|)^n \cdots \max(1, |x_k|)^n \\ &\leq \left(\sum p\right) \max(1, \|x\|)^n \\ &\leq \left(\sum p\right) (1 + |x_1|)^n \cdots (1 + |x_k|)^n\end{aligned}$$

where $\sum p = \sum_{|\alpha| \leq n} |a_\alpha|$. This concludes the proof of the lemma. ■

Let us now assume that the set A considered in this section satisfies the conditions listed above in Definition 33. Then we obtain an equivalence between the two analog classes in the form of the following theorem:

Theorem 39 *If A is a class of functions which satisfies the conditions of Definition 33, then* $\mathrm{ATS_p}(A) = \mathrm{AL}(A)$.

Proof. The proof of this theorem follows essentially the structure of the proof of Theorem 18 in [BGP16a]. Let us first suppose that $f \in \mathrm{ATS_p}(A)$, with $f : \mathbb{R}^n \to \mathbb{R}^m$. We want to show that $f \in \mathrm{AL}(A)$. Since $f \in \mathrm{ATS_p}(A)$, there exist a polynomial p, functions $\Upsilon_1, \Pi_1 \in A$, and polynomials Π_2, Υ_2 satisfying the conditions of Definition 4 when Π and Υ are given by $\Pi(\|x\|, \mu) = \Pi_1(\|x\|)\Pi_2(\mu)$ and $\Upsilon(\|x\|, t) = \Upsilon_1(\|x\|)\Upsilon_2(t)$, respectively. We also assume, without loss of generality, that the polynomials Π_2, Υ_2 are increasing and take non-negative values when they have non-negative arguments. Now consider the system

$$\begin{cases} y(0) = q(x) \\ z(0) = 0 \end{cases} \quad \begin{cases} y'(t) = p(y(t)) \\ z'(t) = 1 \end{cases}$$

We now show that this system computes f according to Definition 36. The first and third condition of Definition 36 are trivially satisfied for the above system. We now have to show that the second condition holds for some function Π^*, where $\Pi^*(\|x\|, \mu) = \Pi_1^*(\|x\|)\Pi_2^*(\mu)$ and $\Pi_1^* \in A$ and Π_2^* is a polynomial.

We begin by noting that if $t < \Pi(\|x\|, \mu) = \Pi_1(\|x\|)\Pi_2(\mu)$, then by the third condition of Definition 4

$$\|y(t)\| \leq \Upsilon(\|x\|, t) = \Upsilon_1(\|x\|)\Upsilon_2(t) \leq \Upsilon_1(\|x\|)\Upsilon_2(\Pi_1(\|x\|)\Pi_2(\mu)). \qquad (6)$$

By Lemma 38 applied to Υ_2, we obtain two polynomials $\bar{P}_1, \bar{P}_2$ such that $0 < \Upsilon_2(\Pi_1(\|x\|)\Pi_2(\mu)) \leq \left(\bar{P}_1 \circ \Pi_1(\|x\|)\right)\left(\bar{P}_2 \circ \Pi_2(\mu)\right)$. This and (6) imply that there is some function $\Psi_1 \in A$ and some polynomial Ψ_2 such that $\|y(t)\| \leq \Psi_1(\|x\|)\Psi_2(\mu)$. Using again lemma 38, we can obtain polynomials $P_1, \ldots, P_k, Q_1, Q_2$ such that

$$\begin{aligned}
\|y'(t)\| &= \|p(y(t))\| \\
&\leq P_1(|y_1(t)|) \ldots P_k(|y_k(t)|) \\
&\leq P_1(\|y(t)\|) \ldots P_k(\|y(t)\|) \\
&\leq P_1(\Psi_1(\|x\|)\Psi_2(\mu)) \ldots P_k(\Psi_1(\|x\|)\Psi_2(\mu)) \\
&\leq (Q_1 \circ \Psi_1(\|x\|))\,(Q_2 \circ \Psi_2(\mu))
\end{aligned}$$

for some polynomials $P_1, \ldots, P_k, Q_1, Q_2$ which we assume without loss of generality to be increasing. We now have (recall that $t < \Pi_1(\|x\|)\Pi_2(\mu)$ by assumption)

$$\begin{aligned}
\mathrm{len}_y(0,t) &= \int_0^t \|y'(u)\|\,du \\
&\leq \int_0^t (Q_1 \circ \Psi_1(\|x\|))\,(Q_2 \circ \Psi_2(\mu))\,du \\
&= (Q_1 \circ \Psi_1(\|x\|))\,(Q_2 \circ \Psi_2(\mu))\,t \\
&< (Q_1 \circ \Psi_1(\|x\|))\,(Q_2 \circ \Psi_2(\mu))\,\Pi_1(\|x\|)\Pi_2(\mu) \\
&\leq \Lambda_1(\|x\|)\Lambda_2(\mu)
\end{aligned}$$

where $\Lambda_1 \in A$ and Λ_2 is a polynomial. We have thus concluded that if $t < \Pi_1(\|x\|)\Pi_2(\mu)$, then $\mathrm{len}_y(0,t) < \Lambda_1(\|x\|)\Lambda_2(\mu)$. This implies that if $\mathrm{len}_y(0,t) \geq \Lambda_1(\|x\|)\Lambda_2(\mu)$, then it must be $t \geq \Pi_1(\|x\|)\Pi_2(\mu)$ which, by assumption, implies that $\|(y_1(t),\ldots,y_m(t)) - f(x)\| \leq e^{-\mu}$, which proves the remaining second condition of Definition 36, thus proving that $f \in \mathrm{AL}(A)$.

For the reverse implication, let us suppose that $f \in \mathrm{AL}(A)$. We now want to show that $f \in \mathrm{ATS_p}(A)$. We first note that $f \in \mathrm{AL}(A)$ does not necessarily imply that $f \in \mathrm{ATS_p}(A)$. For example, consider the following IVP

$$\begin{cases} y_1' = -y_1 y_2 \\ y_2' = y_2 \end{cases} \qquad \begin{cases} y_1(0) = e^{-1} \\ y_2(0) = 1. \end{cases} \tag{7}$$

It is not difficult to see that its solution is $y_2(t) = e^t$ and $y_1(t) = e^{-e^t}$. Hence $y_1(t)$ converges to the value 0. Note that

$$\mathrm{len}_y(0,t) = \int_0^t \|y'(u)\|\, du = \int_0^t e^u du = e^t - 1.$$

Hence, if $t^* \geq 0$ is such that $\mathrm{len}_y(0,t^*) \geq \mu$, we have $e^{t^*} - 1 \geq \mu$ which implies that $t^* \geq \ln(1+\mu)$. When this happens we have that

$$|0 - y_1(t)| \leq y_1(\ln(1+\mu)) = e^{-e^{\ln(1+\mu)}} \leq e^{-\mu}.$$

In other words, the IVP (7) shows that the constant function 0 belongs to ALP. On the other hand, the IVP (7) cannot be used to show that $0 \in \mathrm{ATSP}$, since when $t \geq \Pi(\|x\|,\mu)$ for some polynomial Π, we will have $\|y(t)\| = e^t$ which grows more quickly than any polynomial on t and therefore condition 3 of Definition 4 cannot be satisfied for any polynomial Υ. The solution to this problem is to rescale the time variable of the system (7) so that the condition $\mathrm{len}_y(0,t^*) \geq P(\mu)$, where P is a polynomial, can only achieved for a time t^* greater than a polynomial on μ, instead of a time which is subpolynomial (logarithmic, in the case of the preceding example).

Let us thus assume that $f \in \mathrm{AL}(A)$. This implies that there is some function $q \in \mathrm{GPVAL}$, a polynomial p, a function $\Pi_1 \in A$, and a polynomial Π_2 satisfying the conditions of Definition 36 where Π is given by $\Pi(\|x\|,\mu) = \Pi_1(\|x\|)\Pi_2(\mu)$. Suppose that y is the solution of $y' = p(y)$, $y(0) = q(x)$, with $y(t) \in \mathbb{R}^k$. We now wish to define a time rescaling τ such that if $\hat{y}(u) = y(\tau(u))$, we get that the condition $\mathrm{len}_{\hat{y}}(0,u) \geq P(\mu)$, where P is a polynomial, only holds when $u \geq Q(\mu)$ for some polynomial Q. We begin by noting that there is (see lemma 44 of [BGP17a]) a GPVAL function norm : $\mathbb{R}^k \to \mathbb{R}$ with the property that $\|x\| \leq \mathrm{norm}(x) \leq \|x\| + 1/2$. Furthermore, since $p \in \mathrm{GPVAL}$ and GPVAL is closed under composition, we conclude that $g = \mathrm{norm} \circ p \in \mathrm{GPVAL}$, with $g : \mathbb{R}^k \to \mathbb{R}$. By definition of GPVAL, we conclude that there is a $d \times k$ matrix r consisting of polynomials with coefficients in $\mathbb{R}_P$, with $d \geq 1$ and $x_0 \in \mathbb{R}_G^k, z_0 \in \mathbb{R}_G^d$ such that $g(x) = z_1(x)$, where $z = (z_1,\ldots,z_d)$ is the solution of

$$J_z(x) = r(z),\ z(x_0) = z_0. \tag{8}$$

Now take $\psi : \mathbb{R} \to \mathbb{R}$ as $\psi(t) = g \circ y(t)$. Note that due to condition 3 of Definition 36.

$$\psi(t) = g \circ y(t) = \text{norm}(p(y(t))) = \text{norm}(y'(t)) \geq \|y'(t)\| \geq 1.$$

Next we define the function $\hat{\psi} : \mathbb{R} \to \mathbb{R}$ as $\hat{\psi}(u) = \int_0^u \psi(t)dt$. Note that $\psi(t) \geq 1$ implies that $\hat{\psi}(u)$ is strictly increasing and thus admits an inverse function $\hat{\psi}^{-1}$. Furthermore, since $\psi(t) \geq 1$, we have

$$\hat{\psi}(u) = \int_0^u \psi(t)dt \geq u. \tag{9}$$

Let us now take $\hat{y}(u) = y \circ \hat{\psi}^{-1}(u)$, $\hat{z}(u) = z \circ \hat{y}(u)$, and $\hat{w}(u) = (\hat{\psi}^{-1}(u))'$ where z is the solution of (8). We note that, by using the formula for obtaining the derivative of an inverse function, we obtain

$$\hat{w}(u) = (\hat{\psi}^{-1}(u))' = \frac{1}{\hat{\psi}'(\hat{\psi}^{-1}(u))} = \frac{1}{\psi(\hat{\psi}^{-1}(u))} \tag{10}$$

Using (8), we conclude that

$$\begin{aligned}
\hat{y}'(u) &= y'(\hat{\psi}^{-1}(u))(\hat{\psi}^{-1}(u))' = p(y(\hat{\psi}^{-1}(u)))\hat{w}(u) = p(\hat{y}(u))\hat{w}(u) \\
\hat{z}'(u) &= J_z(\hat{y}(u))\hat{y}'(u) = r(z(\hat{y}(u)))p(\hat{y}(u))\hat{w}(u) = r(\hat{z}(u))p(\hat{y}(u))\hat{w}(u).
\end{aligned}$$

Furthermore, from (10) we get

$$\begin{aligned}
\hat{w}'(u) &= \left(\frac{1}{\psi(\hat{\psi}^{-1}(u))}\right)' \\
&= -\frac{\psi'(\hat{\psi}^{-1}(u)) \cdot (\hat{\psi}^{-1}(u))'}{\left(\psi(\hat{\psi}^{-1}(u))\right)^2} \\
&= -\psi'(\hat{\psi}^{-1}(u))\frac{1}{\left(\psi(\hat{\psi}^{-1}(u))\right)^3} \\
&= -\hat{w}^3(u)\psi'(\hat{\psi}^{-1}(u)).
\end{aligned} \tag{11}$$

Now remark that $(z \circ y)'(\zeta) = J_z(y(\zeta))y'(\zeta) = r(z(y(\zeta)))p(y(\zeta))$. Taking $\zeta = \hat{\psi}^{-1}(u)$, we get

$$\begin{aligned}
(z \circ y)'(\hat{\psi}^{-1}(u)) &= r(z(y(\hat{\psi}^{-1}(u))))p(y(\hat{\psi}^{-1}(u))) \\
&= r(\hat{z}(u))p(\hat{y}(u)).
\end{aligned}$$

In particular, if $r_{(1)}$ denotes the first row of r and since $\psi(t) = g \circ y(t)$ and g is the first component of the solution of z in (8), we get that $\psi'(\hat{\psi}^{-1}(u)) = r_{(1)}(\hat{z}(u))p(\hat{y}(u))$. This last equality and (11) yield

$$\begin{aligned}
\hat{y}'(u) &= p(\hat{y}(u))\hat{w}(u) \\
\hat{z}'(u) &= r(\hat{z}(u))p(\hat{y}(u))\hat{w}(u) \\
\hat{w}'(u) &= -\hat{w}^3(u)r_{(1)}(\hat{z}(u))p(\hat{y}(u)).
\end{aligned}$$

This shows that $\hat{y}, \hat{z}, \hat{w} \in \text{GVAL}$. We now prove that this ODE, with the initial conditions

$$\begin{aligned}
\hat{y}(0) &= y \circ \hat{\psi}^{-1}(0) = y(0) = q(x) \\
\hat{z}(0) &= z \circ \hat{y}(0) = z(q(x)) \\
\hat{w}(u) &= \frac{1}{\psi(\hat{\psi}^{-1}(0))} = \frac{1}{\psi(0)} = \frac{1}{g(y(0))} = \frac{1}{g(q(x))}
\end{aligned}$$

computes the function f according to Definition 4. First we remark that, due to the property that $\|x\| \leq \text{norm}(x) \leq \|x\| + 1/2$, we get

$$\begin{aligned}
\|y'(t)\| = \|p(y(t))\| &\leq \text{norm} \circ p \circ y(t) = \psi(t) \\
\psi(t) = \text{norm} \circ p \circ y(t) &\leq \|p(y(t))\| + 1/2 = \|y'(t)\| + 1/2.
\end{aligned}$$

Therefore

$$\text{len}_y(0,t) = \int_0^t \|y'(u)\|\, du \leq \int_0^t \psi(t) = \hat{\psi}(t) \tag{12}$$

$$\hat{\psi}(t) \leq \int_0^t \|y'(u)\| + 1/2\, du \leq \text{len}_y(0,t) + \frac{t}{2}. \tag{13}$$

Now note that $\text{len}_{\hat{y}}(0,u) = \int_0^u \|\hat{y}'(\zeta)\|\, d\zeta = \int_0^u \|p(\hat{y}(\zeta))\hat{w}(\zeta)\|\, d\zeta$ and by using the variable change $t = \hat{\psi}^{-1}(\zeta)$, we get $\zeta = \hat{\psi}(t)$, $d\zeta = \hat{\psi}'(t)dt$, and

$$\begin{aligned}
\text{len}_{\hat{y}}(0,u) &= \int_0^{\hat{\psi}^{-1}(u)} \left\| p(\hat{y}(\hat{\psi}(t)))\hat{w}(\hat{\psi}(t)) \right\| \hat{\psi}'(t)dt \\
&= \int_0^{\hat{\psi}^{-1}(u)} \left\| p(y(t)) \left(\hat{\psi}^{-1}\right)' (\hat{\psi}(t)) \right\| \hat{\psi}'(t)dt \\
&= \int_0^{\hat{\psi}^{-1}(u)} \|p(y(t))\| \frac{1}{\hat{\psi}'(\hat{\psi}^{-1}(\hat{\psi}(t)))} \hat{\psi}'(t)dt \\
&= \int_0^{\hat{\psi}^{-1}(u)} \|p(y(t))\|\, dt \\
&= \text{len}_y(0, \hat{\psi}^{-1}(u)) \qquad (14) \\
&\leq \hat{\psi}(\hat{\psi}^{-1}(u)) \quad \text{(use (12))} \\
&= u.
\end{aligned}$$

Using the last inequality and noting that $\|\hat{y}(u) - \hat{y}(0)\| = \left\|\int_0^u \hat{y}'(t)dt\right\| \leq \int_0^u \|\hat{y}'(t)\|\, dt = \text{len}_{\hat{y}}(0,u)$, we get that

$$\begin{aligned}
\|\hat{y}(u)\| &\leq \|\hat{y}(0)\| + \|\hat{y}(u) - \hat{y}(0)\| \\
&\leq \|\hat{y}(0)\| + \text{len}_{\hat{y}}(0,u) \\
&\leq \|\hat{y}(0)\| + u \\
&\leq q(x) + u
\end{aligned}$$

and thus $\hat{y}(u)$ is bounded by a polynomial on x and u. Using this result together with the fact that the solution z of (8) belongs to GPVAL, we conclude that $\hat{z}(u) = z \circ \hat{y}(u)$ is also bounded by a polynomial on x and u. Concerning the case of $\hat{w}$, we note that due to (10) we have (recall that $\psi(t) \geq 1$ for any t)

$$\|\hat{w}(u)\| = \left\| \frac{1}{\psi(\hat{\psi}^{-1}(u))} \right\| \leq 1.$$

Therefore $\|\hat{w}(u)\|$ is also polynomially bounded. We have thus shown that condition 3 of Definition 4 holds. We now only have to show that condition 2 holds. Let $t \geq 2\Pi(\|x\|, \mu) = 2\Pi_1(\|x\|)\Pi_2(\mu)$. We note that (9) yields us

$$\hat{\psi}^{-1}(u) \leq u$$

This last inequality, together with (14) and (13), yields $\mathrm{len}_y(0, u) \geq \hat{\psi}(u) - u/2$ and

$$\begin{aligned}
\mathrm{len}_{\hat{y}}(0,t) &= \mathrm{len}_y(0, \hat{\psi}^{-1}(t)) \\
&\geq \left(\hat{\psi}(\hat{\psi}^{-1}(t)) - \frac{\hat{\psi}^{-1}(t)}{2} \right) \\
&\geq t - \frac{t}{2} \\
&= \frac{t}{2} \\
&\geq \Pi(\|x\|, \mu)
\end{aligned}$$

which implies condition 2 of Definition 4 due to condition 2 of Definition 36. This shows that $f \in \mathrm{ATS}_\mathrm{p}(A)$. ■

7 Application to the Grzegorczyk hierarchy

We start this section by briefly recalling the definition of the Grzegorczyk hierarchy. The Grzegorczyk hierarchy, originally proposed by Andrzej Gregorczyk in 1953 in [Grz53], is a hierarchy of classes of functions from the naturals to the naturals, defined recursively. For our specific purpose, the first two levels of the hierarchy are not relevant, since they only include trivial functions such as addition and multiplication, which are obviously computable in polynomial time. The third level, which we indicate with the notation ξ^3 coincides with the set of all elementary functions. The definition of each level of the hierarchy for $n \geq 3$ involves the generator functions, G_n, whose definition is also recursive. Let $G_2 : \mathbb{N} \to \mathbb{N}$ be the exponential function $G_2(x) = 2^x$ and for $n \geq 2$ define: $G_{n+1}(x) = G_n^{[x]}(1)$, where the notation $G_n^{[x]}(1)$ stands for the iteration of the function G_n for x times evaluated on the value 1, i.e. $G_n^{[0]}(x) = x$ and $G_n^{[k+1]}(x) = G_n(G_n^{[k]}(x))$. Then, formally [Odi99, Definition VIII.8.12]:

Definition 40 (Grzegorczyk Hierarchy) *For $n \geq 3$ the nth level of the hierarchy, ξ^n, is the smallest class of functions containing the zero function, the successor function, the projections, cut-off subtraction and G_{n-1} which is closed under composition, bounded sum and bounded product.*

It can be shown that each level is properly included in the next one, $\xi^{n-1} \subsetneq \xi^n \subsetneq \xi^{n+1}$, and that all together they constitute a hierarchy that satisfies $\bigcup_{n \in \mathbb{N}} \xi^n = PR$, where PR is the set of primitive recursive functions. This is the reason way our analog characterization of the hierarchy naturally implies a characterization of the class of primitive recursive functions as well. Moreover, for any function $f \in \xi^n$, there is some $m \in \mathbb{N}$ such that $f(x) \leq G_{n-1}^{[m]}(x)$ (see [Odi99, Theorem VIII.7.8]. Although this theorem is proved for the case of the elementary functions ξ^3, its proof generalizes to ξ^n for any $n \geq 3$). We also note [Odi99, Theorem VIII.8.14] that f belongs to ξ^n iff it is computable in time belonging to ξ^n. Combining the last two facts we conclude that f belongs to ξ^n iff it is computable in time bounded by $G_{n-1}^{[m]}(x)$ for some $m \in \mathbb{N}$. We will use this last characterization to characterize ξ^n in the context of Theorem 35 to avoid having to deal with bounded sums and products.

7.1 Analog characterization of each level

In this section our objective is to use ODEs to characterize the classes ξ^n, $n \geq 3$, defining the Grzegorczyk hierarchy, with the use of Theorem 35.

However, a problem arises if one wants to use Theorem 35 to characterize ξ^n for all $n \geq 3$. When characterizing the classes FP and FEXPTIME, we had to rely on the classes ATSP and ATSE which are defined using the class ATS with polynomial and exponential/polynomial bounds, respectively, which are *defined over* $\mathbb{R}_+ = [0, +\infty[$. In the polynomial and exponential cases, this was not problematic since polynomial and exponential functions over $\mathbb{N}$ admit a trivial extension to $\mathbb{R}$. This is not the case for the time bounds $G_{n-1}^{[m]}$ for the Grzegorczyk hierarchy, which are defined using iteration and hence do not admit a trivial extension to $\mathbb{R}$. To solve this problem, in this section we show how we can obtain functions in GVAL which essentially work as an extension of $G_{n-1}^{[m]}$. To achieve this purpose, we have to be able to iterate a function with ODEs over integers, since the definition of $G_{n-1}^{[m]}$ relies on the use of iterations, as mentioned earlier.

With this objective in mind, we now present a construction to iterate functions with ODEs which is based on the ODE

$$y' = c(b - y)^3 \phi(t). \tag{15}$$

that was already studied in [Bra95], [CMC00], [GCB08]. This ODE is called the targeting equation, for reasons that will become evident in a moment. In (15) we assume that $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ is such that $\int_0^{\frac{1}{2}} \phi(t) dt > 0$ (for example one could consider $\phi(t)=\sin(2\pi t)$ as $\int_0^{\frac{1}{2}}\sin(2\pi t)dt=1>0$), $b\in\mathbb{R}$ is the *target* and

$$c \geq \frac{1}{2\gamma^2\int_0^{1/2}\phi(t)dt} \tag{16}$$

where $\gamma>0$ is the *targeting error*. Let us consider separately two cases:

- Let us first assume that $y(0)\neq b$. Due to (16), we conclude that

  $$\gamma^2 \geq \frac{1}{2c\int_0^{1/2}\phi(t)dt}.$$

  Since (15) is a separable equation, we get that

  $$\frac{1}{(b-y(1/2))^2}-\frac{1}{(b-y(0))^2}=2c\int_0^{1/2}\phi(t)dt \quad\Longrightarrow$$
  $$\frac{1}{2c\int_0^{1/2}\phi(t)dt}>(b-y(1/2))^2.$$

  This implies that $\gamma>|b-y(1/2)|$, i.e. $y(1/2)$ is γ-close to the value b (the target).

- If $y(0)=b$, then it is trivial to notice that $y(t)=b$ is the solution of (15) and thus the condition $\gamma>|b-y(1/2)|$ is true.

One can also consider the more general targeting ODE

$$z'=c(\bar{b}(t)-z)^3\phi(t), \tag{17}$$

where $\bar{b}:\mathbb{R}\to\mathbb{R}$ is a function with the property that $\left|\bar{b}(t)-b\right|\leq\rho$ for all $t\in[0,1/2]$, with $b\in\mathbb{R}$ (the target), $\rho\geq 0$, c satisfies (16) where $\gamma>0$ is the targeting error. A similar analysis (see [GCB08] for more details) would yield that $|z(1/2)-b|<\rho+\gamma$, independently of the initial condition $y(0)\in\mathbb{R}$.

We also note (to our knowledge, this was not mentioned in the literature before) that c might not be a fixed value, but instead some function $c:\mathbb{R}\to\mathbb{R}$. In this case, if $c(t)\geq\left(2\gamma^2\int_0^{1/2}\phi(t)dt\right)^{-1}$ for all $t\in[0,1/2]$, then $\gamma>|b-y(1/2)|$. This follows from known facts of the ODE theory (see e.g. [HW95, pp. 511–514]) and from the conclusions that we obtained for the previous targeting ODE. A variable value for c has the advantage of allowing a *dynamic targeting error*, i.e. γ does not need to be fixed a priori and can be dynamically changed by updating the value of c. However, for the purpose of this paper, it is enough to assume that c is a fixed constant.

We will now show how the targeting ODE (17) can be used to obtain a function $f:\mathbb{R}^3\to\mathbb{R}^2\in\mathrm{GVAL}$ such that the solution of the IVP $y'=f(t,y)$, $y(0)=(1,1)$ satisfies $|y_1(t)-G_2(k)|\leq 1/4$ (i.e. $\left|y_1(t)-2^k\right|\leq 1/4$) for all $t\in[k,k+1/2]$ and $k\in\mathbb{N}$, where $y=(y_1,y_2)$. But before presenting this result,

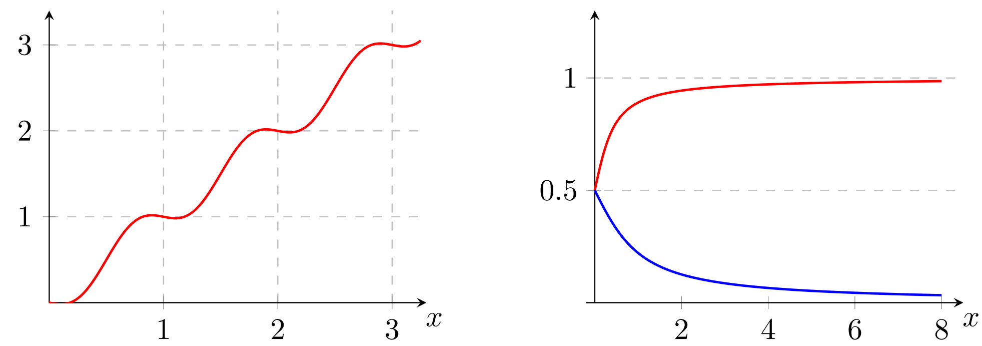

Figure 3: The graph of the function l_2 of Lemma 41 is depicted on the right side (the red graph assumes $x = 1.2$ while the blue graph assumes $x = 0.2$). The left graph depicts the function σ from Lemma 42.

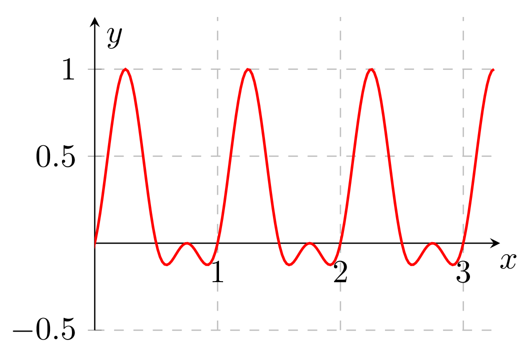

Figure 4: The graph of the function s defined on equation (18).

we have to present several auxiliary results. We begin by introducing the error-correcting function l_2 which was presented in [GCB08, Lemma 9]. Its graph is depicted in Fig. 3.

Lemma 41 *Let $l_2 : \mathbb{R}^2 \to [0,1]$ be given by $l_2(x,y) = \frac{1}{\pi}\arctan(4x(y-1/2)) + \frac{1}{2}$. Suppose also that $a \in \{0,1\}$. Then, for any $\bar{a}, x \in \mathbb{R}$ satisfying $|a - \bar{a}| \leq 1/4$ and $x > 0$, we obtain $|a - l_2(\bar{a}, x)| < 1/x$.*

We can see the function l_2 as a function which reduces the error around a 1/4-neighborhood of the integers 0 and 1 by an amount specified by $1/x$, where $x > 0$ is the first argument of l_2.

Proceeding again as in [GCB08], let us take the function s defined by

$$s(t) = \frac{1}{2}\left(\sin^2(2\pi t) + \sin(2\pi t)\right). \tag{18}$$

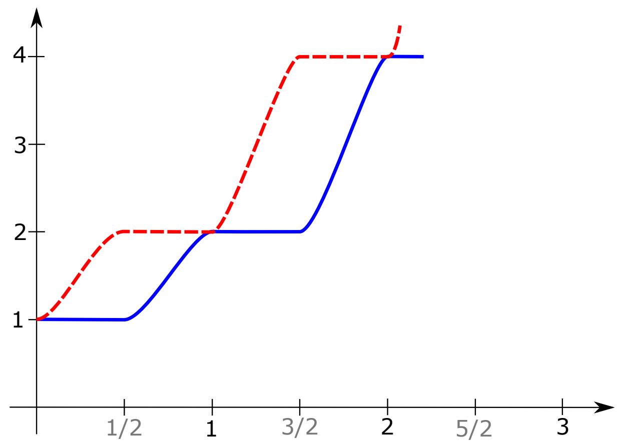

Figure 5: Iterating a function with an ODE. Here the dashed (red) and blue (solid) graphs represent the ideal behavior of y_1 and y_2, respectively, in (20).

The graph of this function is depicted in Fig. 4. A simple analysis shows that this function takes values between 0 and 1 in $[0, 1/2]$ and, in particular, between 3/4 and 1 when $x \in [0.16, 0.34]$, and between $-\frac{1}{8}$ and 0 on the time interval $[1/2, 1]$. Therefore, if we take the GVAL function $\phi : \mathbb{R}^2 \to [0, 1]$, defined by

$$\phi(t, y) = l_2(s(t), y), \tag{19}$$

we conclude that $\int_0^{1/2} \phi(t, y)dt > 3/4 \times (0.34 - 0.16) = 0.135 > 0$ (assuming that $y \geq 4$) and $|\phi(t, y)| < 1/y$ for all $t \in [1/2, 1]$ (i.e. y allows us to provide an upper bound $1/y$ for $\phi(t)$ in the time interval $[1/2, 1]$). Since ϕ has period 1 on t, we conclude that $\int_k^{k+1/2} \phi(t, y)dt > 0.135 > 0$ if $y \geq 4$ and $|\phi(t, y)| < 1/y$ for all $t \in [k + 1/2, k + 1]$, where $k \in \mathbb{N}$ is arbitrary.

The following function σ is also from [GCB08, Proposition 5].

Lemma 42 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a function defined by $\sigma(x) = x - 0.2\sin(2\pi x)$. Then given $\varepsilon \in [0, 1/2)$, there is some contracting factor $\lambda_\varepsilon \in (0, 1)$ such that, $\forall \delta \in [-\varepsilon, \varepsilon]$, $|\sigma(n + \delta) - n| < \lambda_\varepsilon \delta$ where $n \in \mathbb{Z}$ is arbitrary.*

The function σ behaves as a uniform contraction in a neighborhood of the integers $\mathbb{Z}$. Contrarily to the function l_2 which works only for the integer values 0 and 1, σ has the advantage of being a contraction around every integer. However, the rate of contraction is fixed for σ while it can be dynamically prescribed for l_2. We remark, as noted in [GCB08], that we can take $\lambda_{1/4} = 0.4\pi - 1 \approx 0.2566371$.

Let us now show how we can obtain a function $f : \mathbb{R}^3 \to \mathbb{R}^2 \in$ GVAL such that the solution of the IVP $y' = f(t, y)$, $y(0) = (1, 1)$ satisfies $\left|y_1(t) - 2^k\right| \leq 1/4$

for all $t \in [k, k+1/2]$ and $k \in \mathbb{N}$, where $y = (y_1, y_2)$. This will be done by iterating the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(x) = 2x$ via the ODE

$$\begin{cases} y_1' = c(2\sigma(y_2) - y_1)^3 \phi_1(t, y_1, y_2) \\ y_2' = c(\sigma(y_1) - y_2)^3 \phi_2(t, y_1, y_2) \end{cases} \tag{20}$$

with initial condition $y_1(0) = y_2(0) = 1$, where

$$\begin{aligned} \phi_1(t, y_1, y_2) &= \phi(t, 16c((2\sigma(y_2) - y_1)^4 + 1) + 4) \\ \phi_2(t, y_1, y_2) &= \phi(-t, 16c((\sigma(y_1) - y_2)^4 + 1) + 4) \end{aligned} \tag{21}$$

and $c = 1000$. Its (ideal) behavior is depicted in Fig. 5. We note that ϕ_1 is essentially the function ϕ from (19), where the main change is made on the upper bound for ϕ on the time intervals $[k+1/2, k+1]$, where $k \in \mathbb{Z}$. In this case, noting that $\left|x^3\right| \leq x^4 + 1$ for all $x \in \mathbb{R}$, we have that for any $t \in [k+1/2, k+1]$ we have

$$\begin{aligned} |\phi_1(t, y_1, y_2)| &\leq \frac{1}{16c((2\sigma(y_2) - y_1)^4 + 1) + 4} \\ &< \frac{1}{16c(2\sigma(y_2) - y_1)^3} \end{aligned}$$

which implies that $|y_1'(t)| < 1/16$ whenever $t \in [k+1/2, k+1]$ for some $k \in \mathbb{Z}$. Furthermore, since $16c((2\sigma(y_2) - y_1)^4 + 1) + 4 \geq 4$, we conclude that $\int_k^{k+1/2} \phi_1(t, y_1, y_2)dt > 0.135 > 0$ and therefore that the first equation of (20) defines a targeting equation on the time interval $[0, 1/2]$ (or, more generally, on time intervals with the format $[k, k+1/2]$ where $k \in \mathbb{Z}$) with targeting error $1/16$, since according to (16)

$$c = 1000 > \frac{1}{2\left(\frac{1}{16}\right)^2 0.135} > \frac{1}{2\left(\frac{1}{16}\right)^2 \int_0^{1/2} \phi_1(t, y_1, y_2)dt}.$$

Using a similar argument we conclude that $|y_2'(t)| < 1/16$ whenever $t \in [k, k+1/2]$ for some $k \in \mathbb{Z}$ and that the second equation of (20) defines a targeting equation on the time interval $[1/2, 1]$ (or, more generally, on time intervals with the format $[k+1/2, k+1]$ where $k \in \mathbb{Z}$) with targeting error $1/16$.

Let us now analyze in more detail the ODE (20). When $t \in [0, 1/2]$, we have that $|y_2'(t)| \leq 1/16$, which further implies that $|y_2(t) - 1| \leq 1/32$ when $t \in [0, 1/2]$, since

$$\begin{aligned} |y_2(t) - y_2(0)| &= \left| \int_0^t y_2'(t)dt \right| \\ &\leq \int_0^t |y_2'(t)|\, dt \\ &\leq \left(\frac{1}{2} - 0\right) \frac{1}{16} = \frac{1}{32}. \end{aligned}$$

Now notice that, since $|y_2(t) - 1| \leq 1/32$, then

$$\begin{aligned} |2\sigma(y_2(t)) - 2 \cdot 1| &\leq 2\,|y_2(t) - 1| \\ &\leq 1/16. \end{aligned}$$

Hence, since the equation for y_1 in (20) defines a targeting equation with targeting error $\gamma = 1/16$, we get that $\left|y_1(1/2) - 2^1\right| < 1/16 + 1/16 = 1/8$. Now, on the next half-unit interval, we have that $|y_1'(t)| \leq 1/16$ which implies that $\left|y_1(t) - 2^1\right| < 1/8 + 1/32 = 5/32$ for all $t \in [1/2, 1]$. This implies that $\left|\sigma(y_1(t)) - 2^1\right| \leq \lambda_{1/4}\left|y_1(t) - 2^1\right| < 1/24$ for all $t \in [1/2, 1]$. Hence y_2 will become a targeting equation in the time interval $[1/2, 1]$ with targeting error $\gamma = 1/16$ and we will have $\left|y_2(1) - 2^1\right| < 1/24 + 1/16 < 1/8$. Now the procedure repeats itself in subsequent intervals. For example, when $t \in [1, 3/2]$, we will have that $|y_2'(t)| \leq 1/16$, which further implies that $\left|y_2(t) - 2^1\right| < 1/8 + 1/32 < 5/32$ when $t \in [1, 3/2]$. By a similar argument as in the previous case, we conclude that

$$\begin{aligned} \left|2\sigma(y_2(t)) - 2 \cdot 2^1\right| &\leq 2\lambda_{1/4}\left|y_2(t) - 2^1\right| \\ &\leq 2\lambda_{1/4}5/32 \\ &\leq 1/12. \end{aligned}$$

and since the equation for y_1 in (20) defines a targeting equation with targeting error $\gamma = 1/16$, we get that $\left|y_1(3/2) - 2^2\right| < 1/12 + 1/16 = 7/48$. On the next half-unit interval, we have that $|y_1'(t)| \leq 1/16$ which implies that $\left|y_1(t) - 2^2\right| < 7/48 + 1/32 = 17/96$ for all $t \in [3/2, 2]$. This implies that $\left|\sigma(y_1(t)) - 2^2\right| < 1/22$ for all $t \in [3/2, 2]$. Hence y_2 will become a targeting equation in the time interval $[3/2, 2]$ with targeting error $\gamma = 1/16$ and we will have $\left|y_2(2) - 2^2\right| < 1/22 + 1/16 < 1/8$. The procedure repeats itself on subsequent intervals and we conclude that

$$|y_2(t) - G_2(k)| \leq 1/4 \text{ for all } t \in [k, k + 1/2] \text{ and } k \in \mathbb{N}.$$

This result can be generalized as shown in the following theorem.

Theorem 43 *Given the function $G_n : \mathbb{N} \to \mathbb{N}$, $n = 2, 3, \ldots$, there is an IVP $y' = f_n(t, y)$, $y(0) = y_0$, where $f_n \in$ GVAL and $y_0 \in \mathbb{N}^l$, such that $|y_1(t) - G_n(k)| \leq 1/4$ for all $t \in [k, k + 1/2]$ and $k \in \mathbb{N}\backslash\{0\}$.*

Proof. The proof is done by induction on n. The base case was performed just before the proof of this theorem (note that y_2 in the above argument corresponds to y_1 in the context of this theorem).

Now we go to the induction step. Suppose that $y' = f_n(t, y)$, $y(0) = y_0$ satisfies the conditions of the theorem. Then we want to show that there is a system $y' = f_{n+1}(t, y)$ which simulates G_{n+1} in the sense mentioned in the theorem. Consider the ODE

$$\left\{ \begin{array}{l} z' = f_n(\tau, y)\tau' = cf_n(\tau, y)(w + 1/4 - \tau)^3\theta_1(t, \tau, w) \\ \tau' = c(w + 1/4 - \tau)^3\theta_1(t, \tau, w) \\ w' = c(\sigma(z_1) - w)^3\theta_2(t, z_1, w) \end{array} \right. \tag{22}$$

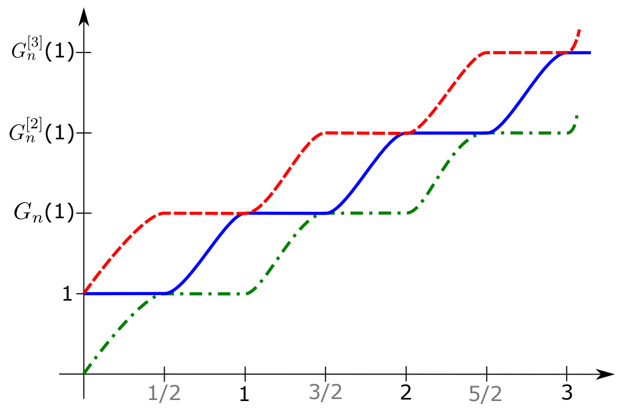

Figure 6: Iterating the function G_n with an ODE. Here the dashed (red), solid (blue), and dotdashed (green) graphs represent the ideal behavior of z_1, w, and τ, respectively, in (22).

with initial condition $z(0) = y_0$, $\tau(0) = 0$, $w(0) = 1$ where

$$\theta_1(t, \tau, w) = \phi(t, 16c((w + 1/4 - \tau)^4 + 1) + 4)$$
$$\theta_2(t, z_1, w) = \phi(-t, 16c((\sigma(z_1) - w)^4 + 1) + 4)$$

and $c = 5000$. Its (idealized) behavior is depicted in Fig. 6. We note that θ_1 and θ_2 behave like ϕ_1 and ϕ_2 of (21), respectively. By using arguments similar to those presented after (21), we conclude that: (i) $|\tau'(t)| \leq 1/16$ whenever $t \in [k + 1/2, k + 1]$ for some $k \in \mathbb{Z}$ and the equation for τ in (22) defines a targeting equation on time intervals with the format $[k, k + 1/2]$ where $k \in \mathbb{Z}$, with targeting error $1/32$ and (ii) $|w'(t)| \leq 1/16$ whenever $t \in [k, k + 1/2]$ for some $k \in \mathbb{Z}$ and that the equation for w in (22) defines a targeting equation on time intervals with the format $[k+1/2, k+1]$ where $k \in \mathbb{Z}$, with targeting error $1/32$.

We note also that, by construction $z(t) = y(\tau(t))$, where y is the solution of $y' = f_n(t, y)$, $y(0) = y_0$. Therefore the value of z only depends on the value of τ. We now have to analyze the behaviour of τ and w. This simulation works again in half-unit time intervals. We break the following analysis in parts, each one related to one half-unit interval.

On the first half-unit interval, where $t \in [0, 1/2]$, we have that w is (almost) constant, since $|w'(t)| \leq 1/16$, which implies that $|w(t) - 1| \leq 1/32$ for all $t \in [0, 1/2]$. Therefore, since the equation governing the behaviour of τ in this time interval $[0, 1/2]$ is a targeting equation, we conclude that $|\tau(1/2) - (1 + 1/4)| \leq 1/32 + 1/32 = 1/16$, since the targeting error is $1/32$. This gives that $\tau(1/2) \in [1 + 3/16, 1 + 5/16]$.

We now proceed with the second time interval. Since $|\tau'(t)| \leq 1/16$ in the time interval $[1/2, 1]$ we conclude that $\tau(t) \in [1 + 1/8, 1 + 3/8] \subseteq [1, 3/2]$ for

$t \in [1/2, 1]$. This implies, by the induction hypothesis, that $z(t) = y(\tau(t))$ is such that $|z_1(t) - G_n(1)| \leq 1/4$ for all $t \in [1/2, 1]$. Note that $|\sigma(z_1(t)) - G_n(1)| \leq \lambda_{1/4} |z_1(t) - G_n(1)| \leq \lambda_{1/4}/4 < 1/15$ for all $t \in [1/2, 1]$. Since the targeting error is $1/32$, we conclude that $|w(1) - G_n(1)| \leq 1/32 + 1/15$.

Now the procedure repeats itself on the time interval $[1, 3/2]$. On this interval, we have $|w'(t)| < 1/16$ and thus $|w(t) - G_n(1)| = |w(t) - G_n(1)| \leq 1/32 + 1/15 + 1/32 = 31/240$ for all $t \in [1, 3/2]$. Therefore, the targeting equation for τ yields that $|\tau(3/2) - (G_n(1) + 1/4)| \leq 31/240 + 1/32 = 77/480$. Since $|\tau'(t)| \leq 1/16$ in the time interval $[3/2, 2]$, we have that $|\tau(t) - (G_n(1) + 1/4)| \leq 77/480 + 1/16 = 107/480 < 1/4$ and therefore $\tau(t) \in [G_n(1), G_n(1) + 1/2]$ for $t \in [3/2, 2]$. This implies, by the induction hypothesis, that $z(t) = y(\tau(t))$ is such that $|z_1(t) - G_n(G_n(1))| = |z_1(t) - G_{n+1}(2)| \leq 1/4$ for all $t \in [3/2, 2]$. Note that $|\sigma(z_1) - G_{n+1}(2)| \leq \lambda_{1/4} |z_1(t) - G_{n+1}(2)| \leq \lambda_{1/4}/4 < 1/15$ for all $t \in [3/2, 2]$. Considering the targeting equation for w in $[3/2, 2]$, we conclude that $|w(2) - G_{n+1}(2)| \leq 1/32 + 1/15$ since the targeting error is $1/32$. Notice again that, for all $t \in [2, 5/2]$, since $|w'(t)| < 1/16$, we have $|w(t) - G_{n+1}(2)| \leq 1/32 + 1/15 + 1/32 = 31/240 < 1/4$.

By repeating this procedure on subsequent intervals and by considering w as the variable y_1 of the statement of the theorem, we conclude the desired result. ■

We now know from the previous theorem that each function G_n admits an GVAL-extension t_n in the sense of Theorem 43. In other words, there is some GVAL function t_n such that $|t_n(t) - G_n(k)| \leq 1/4$ for all $t \in [k, k + 1/2]$ and $k \in \mathbb{N} \backslash \{0\}$.

Definition 44 *For each $n \geq 3$, we define the class $\mathbb{T}^n$ as the smallest class of functions $f : \mathbb{R} \to \mathbb{R}$ containing t_{n-1}, the identity, $\mathbb{R}_G$, and which is closed under sum, difference, product, and composition for $n \geq 3$.*

Note that this definition of $\mathbb{T}^n$ implies that condition 1 and 2 in the list of Definition 33 are immediately satisfied (for condition 2 remark that $\mathbb{T}^n$ includes t_2 which grows exponentially fast, and hence which grows more quickly than any polynomial). Furthermore, it is also not difficult to see that condition 4 is satisfied. Indeed, we have that $t_{n-1}(k) \leq G_{n-1}(k) + 1$ and since ξ^n is closed under composition and arithmetic operations, this shows that any function in $\mathbb{T}^n$ is dominated, over the naturals, by a function in ξ^n (note that all function in ξ^n are time-constructible). Reciprocally, if $f \in \xi^n$, then there is some $m \in \mathbb{N}$ such that $f(k) \leq G_{n-1}^{[m]}(k) \leq t_{n-1}(\ldots t_{n-1}(k) + 1 \ldots) + 1$, where $t_{n-1} + 1$ is composed m times. Since $(t_{n-1} + 1) \circ (t_{n-1} + 1) \circ \ldots \circ (t_{n-1} + 1) \in \mathbb{T}^n$, we conclude that condition 4 is satisfied. Moreover, this also shows the following lemma.

Lemma 45 $\mathrm{FTIME}(\mathbb{T}^n) = \mathrm{FTIME}(\xi^n) = \xi^n$.

It is then left to prove condition 3, which requires that given any function $f \in \xi^n$ there is a function $g \in \mathbb{T}^n$ such that $f \leq g$ and $g \in \mathrm{ATS_p}(\mathbb{T}^n)$, for all $n \geq 3$. We will now show that this condition is satisfied, in a multi-step argument.

From the above argument, we have just showed by means of the above proof of Theorem 43 that each t_n is generable, meaning that $t_n \in \mathrm{GVAL}$ for each $n \geq 3$. We now show that (i) we have $\mathbb{T}^n \subseteq \mathrm{GVAL}(\mathbb{T}^n)$, where

$$\mathrm{GVAL}(\mathbb{T}^n) = \bigcup_{g \in \mathbb{T}^n} \mathrm{GVAL}(g),$$

and (ii) $\mathrm{GVAL}(\mathbb{T}^n) \subseteq \mathrm{ATS}_{\mathrm{p}}(\mathbb{T}^n)$. This will show condition 3 of Definition 33, since any function $f \in \xi^n$ is bounded by a function in $\mathbb{T}^n$, as we have already seen.

To show condition (i), we first present the two following lemmas taken from [BGP17a, Lemma 24, Corollary 26]:

Lemma 46 ([BGP17a], Arithmetic on bounded generable functions) *Let $d, l, n, m \in \mathbb{N}$, $sp, \overline{sp} : \mathbb{R}_+ \to \mathbb{R}_+$, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^n \in \mathrm{GVAL}(sp)$ and $g :\subseteq \mathbb{R}^l \to \mathbb{R}^m \in \mathrm{GVAL}(\overline{sp})$. Then:*

- $f + g, f - g \in \mathrm{GVAL}(sp + \overline{sp})$ *over* $\mathrm{dom}\, f \cap \mathrm{dom}\, g$ *if* $d = l$ *and* $n = m$
- $f \cdot g \in \mathrm{GVAL}(\max(sp, \overline{sp}, sp \cdot \overline{sp}))$ *if* $d = l$ *and* $n = m$
- $f \circ g \in \mathrm{GVAL}(\max(\overline{sp}, sp \circ \overline{sp}))$ *if* $m = d$ *and* $g(\mathrm{dom}\, g) \subseteq \mathrm{dom}\, f$

Lemma 47 ([BGP17a], Generable functions are closed under ODE) *Let $d \in \mathbb{N}$, $J \subseteq \mathbb{R}$ an interval, $sp, \overline{sp} : \mathbb{R}_+ \to \mathbb{R}_+$, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^d \in \mathrm{GVAL}(sp)$, $t_0 \in \mathbb{K} \cap J$ and $y_0 \in \mathbb{R}^d_G \cap \mathrm{dom}\, f$. Assume there exists $y : J \to \mathrm{dom}\, f$ satisfying for all $t \in J$:*

- $y(t_0) = y_0$
- $y'(t) = f(y(t))$
- $\|y(t)\| \leq \overline{sp}(t)$

Then $y \in \mathrm{GVAL}(\max(\overline{sp}, sp \circ \overline{sp}))$ and is unique.

Since lemmas 46 and 47 show closure of $\mathrm{GVAL}(\mathbb{T}^n)$ under the operations used to define $\mathbb{T}^n$, to show (i), i.e. that $\mathbb{T}^n \subseteq \mathrm{GVAL}(\mathbb{T}^n)$ it is enough to show that $t_n \in \mathrm{GVAL}(\mathbb{T}^{n+1})$ (note that the identity and all elements of $\mathbb{R}_G$ belong to $\mathrm{GPVAL} \subseteq \mathrm{GVAL}(\mathbb{T}^n)$).

First, define $\mathbb{T}^2 = \{p \mid p \in \mathit{poly}\}$ as the class of polynomials over $\mathbb{R}_G$. We show that the above proof of Theorem 43 implies that $f_n \in \mathrm{GVAL}(\mathbb{T}^2)$ and $t_n = y_1 \in \mathrm{GVAL}(\mathbb{T}^{n+1})$ for each $n \geq 2$, where f_n and y_1 are defined in the statement of this theorem. This result can be showed by induction on n thanks to the closure by composition of each class $\mathbb{T}^n$.

To show that $f_n \in \mathrm{GVAL}(\mathbb{T}^2)$ and $t_n = y_1 \in \mathrm{GVAL}(\mathbb{T}^{n+1})$ for each $n \geq 2$, we proceed by induction. For the base case $n = 2$, consider f_2 as defined in (20) and note that $f_2 \in \mathrm{GVAL}(\mathbb{T}^2) = \mathrm{GPVAL}$ since all the right-hand terms in the system (20) belong to GPVAL. Moreover, note that, from the arguments which follow (20), the norm of the solution y of that dynamical system is bounded by a function $sp_2 \in \mathbb{T}^3$, where $sp_2(k) = t_2(k+1)$. Therefore, since $\mathbb{T}^2 \subset \mathbb{T}^3$ and each class is closed by composition, applying lemma 47 above yields $y \in \mathrm{GVAL}(\mathbb{T}^3)$ and, in particular, $t_2 = y_1 \in \mathbb{T}^3$. This proves the base case.

Let us now assume that $f_n \in \mathrm{GVAL}(\mathbb{T}^2)$ and that $t_n \in \mathrm{GVAL}(\mathbb{T}^{n+1})$. We now want to show that $f_{n+1} \in \mathrm{GVAL}(\mathbb{T}^2)$ and $t_{n+1} \in \mathrm{GVAL}(\mathbb{T}^{n+2})$. First we note that f_{n+1} is defined as the function used in the right-hand side of the ODE (22) and applied to the variables z, τ, and w. Since f_{n+1} is built using arithmetic operations, elements of $\mathbb{R}_G$, and the functions $f_n, \theta_1, \theta_2, \sigma \in \mathrm{GVAL}(\mathbb{T}^2)$, we conclude by lemma 46 that $f_{n+1} \in \mathrm{GVAL}(\mathbb{T}^2)$. Also, from the analysis done in the proof of Theorem 43, we conclude that the solution $x(t)$ of (22) is bounded by $t_{n+1}(t+1) \in \mathrm{GVAL}(\mathbb{T}^{n+2})$, which shows by lemma 47 that x and hence t_{n+1} belongs to $\mathrm{GVAL}(\mathbb{T}^{n+2})$.

At this point we have shown that $t_n \in \mathrm{GVAL}(\mathbb{T}^{n+1})$ and hence we have proved (i). Now the last element left is condition (ii), which follows from the following theorem, that is a variant of Theorem 11, but now applied to $\mathbb{T}^n$:

Theorem 48 *If $f \in \mathrm{GVAL}(\mathbb{T}^n)$ has a star domain with a generable vantage point, then we have $f \in \mathrm{ATS}(\mathbb{T}^n)$.*

We note that (necessarily univariate) functions of $\mathrm{GVAL}(\mathbb{T}^{n+1})$ are always defined at any point of $\mathbb{R}_+ = [0, +\infty[$. This implies that such functions have a vantage point over this domain (e.g. 0 or 1 may be used as vantage points). Hence, we can conclude from Theorem 48 that if $f \in \mathrm{GVAL}(\mathbb{T}^n)$, then $f \in \mathrm{ATS}(\mathbb{T}^n)$, which immediately implies condition 3 of Definition 33. To prove Theorem 48, we need another result from [BGP17a, Proposition 28].

Lemma 49 *Let $sp : \mathbb{R}_+ \to \mathbb{R}_+$, $f \in \mathrm{GVAL}(sp)$. There exists a polynomial $q : \mathbb{R} \to \mathbb{R}$, with coefficients in $\mathbb{R}_G$, such that for any $x_1, x_2 \in \operatorname{dom} f$ then $\|f(x_1) - f(x_2)\| \le \|x_1 - x_2\| \, q(sp(\max(\|x_1\|, \|x_2\|)))$.*

This lemma tells us that each function in $\mathrm{GVAL}(sp)$ has a modulus of continuity expressed by the function $q \circ sp$ where q is a polynomial; therefore, since $\mathrm{GVAL}(\mathbb{T}^n)$ is closed by composition with polynomials, this implies that each function in $\mathrm{GVAL}(\mathbb{T}^n)$ has modulus of continuity in $\mathbb{T}^n$.

Proof of Theorem 48. Let $f : \mathbb{R}_+ \to \mathbb{R}$ be a function in $\mathrm{GVAL}(\mathbb{T}^n)$ and let $0 \in \mathbb{R}_G^n \cap \operatorname{dom} f$ be a generable vantage point. By definition of $\mathrm{GVAL}(\mathbb{T}^n)$ we know that there exist a function $sp \in \mathbb{T}^n$, a polynomial p, two initial points t_0, y_0 and a solution y that satisfy Definition 2. In particular, y is the solution of the ODE $y' = p(y)$ and we have $y(t) \le sp(t)$ for all $t \ge 0$. Moreover, since the point 0 belongs to the generable field $\mathbb{R}_G$, and by definition of a generable field, we know that $y(0) \in \mathbb{R}_G^d$ (note that since $y \in \mathrm{GVAL}(\mathbb{T}^n)$, we have that y is defined over $[0, +\infty[$ and hence $y(0)$ is defined). Let $x \in \operatorname{dom} f$, which implies that $x \ge 0$, and consider the system:

$$\begin{cases} x(0) = x \\ \gamma(0) = 0 \\ z(0) = y(0) \end{cases} \quad \begin{cases} x'(t) = 0 \\ \gamma'(t) = x(t) - \gamma(t) \\ z'(t) = p(z(t))\,(x(t) - \gamma(t)) \end{cases} \tag{23}$$

We notice that all the differential equations of (23) are expressed as polynomials of the variables $x(t), \gamma(t), z(t)$. It is immediate to check that the variable $x(t)$ is constant and takes the value $x \geq 0$. This implies that the ODE for γ is separable and can be explicitly solved, yielding $\gamma(t) = x(1 - e^{-t})$. Hence $\gamma(t) \subseteq [0, x] \subseteq \operatorname{dom} f$ for all $t \geq 0$. It is also not difficult to directly check that $z(t) = y(\gamma(t))$ is the solution of the last ODE of (23), which also implies that $\|z(t)\| \leq \|y(\gamma(t))\| \leq sp(\gamma(t)) \leq sp(x)$ for all $t \geq 0$ assuming, without loss of generality, that sp is an increasing function. This shows that $\|(x(t), \gamma(t), z(t))\| \leq \overline{sp}(x)$, where $\overline{sp}$ is some function satisfying $\overline{sp}(x) \geq \max(x, sp(x))$ which belongs to $\mathbb{T}^n$. This shows that $\overline{sp} \in \mathbb{T}^n$ defines a space bound for the solution of (23), where the only argument of $\overline{sp}$ is the initial condition x of (23) (there is no dependence of $\overline{sp}$ on the time variable t). Note also that, due to Definition 2 and because the first m components of the solution $y(t)$ of $y' = p(y)$ yield $f(t)$, we have $(z_1(t), \ldots, z_m(t)) = (y_1(\gamma(t)), \ldots, y_m(\gamma(t))) = f(\gamma(t))$ and this last value converges to $f(x)$ by continuity of f.

Therefore, according to Definition 31, we have just shown that the second condition of this definition is satisfied when using (23) to approach $f(x)$, i.e. we can take $\Upsilon(\|x\|, t) = \Upsilon_1(\|x\|)\Upsilon_2(t)$ in this definition, where $\Upsilon_1 = \overline{sp} \in \mathbb{T}^n$ and $\Upsilon_2(t) = 1$. We now just have to show the first condition of that definition. In other words, we have to show that for any $\mu > 0$, there is some $\Pi_1 \in \mathbb{T}^n$ and a polynomial function Π_2 such that if $t \geq \Pi_1(\|x\|)\Pi_2(\mu)$ then $\|(z_1(t), \ldots, z_m(t)) - f(x)\| \leq e^{-\mu}$.

By lemma 49, we know that there exists $h \in \mathbb{T}^n$ such that, $\forall x_1, x_2 \geq 0$, we have:

$$\|f(x_1) - f(x_2)\| \leq \|x_1 - x_2\|\, h(\max(\|x_1\|, \|x_2\|)).$$

In particular, since $\|\gamma(t)\| \leq x$ and $(z_1(t), \ldots, z_m(t)) = f(\gamma(t))$, we get

$$\begin{aligned} \|f(x) - (z_1(t), \ldots, z_m(t))\| &= \|f(x) - f(\gamma(t))\| \\ &\leq \|x - \gamma(t)\|\, h(x) \\ &= xe^{-t}h(x) \end{aligned}$$

Hence, we have that if

$$t \geq x + h(x) + \mu$$

then

$$\begin{aligned} \|f(x) - (z_1(t), \ldots, z_m(t))\| &\leq xe^{-t}h(x) \\ &\leq xe^{-(x+h(x)+\mu)}h(x) \\ &\leq \frac{x}{e^x}\frac{h(x)}{e^{h(x)}}e^{-\mu} \\ &\leq e^{-\mu}. \end{aligned}$$

By taking $\Pi_1(x) = 1+x+h(x)$ and $\Pi_2(\mu) = (1+\mu)$, we easily see that, by assuming without loss of generality that $x, h(x), \mu \geq 0$, that we have $\Pi_1(x)\Pi_2(\mu) \geq x + h(x) + \mu$ and $\Pi_1 \in \mathbb{T}^n$ and Π_2 is a polynomial. This implies from the above argument that if $t \geq \Pi_1(x)\Pi_2(\mu)$, then $\|f(x) - (z_1(t), \ldots, z_m(t))\| \leq e^{-\mu}$, thus showing that $f \in \mathrm{ATS_p}(\mathbb{T}^n)$. ■

Hence, the classes $\mathbb{T}^n$ defined above satisfy all the four conditions in the list of Definition 33, and therefore Theorem 50 holds. We then have the following result.

Theorem 50 (Analog characterization of the Grzegorczyk hierarchy) *Let $n \in \mathbb{N}$. Let $f : \Gamma^* \to \Gamma^*$, then $f \in \xi^n$ if and only if is emulable under* $\mathrm{ATS}(\mathbb{T}^n)$*.*

8 Characterizing the class EXPTIME

Until now we have described connections between analog classes of functions (e.g. ATSE) and discrete classes of computable functions such as FEXPTIME. At this point a very natural question arises: is it possible to extend these results to discrete classes of computable sets as well, such as EXPTIME? The answer to this important question is yes, and the latter can be done by providing a criterion of acceptance/rejection in the definition of the analog classes. Indeed, we will not require any convergence of the dynamical system, but instead that the solution of the PIVP, with initial condition dependent from the input word as usual, reaches an accepting or rejecting region in a certain amount of time to determine whether the input word w belongs or not to the considered set. This modification will allow solutions of ODEs to *decide* a certain class of sets of words over the considered alphabet Γ, and therefore it will be possible to introduce new classes of sets closely related to EXPTIME. A very similar procedure has been followed by the authors of [BGP17b] to extend their results obtained for polynomial computable functions to the complexity class P.

8.1 Definitions of the analog classes

We start with a definition of a new class of sets, that we will call Exponential-Analog-Recognizable, or EAR.

Definition 51 (EAR) *A language $L \subseteq \Gamma^*$ is called exponential analog recognizable if there exist a vector q of bivariate polynomials and a vector p of polynomials with d variables, both with coefficients in $\mathbb{R}_G$, and an exponential boundary function $\Pi : \mathbb{R} \to \mathbb{R}$ such that for all $w \in \Gamma^*$ there exists (a unique) $y : \mathbb{R} \to \mathbb{R}^d$ satisfying for all $t \geq 0$ (see Fig. 7):*

1. *$y(0) = q(\Psi_k(w))$ and $y'(t) = p(y(t))$;*
2. *if $|y_1(t)| \geq 1$ then $|y_1(u)| \geq 1$ for all $u \geq t$;*

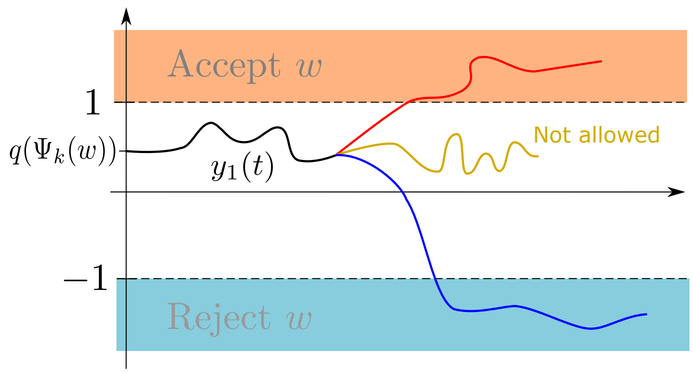

Figure 7: Graphical depiction of Definition 51. A word w is accepted (rejected) if after following the solution curve for a length exponential in $|w|$ the first component y_1 of the solution curve takes values ≥ 1 (≤ -1, respectively).

3. *if* $w \in L$ *(respectively,* $w \notin L$*) and* $len_y(0,t) \geq \Pi(|w|)$ *then* $y_1(t) \geq 1$ *(respectively,* $y_1(t) \leq -1$*);*

4. $len_y(0,t) \geq t$*, where* $len_y(0,t)$ *represents the length of the solution* y *in the interval* $[0,t]$.

Informally, the definition above states that a language L belong to the EAR class if there exists a dynamical system, whose evolution is ruled by a polynomial differential equation, such that, for each word of the alphabet representing a starting point for the system, the trajectory described by the solution enters either the accepting region or the rejecting region as soon as the length of the solution is exponential on the size of the input word. Specifically, condition (1) defines the dynamical system, condition (2) ensures stability once the decision is taken, condition (3) enforces the entrance within the accepting or rejecting region and condition (4) excludes from the classes pathological cases in which the evolution of the system is too slow.

8.2 Equivalence relation between EXPTIME and EAR

We can now state the equivalence result related to the EAR class:

Theorem 52 (EXPTIME equivalence) *For any language* $L \subseteq \Gamma^*$*,* $L \in$ EXPTIME *if and only if* L *is exponential analog recognizable.*

Proof. Let $L \in$ EXPTIME. Then there exists a function $f \in$ FEXPTIME and two distinct symbols $\bar{0}, \bar{1} \in \Gamma$ such that for any $w \in \Gamma^*$, $f(w) = \bar{1}$ if $w \in L$ and $f(w) = \bar{0}$ otherwise. By Theorem 29 there is some $g \in$ ATSE that emulates f. Since for any $w \in \Gamma^*$ one has $f(w) \in \{\bar{0}, \bar{1}\}$, this implies that $\Psi_k(f(w)) = (k^{-1}\gamma(\bar{0}), 1)$ or $\Psi_k(f(w)) = (k^{-1}\gamma(\bar{1}), 1)$. Next define a function

$res : \{k^{-1}\gamma(\bar{0}), k^{-1}\gamma(\bar{1})\} \to \{-2, 2\}$ which is defined by $res(k^{-1}\gamma(\bar{0})) = -2$ and $res(k^{-1}\gamma(\bar{1})) = 2$. Using Lagrange interpolation, by proposition 12 we can extend res to a function $L_{res} \in$ ATSP which extends res to $\mathbb{R}$. Now take $g^*(x) = L_{res}(g_1(x))$, where $g(x) = (g_1(x), g_2(x))$. Since g^* is the composition of an ATSP function with an ATSE function, by Theorem 28 we conclude that $g^* \in$ ATSE. Moreover, $g^*(\Psi_k(w)) = 2$ if $w \in L$ and $g^*(\Psi_k(w)) = -2$ if $w \notin L$.

From the definition of the ATSE class we know that $g^* \in$ ATSE $(\Pi_1\Pi_2, \Upsilon_1\Upsilon_2)$ for some exponential boundary functions Π_1, Υ_1 and some polynomials Π_2, Υ_2 with corresponding d, p, q as parameters and functions defining the dynamical system. Assume, without loss of generality, that these four functions used as boundaries are increasing functions. Let $w \in \Gamma^*$ and consider the following system, where v is a constant variable used to store the input and in particular the input length ($v_2(t) = |w|$), $\tau(t) = t$ is used to keep the time, z is the decision variable, and $\tau^* = \Pi_1(v_2(t))\Pi_2(\ln 2) = \Pi_1(|w|)\Pi_2(\ln 2)$

$$\begin{cases} y(0) = q(\Psi_k(w)) \\ v(0) = \Psi_k(w) \\ z(0) = 0 \\ \tau(0) = 0 \end{cases} \qquad \begin{cases} y'(t) = p(y(t)) \\ v'(t) = 0 \\ z'(t) = \text{lxh}_{[0,1]}(\tau(t) - \tau^*, 1 + \tau(t), y_1(t) - z(t)) \\ \tau'(t) = 1 \end{cases}$$

where y is the solution of the ODE computing g^*. Let $t \in [0, \tau^*]$. Then, by the properties of lxh (see proposition 14), one has $|z'| \leq e^{-1-t}$. This implies that

$$\begin{aligned} z(t) &= \int_0^t z'(u)du \quad \Rightarrow \\ |z(t)| &\leq \int_0^t |z'(u)|\, du \leq \int_0^t e^{-1-u} du \leq e^{-1} - e^{-1-t} \leq e^{-1}. \end{aligned} \tag{24}$$

In particular we conclude that $|z(t)| < 1$ for $t \in [0, \tau^*]$ and therefore that the system has not decided whether w should be accepted or rejected for times $\leq \tau^*$.

Let us now consider the case when $t \geq \tau^*$. By definition of ATSE , we have $\|y_1(t) - g^*(\Psi_k(w))\| \leq e^{-\ln 2}$. Recall that $g^*(\Psi_k(w)) \in \{-2, 2\}$ and let $s \in \{-1, 1\}$ be such that $g^*(\Psi_k(w)) = 2s$. Then $\|y_1(t) - 2s\| \leq \frac{1}{2}$, which means that $y_1(t) = s\lambda(t)$, where $\lambda(t) \geq \frac{3}{2}$. By (24), we conclude that $z(\tau^*) \in [-e^{-1}, e^{-1}]$. From proposition 14, we also conclude that z satisfies, for $t \geq \tau^*$

$$z'(t) = \phi(t)(s\lambda(t) - z(t)),$$

where $1 > \phi(t) > 0$. Let us assume, without loss of generality, that $s = 1$ (a similar reasoning can be applied for the case $s = -1$). Then the previous equation gives us, for $t \geq \tau^*$,

$$z'(t) = \phi(t)(\lambda(t) - z(t)) \geq \phi(t)\left(\frac{3}{2} - z(t)\right). \tag{25}$$

Furthermore, since $z(\tau^*) \in [-e^{-1}, e^{-1}]$, we conclude that z is strictly increasing when $t \geq \tau^*$. To see this, consider a variable $r(t)$ defined by $r'(t) = 3/2 - r(t)$

and $r(\tau^*) = z(\tau^*)$. We can explicitly solve the ODE for r and conclude that it converges to $3/2$ with a rate of convergence of the order of e^{-t} and stays below $3/2$ for all $t \geq \tau^*$. Furthermore, using standard results from ODEs we can conclude that $3/2 > r(t) \geq z(t)$ for all $t \geq \tau^*$. Knowing that $z(\tau^*) \in [-e^{-1}, e^{-1}]$ this implies that $z(t)$ is strictly increasing for all $t \geq \tau^*$. By proposition 14, we have that for $t \geq \tau^* + 1$

$$\left|y_1(t) - z(t) - \mathrm{lxh}_{[0,1]}(\tau(t) - \tau^*, 1 + \tau(t), y_1(t) - z(t))\right| \leq e^{-1-\tau(t)} \leq e^{-1}.$$

This inequality and the definition of $z(t)$ yield that

$$|y_1(t) - z(t) - z'(t)| \leq e^{-1}. \tag{26}$$

We will now show that if $t \geq \tau^{**} = \tau^* + 4/(1 - 2e^{-1})$, then $z(t) \geq 1$. To show that it suffices to show that $z(\tau^{***}) \geq 1$ for some $\tau^{***} \in [\tau^*, \tau^{**}]$ since z is increasing for $t \geq \tau^*$. Suppose, by absurd, that there is no such τ^{***}. Then $z(t) < 1$ for all $t \in [\tau^*, \tau^{**}]$ which implies that $y_1(t) - z(t) > 3/2 - 1 = 1/2$. Then using this last inequality and (26), we conclude that $z'(t) \geq 1/2 - e^{-1}$. Since $z(\tau^*) \in [-e^{-1}, e^{-1}]$, this implies that

$$\begin{aligned}
z(\tau^{**}) &= z(\tau^*) + \int_{\tau^*}^{\tau^{**}} z'(t)dt \\
&\geq -e^{-1} + (\tau^{**} - \tau^*)\left(1/2 - e^{-1}\right) \\
&\geq -1 + \frac{4}{1 - 2e^{-1}}\left(\frac{1}{2} - e^{-1}\right) \\
&= -1 + \frac{2}{\frac{1}{2} - e^{-1}}\left(\frac{1}{2} - e^{-1}\right) \\
&= 1.
\end{aligned}$$

which is absurd. Therefore $z(t) \geq 1$ for all $t \geq \tau^{**}$. This proves conditions 1 and 2 of Definition 51.

Note that $\|(y, v, z, \tau)'(t)\| \geq 1$ for all $t \geq 1$ so condition 4 of Definition 51 is also satisfied. To show condition 3, recall that $|g^*(\Psi_k(w))| = 2$. Therefore from (24) for $t < \tau^*$ and from the previous analysis for $t \geq \tau^*$, it is possible to conclude that $|z(t)| \leq |g^*(\Psi_k(w))| + 1/2 = 5/2$ for all $t \geq 0$. This shows that if $Y = (y, v, z, \tau)$, then $\|Y(t)\|$ is bounded by an exponential boundary function on $\|\Psi_k(w)\|$ and by a polynomial on t because $\|y(t)\| \leq \Upsilon_1(\|\Psi_k(w)\|)\Upsilon_2(t)$ for all $t \geq 0$. Because $y'(t) = p(y(t))$ and, by proposition 14 $|z'(t)| \leq |y_1(t) - z(t)| \leq |y_1(t)| + 5/2$, we conclude that there are an exponential boundary function Υ_1^* and a polynomial Υ_2^* such that $\|Y'(t)\| \leq \Upsilon_1^*(\|\Psi_k(w)\|)\Upsilon_2^*(t)$ and, without loss of generality, we can assume that Υ_1^* and Υ_2^* are increasing functions. Now, since $\|Y'(t)\| \geq 1$, we have that

$$t \leq len_Y(0, t) \leq t \sup_{u \in [0,t]} \|Y'(u)\| \leq t\Upsilon_1^*(\|\Psi_k(w)\|)\Upsilon_2^*(t). \tag{27}$$

Define the function Π^* by $\Pi^*(|w|) \equiv \tau^{**}\Upsilon_1^*(|w|)\Upsilon_2^*(\tau^{**})$ which is an exponential boundary function in $\|\Psi_k(w)\| = |w|$, because τ^{**} is an exponential boundary function on $|w|$, Υ_2^* is a polynomial, and Υ_1^* is an exponential boundary function. Let t be such that $len_Y(0,t) \geq \Pi^*(|w|)$. Then, by (27),

$$t\Upsilon_1^*(|w|)\Upsilon_2^*(t) \geq \Pi^*(|w|) = \tau^{**}\Upsilon_1^*(|w|)\Upsilon_2^*(\tau^{**})$$

which implies that $t\Upsilon_2^*(t) \geq \tau^{**}\Upsilon_2^*(\tau^{**})$. Since Υ_2^* is increasing, this last condition is only true when $t \geq \tau^{**}$ which, by the previous analysis, implies that $|z(t)| \geq 1$, that is, the system has decided. This concludes the direct direction of the proof of Theorem 52.

We will now proceed with the reverse direction of the proof of Theorem 52. Assume that $L \in \mathrm{EAR}$. Apply the definition of the class EAR to get the parameters and polynomials d, p, q characterizing the dynamical system and an exponential boundary function Π which satisfies the third condition of the definition of the class. Let $w \in \Gamma^*$ and consider the following system

$$y(0) = q(\Psi_k(w)),\ y'(t) = p(y(t)).$$

We will show that we can decide in time exponential in $|w|$ whether $w \in L$ or not. Note that q is a polynomial with coefficients in $\mathbb{R}_\mathrm{P}$ and $\Psi_k(w)$ is a rational number. Therefore $q(\Psi_k(w)) \in \mathbb{R}_\mathrm{P}^d$. Finally, note that

$$\begin{aligned}
PsLen_{y,p}(0,t) &= \int_0^t \Sigma p \max(1, \|y(u)\|)^{\deg(p)} du \\
&\leq t\Sigma p \max(1, \sup_{u\in[0,t]} \|y(u)\|^{\deg(p)}) \\
&\leq t\Sigma p \max(1, \sup_{u\in[0,t]} (\|y(0)\| + len_y(0,t))^{\deg(p)}) \\
&\leq t\, poly(len_y(0,t)) \\
&\leq poly(len_y(0,t))
\end{aligned}$$

where the last inequality holds because $len_y(0,t) \geq t$. We can now apply Theorem 30 to conclude that we are able to compute $y(t) \pm 2^{-\mu}$ in time polynomial in t, μ and $len_y(0,t)$.

At this point some extra care is necessary. Indeed, the temptation is to use Theorem 30 to compute with a desired precision the value of the curve $y(t)$ at time $\Pi(|w|)$. Nevertheless, it is possible that, at time $\Pi(|w|)$, the length of the solution could be already over exponential in $|w|$. Therefore, it is essential to use carefully the algorithm developed for the proof of Theorem 30 and stop the computation as soon as the length of the solution is greater than $\Pi(|w|)$. This is possible due to the particular nature of the algorithm developed for the proof of Theorem 30 in [PG16]. Let t^* be the time at which the algorithm stops. Then, the running time of the algorithm will be polynomial in t^*, μ and $len_y(0,t^*) \leq \Pi(|w|) + O(1)$. Finally, by definition of the EAR class, we have $t^* \leq len_y(0,t^*)$ and so, because $len_y(0,t^*) \leq \Pi(|w|)$ this algorithm has running time exponential in $|w|$ and polynomial in μ. Take $\mu = \log 2$. Then we can obtain $\tilde{y}$ such that $\|y(t^*) - \tilde{y}\| \leq \frac{1}{2}$. By definition of Π we have that $y_1(t^*) \geq 1$ or $y_1(t^*) \leq -1$, so we can decide from $\tilde{y}$ if $w \in L$ or not. This finishes the proof of the theorem. ∎

9 Conclusions and open problems

In this paper we have showed that the analog characterization obtained in [BGP17b] for polynomial complexity classes can be applied to other classes of computable functions. To reach this conclusion we had to overcome the fact that the equivalence established at a polynomial level was tailored over properties of polynomials, and therefore not trivially applicable to more general classes of functions. We started by analyzing the exponential case and we proved that the key ingredient to obtain the analog characterization was to modify the analog classes involved in such a way that could ensure closure by basic arithmetic operations and closure by composition with functions in ATSP. This led us to Definition 23 of the ATSE class and provided us enough elements to be able to apply the characterization for both the class of functions FEXPTIME and the class of sets EXPTIME by means of Theorems 29 and 52 respectively. Moreover, taking the exponential case as an inspiration and as a starting point, we established which set of conditions on the boundary functions are sufficient for being able to repeat the process applied to the exponential functions and obtain characterizations of greater complexity classes. This analysis has brought us to the formulation of the set of conditions 33 that implied Theorem 35. Then, we applied this generalization to the concrete case of the Grzegorczyk hierarchy, which implies a characterization of the class of elementary function as well as the class of primitive recursive functions.

As a natural consequence of the work presented with this paper, one can wonder whether a similar characterization can be obtained for complexity classes such as NP. The framework used in this paper only deals which classes of languages decided in deterministic time. It would be interesting to know if this framework could be extended to non-deterministic time.

Acknowledgments. The authors wish to thank Olivier Bournez and Amaury Pouly for helpful discussions. R. Gozzi was supported by the Fundação para a Ciência e a Tecnologia (FCT) via grant PD/BD/135190/2017. R. Gozzi and D. Graça were partially funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/50008/2020.

References

[BCGH06] O. Bournez, M. L. Campagnolo, D. S. Graça, and E. Hainry. The General Purpose Analog Computer and Computable Analysis are two equivalent paradigms of analog computation. In J.-Y. Cai, S. B. Cooper, and A. Li, editors, *Proc. Theory and Applications of Models of Computation (TAMC 06)*, volume 3959 of *Lecture Notes in Computer Science*, pages 631–643. Springer, 2006.

[BCGH07] O. Bournez, M. L. Campagnolo, D. S. Graça, and E. Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. *Journal of Complexity*, 23(3):317–335, 2007.

[BGP16a] O. Bournez, D. S. Graça, and A. Pouly. Computing with polynomial ordinary differential equations. *Journal of Complexity*, 36:106–140, 2016.

[BGP16b] O. Bournez, D. S. Graça, and A. Pouly. Polynomial time corresponds to solutions of polynomial ordinary differential equations of polynomial length – the general purpose analog computer and computable analysis are two efficiently equivalent models of computations. In I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, editors, *Proc. 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 109:1–109:15. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[BGP17a] O. Bournez, D. S. Graça, and A. Pouly. On the functions generated by the general purpose analog computer. *Information and Computation*, 257:34 – 57, 2017.

[BGP17b] O. Bournez, D. S. Graça, and A. Pouly. Polynomial time corresponds to solutions of polynomial ordinary differential equations of polynomial length. *Journal of the ACM*, 64(6):38:1–38:76, 2017.

[BH04] O. Bournez and E. Hainry. An analog characterization of elementarily computable functions over the real numbers. In *Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP 2004)*, volume 3142 of *Lecture Notes in Computer Science*, pages 269–280. Springer, 2004.

[BH05] 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.

[BHW08] V. Brattka, P. Hertling, and K. Weihrauch. A tutorial on computable analysis. In S. B. Cooper, , B. Löwe, and A. Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 425–491. Springer, 2008.

[Bra95] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, 1995.

[Bus31] V. Bush. The differential analyzer. A new machine for solving differential equations. *Journal of the Franklin Institute*, 212:447–488, 1931.

[CM01] M. Campagnolo and C. Moore. Upper and lower bounds on continuous-time computation. In I. Antoniou, C. Calude, and M. Dinneen, editors, *Proc. 2nd International Conference on Unconventional Models of Computation - UMC'2K*, pages 135–153. Springer, 2001.

[CMC00] M. L. Campagnolo, C. Moore, and J. F. Costa. Iteration, inequalities, and differentiability in analog computers. *Journal of Complexity*, 16(4):642–660, 2000.

[CMC02] M. L. Campagnolo, C. Moore, and J. F. Costa. An analog characterization of the Grzegorczyk hierarchy. *Journal of Complexity*, 18(4):977–1000, 2002.

[GC03] D. S. Graça and J. F. Costa. Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5):644–664, 2003.

[GCB08] D. S. Graça, M. L. Campagnolo, and J. Buescu. Computability with polynomial differential equations. *Advances in Applied Mathematics*, 40(3):330–349, 2008.

[Gol08] O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, revised first edition, 2008.

[Gra04] D. S. Graça. Some recent developments on Shannon's General Purpose Analog Computer. *Mathematical Logic Quarterly*, 50(4-5):473–485, 2004.

[Grz53] A. Grzegorczyk. Some classes of recursive functions. *Rozprawy matematyczne*, 4:1–45, 1953.

[HW95] J. H. Hubbard and B. H. West. *Differential Equations: A Dynamical Systems Approach — Higher-Dimensional Systems*. Springer, 1995.

[KF82] K.-I Ko and H. Friedman. Computational complexity of real functions. *Theoretical Computer Science*, 20:323–352, 1982.

[Odi99] P. Odifreddi. *Classical Recursion Theory*, volume 2. Elsevier, 1999.

[PG16] Amaury Pouly and Daniel S. Graça. Computational complexity of solving polynomial differential equations over unbounded domains. *Theoretical Computer Science*, 626(2):67–82, 2016.

[Sha41] C. E. Shannon. Mathematical theory of the differential analyzer. *Journal of Mathematics and Physics*, 20:337–354, 1941.

[Wei00] K. Weihrauch. *Computable Analysis: an Introduction*. Springer, 2000.