Polynomial Time Corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length

Olivier Bournez
École Polytechnique, LIX, 91128 Palaiseau Cedex, France

Daniel Graça
FCT, Universidade do Algarve, C. Gambelas
8005-139 Faro, Portugal
& Instituto de Telecomunicações, Lisbon, Portugal

Amaury Pouly*
Department of Computer Science, University of Oxford
Wolfson Building, Parks Rd, OX1 3QD Oxford, United Kingdom

September 29, 2017

Abstract

The outcomes of this paper are twofold.

Implicit complexity. We provide an implicit characterization of polynomial time computation in terms of ordinary differential equations: we characterize the class P of languages computable in polynomial time in terms of differential equations with polynomial right-hand side. This result gives a purely continuous elegant and simple characterization of P. We believe it is the first time complexity classes are characterized using only ordinary differential equations. Our characterization extends to functions computable in polynomial time over the reals in the sense of Computable Analysis.

Our results may provide a new perspective on classical complexity, by giving a way to define complexity classes, like P, in a very simple way, without any reference to a notion of (discrete) machine. This may also provide ways to state classical questions about computational complexity via ordinary differential equations.

Continuous-Time Models of Computation. Our results can also be interpreted in terms of analog computers or analog models of computation: As a side effect, we get that the 1941 General Purpose Analog Computer (GPAC) of Claude Shannon is provably equivalent to Turing machines both in terms of computability and complexity, a fact that has never been established before. This result provides arguments in favour of a generalised form of the Church-Turing Hypothesis, which states that any physically realistic (macroscopic) computer is equivalent to Turing machines both in terms of computability and complexity.

Contents

1 Introduction

The current article is a journal extended version of our paper presented at 43rd International Colloquium on Automata, Languages and Programming ICALP'2016 (Track B best paper award).

The outcomes of this paper are twofold, and concern a priori not closely related topics.

1.1 Implicit Complexity

Since the introduction of the P and NP complexity classes, much work has been done to build a well-developed complexity theory based on Turing Machines. In particular, classical computational complexity theory is based on limiting resources used by Turing machines, such as time and space. Another approach is implicit computational complexity. The term "implicit" in this context can be understood in various ways, but a common point of these characterizations is that they provide (Turing or equivalent) machine-independent alternative definitions of classical complexity.

Implicit complexity theory has gained enormous interest in the last decade. This has led to many alternative characterizations of complexity classes using recursive functions, function algebras, rewriting systems, neural networks, lambda calculus and so on.

However, most of — if not all — these models or characterizations are essentially discrete: in particular they are based on underlying discrete-time models working on objects which are essentially discrete, such as words, terms, etc.

Models of computation working on a continuous space have also been considered: they include Blum Shub Smale machines [BCSS98], Computable Analysis [Wei00], and quantum computers [Fey82] which usually feature discrete-time and continuous-space. Machine-independent characterizations of the corresponding complexity classes have also been devised: see e.g. [BCdNM05, GM95]. However, the resulting characterizations are still essentially discrete, since time is still considered to be discrete.

In this paper, we provide a purely analog machine-independent characterization of the class P. Our characterization relies only on a simple and natural class of ordinary differential equations: P is characterized using ordinary differential equations (ODEs) with polynomial right-hand side. This shows first that (classical) complexity theory can be presented in terms of ordinary differential equations problems. This opens the way to state classical questions, such as P vs NP, as questions about ordinary differential equations, assuming one can also express NP this way.

1.2 Analog Computers

Our results can also be interpreted in the context of analog models of computation and actually originate as a side effect of an attempt to understand the power of continuous-time analog models relative to classical models of computation. Refer to [Ulm13] for a very instructive historical account of the history of Analog computers. See also [Mac09, BC08] for further discussions.

Indeed, in 1941, Claude Shannon introduced in [Sha41] the General Purpose Analog Computer (GPAC) model as a model for the Differential Analyzer [Bus31], a mechanical programmable machine, on which he worked as an operator. The GPAC model was later refined in [PE74], [GC03]. Originally it was presented as a model based on circuits (see Figure 1), where several units performing basic operations (e.g. sums, integration) are interconnected (see Figure 2).

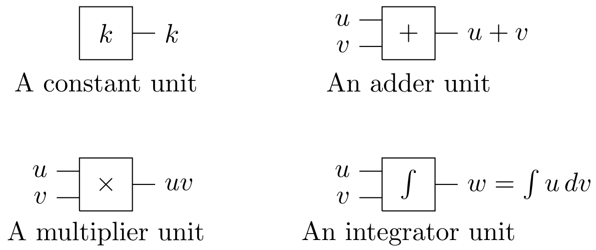
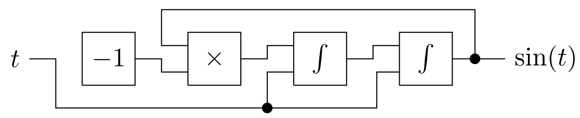


Figure 1: Circuit presentation of the GPAC: a circuit built from basic units



$$\begin{cases} y'(t) = z(t) \\ z'(t) = -y(t) \\ y(0) = 0 \\ z(0) = 1 \end{cases} \Rightarrow \begin{cases} y(t) = \sin(t) \\ z(t) = \cos(t) \end{cases}$$

Figure 2: Example of GPAC circuit: computing sine and cosine with two variables

However, Shannon himself realized that functions computed by a GPAC are nothing more than solutions of a special class of polynomial differential equations. In particular it can be shown that a function is computed by a GPAC if and only if it is a (component of the) solution of a system of ordinary differential equations (ODEs) with polynomial right-hand side [Sha41], [GC03]. In this paper, we consider the refined version presented in [GC03].

We note that the original notion of computation in the model of the GPAC presented in [Sha41], [GC03] is known not to be equivalent to Turing machine based models, like Computable Analysis. However, the original GPAC model only allows for functions in one continuous variable and in real-time: at time t the output is $f(t)$, which is different from the notion used by Turing machines. This prevents the original GPAC model from computing functions on several variables and from computing functions like the Gamma function Γ. Moreover, the model from [Sha41] only considers differential equations which are assumed to have unique solutions, while in general it is not trivial to know when a differential equation has a unique solution or not (this problem was solved in [GC03]).

In [Gra04] a new notion of computation for the GPAC, which uses "converging computations" as done by Turing machines was introduced and it was shown in [BCGH06], [BCGH07] that using this new notion of computation, the GPAC and Computable Analysis are two equivalent models of computation, at the computability level.

Our paper extends this latter result and proves that the GPAC and Computable Analysis are two equivalent models of computation, both in terms of computability and complexity. As a consequence of this work, we also provide a robust way to measure time in the GPAC, or more generally in computations performed by ordinary differential equations: essentially by considering the length of the solution curve.

2 Results and discussion

2.1 Our results

The first main result of this paper shows that the class P can be characterized using ODEs. In particular this result uses the following class of differential equations:

$$y(0) = y_0 \qquad y'(t) = p(y(t)) \tag{1}$$

where p is a vector of polynomials and $y : I \to \mathbb{R}^d$ for some interval $I \subset \mathbb{R}$. Such systems are sometimes called PIVP, for polynomial initial value problems [GBC09]. Observe that, as opposed to the differential algebraic equations describing a GPAC, as used in [Sha41], there is always a unique solution to the PIVP (the approach used in [GC03]), which is analytic, defined on a maximum domain I containing 0, which we refer to as "the solution".

To state complexity results via ODEs, we need to introduce some kind of complexity measure for ODEs and, more concretely, for PIVPs. This is a nontrivial task since, contrarily to discrete models of computation, continuous models of computation (not only the GPAC, but many others) usually exhibit the so-called "Zeno phenomena", where time can be arbitrarily contracted in a continuous system, thus allowing an arbitrary speed-up of the computation, if we take the naive approach of using the time variable of the ODE as a measure of "time complexity" (see Section 2.3 for more details).

Our crucial and key idea to solve this problem is that, when using PIVPs (in principle this idea can also be used for others continuous models of computation) to compute a function f, the cost should be measured as a function of the length of the solution curve of the PIVP computing the function f. We recall that the length of a curve $y \in C^1(I, \mathbb{R}^n)$ defined over some interval $I = [a, b]$ is given by $\text{len}_y(a, b) = \int_I \|y'(t)\|\, dt$, where $\|y\|$ refers to the infinity norm of y.

Since a language is made up of words, we need to discuss how to represent (encode) a word into a real number to decide a language with a PIVP. We fix a finite alphabet $\Gamma = \{0, .., k-2\}$ and define the encoding[1] $\psi(w) = \left(\sum_{i=1}^{|w|} w_i k^{-i}, |w|\right)$ for a word $w = w_1 w_2 \dots w_{|w|}$. We also take $\mathbb{R}_+ = [0, +\infty[$.

[1]Other encodings may be used, however, two crucial properties are necessary: (i) $\psi(w)$ must provide a way to recover the length of the word, (ii) $\|\psi(w)\| \approx \text{poly}(|w|)$ in other words, the norm of the encoding is roughly the length of the word. For technical reasons, we need to encode the number in basis one more than the number of symbols.

Definition 1 (Discrete recognizability) poly-length-analog-recognizable *if there exists a vector q of bivariate polynomials and a vector p of polynomials with d variables, both with coefficients in $\mathbb{Q}$, and a polynomial $\Pi : \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $w \in \Gamma^*$, there is a (unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ such that for all $t \in \mathbb{R}_+$:*

- $y(0) = q(\psi_k(w))$ *and* $y'(t) = p(y(t))$ ▶ *y satisfies a differential equation*
- *if* $|y_1(t)| \geqslant 1$ *then* $|y_1(u)| \geqslant 1$ *for all* $u \geqslant t$ ▶ *decision is stable*
- *if* $w \in \mathcal{L}$ *(resp.* $\notin \mathcal{L}$*) and* $\operatorname{len}_y(0,t) \geqslant \Pi(|w|)$ *then* $y_1(t) \geqslant 1$ *(resp.* $\leqslant -1$*)* ▶ *decision*
- $\operatorname{len}_y(0,t) \geqslant t$ ▶ *technical condition*[2]

Intuitively (see Fig. 3) this definition says that a language is poly-length-analog-recognizable if there is a PIVP such that, if the initial condition is set to be (the encoding of) some word $w \in \Gamma^*$, then by using a *polynomial length* portion of the curve, we are able to tell if this word should be accepted or rejected, by watching to which region of the space the trajectory goes: the value of y_1 determines if the word has been accepted or not, or if the computation is still in progress. See Figure 3 for a graphical representation of Definition 1.

Theorem 2 (A characterization of P) *A decision problem (language) $\mathcal{L}$ belongs to the class* P *if and only if it is poly-length-analog-recognizable.*

A slightly more precise version of this statement is given at the end of the paper, in Theorem 68. A characterization of the class FP of polynomial-time computable functions is also given in Theorem 52.

Concerning the second main result of this paper, we assume the reader is familiar with the notion of a polynomial-time computable function $f : [a, b] \to \mathbb{R}$ (see [Ko91], [Wei00] for an introduction to Computable Analysis). We denote by $\mathbb{R}_P$ the set of polynomial-time computable reals. For any vector y, $y_{i\ldots j}$ refers to the vector $(y_i, y_{i+1}, \ldots, y_j)$. For any sets X and Z, $f :\subseteq X \to Z$ refers to any function $f : Y \to Z$ where $Y \subseteq X$ and $\operatorname{dom} f$ refers to the domain of definition of f.

Our second main result is an analog characterization of polynomial-time computable real functions. More precisely, we show that the class of poly-length-computable functions (defined below), when restricted to domains of the form $[a, b]$, is the same as the class of polynomial-time computable real functions of Computable Analysis over $[a, b]$, sometimes denoted by $\mathrm{P}_{C[a,b]}$, as defined in [Ko91]. It is well-known that all computable functions (in the Computable Analysis setting) are continuous. Similarly, all poly-length-computable functions (and more generally GPAC-computable functions) are continuous (see Theorem 22).

Definition 3 (Poly-Length-Computable Functions) *We say that $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ is poly-length-computable if and only if there exists a vector p of polynomials with $d \geqslant m$ variables and a vector q of polynomials with n variables, both with coefficients in $\mathbb{Q}$, and a bivariate polynomial Π such that for any $x \in \operatorname{dom} f$, there exists (a unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ satisfying for all $t \in \mathbb{R}_+$:*

[2] This could be replaced by only assuming that we have somewhere the additional ordinary differential equation $y_0' = 1$.

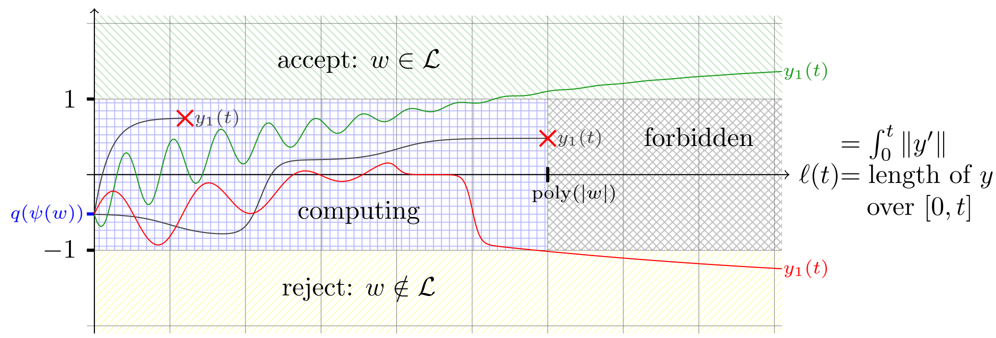


Figure 3: Graphical representation of poly-length-analog-recognizability (Definition 1). The green trajectory represents an accepting computation, the red a rejecting one, and the gray are invalid computations. An invalid computation is a trajectory that is too slow (or converges) (thus violating the technical condition), or that does not accept/reject in polynomial length. Note that we only represent the first component of the solution, the other components can have arbitrary behaviors.

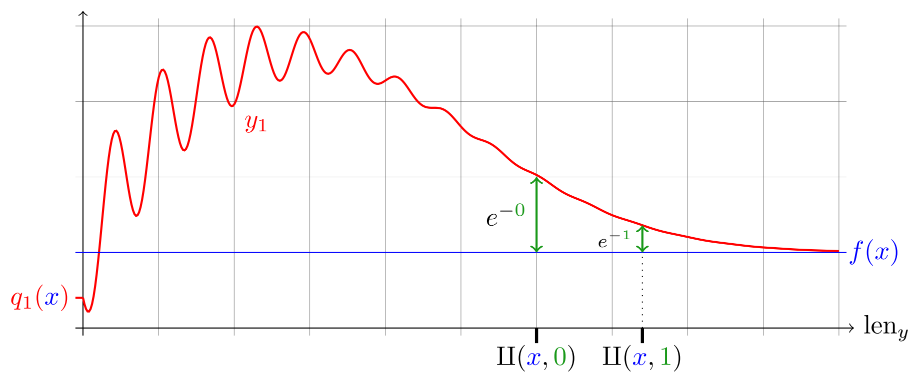


Figure 4: Poly-length-computability: on input x, starting from initial condition $q(x)$, the PIVP $y' = p(y)$ ensures that $y_1(t)$ gives $f(x)$ with accuracy better than $e^{-\mu}$ as soon as the length of y (from 0 to t) is greater than $\Pi(\|x\|, \mu)$. Note that we did not plot the other variables $y_2, \ldots, y_d$ and the horizontal axis measures the length of y (instead of the time t).

- $y(0) = q(x)$ *and* $y'(t) = p(y(t))$ ▶ *y satisfies a PIVP*

- $\forall \mu \in \mathbb{R}_+$, *if* $\operatorname{len}_y(0,t) \geqslant \Pi(\|x\|, \mu)$ *then* $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ ▶ $y_{1..m}$ *converges to* $f(x)$

- $\operatorname{len}_y(0,t) \geqslant t$ ▶ *technical condition: the length grows at least linearly with time*[3][4]

Intuitively, a function f is poly-length-computable if there is a PIVP that approximates f with a polynomial length to reach a given level of approximation. See Figure 4 for a graphical representation of Definition 3 and Section 3.2 for more background on analog computable functions.

Theorem 4 (Equivalence with Computable Analysis) *For any* $a, b \in \mathbb{R}_P$ *and* $f \in C^0([a,b],\mathbb{R})$, *f is polynomial-time computable if and only if it is poly-length-computable.*

A slightly more precise version of this statement is given at the end of the paper, in Theorem 79.

2.2 Applications to computational complexity

We believe these characterizations to open a new perspective on classical complexity, as we indeed provide a natural definition (through previous definitions) of P for decision problems and of polynomial time for functions over the reals using analysis only i.e. ordinary differential equations and polynomials, no need to talk about any (discrete) machinery like Turing machines. This may open ways to characterize other complexity classes like NP or PSPACE. In the current settings of course NP can be viewed as an existential quantification over our definition, but we are obviously talking about "natural" characterizations, not involving unnatural quantifiers (for e.g. a concept of analysis like ordinary differential inclusions).

As a side effect, we also establish that solving ordinary differential equations with polynomial right-hand side leads to P-complete problems, when the length of the solution curve is taken into account. In an less formal way, this is stating that ordinary differential equations can be solved by following the solution curve (as most numerical analysis method do), but that for general (and even right-hand side polynomial) ODEs, no better method can work. Note that our results only deal with ODEs with a polynomial right-hand side and that we do not know what happens for ODEs with analytic right-hand sides over unbounded domains. There are some results (see e.g. [MM93]) which show that ODEs with analytic right-hand sides can be computed locally in polynomial time. However these results do not apply to our setting since we need to compute the solution of ODEs over arbitrary large domains, and not only locally.

[3]This is a technical condition required for the proof. This can be weakened, for example to $\|y'(t)\| = \|p(y(t))\| \geqslant \frac{1}{\operatorname{poly}(t)}$. The technical issue is that if the speed of the system becomes extremely small, it might take an exponential time to reach a polynomial length, and we want to avoid such "unnatural" cases. This is satisfied by all examples of computations we know [Ulm13]. It also avoids pathological cases where the system would "stop" (i.e. converge) before accepting/rejecting, as depicted in Figure 3.

[4]This could also be replaced by only assuming that we have somewhere the additional ordinary differential equation $y_0' = 1$.

Figure 5: A continuous system before and after an exponential speed-up.

2.3 Applications to continuous-time analog models

PIVPs are known to correspond to functions that can be generated by the GPAC of Claude Shannon [Sha41], which is itself a model of the analog computers (differential analyzers) in use in the first half of the XXth century [Bus31].

As we have mentioned previously, defining a robust (time) complexity notion for continuous time systems was a well known open problem [BC08] with no generic solution provided to this day. In short, the difficulty is that the naive idea of using the time variable of the ODE as a measure of "time complexity" is problematic, since time can be arbitrarily contracted in a continuous system due to the "Zeno phenomena". For example, consider a continuous system defined by an ODE

$$y' = f(y)$$

where $f : \mathbb{R} \to \mathbb{R}$ and with solution $\theta : \mathbb{R} \to \mathbb{R}$. Now consider the following system

$$\begin{cases} y' = f(y)z \\ z' = z \end{cases}$$

with solution $\phi : \mathbb{R}^2 \to \mathbb{R}^2$. It is not difficult to see that this systems re-scales the time variable and that its solution $\phi = (\phi_1, \phi_2)$ is given by $\phi_2(t) = e^t$ and $\phi_1(t) = \theta(e^t)$ (see Figure 5). Therefore, the second ODE simulates the first ODE, with an exponential acceleration. In a similar manner, it is also possible to present an ODE which has a solution with a component $\varphi_1 : \mathbb{R} \to \mathbb{R}$ such that $\varphi_1(t) = \phi(\tan t)$, i.e. it is possible to contract the whole real line into a bounded set. Thus any language computable by the first system (or, in general, by a continuous system) can be computed by another continuous system in time $O(1)$. This problem appears not only for PIVPs (or, equivalently, GPACs), but also for many continuous models (see e.g. [Ruo93], [Ruo94], [Moo96], [Bou97], [Bou99], [AD90], [CP02], [Dav01], [Cop98], [Cop02]).

With that respect, we solve this open problem by stating that the "time complexity" should be measured by the length of the solution curve of the ODE. Doing so, we get a robust notion of time complexity for PIVP systems. Indeed, the length is a geometric property of the curve and is thus "invariant" by rescaling.

Using this notion of complexity, we are then able to show that functions computable by a GPAC in polynomial time are exactly the functions computable in polynomial time in the sense of Computable Analysis (see Section 2.1 or [Ko91]). It was already previously shown in [BCGH06], [BCGH07] that functions computable by a GPAC are exactly those computable in the sense of Computable Analysis. However this result was only pertinent for computability. Here we show that this equivalence holds also at a computational complexity level.

Stated otherwise, analog computers (as used before the advent of the digital computer) are theoretically equivalent to (and not more powerful than) Turing machine based models, both at a computability and complexity level. Note that this is a new result since, although digital computers are usually more powerful than analog computers at our current technological stage, it was not known what happened at a fundamental level.

This result leave us to conjecture the following generalization of the Church-Turing thesis: any physically realistic (macroscopic) computer is equivalent to Turing machines both in terms of computability and computational complexity.

2.4 Applications to algorithms

We believe that transferring the notion of time complexity to a simple consideration about length of curves allows for very elegant and nice proofs of polynomiality of many methods for solving both continuous and discrete problems. For example, the zero of a function f can easily be computed by considering the solution of $y' = -f(y)$ under reasonable hypotheses on f. More interestingly, this may also cover many interior-point methods or barrier methods where the problem can be transformed into the optimization of some continuous function (see e.g. [Kar84, Fay91, BHFFS03, KMNY91]).

2.5 Related work

We believe that no purely continuous-time definition of P has ever been stated before. One direction of our characterization is based on a polynomial-time algorithm (in the length of the curve) to solve PIVPs over unbounded time domains, and strengthens all existing results on the complexity of solving ODEs over unbounded time domains. In the converse direction, our proof requires a way to simulate a Turing machine using PIVP systems of polynomial length, a task whose difficulty is discussed below, and still something that has never been done up to date.

Attempts to derive a complexity theory for continuous-time systems include [GM02]. However, the theory developed there is not intended to cover generic dynamical systems but only specific systems that are related to Lyapunov theory for dynamical systems. The global minimizers of particular energy functions are supposed to give solutions of the problem. The structure of such energy functions leads to the introduction of problem classes U and NU, with the existence of complete problems for theses classes.

Another attempt is [BHSF02], which also focused on a very specific type of systems: dissipative flow models. The proposed theory is nice but non-generic. This theory has been used in several papers from the same authors to study a particular class of flow dynamics [BHFFS03] for solving linear programming problems.

Neither of the previous two approaches is intended to cover generic ODEs, and none of them is able to relate the obtained classes to classical classes from computational complexity.

To the best of our knowledge, the most up to date surveys about continuous time computation are [BC08, Mac09].

Relating computational complexity problems (such as the P vs NP question) to problems of analysis has already been the motivation of other papers. In particular, Félix Costa and Jerzy Mycka have a series of work (see e.g. [MC06]) relating the P vs NP question to questions in the context of real and complex analysis. Their approach is very different: they do so at the price of introducing a whole hierarchy of functions and operators over functions. In particular, they can use multiple times an operator which solves ordinary differential equations before defining an element of $DAnalog$ and $NAnalog$ (the counterparts of P and NP introduced in their paper), while in our case we do not need the multiple application of this kind of operator: we only need to use *one* application of such an operator (i.e. we only need to solve one ordinary differential equations with polynomial right-hand side).

It its true that one can sometimes convert the multiple use of operators solving ordinary differential equations into a single application [GC03], but this happens only in very specific cases, which do not seem to include the classes $DAnalog$ and $NAnalog$. In particular, the application of nested continuous recursion (i.e. nested use of solving ordinary differential equations) may be needed using their constructions, whereas we define P using only a simple notion of acceptance and only *one* system of ordinary differential equations.

We also mention that Friedman and Ko (see [Ko91]) proved that polynomial time computable functions are closed under maximization and integration if and only if some open problems of computational complexity (like P = NP for the maximization case) hold. The complexity of solving Lipschitz continuous ordinary differential equations has been proved to be polynomial-space complete by Kawamura [Kaw10].

This paper contains mainly original contributions. We however make references to results established in:

1. [BGP16a], under revision for publication in *Information and Computation*, devoted to properties of generable functions.

2. [BGP16c], published in *Journal of Complexity*, devoted to the proof of Proposition 12.

3. [PG16], published in *Theoretical Computer Science*, devoted to providing a polynomial time complexity algorithm for solving polynomially bounded polynomial ordinary differential equations.

None of these papers establishes relations between polynomial-length-analog-computable-functions and classical computability/complexity. This is precisely the objective of the current article.

2.6 Organization of the remainder of the paper

In Section 3, we introduce generable functions and computable functions. Generable functions are functions computable by PIVPs (GPACs) in the classical sense of [Sha41]. They will be a crucial tool used in the paper to simplify the construction of polynomial differential equations. Computable functions were introduced in [BGP16c]. This section does not contain any new original result, but only recalls already known results about these classes of functions.

Section 4 establishes some original preliminary results needed in the rest of the paper: First we relate generable functions to computable functions under some basic conditions about their domain. Then we show that the class of computable functions is closed under arithmetic operations and composition. We then provide several growth and continuity properties. We then prove that absolute value, min, max, and some rounding functions, norm, and bump function are computable.

In Section 5, we show how to efficiently encode the step function of Turing machines using a computable function.

In Section 6, we provide a characterization of FP. To obtain this characterization, the idea is basically to iterate the functions of the previous section using ordinary differential equations in one direction, and to use a numerical method for solving polynomial ordinary differential equations in the reverse direction.

In Section 7, we provide a characterization of P.

In Section 8, we provide a characterization of polynomial time computable functions over the real in the sense of Computable Analysis.

On purpose, to help readability of the main arguments of the proof, we postpone the most technical proofs to Section 9. This latter section is devoted to proofs of some of the results used in order to establish previous characterizations.

Up to Section 9, we allow coefficients that maybe possibly non-rational numbers. In Section 10, we prove that all non -rationnal coefficients can be eliminated. This proves our main results stated using only rational coefficients.

A list of notations used in this paper as well as in the above mentioned related papers can be found in Appendix A.

3 Generable and Computable Functions

In this section we define the main classes of functions considered in this paper and state some of their properties. Results and definitions from this section have already been obtained in other articles: They are taken from [BGP16a],[BGP16c]. The material of this section is however needed for what follows.

3.1 Generable functions

The following concept can be attributed to [Sha41]: a function $f : \mathbb{R} \to \mathbb{R}$ is said to be a PIVP function if there exists a system of the form (1) with $f(t) = y_1(t)$ for all t, where y_1 denotes the first component of the vector y defined in $\mathbb{R}^d$. In our proofs, we needed to extend Shannon's notion to talk about (i) multivariable functions and (ii) the growth of these functions. To this end, we introduced an extended class of generable functions in [BGP16b].

We will basically be interested with the case $\mathbb{K} = \mathbb{Q}$ in the following definition. However, for reasons explained in a few lines, we will need to consider larger fields $\mathbb{K}$.

Definition 5 (Polynomially bounded generable function) *Let $\mathbb{K}$ be a field. Let I be an open and connected subset of $\mathbb{R}^d$ and $f : I \to \mathbb{R}^e$. We say that $f \in \mathrm{GPVAL}_{\mathbb{K}}$ if and only if there exists a polynomial $\mathtt{sp} : \mathbb{R} \to \mathbb{R}_+$, $n \geqslant e$, a $n \times d$ matrix p consisting of polynomials with coefficients in $\mathbb{K}$, $x_0 \in \mathbb{K}^d \cap I$, $y_0 \in \mathbb{K}^n$ and $y : I \to \mathbb{R}^n$ satisfying for all $x \in I$:*

- $y(x_0) = y_0$ *and* $J_y(x) = p(y(x))$ ► *y satisfies a differential equation*[5]
- $f(x) = y_{1..e}(x)$ ► *f is a component of y*

[5] J_y denotes the Jacobian matrix of y.

- $\|y(x)\| \leqslant \mathtt{sp}(\|x\|)$ ► *y is polynomially bounded*

This class can be seen as an extended version of PIVPs. Indeed, when I is an interval, the Jacobian of y simply becomes the derivative of y and we get the solutions of $y' = p(y)$ where p is a vector of polynomials.

Note that, although functions in GPVAL$_{\mathbb{K}}$ can be viewed as solutions of partial differential equations (PDEs) (as we use a Jacobian), we will never have to deal with classical problems related to PDEs: PDEs have no general theory about the existence of solutions, etc. This comes from the way how we define functions in GPVAL$_{\mathbb{K}}$. Namely, in this paper, we will explictly present the functions in GPVAL$_{\mathbb{K}}$ which we will be used and we will show that they satisfy the conditions of Definition 5. Note also that it can be shown [BGP16a, Remark 15] that a solution to the PDE defined with the Jacobian is unique, because the condition $J_y(x) = p(y(x))$ is not general enough to capture the class of all PDEs. We also remark that, because a function in GPVAL$_{\mathbb{K}}$ must be polynomially bounded, it is defined everywhere on I.

A much more detailed discussion of this extension (which includes the results stated in this section) can be found in [BGP16b]. The key property of this extension is that it yields a much more stable class of functions than the original class considered in [Sha41]. In particular, we can add, subtract, multiply generable functions, and we can even do so while keeping them polynomially bounded.

Lemma 6 (Closure properties of GPVAL**)** *Let* $(f :\subseteq \mathbb{R}^d \to \mathbb{R}^n), (g :\subseteq \mathbb{R}^e \to \mathbb{R}^m) \in$ GPVAL$_{\mathbb{K}}$. *Then*[6] $f+g$, $f-g$, fg *are in* GPVAL$_{\mathbb{K}}$.

As we said, we are basically mostly interested by the case $\mathbb{K} = \mathbb{Q}$, but unfortunately, it turns out that GPVAL$_{\mathbb{Q}}$ is not closed by composition[7], while GPVAL$_{\mathbb{K}}$ is closed by composition for particular fields $\mathbb{K}$: An interesting case is when $\mathbb{K}$ is supposed to be a *generable field* as introduced in [BGP16b]. All the reader needs to know about generable fields is that they are fields and are stable by generable functions (introduced in Section 3.1). More precisely,

Proposition 7 (Generable field stability) *Let $\mathbb{K}$ be a generable field. If $\alpha \in \mathbb{K}$ and f is generable using coefficients in $\mathbb{K}$ (i.e. $f \in$ GPVAL$_{\mathbb{K}}$) then $f(\alpha) \in \mathbb{K}$.*

It is shown in [BGP16b] that there exists a smallest generable field $\mathbb{R}_G$ lying somewhere between $\mathbb{Q}$ and $\mathbb{R}_P$.

Lemma 8 (Closure properties of GPVAL**)** *Let $\mathbb{K}$ be a generable field. Let* $(f :\subseteq \mathbb{R}^d \to \mathbb{R}^n), (g :\subseteq \mathbb{R}^e \to \mathbb{R}^m) \in$ GPVAL$_{\mathbb{K}}$. *Then*[8] $f \circ g$ *in* GPVAL$_{\mathbb{K}}$.

As dealing with a class of functions closed by composition helps in many constructions, we will first reason assuming that $\mathbb{K}$ is a generable field with $\mathbb{R}_G \subseteq \mathbb{K} \subseteq \mathbb{R}_P$: From now on, $\mathbb{K}$ always denotes such a generable field, and we

[6]For matching dimensions of course.

[7]To observe that GPVAL$_{\mathbb{Q}}$ is not closed by composition, see for example that π is not rational and hence the constant function π does not belong to GPVAL$_{\mathbb{Q}}$. However it can be obtained from $\pi = 4\arctan 1$.

[8]For matching dimensions of course.

write GPVAL for $\mathrm{GPVAL}_{\mathbb{K}}$. We will later prove that non-rational coefficients can be eliminated in order to come back to the case $\mathbb{K} = \mathbb{Q}$. Up to Section 9 we allow coefficients in $\mathbb{K}$. Section 10 is devoted to prove than their can then be eliminated.

As $\mathbb{R}_P$ is generable, if this helps, the reader can consider that $\mathbb{K} = \mathbb{R}_P$ without any significant loss of generality.

Another crucial property of class GPVAL is that it is closed under solutions of ODE. In practice, this means that we can write differential equations of the form $y' = g(y)$ where g is generable, knowing that this can always be rewritten as a PIVP.

Lemma 9 (Closure by ODE of GPVAL**)** *Let $J \subseteq \mathbb{R}$ be an interval, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^d$ in* GPVAL*, $t_0 \in \mathbb{Q} \cap J$ and $y_0 \in \mathbb{Q}^d \cap \operatorname{dom} f$. Assume there exists $y : J \to \operatorname{dom} f$ and a polynomial $\mathtt{sp} : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying for all $t \in J$:*

$$y(t_0) = y_0 \qquad y'(t) = f(y(t)) \qquad \|y(t)\| \leqslant \mathtt{sp}(t)$$

Then y is unique and belongs to GPVAL*.*

The class GPVAL contains many classic polynomially bounded analytic[9] functions. For example, all polynomials belong to GPVAL, as well as sine and cosine. Mostly notably, the hyperbolic tangent (tanh) also belongs to GPVAL. This function appears very often in our constructions. Lemmas 6 and 9 are very useful to build new generable functions.

Functions from GPVAL are also known to have a polynomial modulus of continuity.

Proposition 10 (Modulus of continuity) *Let $f \in$ GPVAL with corresponding polynomial $\mathtt{sp} : \mathbb{R}_+ \to \mathbb{R}_+$. There exists $q \in \mathbb{K}[\mathbb{R}]$ such that for any $x_1, x_2 \in \operatorname{dom} f$, if $[x_1, x_2] \subseteq \operatorname{dom} f$ then $\|f(x_1) - f(x_2)\| \leqslant \|x_1 - x_2\| \, q(\mathtt{sp}(\max(\|x_1\|, \|x_2\|)))$. In particular, if $\operatorname{dom} f$ is convex then f has a polynomial modulus of continuity.*

3.2 Computable functions

In [BGP16c], we introduced several notions of computation based on polynomial differential equations extending the one introduced by [BCGH07] by adding a measure of complexity. The idea, illustrated in Figure 4 is to put the input value x as part of the initial condition of the system and to look at the asymptotic behavior of the system.

Our key insight to have a proper notion of complexity is to measure the *length* of the curve, instead of the time. Alternatively, a proper notion of complexity is achieved by considering both time *and* space, where space is defined as the maximum value of all components of the system.

Earlier attempts at defining a notion of complexity for the GPAC based on other notions failed because of time-scaling. Indeed, given a solution y of a PIVP, the function $z = y \circ \exp$ is also solution of a PIVP, but converges exponentially faster. A longer discussion on this topic can be found in [BGP16c]. In this section, we recall the main complexity classes and restate the main

[9] Functions from GPVAL are necessarily analytic, as solutions of an analytic ODE are analytic.

equivalence theorem. We denote by $\mathbb{K}[\mathbb{A}^n]$ the set of polynomial functions with n variables, coefficients in $\mathbb{K}$ and domain of definition $\mathbb{A}^n$.

The following definition is a generalization (to general length bound Π and field $\mathbb{K}$) of Definition 3: Following class ALP when $\mathbb{K} = \mathbb{Q}$, i.e. $\mathrm{ALP}_{\mathbb{Q}}$, corresponds of course to poly-length-computable functions (Definition 3).

Definition 11 (Analog Length Computability) *Let $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ and $\Pi : \mathbb{R}^2_+ \to \mathbb{R}_+$. We say that f is Π-length-computable if and only if there exist $d \in \mathbb{N}$, and $p \in \mathbb{K}^d[\mathbb{R}^d], q \in \mathbb{K}^d[\mathbb{R}^n]$ such that for any $x \in \operatorname{dom} f$, there exists (a unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ satisfying for all $t \in \mathbb{R}_+$:*

- *$y(0) = q(x)$ and $y'(t) = p(y(t))$* ► *y satisfies a PIVP*
- *for any $\mu \in \mathbb{R}_+$, if $\operatorname{len}_y(0,t) \geqslant \Pi(\|x\|, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$* ► *$y_{1..m}$ converges to $f(x)$*
- *$\|y'(t)\| \geqslant 1$* ► *technical condition: the length grows at least linearly with time*[10]

We denote by $\mathrm{ALC}(\Pi)$ the set of Π-length-computable functions, and by ALP the set of Π-length-computable functions where Π is a polynomial, and more generally by ALC the length-computable functions (for some Π). If we want to explicitly mention the set $\mathbb{K}$ of the coefficients, we write $\mathrm{ALC}_{\mathbb{K}}(\Pi)$, $\mathrm{ALP}_{\mathbb{K}}$ and $\mathrm{ALC}_{\mathbb{K}}$.

This notion of computation turns out to be equivalent to various other notions: The following equivalence result is proved in [BGP16c].

Proposition 12 (Main equivalence, [BGP16c]) *Let $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$. Then the following are equivalent for any generable field $\mathbb{K}$:*

1. *(illustrated by Figure 4) $f \in \mathrm{ALP}$;*
2. *(illustrated by Figure 6) There exist $d \in \mathbb{N}$, and $p, q \in \mathbb{K}^d[\mathbb{R}^n]$, polynomials $\Pi : \mathbb{R}^2_+ \to \mathbb{R}_+$ and $\Upsilon : \mathbb{R}^2_+ \to \mathbb{R}_+$ such that for any $x \in \operatorname{dom} f$, there exists (a unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ satisfying for all $t \in \mathbb{R}_+$:*
   - *$y(0) = q(x)$ and $y'(t) = p(y(t))$* ► *y satisfies a PIVP*
   - *$\forall \mu \in \mathbb{R}_+$, if $t \geqslant \Pi(\|x\|, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$* ► *$y_{1..m}$ converges to $f(x)$*
   - *$\|y(t)\| \leqslant \Upsilon(\|x\|, t)$* ► *$y$ is bounded*
3. *There exist $d \in \mathbb{N}$, and $p \in \mathbb{K}^d[\mathbb{R}^d], q \in \mathbb{K}^d[\mathbb{R}^{n+1}]$, and polynomial $\Pi : \mathbb{R}^2_+ \to \mathbb{R}_+$ and $\Upsilon : \mathbb{R}^3_+ \to \mathbb{R}_+$ such that for any $x \in \operatorname{dom} f$ and $\mu \in \mathbb{R}_+$, there exists (a unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ satisfying for all $t \in \mathbb{R}_+$:*
   - *$y(0) = q(x, \mu)$ and $y'(t) = p(y(t))$* ► *y satisfies a PIVP*

[10] This is a technical condition required for the proof. This can be weakened, for example to $\|p(y(t))\| \geqslant \frac{1}{\operatorname{poly}(t)}$. The technical issue is that if the speed of the system becomes extremely small, it might take an exponential time to reach a polynomial length, and we want to avoid such "unnatural" cases. This could be replaced by only assuming that we have somewhere the additional ordinary differential equation $y_0' = 1$.

- *if* $t \geqslant \Pi(\|x\|, \mu)$ *then* $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ ► $y_{1..m}$ *approximates* $f(x)$
- $\|y(t)\| \leqslant \Upsilon(\|x\|, \mu, t)$ ► *y is bounded*

4. *(illustrated by Figure 7) There exist* $\delta \geqslant 0$, $d \in \mathbb{N}$ *and* $p \in \mathbb{K}^d[\mathbb{R}^d \times \mathbb{R}^n]$, $y_0 \in \mathbb{K}^d$ *and polynomials* $\Upsilon, \Pi, \Lambda : \mathbb{R}_+^2 \to \mathbb{R}_+$, *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* $t \in \mathbb{R}_+$*:*
   - $y(0) = y_0$ *and* $y'(t) = p(y(t), x(t))$ ► *y satisfies a PIVP (with input)*
   - $\|y(t)\| \leqslant \Upsilon\big(\sup_\delta \|x\|(t), t\big)$ ► *y is bounded*
   - *for any* $I = [a, b] \subseteq \mathbb{R}_+$, *if there exist* $\bar{x} \in \operatorname{dom} f$ *and* $\bar{\mu} \geqslant 0$ *such that for all* $t \in I$, $\|x(t) - \bar{x}\| \leqslant e^{-\Lambda(\|\bar{x}\|, \bar{\mu})}$ *then* $\|y_{1..m}(u) - f(\bar{x})\| \leqslant e^{-\bar{\mu}}$ *whenever* $a + \Pi(\|\bar{x}\|, \bar{\mu}) \leqslant u \leqslant b$. ► *y converges to* $f(x)$ *when input* x *is stable*

5. *There exist* $\delta \geqslant 0$, $d \in \mathbb{N}$ *and* $(g : \mathbb{R}^d \times \mathbb{R}^{n+1} \to \mathbb{R}^d) \in \mathrm{GPVAL}_{\mathbb{K}}$ *and polynomials* $\Upsilon : \mathbb{R}_+^3 \to \mathbb{R}_+$ *and* $\Pi, \Lambda, \Theta : \mathbb{R}_+^2 \to \mathbb{R}_+$ *such that for any* $x \in C^0(\mathbb{R}_+, \mathbb{R}^n)$, $\mu \in C^0(\mathbb{R}_+, \mathbb{R}_+)$, $y_0 \in \mathbb{R}^d$, $e \in C^0(\mathbb{R}_+, \mathbb{R}^d)$ *there exists (a unique)* $y : \mathbb{R}_+ \to \mathbb{R}^d$ *satisfying for all* $t \in \mathbb{R}_+$*:*
   - $y(0) = y_0$ *and* $y'(t) = g(t, y(t), x(t), \mu(t)) + e(t)$
   - $\|y(t)\| \leqslant \Upsilon\left(\sup_\delta \|x\|(t), \sup_\delta \mu(t), \|y_0\| \mathbb{1}_{[1,\delta]}(t) + \int_{\max(0,t-\delta)}^t \|e(u)\| \, du\right)$
   - *For any* $I = [a, b]$, *if there exist* $\bar{x} \in \operatorname{dom} f$ *and* $\check{\mu}, \hat{\mu} \geqslant 0$ *such that for all* $t \in I$*:*

     $$\mu(t) \in [\check{\mu}, \hat{\mu}] \text{ and } \|x(t) - \bar{x}\| \leqslant e^{-\Lambda(\|\bar{x}\|, \hat{\mu})} \text{ and } \int_a^b \|e(u)\| \, du \leqslant e^{-\Theta(\|\bar{x}\|, \hat{\mu})}$$

     then

     $$\|y_{1..m}(u) - f(\bar{x})\| \leqslant e^{-\check{\mu}} \text{ whenever } a + \Pi(\|\bar{x}\|, \hat{\mu}) \leqslant u \leqslant b.$$

Note that (1) and (2) in the previous proposition are very closely related, and only differ in how the complexity is measured. In (1), based on length, we measure the length required to reach precision $e^{-\mu}$. In (2), based on time+space, we measure the time t required to reach precision $e^{-\mu}$ and the space (maximum value of all components) during the time interval $[0, t]$.

Item (3) in the previous proposition gives an apparently weaker form of computability where the system is no longer required to converge to $f(x)$ on input x. Instead, we give the system an input x and a precision μ, and ask that the system stabilizes within $e^{-\mu}$ of $f(x)$.

Item (4) in the previous proposition is a form of online-computability: the input is no longer part of the initial condition but rather given by an external input $x(t)$. The intuition is that if $x(t)$ approaches a value $\bar{x}$ sufficiently close, then by waiting long enough (and assuming that the external input stays near the value $\bar{x}$ during that time interval), we will get an approximation of $f(\bar{x})$ with some desired accuracy. This will be called online-computability.

Item (5) is a version robust with respect to perturbations. This notion will only be used in some proofs, and will be called extreme computability.

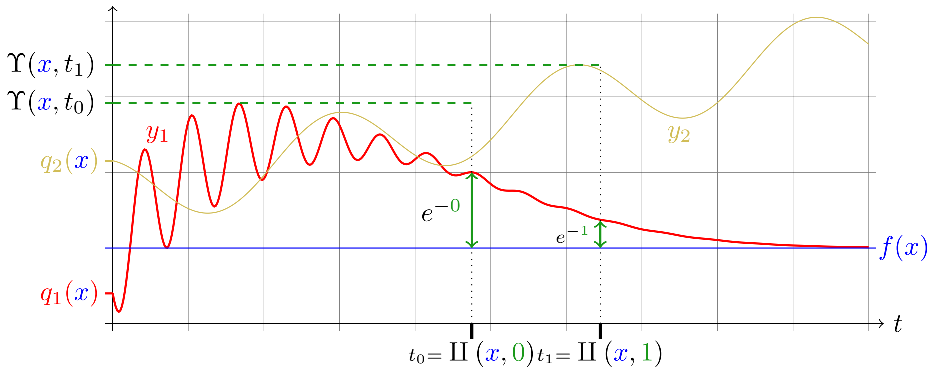


Figure 6: $f \in \mathrm{ATSC}(\Upsilon, \amalg)$: On input x, starting from initial condition $q(x)$, the PIVP $y' = p(y)$ ensures that $y_1(t)$ gives $f(x)$ with accuracy better than $e^{-\mu}$ as soon as the time t is greater than $\amalg(\|x\|, \mu)$. At the same time, all variables y_j are bounded by $\Upsilon(\|x\|, t)$. Note that the variables $y_2, \dots, y_d$ need not converge to anything.

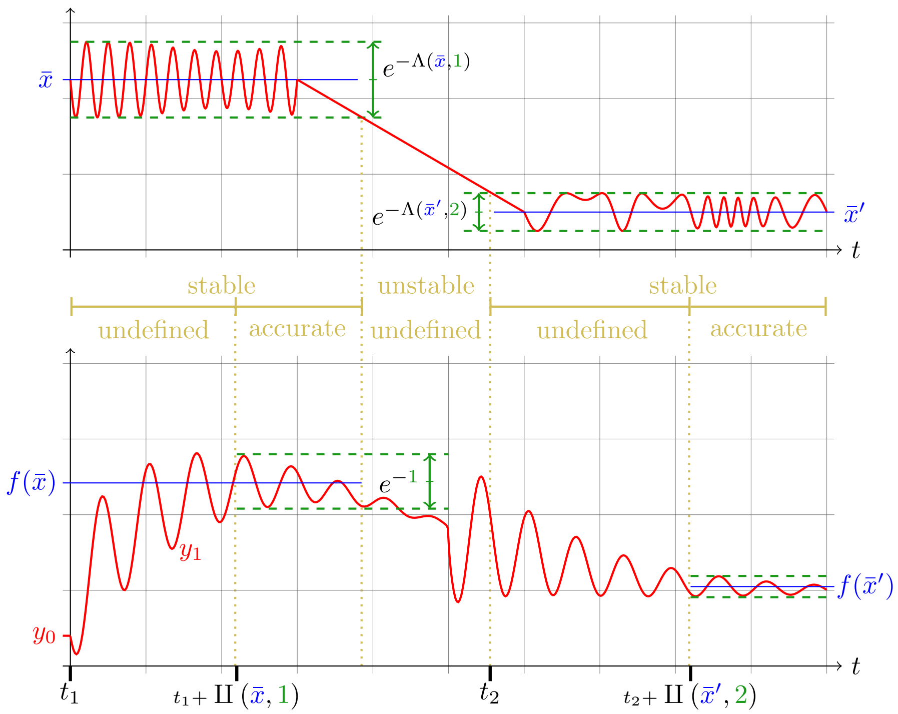


Figure 7: $f \in \mathrm{AOC}(\Upsilon, \amalg, \Lambda)$: starting from the (constant) initial condition y_0, the PIVP $y'(t) = p(y(t), x(t))$ has two possible behaviors depending on the input signal $x(t)$. If $x(t)$ is unstable, the behavior of the PIVP $y'(t) = p(y(t), x(t))$ is undefined. If $x(t)$ is stable around $\bar{x}$ with error at most $e^{-\Lambda(\|\bar{x}\|,\mu)}$ then $y(t)$ is initially undefined, but after a delay of at most $\amalg(\|\bar{x}\|, \mu)$, $y_1(t)$ gives $f(\bar{x})$ with accuracy better than $e^{-\mu}$. In all cases, all variables $y_j(t)$ are bounded by a function (Υ) of the time t and the supremum of $\|x(u)\|$ during a small time interval $u \in [t - \delta, t]$.

Remark 13 (Effective Limit computability) *A careful look at Item (3) of the previous Proposition shows that it corresponds to a form of effective limit computability. Formally, let* $f : I \times \mathbb{R}_+^* \to \mathbb{R}^n$*,* $g : I \to \mathbb{R}^n$ *and* $\mho : \mathbb{R}_+^2 \to \mathbb{R}_+$ *a polynomial. Assume that* $f \in \mathrm{ALP}$ *and that for any* $x \in I$ *and* $\tau \in \mathbb{R}_+^*$*, if* $\tau \geqslant \mho(\|x\|, \mu)$ *then* $\|f(x,\tau) - g(x)\| \leqslant e^{-\mu}$*. Then* $g \in \mathrm{ALP}$ *because the analog system for* f *satisfies all the items of the definition.*

Remark 14 (Comparison with classical complexity, unary and binary encodings) *Our notion of complexity is very similar to that of Computable Analysis over compact domains (indeed the goal of this paper is to show they are equivalent). However, there is a significant difference over unbounded domains: given* $x \in \mathbb{R}$*, we measure the complexity in terms of* $|x|$*, the absolute value of* x*, and the precision requested (*μ *in Definition 11). On the other hand, complexity in Computable Analysis is typically measured in terms of* $k \in \mathbb{N}$ *such that* $x \in [-2^k, 2^k]$*, and the precision requested. In particular, note that* $k \approx \log_2 |x|$*. A consequence of this fact is that the two frameworks have a different notion of "unary" and "binary":*

- *In the framework of Computable Analysis, the fractional part of the input/output (related to the precision) is measured "in unary": doubling the precision doubles the size and time allowed to compute. On the other hand, the integer part of input is measured "in binary": doubling the input only increases the size by* 1*. Note that this matches Turing complexity where we only deal with integers.*

- *In our framework, the fractional part of the input/output (related to the precision* μ*) is also measured "in unary": doubling the precision doubles the size and time allowed to compute. But the integer part of the input is also measured "in unary" because we measure* $|x|$*: doubling the input doubles the norm.*

This difference results in some surprising facts for the reader familiar with Computable Analysis:

- *The modulus of continuity is exponential in* $|x|$ *for polynomial-length computable functions, whereas it is pseudo-polynomial in Computable Analysis: see Theorem 22.*

- *A function* $f : \mathbb{N} \to \mathbb{N}$ *that is polynomial-length computable typically corresponds to a polynomial-time computable function* with unary encoding. *More generally, integer arguments in our framework should be considered as unary argument.*

For notational purpose, we will write $f \in \mathrm{ATSC}(\Upsilon, \Pi)$ when f satisfies (2) with corresponding polynomials Υ and Π, $f \in \mathrm{AWC}(\Upsilon, \Pi)$ when when f satisfies (3) with corresponding polynomials Υ and Π, $f \in \mathrm{AOC}(\Upsilon, \Pi, \Lambda)$ when f satisfies (4) with corresponding polynomials Υ, Π and Λ, and we will write $f \in \mathrm{AXC}(\Upsilon, \Pi, \Lambda, \Theta)$ when f satisfies (5) with corresponding polynomials Υ, Π, Λ, Θ.

3.3 Dynamics and encoding can be assumed generable

Before moving on to some basic properties of computable functions, we observe that a certain aspect of the definitions does not really matter: In Item (2) of Proposition 12, we required that p and q be polynomials. It turns out, surprisingly, that the class is the same if we only assume that $p, q \in \mathrm{GPVAL}$. This remark also applies to the Item (3). This turns out to be very useful when defining computable function.

Following proposition follows from Remark 26 of [BGP16c].

Remark 15 *Notice that this also holds for class* ALP*, even if not stated explicitely in [BGP16c]. Indeed, in Theorem 20 of [BGP16c] (*$\mathrm{ALP} = \mathrm{ATSP}$*), the inclusion* $\mathrm{ATSP} \subseteq \mathrm{ALP}$ *is trivial. Now, when proving that* $\mathrm{ALP} \subseteq \mathrm{ATSP}$*, the considered* p *and* g *could have been assumed generable without any difficulty.*

Proposition 16 (Polynomial versus generable) *Theorem 12 is still true if we only assume that* $p, q \in \mathrm{GPVAL}$ *in Item (2) or (3) (instead of* p, q *polynomials).*

We will use intensively this remark from now on. Actually, in several of the proofs, given a function from ALP, we will use the fact that it satisfies item (2) (the stronger notion) to build another function satifying item (3) with functions p and q in GPVAL (the weaker notion). From Proposition 12, this proves that the constructed function is indeed in ALP.

4 Some preliminary results

In this section, we present new and original results the exception being in subsection 4.4.3. First we relate generability to computability. Then, we prove some closure results for the class of computable functions. Then, we discuss their continuity and growth. Finaly, we prove that some basic functions such as min, max and absolute value, and rounding functions are in ALP.

4.1 Generable implies computable over star domains

We introduced the notion of GPAC generability and of GPAC computability. The latter can be considered as a generalization of the first, and as such, it may seem natural that any generable function must be computable: The intuition tells us that computing the value of f, a generable function, at point x is only a matter of finding a path *in the domain of definition* from the initial value x_0 to x, and simulating the differential equation along this path.

This however requires some discussions and hypotheses on the domain of definition of the function: We recall that a function is generable if it satisfies a PIVP over an open connected subset. We proved in [BGP16b] that there is always a path between x_0 to x and it can even be assumed to be generable.

Proposition 17 (Generable path connectedness) *An open, connected subset* U *of* $\mathbb{R}^n$ *is always* generable-path-connected*: for any* $a, b \in (U \cap \mathbb{K}^n)$*, there exists* $(\phi : \mathbb{R} \to U) \in \mathrm{GPVAL}_{\mathbb{K}}$ *such that* $\phi(0) = a$ *and* $\phi(1) = b$*.*

However, the proof is not constructive and we have no easy way of computing such a path given x.

For this reason, we restrict ourselves to the case where finding the path is trivial: star domains with a generable vantage point.

Definition 18 (Star domain) *A set $X \subseteq \mathbb{R}^n$ is called a* star domain *if there exists $x_0 \in X$ such that for all $x \in U$ the line segment from x_0 to x is in X, i.e $[x_0, x] \subseteq X$. Such an x_0 is called a* vantage point*.*

The following result is true, where a generable vantage point means a vantage point which belongs to a generable field. We will mostly need this theorem for domains of the form $\mathbb{R}^n \times \mathbb{R}_+^m$, which happen to be star domains.

Theorem 19 (GPVAL $\subseteq$ ALP over star domains) *If $f \in$ GPVAL has a star domain with a generable vantage point then $f \in$ ALP.*

Proof. Let $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in$ GPVAL and $z_0 \in \operatorname{dom} f \cap \mathbb{K}^n$ a generable vantage point. Apply Definition 5 to get $\mathtt{sp}, d, p, x_0, y_0$ and y. Since y is generable and $z_0 \in \mathbb{K}^d$, apply Proposition 7 to get that $y(z_0) \in \mathbb{K}^d$. Let $x \in \operatorname{dom} f$ and consider the following system:

$$\begin{cases} x(0) = x \\ \gamma(0) = x_0 \\ z(0) = y(z_0) \end{cases} \qquad \begin{cases} x'(t) = 0 \\ \gamma'(t) = x(t) - \gamma(t) \\ z'(t) = p(z(t))(x(t) - \gamma(t)) \end{cases}$$

First note that $x(t)$ is constant and check that $\gamma(t) = x + (x_0 - x)e^{-t}$ and note that $\gamma(\mathbb{R}_+) \subseteq [x_0, x] \subseteq \operatorname{dom} f$ because it is a star domain. Thus $z(t) = y(\gamma(t))$ since $\gamma'(t) = x(t) - \gamma(t)$ and $J_y = p$. It follows that $\|f(x) - z_{1..m}(t)\| = \|f(x) - f(\gamma(t))\|$ since $z_{1..m} = f$. Apply Proposition 10 to f to get a polynomial q such that

$$\forall x_1, x_2 \in \operatorname{dom} f, [x_1, x_2] \subseteq \operatorname{dom} f \ \Rightarrow \ \|f(x_1) - f(x_2)\| \leqslant \|x_1 - x_2\| \, q(\mathtt{sp}(\max(\|x_1\|, \|x_2\|))).$$

Since $\|\gamma(t)\| \leqslant \|x_0, x\|$ we have

$$\|f(x) - z_{1..m}(t)\| \leqslant \|x - x_0\| \, e^{-t} q(\|x_0, x\|) \leqslant e^{-t} \operatorname{poly}(\|x\|).$$

Finally, $\|z(t)\| \leqslant \mathtt{sp}(\gamma(t)) \leqslant \operatorname{poly}(\|x\|)$ because $\mathtt{sp}$ is a polynomial. Then, by Proposition 12, $f \in$ ALP.

■

4.2 Closure by arithmetic operations and composition

The class of polynomial time computable function is stable under addition, subtraction and multiplication, and composition.

Theorem 20 (Closure by arithmetic operations) *If $f, g \in$ ALP then $f \pm g, fg \in$ ALP, with the obvious restrictions on the domains of definition.*

Proof. We do the proof for the case of $f + g$ in detail. The other cases are similar. Apply Proposition 12 to get polynomials $\Pi, \Upsilon, \Pi^*, \Upsilon^*$ such that

$f \in \mathrm{ATSC}(\Upsilon, \Pi)$ and $g \in \mathrm{ATSC}(\Upsilon^*, \Pi^*)$ with corresponding d, p, q and d^*, p^*, q^* respectively. Let $x \in \operatorname{dom} f \cap \operatorname{dom} g$ and consider the following system:

$$\begin{cases} y(0)= q(x) \\ z(0)= q^*(x) \\ w(0)= q(x) + q^*(x) \end{cases} \qquad \begin{cases} y'(t)= p(y(t)) \\ z'(t)= p^*(z(t)) \\ w'(t)= p(y(t)) + p^*(z(t)) \end{cases} . \tag{2}$$

Notice that w was built so that $w(t) = y(t) + z(t)$. Let

$$\hat{\Pi}(\alpha, \mu) = \max(\Pi(\alpha, \mu + \ln 2), \Pi^*(\alpha, \mu + \ln 2))$$

and

$$\hat{\Upsilon}(\alpha, t) = \Upsilon(\alpha, t) + \Upsilon^*(\alpha, t).$$

Since, by construction, $w(t) = y(t)+z(t)$, if $t \geqslant \hat{\Pi}(\alpha, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu-\ln 2}$ and $\|z_{1..m}(t) - g(x)\| \leqslant e^{-\mu-\ln 2}$ thus $\|w_{1..m}(t) - f(x) - g(x)\| \leqslant e^{-\mu}$. Furthermore, $\|y(t)\| \leqslant \Upsilon(\|x\|, t)$ and $\|z(t)\| \leqslant \Upsilon^*(\|x\|, t)$ thus $\|w(t)\| \leqslant \hat{\Upsilon}(\|x\|, t)$.

The case of $f - g$ is exactly the same. The case of fg is slightly more involved. Since the standard product is defined on $\mathbb{R}$, the images of f and g are assumed to be on $\mathbb{R}$. Hence, instead of adding a vectorial w in (2), which could potentially have several components, we use a w composed by the single component given by

$$w'(t) = y_1'(t)z_1(t) + y_1(t)z_1'(t) = p_1(y(t))z_1(t) + y_1(t)p_1^*(z(t))$$

and $w(0) = q_1(x)q_1^*(x)$ so that $w(t) = y_1(t)z_1(t)$. The error analysis is a bit more complicated because the speed of convergence now depends on the length of the input.

First note that $\|f(x)\| \leqslant 1+\Upsilon(\|x\|, \Pi(\|x\|, 0))$ and $\|g(x)\| \leqslant 1+\Upsilon^*(\|x\|, \Pi^*(\|x\|, 0))$, and denote by $\ell(\|x\|)$ and $\ell^*(\|x\|)$ those two bounds respectively. If $t \geqslant \Pi(\|x\|, \mu + \ln 2\ell^*(\|x\|))$ then $\|y_1(t) - f(x)\| \leqslant e^{-\mu-\ln 2\|g(x)\|}$ and similarly if $t \geqslant \Pi^*(\|x\|, \mu + \ln 2(1 + \ell^*(\|x\|)))$ then $\|z_1(t) - g(x)\| \leqslant e^{-\mu-\ln 2(1+\|f(x)\|)}$. Thus for t greater than the maximum of both bounds,

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

because $\|y_1(t)\| \leqslant 1 + \|f(x)\| \leqslant 1 + \ell(\|x\|)$. ■

Recall that we assume we are working over a generable $\mathbb{K}$.

Theorem 21 (Closure by composition) *If $f, g \in \mathrm{ALP}$ and $f(\operatorname{dom} f) \subseteq \operatorname{dom} g$ then $g \circ f \in \mathrm{ALP}$.*

Proof. Let $f : I \subseteq \mathbb{R}^n \to J \subseteq \mathbb{R}^m$ and $g : J \to K \subseteq \mathbb{R}^l$. We will show that $g \circ f$ is computable by using the fact that g is online-computable. We could show directly that $g \circ f$ is online-computable but this would only complicate the proof for no apparent gain.

Apply Proposition 12 to get that $g \in \mathrm{AOC}(\Upsilon, \Pi, \Lambda)$ with corresponding r, Δ, z_0. Apply Proposition 12 to get that $f \in \mathrm{ATSC}(\Upsilon', \Pi')$ with corresponding d, p, q. Let $x \in I$ and consider the following system:

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

Define $v(t)=(x(t),y(t),z(t))$. Then it immediately follows that v satisfies a PIVP of the form $v(0)=\operatorname{poly}(x)$ and $v'(t)=\operatorname{poly}(v(t))$. Furthermore, by definition:

$$\begin{aligned}\|v(t)\| &= \max(\|x\|,\|y(t)\|,\|z(t)\|)\\ &\leqslant \max\left(\|x\|,\|y(t)\|,\Upsilon\left(\sup_{u\in[t,t-\Delta]\cap\mathbb{R}_+}\|y_{1..m}(t)\|,t\right)\right)\\ &\leqslant \operatorname{poly}\left(\|x\|,\sup_{u\in[t,t-\Delta]\cap\mathbb{R}_+}\|y(t)\|,t\right)\\ &\leqslant \operatorname{poly}\left(\|x\|,\sup_{u\in[t,t-\Delta]\cap\mathbb{R}_+}\Upsilon'(\|x\|,u),t\right)\\ &\leqslant \operatorname{poly}(\|x\|,t)\,.\end{aligned}$$

Define $\bar{x}=f(x)$, $\Upsilon^*(\alpha)=1+\Upsilon'(\alpha,0)$ and $\amalg''(\alpha,\mu)=\amalg'(\alpha,\Lambda(\Upsilon^*(\alpha),\mu))+\amalg(\Upsilon^*(\alpha),\mu)$. By definition of Υ', $\|\bar{x}\|\leqslant 1+\Upsilon'(\|x\|,0)=\Upsilon^*(\|x\|)$. Let $\mu\geqslant 0$ then by definition of $\amalg'$, if $t\geqslant\amalg'(\|x\|,\Lambda(\Upsilon^*(\|x\|),\mu))$ then $\|y_{1..m}(t)-\bar{x}\|\leqslant e^{-\Lambda(\Upsilon^*(\|x\|),\mu)}\leqslant e^{-\Lambda(\|\bar{x}\|,\mu)}$. For $a=\amalg'(\|x\|,\Lambda(\Upsilon^*(\|x\|),\mu))$ we get that $\|z_{1..l}(t)-g(f(x))\|\leqslant e^{-\mu}$ for any $t\geqslant a+\amalg(\bar{x},\mu)$. And since $t\geqslant a+\amalg(\bar{x},\mu)$ whenever $t\geqslant\amalg''(\|x\|,\mu)$, we get that $g\circ f\in\mathrm{ATSC}(\operatorname{poly},\amalg'')$. This concludes the proof because $\amalg''$ is a polynomial. ■

4.3 Continuity and growth

All computable functions are continuous. More importantly, they admit a polynomial modulus of continuity, in a similar spirit as in Computable Analysis.

Theorem 22 (Modulus of continuity) *If $f\in\mathrm{ALP}$ then f admits a polynomial modulus of continuity: there exists a polynomial $\mho:\mathbb{R}_+^2\to\mathbb{R}_+$ such that for all $x,y\in\operatorname{dom}f$ and $\mu\in\mathbb{R}_+$,*

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

In particular f is continuous.

Proof. Let $f\in\mathrm{ALP}$, apply Proposition 12 to get that $f\in\mathrm{AOC}(\Upsilon,\amalg,\Lambda)$ with corresponding δ,d,p and y_0. Without loss of generality, we assume polynomial $\amalg$ to be an increasing function. Let $u,v\in\operatorname{dom}f$ and $\mu\in\mathbb{R}_+$. Assume that $\|u-v\|\leqslant e^{-\Lambda(\|u\|+1,\mu+\ln 2)}$ and consider the following system:

$$y(0)=y_0\qquad y'(t)=p(y(t),u).$$

This is simply the online system where we hardwired the input of the system to the constant input u. The idea is that the definition of online computability can be applied to both u with 0 error, or v with error $\|u-v\|$.

By definition, $\|y_{1..m}(t)-f(u)\|\leqslant e^{-\mu-\ln 2}$ for all $t\geqslant\amalg(\|u\|,\mu+\ln 2)$. For the same reason, $\|y_{1..m}(t)-f(v)\|\leqslant e^{-\mu-\ln 2}$ for all $t\geqslant\amalg(\|v\|,\mu+\ln 2)$ because $\|u-v\|\leqslant e^{-\Lambda(\|u\|+1,\mu+ln2)}\leqslant e^{-\Lambda(\|v\|,\mu+\ln 2)}$ and $\|v\|\leqslant\|u\|+1$. Combine both results at $t=\amalg(\|u\|+1,\mu+\ln 2)$ to get that $\|f(u)-f(v)\|\leqslant e^{-\mu}$. ■

It is is worth observing that all functions in ALP are polynomially bounded (this follows trivially from condition (2) of Proposition 12).

Proposition 23 *Let $f \in \mathrm{ALP}$, there exists a polynomial P such that $\|f(x)\| \leqslant P(\|x\|)$ for all $x \in \operatorname{dom} f$.*

4.4 Some basic functions proved to be in ALP

4.4.1 Absolute, minimum, maximum value

We will now show that basic functions like the absolute value, the minimum and maximum value are computable. We will also show a powerful result when limiting a function to a computable range. In essence all these result follow from the fact that the absolute value belongs to ALP, which is a surprisingly non-trivial result (see the example below).

Example 24 (Broken way of computing the absolute value) *Computing the absolute value in polynomial length, or equivalently in polynomial time with polynomial bounds, is a surprisingly difficult operation, for unintuitive reasons. This example illustrates the problem. A natural idea to compute the absolute value is to notice that $|x| = x\operatorname{sgn}(x)$, where $\operatorname{sgn}(x)$ denotes the sign function (with conventionally $\operatorname{sgn}(0) = 0$). To this end, define $f(x,t) = x\tanh(xt)$ which works because $\tanh(xt) \to \operatorname{sgn}(x)$ when $t \to \infty$. Unfortunately, $\big||x| - f(x,t)\big| \sim \frac{1}{2}|x|e^{-2|x|t}$ as $t \to \infty$, which* ***converges very slowly for small*** *x. Indeed, if $x = e^{-\alpha}$ then $\big||x|-f(x,t)\big| \sim \frac{1}{2}e^{-\alpha-2e^{-\alpha}t}$ as $t \to \infty$, so we must take $t(\mu) = e^{\alpha}\mu$ to reach a precision of $e^{-\mu}$. This is unacceptable because it grows as $\frac{1}{|x|}$ instead of $|x|$. In particular, it is unbounded when $x \to 0$ which is clearly wrong.*

The sign function is not computable because it not continuous. However, if f is a continuous function that is zero at 0 then $\operatorname{sgn}(x)f(x)$ is continuous, and polynomial length computable under some conditions that we explore below. This simple remark is quite powerful because some continuous functions can be easily put in the form $\operatorname{sgn}(x)f(x)$. For example, the absolute value corresponds to $f(x) = x$.

The proof is not difficult but the idea is not very intuitive. As the example above outlines, the naive idea of computing $x\tanh(xt)$ and hope that it converges quickly enough when $t \to \infty$ does not work because the convergence speed is too slow for small x. However if we could somehow compute $x\tanh(xe^t)$, our problem would be solved. To understand why, write $x = e^{-\alpha}$ and consider the following two phases. For $t \leqslant \alpha$, $|x\tanh(xe^t) - |x|| \leqslant |x| \leqslant e^{-\alpha} \leqslant e^{-t}$, in other words $|x|$ is so small that any value in $[0,|x|]$ is a good approximation. For $t \geqslant \alpha$, use $|\operatorname{sgn}(u) - \tanh(u)| \leqslant e^{-|u|}$ to get that $||x| - x\tanh(xe^t)| \leqslant |x|e^{-xe^t} \leqslant e^{-\alpha-e^{t-\alpha}} \leqslant e^{-\alpha-t+\alpha+1} \leqslant e^{1-t}$.

Unfortunately, we cannot compute xe^t in polynomial length, but we can work around it by noticing that we do not really need to compute xe^t but rather $s(x,t)$ such that $s(x,t) \approx xe^t$ for small t, and $s(x,t) \approx t$ for large t. To do so, we use the following differential equation:

$$s(0) = x, \qquad s'(t) = \tanh(s(t)).$$

Note that since tanh is bounded by 1, $|s(t)| \leqslant |x| + t$ thus it is bounded by a polynomial in x and t. However, note that if $s(t) \approx 0$ then $\tanh(s(t)) \approx s(t)$ thus the differential equation becomes $s'(t) \approx s(t)$, *i.e.* $s(t) \approx xe^t$ which remains a valid approximation as long as $s(t) \ll 1$. Thus at $t \approx \ln \frac{1}{|x|}$, we have $s(t) \approx \operatorname{sgn}(x)$ and then $s(t) \propto t \operatorname{sgn}(x)$ for $t \gg \ln \frac{1}{|x|}$.

The following lemma uses those ideas and generalizes them to compute $(x, z) \mapsto \operatorname{sgn}(x)z$. In order to generalize the result to two variables, we need to add some constraint on the domain of definition: z needs to be small enough relative to x to give the system enough time for $\tanh(xe^t)$ to be a good approximation for $\operatorname{sgn}(x)$. Using a similar reasoning as above, we want $|\operatorname{sgn}(x)z - z\tanh(xe^t)| \leqslant e^{p(\|x,z\|)-t}$ for $t \geqslant -\ln|z|$ for some polynomial p. We leave the details of the computation to the reader.

Proposition 25 (Smooth sign is computable) *For any polynomial* $p : \mathbb{R}_+ \to \mathbb{R}_+$, $H_p \in \mathrm{ALP}$ *where*

$$H_p(x,z) = \operatorname{sgn}(x)z \quad \textit{for all} \quad (x,z) \in U_p := \big\{(0,0)\big\} \cup \Big\{(x,z) \in \mathbb{R}^* \times \mathbb{R} : \left|\tfrac{z}{x}\right| \leqslant e^{p(\|x,z\|)}\Big\}.$$

Proof. Let $(x, z) \in U$ and consider the following system:

$$\begin{cases} s(0) = x \\ y(0) = z\tanh(x) \end{cases} \qquad \begin{cases} s'(t) = \tanh(s(t)) \\ y'(t) = \left(1 - \tanh(s(t))^2\right) y(t) \end{cases}$$

First check that $y(t) = z\tanh(s(t))$. The case of $x = 0$ is trivial because $s(t) = 0$ and $y(t) = 0 = H(x, z)$. If $x < 0$ then check that the same system for $-x$ has the opposite value for s and y so all the convergence result will the exactly the same and will be correct because $H(x, z) = -H(-x, z)$. Thus we can assume that $x > 0$. We will need the following elementary property of the hyperbolic tangent for all $t \in \mathbb{R}$:

$$1 - \operatorname{sgn}(t)\tanh(t) \leqslant e^{-|t|}.$$

Apply the above formula to get that $1 - e^{-u} \leqslant \tanh(u) \leqslant 1$ for all $u \in \mathbb{R}_+$. Thus $\tanh(s(t)) \geqslant 1 - e^{-s(t)}$ and by a classical result of differential inequalities, $s(t) \geqslant w(t)$ where $w(0) = s(0) = x$ and $w'(t) = 1 - e^{-w(t)}$. Check that $w(t) = \ln\left(1 + (e^x - 1)e^t\right)$ and conclude that

$$|z - y(t)| \leqslant |z|(1 - \tanh(s(t))| \leqslant |z|e^{-s(t)} \leqslant \frac{|z|}{1 + (e^x - 1)e^t} \leqslant \frac{|z|e^{-t}}{e^x - 1} \leqslant \frac{|z|}{x}e^{-t} \leqslant e^{p(\|x,z\|)-t}.$$

Thus $|z - y(t)| \leqslant e^{-\mu}$ for all $t \geqslant \mu + p(\|x, z\|)$ which is polynomial in $\|x, z, \mu\|$. Furthermore, $|s(t)| \leqslant |x| + t$ because $|s'(t)| \leqslant 1$. Similarly, $|y(t)| \leqslant |z|$ so the system is polynomially bounded. Finally, the system is of the form $(s, y)(0) = f(x)$ and $(s, y)'(t) = g((s, y)(t))$ where $f, g \in \mathrm{GPVAL}$ so $H_p \in \mathrm{ALP}$ with generable functions. Apply Proposition 16 to conclude. ■

Theorem 26 (Absolute value is computable) $(x \mapsto |x|) \in \mathrm{ALP}$.

Proof. Let $p(x) = 0$ which is a polynomial, and $a(x) = H_p(x, x)$ where $H_p \in \mathrm{ALP}$ comes from Proposition 25. It is not hard to see that a is defined over $\mathbb{R}$ because $(0, 0) \in U_p$ and for any $x \neq 0$, $\left|\frac{x}{x}\right| \leqslant 1 = e^{p(|x|)}$ thus $(x, x) \in U_p$. Consequently $a \in \mathrm{ALP}$ and for any $x \in \mathbb{R}$, $a(x) = \operatorname{sgn}(x)x = |x|$ which concludes. ■

Corollary 27 (Max, Min are computable) $\max, \min \in \mathrm{ALP}$.

Proof. Use that $\max(a, b) = \frac{a+b}{2} + \left|\frac{a+b}{2}\right|$ and $\min(a, b) = -\max(-a, -b)$. Conclude with Theorem 26 and closure by arithmetic operations and composition of ALP. ■

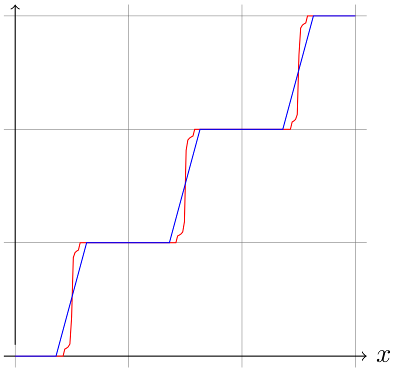


Figure 8: Graph of rnd and rnd*.

4.4.2 Rounding

In [BGP16a] we showed that it is possible to build a generable rounding function rnd of very good quality. See Figure 8 for an illustration.

Lemma 28 (Round) *There exists* $\mathrm{rnd} \in \mathrm{GPVAL}$ *such that for any* $n \in \mathbb{Z}$, $\lambda \geqslant 2$, $\mu \geqslant 0$ *and* $x \in \mathbb{R}$ *we have:*

- $|\,\mathrm{rnd}(x,\mu,\lambda) - n| \leqslant \frac{1}{2}$ *if* $x \in \left[n - \frac{1}{2}, n + \frac{1}{2}\right]$,
- $|\,\mathrm{rnd}(x,\mu,\lambda) - n| \leqslant e^{-\mu}$ *if* $x \in \left[n - \frac{1}{2} + \frac{1}{\lambda}, n + \frac{1}{2} - \frac{1}{\lambda}\right]$.

In this section, we will see that we can do even better with computable functions. More precisely, we will build a computable function rnd* that rounds perfectly everywhere, except on a small, periodic, interval of length $e^{-\mu}$ where μ is a parameter. This is the best can do because of the continuity and modulus of continuity requirements of computable functions, as shown in Theorem 22. We will need a few technical lemmas before getting to the rounding function itself. We start by a small remark that will be useful later on. See Figure 8 for an illustration of rnd*.

Remark 29 (Constant function) *Let* $f \in \mathrm{ALP}$, I *a convex subset of* $\operatorname{dom} f$ *and assume that* f *is constant over* I, *with value* α. *From Proposition 12, we have* $f \in \mathrm{AOC}(\Upsilon, \amalg, \Lambda)$ *for some polynomials* $\Upsilon, \amalg, \Lambda$ *with corresponding* d, δ, p *and* y_0. *Let* $x \in C^0(\mathbb{R}_+, \operatorname{dom} f)$ *and consider the system:*

$$y(0) = y_0 \qquad y'(t) = p(y(t), x(t))$$

If there exists $J = [a,b]$ *and* M *such that for all* $x(t) \in I$ *and* $\|x(t)\| \leqslant M$ *for all* $t \in J$, *then* $\|y_{1..m}(t) - \alpha\| \leqslant e^{-\mu}$ *for all* $t \in [a + \amalg(M,\mu), b]$. *This is unlike the case where the input must be nearly constant and it is true because whatever the system can sample from the input* $x(t)$, *the resulting output will be the same. Formally, it can shown by building a small system around the online-system that samples the input, even if it unstable.*

Proposition 30 (Clamped exponential) *For any* $a, b, c, d \in \mathbb{K}$ *and* $x \in \mathbb{R}$ *such that* $a \leqslant b$, *define* h *as follows. Then* $h \in \mathrm{ALP}$*:*

$$h(a,b,c,d,x) = \max(a, \min(b, ce^x + d)).$$

Proof. First note that we can assume that $d = 0$ because $h(a, b, c, d, x) = h(a-d, b-d, c, 0, x)+d$. Similarly, we can assume that $a = -b$ and $b \geqslant |c|$ because $h(a, b, c, d, x) = \max(a, \min(b, h(-|c| - \max(|a|, |b|), |c| + \max(|a|, |b|), c, d, x)))$ and $\min, \max, |\cdot| \in \text{ALP}$. So we are left with $H(\ell, c, x) = \max(-\ell, \min(\ell, ce^x))$ where $\ell \geqslant |c|$ and $x \in \mathbb{R}$. Furthermore, we can assume that $c \geqslant 0$ because $H(\ell, c, x) = \operatorname{sgn}(c)H(\ell, |c|, x)$ and it belongs to ALP for all $\ell \geqslant |c|$ and $x \in \mathbb{R}$ thanks to Proposition 25. Indeed, if $c = 0$ then $H(\ell, |c|, x) = 0$ and if $c \neq 0$, $\ell \geqslant |c|$ and $x \in \mathbb{R}$, then $\left|\frac{c}{H(\ell,|c|,x)}\right| \geqslant e^{-|x|}$.

We will show that $H \in \text{ALP}$.Let $\ell \geqslant c \geqslant 0$, $\mu \in \mathbb{R}_+$, $x \in \mathbb{R}$ and consider the following system:

$$\begin{cases} y(0) = c \\ z(0) = 0 \end{cases} \qquad \begin{cases} y'(t) = z'(t)y(t) \\ z'(t) = (1 + \ell - y(t))(x - z(t)) \end{cases}$$

Note that formally, we should add extra variables to hold x, μ and ℓ (the inputs). Also note that to make this a PIVP, we should replace $z'(t)$ by its expression in the right-hand side, but we kept $z'(t)$ to make things more readable. By construction $y(t) = ce^{z(t)}$, and since $\ell \geqslant c \geqslant 0$, by a classical differential argument, $z(t) \in [0, x]$ and $y(t) \in [0, \min(ce^x, \ell + 1)]$. This shows in particular that the system is polynomially bounded in $\|\ell, x, c\|$. There are two cases to consider.

- If $\ell \geqslant ce^x$ then $\ell - y(t) = \ell - ce^{z(t)} \geqslant c(e^x - e^{z(t)}) \geqslant c(x - z(t)) \geqslant 0$ thus by a classical differential inequalities reasoning, $z(t) \geqslant w(t)$ where w satisfies $w(0) = 0$ and $w'(t) = (x - w(t))$. This system can be solved exactly and $w(t) = x(1 - e^{-t})$. Thus

  $$y(t) \geqslant ce^{w(t)} \geqslant ce^x e^{-xe^{-t}} \geqslant ce^x(1 - xe^{-t}) \geqslant ce^x - cxe^{x-t}.$$

  So if $t \geqslant \mu + x + c$ then $y(t) \geqslant ce^x - e^{-\mu}$. Since $y(t) \leqslant ce^x$ it shows that $|y(t) - ce^x| \leqslant e^{-\mu}$.

- If $\ell \leqslant ce^x$ then by the above reasoning, $\ell + 1 \geqslant y(t) \geqslant \ell$ when $t \geqslant \mu + x + c$.

We will modify this sytem to feed y to an online-system computing $\min(-\ell, \max(\ell, \cdot))$. The idea is that when $y(t) \geqslant \ell$, this online-system is constant so the input does not need to be stable.

Let $G(x) = \min(\ell, x)$ then $G \in \text{AOC}(\Upsilon, \Pi, \Lambda)$ with polynomials Λ, Π, Υ are polynomials and corresponding d, δ, p and y_0. Let x, c, ℓ, μ and consider the following system (where y and z are from the previous system):

$$w(0) = y_0 \qquad w'(t) = p(w(t), y(t))$$

Again, there are two cases.

- If $\ell \geqslant ce^x$ then $|y(t) - ce^x| \leqslant e^{-\Lambda(\ell,\mu)} \leqslant e^{-\Lambda(ce^x,\mu)}$ when $t \geqslant \Lambda(\ell, \mu) + x + c$, thus $|w_1(t) - G(ce^x)| \leqslant e^{-\mu}$ when $t \geqslant \Lambda(\ell, \mu) + x + c + \Pi(\ell, \mu)$ and this concludes because $G(ce^x) = ce^x$.

- If $\ell \leqslant ce^x$ then by the above reasoning, $\ell + 1 \geqslant y(t) \geqslant \ell$ when $t \geqslant \Lambda(\ell, \mu) + x + c$ and thus $|w_1(t) - \ell| \leqslant e^{-\mu}$ when $t \geqslant \Lambda(\ell, \mu) + x + c + \Pi(\ell, \mu)$ by Remark 29 because $G(x) = \ell$ for all $x \geqslant \ell$.

To conclude the proof that $H \in \text{ALP}$,note that w is also polynomially bounded . ■

Definition 31 (Round) *Let* $\text{rnd}^* \in C^0(\mathbb{R}, \mathbb{R})$ *be the unique function such that:*

- $\text{rnd}^*(x, \mu) = n$ *for all* $x \in \left[n - \frac{1}{2} + e^{-\mu}, n + \frac{1}{2} - e^{-\mu}\right]$ *for all* $n \in \mathbb{Z}$
- $\text{rnd}^*(x, \mu)$ *is affine over* $\left[n + \frac{1}{2} - e^{-\mu}, n + \frac{1}{2} + e^{-\mu}\right]$ *for all* $n \in \mathbb{Z}$

Theorem 32 (Round) $\text{rnd}^* \in \text{ALP}$.

Proof. The idea of the proof is to build a function computing the "fractional part" function, by this we mean a 1-periodic function that maps x to x over $[-1 + e^{-\mu}, 1 - e^{-\mu}]$ and is affine at the border to be continuous. The rounding function immediately follows by subtracting the fractional of x to x. Although the idea behind this construction is simple, the details are not so immediate. The intuition is that $\frac{1}{2\pi}\arccos(\cos(2\pi x))$ works well over $[0, 1/2 - e^{-\mu}]$ but needs to be fixed at the border (near 1/2), and also its parity needs to be fixed based on the sign of $\sin(2\pi x)$.

Formally, define for $c \in [-1, 1]$, $x \in \mathbb{R}$ and $\mu \in \mathbb{R}_+$:

$$g(c, \mu) = \max(0, \min((1 - \tfrac{e^{\mu}}{2})(\arccos(c) - \pi), \arccos(c))),$$

$$f(x, \mu) = \frac{1}{2\pi}\,\text{sgn}(\sin(2\pi x))g(\cos(2\pi x), \mu).$$

Remark that $g \in \text{ALP}$ because of Theorem 30 and that $\arccos \in \text{ALP}$ because $\arccos \in \text{GPVAL}$. Then $f \in \text{ALP}$ by Proposition 25. Indeed, if $\sin(2\pi x) = 0$ then $g(\cos(2\pi x), \mu) = 0$ and if $\sin(2\pi x) \neq 0$, a tedious computation shows that $\left|\frac{g(\cos(2\pi x), \mu)}{\sin(2\pi x)}\right| = \min\left((1 - \frac{e^{\mu}}{2})\frac{\arccos(\cos(2\pi x)) - \pi}{\sin(2\pi x)}, \frac{\arccos(\cos(2\pi x))}{\sin(2\pi x)}\right) \leqslant 2\pi e^{\mu}$ because $g(\cos(2\pi x), \mu)$ is piecewise affine with slope e^{μ} at most (see below for more details).

Note that f is 1-periodic because of the sine and cosine so we only need to analyze if over $[-\frac{1}{2}, \frac{1}{2}]$, and since f is an odd function, we only need to analyze it over $[0, \frac{1}{2}]$. Let $x \in [0, \frac{1}{2}]$ and $\mu \in \mathbb{R}_+$ then $2\pi x \in [0, \pi]$ thus $\arccos(\cos(2\pi x)) = 2\pi x$ and $f(x, \mu) = \min((1 - \frac{e^{\mu}}{2})(x - \frac{1}{2}), \frac{x}{2\pi})$. There are two cases.

- If $x \in [0, \frac{1}{2} - e^{-\mu}]$ then $x - \frac{1}{2} \leqslant -e^{-\mu}$ thus $(1 - \frac{e^{\mu}}{2})(x - \frac{1}{2}) \geqslant \frac{1}{2} - e^{-\mu} \geqslant \frac{x}{2\pi}$ so $f(x, \mu) = x$.
- If $x \in [\frac{1}{2} - e^{-\mu}, \frac{1}{2}]$ then $0 \geqslant x - \frac{1}{2} \geqslant -e^{-\mu}$ thus $(1 - \frac{e^{\mu}}{2})(x - \frac{1}{2}) \leqslant \frac{1}{2} - e^{-\mu} \leqslant \frac{x}{2\pi}$ so $f(x, \mu) = (1 - \frac{e^{\mu}}{2})(x - \frac{1}{2})$ which is affine.

Finally define $\text{rnd}^*(x, \mu) = x - f(x, \mu)$ to get the desired function. ■

4.4.3 Some functions considered elsewhere: Norm, and Bump functions

The following functions have already been considered in some other articles, and proved to be in GPVAL (and hence in ALP).

A useful function when dealing with error bound is the norm function. Although it would be possible to build a very good infinity norm, in practice we will only need a constant *overapproximation* of it. The following results can be found in [BGP16a, Lemma 44 and 46].

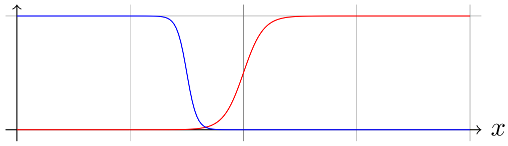


Figure 9: Graph of $\mathrm{lxh}_{[1,3]}$ and $\mathrm{hxl}_{[1,2]}$

Lemma 33 (Norm function) *For every $\delta \in]0,1]$, there exists $\mathrm{norm}_{\infty,\delta} \in \mathrm{GPVAL}$ such that for any $x \in \mathbb{R}^n$ we have*

$$\|x\| \leqslant \mathrm{norm}_{\infty,\delta}(x) \leqslant \|x\| + \delta.$$

A crucial function when simulating computation is a "step" or "bump" function. Unfortunately, for continuity reasons, it is again impossible to build a perfect one but we can achieve a good accuracy except on a small transition interval. Figure 9 illustrates both functions.

Lemma 34 ("low-X-high" and "high-X-low") *For every $I = [a,b]$, $a,b \in \mathbb{K}$, there exists $\mathrm{lxh}_I, \mathrm{hxl}_I \in \mathrm{GPVAL}$ such that for every $\mu \in \mathbb{R}_+$ and $t,x \in \mathbb{R}$ we have:*

- lxh_I *is of the form* $\mathrm{lxh}_I(t,\mu,x) = \phi_1(t,\mu,x)x$ *where* $\phi_1 \in \mathrm{GPVAL}$*,*
- hxl_I *is of the form* $\mathrm{lxh}_I(t,\mu,x) = \phi_2(t,\mu,x)x$ *where* $\phi_2 \in \mathrm{GPVAL}$*,*
- *if* $t \leqslant a$, $|\,\mathrm{lxh}_I(t,\mu,x)| \leqslant e^{-\mu}$ *and* $|x - \mathrm{hxl}_I(t,\mu,x)| \leqslant e^{-\mu}$*,*
- *if* $t \geqslant b$, $|x - \mathrm{lxh}_I(t,\mu,x)| \leqslant e^{-\mu}$ *and* $|\,\mathrm{hxl}_I(t,\mu,x)| \leqslant e^{-\mu}$*,*
- *in all cases,* $|\,\mathrm{lxh}_I(t,\mu,x)| \leqslant |x|$ *and* $|\,\mathrm{hxl}_I(t,\mu,x)| \leqslant |x|$*.*

5 Encoding The Step Function of a Turing machine

In this section, we will show how to encode and simulate one step of a Turing machine with a computable function in a robust way. The empty word will be denoted by λ. We define the integer part function $\mathrm{int}(x)$ by $\max(0, \lfloor x \rfloor)$ and the fractional part function $\mathrm{frac}(x)$ by $x - \mathrm{int}\, x$. We also denote by $\#S$ the cardinal of a finite set S.

5.1 Turing Machine

There are many possible definitions of Turing machines. The exact kind we pick is usually not important but since we are going to simulate one with differential equations, it is important to specify all the details of the model. We will simulate deterministic, one-tape Turing machines, with complete transition functions.

Definition 35 (Turing Machine) *A* Turing Machine *is a tuple $\mathcal{M} = (Q, \Sigma, b, \delta, q_0, q_\infty)$ where $Q = [\![0, m-1]\!]$ are the states of the machines, $\Sigma = [\![0, k-2]\!]$ is the alphabet and $b = 0$ is the blank symbol, $q_0 \in Q$ 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* $L = -1$, $S = 0$ *and* $R = 1$. *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* $\delta(q_\infty, \sigma) = (q_\infty, \sigma, S)$.

Remark 36 (Choice of k) *The choice of* $\Sigma = [\![0, k-2]\!]$ *will be crucial for the simulation, to ensure that the transition function is continuous. See Lemma 45.*

For completeness, and also to make the statements of the next theorems easier, we introduce the notion of configuration of a machine, and define one step of a machine on configurations. This allows us to define the result of a computation. Since we will characterize FP, our machines not only accept or reject a word, but compute an output word.

Definition 37 (Configuration) *A* configuration *of* $\mathcal{M}$ *is a tuple* $c = (x, \sigma, y, q)$ *where* $x \in \Sigma^*$ *is the part of the tape at left of the head,* $y \in \Sigma^*$ *is the part at the right,* $\sigma \in \Sigma$ *is the symbol under the head and* $q \in Q$ *the current state. More precisely* x_1 *is the symbol immediately at the left of the head and* y_1 *the symbol immediately at the right. See Figure 10 for a graphical representation. The set of configurations of* $\mathcal{M}$ *is denoted by* $\mathcal{C}_\mathcal{M}$. *The* initial configuration *is defined by* $c_0(w) = (\lambda, b, w, q_0)$ *and the* final configuration *by* $c_\infty(w) = (\lambda, b, w, q_\infty)$ *where* λ *is the empty word.*

Definition 38 (Step) *The* step *function of a Turing machine* $\mathcal{M}$ *is the function, acting on configurations, denoted by* $\mathcal{M}$ *and defined by:*

$$\mathcal{M}(x, \sigma, y, q) = \begin{cases} (\lambda, b, \sigma' y, q') & \text{if } d = L \text{ and } x = \lambda \\ (x_{2..|x|}, x_1, \sigma' y, q') & \text{if } d = L \text{ and } x \neq \lambda \\ (x, \sigma', y, q') & \text{if } d = S \\ (\sigma' x, b, \lambda, q') & \text{if } d = R \text{ and } y = \lambda \\ (\sigma' x, y_1, y_{2..|y|}, q') & \text{if } d = R \text{ and } y \neq \lambda \end{cases} \quad \text{where } \begin{cases} q' = \delta_1(q, \sigma) \\ \sigma' = \delta_2(q, \sigma) \\ d = \delta_3(q, \sigma) \end{cases}.$$

Definition 39 (Result of a computation) *The* result of a computation *of* $\mathcal{M}$ *on a word* $w \in \Sigma^*$ *is defined by:*

$$\mathcal{M}(w) = \begin{cases} x & \text{if } \exists n \in \mathbb{N}, \mathcal{M}^{[n]}(c_0(w)) = c_\infty(x) \\ \perp & \text{otherwise} \end{cases}$$

Remark 40 *The result of a computation is well-defined because we imposed that when a machine reaches a halting state, it does not move, change state or change the symbol under the head.*

5.2 Finite set interpolation

In order to implement the transition function of the Turing Machine, we will use an interpolation scheme.

Lemma 41 (Finite set interpolation) *For any finite* $G \subseteq \mathbb{K}^d$ *and* $f : G \to \mathbb{K}$, *there exists* $\mathbb{1}_f \in \mathrm{ALP}$ *with* $\mathbb{1}_f\restriction_G = f$, *where* $\mathbb{1}_f\restriction_G$ *denotes the restriction of* $\mathbb{1}_f$ *to* G.

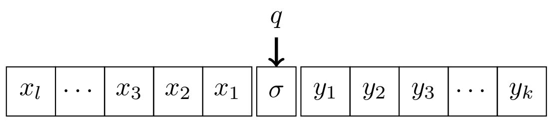


Figure 10: Example of generic configuration $c = (x, \sigma, y, q)$

Proof. For $d = 1$, consider for example Lagrange polynomial

$$\mathbb{1}_f(x) = \sum_{\bar{x} \in G} f(\bar{x}) \prod_{\substack{y \in G \\ y \neq \bar{x}}} \prod_{i=1}^{d} \frac{x_i - y_i}{\bar{x}_i - y_i}.$$

The fact that $\mathbb{1}_f$ matches f on G is a classical calculation. Also $\mathbb{1}_f$ is a polynomial with coefficients in $\mathbb{K}$ so clearly it belongs to ALP. The generalization to $d > 1$ is clear, but tedious to be fully detailed so we leave it to the reader. ■

It is customary to prove robustness of the interpolation, which means that on the neighborhood of G, $\mathbb{1}_f$ is nearly constant. However this result is a byproduct of the effective continuity of $\mathbb{1}_f$, thanks to Theorem 22.

We will often need to interpolate characteristic functions, that is polynomials that value 1 when $f(x) = a$ and 0 otherwise. For convenience we define a special notation for it.

Definition 42 (Characteristic interpolation) *Let $f : G \to \mathbb{R}$ where G is a finite subset of $\mathbb{R}^d$, $\alpha \in \mathbb{R}$, and define functions $\mathbb{D}_{f=\alpha}, \mathbb{D}_{f\neq\alpha} : \mathbb{R}^d \to \mathbb{R}$ in the following manner*

$$\mathbb{D}_{f=\alpha}(x) = \mathbb{1}_{f_\alpha}(x) \quad \textit{and} \quad \mathbb{D}_{f\neq\alpha}(x) = \mathbb{1}_{1-f_\alpha}(x)$$

where

$$f_\alpha(x) = \begin{cases} 1 & \textit{if } f(x) = \alpha \\ 0 & \textit{otherwise} \end{cases}.$$

Lemma 43 (Characteristic interpolation) *For any finite set $G \subseteq \mathbb{K}^d$, $f : G \to \mathbb{K}$ and $\alpha \in \mathbb{K}$, $\mathbb{D}_{f=\alpha}, \mathbb{D}_{f\neq\alpha} \in$ ALP.*

Proof. Observe that $f_\alpha : G \to \{0, 1\}$ and $\{0, 1\} \subseteq \mathbb{K}$. Apply Lemma 41. ■

5.3 Encoding

In order to simulate a machine, we will need to encode configurations with real numbers. There are several ways of doing so but not all of them are suitable for use when proving complexity results. This particular issue is discussed in Remark 51. For our purpose, it is sufficient to say that we will encode a configuration as a tuple, we store the state and current letter as integers and the left and right parts of the tape as real numbers between 0 and 1. Intuitively, the tape is represented as two numbers whose digits in a particular basis are the letters of the tape. Recall that the alphabet is $\Sigma = [\![0, k-2]\!]$.

Definition 44 (Real encoding) *Let $c = (x, \sigma, y, q)$ be a configuration of $\mathcal{M}$, the* real encoding *of c is $\langle c \rangle = (0.x, \sigma, 0.y, q) \in \mathbb{Q} \times \Sigma \times \mathbb{Q} \times Q$ where $0.x = x_1 k^{-1} + x_2 k^{-2} + \cdots + x_{|w|} k^{-|w|} \in \mathbb{Q}$.*

Lemma 45 (Encoding range) *For any word $x \in [\![0, k-2]\!]^*$, $0.x \in \left[0, \frac{k-1}{k}\right]$.*

Proof. $0 \leqslant 0.x = \sum_{i=1}^{|x|} x_i k^{-i} \leqslant \sum_{i=1}^{\infty} (k-2) k^{-i} \leqslant \frac{k-2}{k-1} \leqslant \frac{k-1}{k}$. ■

The same way we defined the step function for Turing machines on configurations, we have to define a step function that works directly the encoding of configuration. This function is ideal in the sense that it is only defined over real numbers that are encoding of configurations.

Definition 46 (Ideal real step) *The* ideal real step *function of a Turing machine $\mathcal{M}$ is the function defined over $\langle \mathcal{C}_{\mathcal{M}} \rangle$ by:*

$$\langle \mathcal{M} \rangle_\infty (\tilde{x}, \sigma, \tilde{y}, q) = \begin{cases} \left(\operatorname{frac}(k\tilde{x}), \operatorname{int}(k\tilde{x}), \frac{\sigma' + \tilde{y}}{k}, q'\right) & \text{if } d = L \\ (\tilde{x}, \sigma', \tilde{y}, q') & \text{if } d = S \\ \left(\frac{\sigma' + \tilde{x}}{k}, \operatorname{int}(k\tilde{y}), \operatorname{frac}(k\tilde{y}), q'\right) & \text{if } d = R \end{cases} \quad \text{where } \begin{cases} q' = \delta_1(q, \sigma) \\ \sigma' = \delta_2(q, \sigma) \\ d = \delta_3(q, \sigma) \end{cases}.$$

Lemma 47 ($\langle \mathcal{M} \rangle_\infty$ is correct) *For any machine $\mathcal{M}$ and configuration c, $\langle \mathcal{M} \rangle_\infty (\langle c \rangle) = \langle \mathcal{M}(c) \rangle$.*

Proof. Let $c = (x, \sigma, y, q)$ and $\tilde{x} = 0.x$. The proof boils down to a case analysis (the analysis is the same for x and y):

- If $x = \lambda$ then $\tilde{x} = 0$ so $\operatorname{int}(k\tilde{x}) = b$ and $\operatorname{frac}(k\tilde{x}) = 0 = 0.\lambda$ because $b = 0$.
- If $x \neq \lambda$, $\operatorname{int}(k\tilde{x}) = x_1$ and $\operatorname{frac}(k\tilde{x}) = 0.x_{2..|x|}$ because $k\tilde{x} = x_1 + 0.x_{2..|x|}$ and Lemma 45.

■

The previous function was ideal but this is not enough to simulate a machine: We need a step function robust to small perturbations and computable. For this reason, we define a new step function with both features and that relates closely to the ideal function.

Definition 48 (Real step) *For any $\bar{x}, \bar{\sigma}, \bar{y}, \bar{q} \in \mathbb{R}$ and $\mu \in \mathbb{R}_+$, define the* real step *function of a Turing machine $\mathcal{M}$ by:*

$$\langle \mathcal{M} \rangle (\bar{x}, \bar{\sigma}, \bar{y}, \bar{q}, \mu) = \langle \mathcal{M} \rangle^* (\bar{x}, \operatorname{rnd}^*(\bar{\sigma}, \mu), \bar{y}, \operatorname{rnd}^*(\bar{q}, \mu), \mu)$$

where

$$\langle \mathcal{M} \rangle^* (\bar{x}, \bar{\sigma}, \bar{y}, \bar{q}, \mu) = \langle \mathcal{M} \rangle^\star \left(\bar{x}, \bar{y}, \mathbb{1}_{\delta_1}(\bar{q}, \bar{\sigma}), \mathbb{1}_{\delta_2}(\bar{q}, \bar{\sigma}), \mathbb{1}_{\delta_2}(\bar{q}, \bar{\sigma}), \mu\right)$$

where

$$\langle \mathcal{M} \rangle^\star \left(\bar{x}, \bar{y}, \bar{q}, \bar{\sigma}, \bar{d}, \mu\right) = \begin{pmatrix} \operatorname{choose}\left[\operatorname{frac}^*(k\bar{x}), \bar{x}, \frac{\bar{\sigma} + \bar{x}}{k}\right] \\ \operatorname{choose}\left[\operatorname{int}^*(k\bar{x}), \bar{\sigma}, \operatorname{int}^*(k\bar{y})\right] \\ \operatorname{choose}\left[\frac{\bar{\sigma} + \bar{y}}{k}, \bar{y}, \operatorname{frac}^*(k\bar{y})\right] \\ \bar{q} \end{pmatrix}$$

where

$$\text{choose}[l,s,r] = \mathbb{D}_{\text{id}=L}(\bar{d})l + \mathbb{D}_{\text{id}=S}(\bar{d})s + \mathbb{D}_{\text{id}=R}(\bar{d})r,$$

$$\text{int}^*(x) = \text{rnd}^*\left(x - \tfrac{1}{2} + \tfrac{1}{2k}, \mu + \ln k\right) \qquad \text{frac}^*(x) = x - \text{int}^*(x),$$

$$\text{rnd}^* \textit{ is defined in Definition 31.}$$

Theorem 49 (Real step is robust) *For any machine $\mathcal{M}$, $c \in \mathcal{C}_{\mathcal{M}}$, $\mu \in \mathbb{R}_+$ and $\bar{c} \in \mathbb{R}^4$, if $\|\langle c\rangle - \bar{c}\| \leqslant \frac{1}{2k^2} - e^{-\mu}$ then $\|\langle\mathcal{M}\rangle(\bar{c},\mu) - \langle\mathcal{M}(c)\rangle\| \leqslant k\|\langle c\rangle - \bar{c}\|$. Furthermore $\langle\mathcal{M}\rangle \in \text{ALP}$.*

Proof. We begin by a small result about int^* and frac^*: if $\|\bar{x} - 0.x\| \leqslant \frac{1}{2k^2} - e^{-\mu}$ then $\text{int}^*(k\bar{x}) = \text{int}(k0.x)$ and $\|\text{frac}^*(k\bar{x}) - \text{frac}(k0.x)\| \leqslant k\|\bar{x} - 0.x\|$. Indeed, by Lemma 45, $k0.x = n + \alpha$ where $n \in \mathbb{N}$ and $\alpha \in \left[0, \frac{k-1}{k}\right]$. Thus $\text{int}^*(k\bar{x}) = \text{rnd}^*\left(k\bar{x} - \frac{1}{2} + \frac{1}{2k}, \mu\right) = n$ because $\alpha + k\|\bar{x} - 0.x\| - \frac{1}{2} + \frac{1}{2k} \in \left[-\frac{1}{2} + ke^{-\mu}, \frac{1}{2} - ke^{-\mu}\right]$. Also, $\text{frac}^*(k\bar{x}) = k\bar{x} - \text{int}^*(k\bar{x}) = k\|\bar{x} - 0.x\| + kx - \text{int}(kx) = \text{frac}(kx) + k\|\bar{x} - 0.x\|$.

Write $\langle c\rangle = (x, \sigma, y, q)$ and $\bar{c} = (\bar{x}, \bar{\sigma}, \bar{y}, \bar{q})$. Apply Definition 31 to get that $\text{rnd}^*(\bar{\sigma},\mu) = \sigma$ and $\text{rnd}^*(\bar{q},\mu) = q$ because $\|(\bar{\sigma},\bar{q}) - (\sigma, q)\| \leqslant \frac{1}{2} - e^{-\mu}$. Consequently, $\mathbb{1}_{\delta_i}(\bar{q},\bar{\sigma}) = \delta_i(q,\sigma)$ and $\langle\mathcal{M}\rangle(\bar{c},\mu) = \langle\mathcal{M}\rangle^\star(\bar{x},\bar{y},q',\sigma',d')$ where $q' = \delta_1(q,\sigma)$, $\sigma' = \delta_2(q,\sigma)$ and $d' = \delta_3(q,\sigma)$. In particular $d' \in \{L, S, R\}$ so there are three cases to analyze.

- **If** $d' = L$ **then** $\text{choose}[l,s,r] = l$, $\text{int}^*(k\bar{x}) = \text{int}(kx)$, $\|\text{frac}^*(k\bar{x}) - \text{frac}(kx)\| \leqslant k\|\bar{x} - x\|$ and $\left\|\frac{\sigma'+\bar{y}}{k} - \frac{\sigma'+y}{k}\right\| \leqslant \|\bar{x} - x\|$. Thus $\|\langle\mathcal{M}\rangle(\bar{c},\mu) - \langle\mathcal{M}\rangle_\infty(\langle c\rangle)\| \leqslant k\|\bar{c} - \langle c\rangle\|$. Conclude using Lemma 47.
- **If** $d' = S$ **then** $\text{choose}[l,s,r] = s$ so we immediately have that $\|\langle\mathcal{M}\rangle(\bar{c},\mu) - \langle\mathcal{M}\rangle_\infty(\langle c\rangle)\| \leqslant \|\bar{c} - \langle c\rangle\|$. Conclude using Lemma 47.
- **If** $d' = R$ **then** $\text{choose}[l,s,r] = r$ and everything else is similar to the case of $d' = L$.

Finally apply Lemma 41, Theorem 32, Theorem 20 and Theorem 21 to get that $\langle\mathcal{M}\rangle \in \text{ALP}$. ■

6 A Characterization of FP

We will now provide a characterization of FP by introducing a notion of function *emulation*. This characterization builds on our notion of computability introduced previously.

In this section, we fix an alphabet Γ and all languages are considered over Γ. It is common to take $\Gamma = \{0, 1\}$ but the proofs work for any finite alphabet. We will assume that Γ comes with an injective mapping $\gamma : \Gamma \to \mathbb{N} \setminus \{0\}$, in other words every letter has an uniquely assigned positive number. By extension, γ applies letterwise over words.

6.1 Main statement

Definition 50 (Discrete emulation) *$f : \Gamma^* \to \Gamma^*$ is called* $\mathbb{K}$-emulable *if there exists $g \in \mathrm{ALP}_{\mathbb{K}}$ and $k \geqslant 1 + \max(\gamma(\Gamma))$ such that for any word $w \in \Gamma^*$:*

$$g(\psi_k(w)) = \psi_k(f(w)) \qquad \textit{where} \quad \psi_k(w) = \left(\sum_{i=1}^{|w|} \gamma(w_i)k^{-i}, |w|\right).$$

We say that g $\mathbb{K}$-emulates *f with k. When the field $\mathbb{K}$ is unambiguous, we will simply say that f is emulable.*

Remark 51 (Encoding length) *The exact details of the encoding ψ chosen in the definition above are not extremely important, however the length of the encoding is crucial. More precisely, the proof heavily relies on the fact that $\|\psi(w)\| \approx |w|$. Note that this works both ways:*

- *$\|\psi(w)\|$ must be polynomially bounded in $|w|$ so that a simulation of the system runs in polynomial time in $|w|$.*
- *$\|\psi(w)\|$ must be polynomially lower bounded in $|w|$ so that we can recover the output length from the length of its encoding.*

The sef FP of polynomial-time computable functions can then be characterized as follows.

Theorem 52 (FP equivalence) *For any generable field $\mathbb{K}$ such that $\mathbb{R}_G \subseteq \mathbb{K} \subseteq \mathbb{R}_P$ and $f : \Gamma^* \to \Gamma^*$, $f \in \mathrm{FP}$ if and only if f is $\mathbb{K}$-emulable (with $k = 2 + \max(\gamma(\Gamma))$).*

The rest of this section is devoted to the proof of Theorem 52

6.2 Reverse direction of Theorem 52

The reverse direction of the equivalence between Turing machines and analog systems will involve polynomial initial value problems such as (1).

6.2.1 Complexity of solving polynomial differential equations

The complexity of solving this kind of differential equation has been heavily studied over compact domains but there are few results over unbounded domains. In [PG16] we studied the complexity of this problem over unbounded domains and obtained a bound that involved the length of the solution curve. In [Pou16], we extended this result to work with any real inputs (and not just rationals) in the framework of Computable Analysis.

We need a few notations to state the result. For any multivariate polynomial $p(x) = \sum_{|\alpha|\leqslant k} a_\alpha x^\alpha$, we call k the degree if k is the minimal integer k for which the condition $p(x) = \sum_{|\alpha|\leqslant k} a_\alpha x^\alpha$ holds and we denote the sum of the norm of the coefficients by $\Sigma p = \sum_{|\alpha|\leqslant k} |a_\alpha|$ (also known as the length of p). For a vector of polynomials, we define the degree and Σp as the maximum over all components. For any continuous function y and polynomial p define the *pseudo-length*

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

Theorem 53 ([PG16], [Pou16]) *Let $I = [a, b]$ be an interval, $p \in \mathbb{R}^n[\mathbb{R}^n]$ and k its degree and $y_0 \in \mathbb{R}^n$. Assume that $y : I \to \mathbb{R}^n$ satisfies for all $t \in I$ that*

$$y(a) = y_0 \qquad y'(t) = p(y(t)), \tag{3}$$

then $y(b)$ can be computed with precision $2^{-\mu}$ in time bounded by

$$\mathrm{poly}(k, \mathrm{PsLen}_{y,p}(a, b), \log \|y_0\|, \log \Sigma p, \mu)^n. \tag{4}$$

More precisely, there exists a Turing machine $\mathcal{M}$ such that for any oracle $\mathcal{O}$ representing[11] *(a, y_0, p, b) and any $\mu \in \mathbb{N}$, $\left\|\mathcal{M}^{\mathcal{O}}(\mu) - y(b)\right\| \leqslant 2^{-\mu}$ where y satisfies (3), and the number of steps of the machine is bounded by (4) for all such oracles.*

Finally, we would like to remind the reader that the existence of a solution y of a PIVP up to a given time is undecidable, see [GBC07] for more details. This explains why, in the previous theorem, we have to assume the existence of the solution if we want to have any hope of computing it.

6.2.2 Proof of Reverse direction of Theorem 52

Assume that f is $\mathbb{R}_P$-emulable and apply Definition 50 to get $g \in \mathrm{ATSC}(\Upsilon, \amalg)$ where $\Upsilon, \amalg$ are polynomials, with respective d, p, q. Let $w \in \Gamma^*$: we will describe an FP algorithm to compute $f(w)$. Consider the following system:

$$y(0) = q(\psi_k(w)) \qquad y'(t) = p(y(t)).$$

Note that, by construction, y is defined over $\mathbb{R}_+$. Also note, that the coefficients of p, q belong to $\mathbb{R}_P$ which means that they are polynomial time computable. And since $\psi_k(w)$ is a pair of rational numbers with polynomial length (with respect to $|w|$), then $q(\psi_k(w)) \in \mathbb{R}_P^d$.

The algorithm works in two steps: first we compute a rough approximation of the output to guess the length of the output. Then we rerun the system with enough precision to get the full output.

Let $t_w = \amalg(|w|, 2)$ for any $w \in \Sigma^*$. Note that $t_w \in \mathbb{R}_P$ and that it is polynomially bounded in $|w|$ because $\amalg$ is a polynomial. Apply Theorem 53 to compute $\tilde{y}$ such that $\|\tilde{y} - y(t_w)\| \leqslant e^{-2}$: this takes a time polynomial in $|w|$ because t_w is polynomially bounded and because[12] $\mathrm{PsLen}_{y,p}(0, t_w) \leqslant \mathrm{poly}(t_w, \sup_{[0,t_w]} \|y\|)$ and by construction, $\|y(t)\| \leqslant \Upsilon(\|\psi_k(w)\|, t_w)$ for $t \in [0, t_w]$ where Υ is a polynomial. Furthermore, by definition of t_w, $\|y(t_w) - g(\psi_k(w))\| \leqslant e^{-2}$ thus $\|\tilde{y} - \psi_k(f(w))\| \leqslant 2e^{-2} \leqslant \frac{1}{3}$. But since $\psi_k(f(w)) = (0.\gamma(f(w)), |f(w)|)$, from $\tilde{y}_2$ we can find $|f(w)|$ by rounding to the closest integer (which is unique because it is within distance at most $\frac{1}{3}$). In other words, we can compute $|f(w)|$ in polynomial time in $|w|$. Note that this implies that $|f(w)|$ is at most polynomial in $|w|$.

Let $t'_w = \amalg(|w|, 2 + |f(w)| \ln k)$ which is polynomial in $|w|$ because $\amalg$ is a polynomial and $|f(w)|$ is at most polynomial in $|w|$. We can use the same reasoning and apply Theorem 53 to get $\tilde{y}$ such that $\|\tilde{y} - y(t'_w)\| \leqslant e^{-2-|f(w)| \ln k}$. Again this takes a time polynomial in $|w|$. Furthermore, $\|\tilde{y}_1 - 0.\gamma(f(w))\| \leqslant$

[11] See [Ko91] for more details. In short, the machine can ask arbitrary approximations of a, y_0, p and b to the oracle. The polynomial is represented by the finite list of coefficients.

[12] See Section 6.2.1 for the expression PsLen.

$2e^{-2-|f(w)|\ln k} \leqslant \frac{1}{3}k^{-|f(w)|}$. We claim that this allows to recover $f(w)$ unambiguously in polynomial time in $|f(w)|$. Indeed, it implies that $\left\|k^{|f(w)|}\tilde{y}_1 - k^{|f(w)|}0.\gamma(f(w))\right\| \leqslant \frac{1}{3}$. 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 nearest integer, we recover $\gamma(f(w))$, and then $f(w)$. This is all done in polynomial time in $|f(w)|$, which proves that f is polynomial time computable.

6.3 Direct direction of Theorem 52

6.3.1 Iterating a function

The direct direction of of the equivalence between Turing machines and analog systems will involve iterations of the robust real step associated to a Turing machine of the previous section.

We now state that iterating a function is computable under reasonable assumptions. Iteration is a powerful operation, which is why reasonable complexity classes are never closed under unrestricted iteration. If we want to keep to polynomial-time computability for Computable Analysis, there are at least two immediate necessary conditions: the iterates cannot grow faster than a polynomial and the iterates must keep a polynomial modulus of continuity. The optimality of the conditions of next theorem is discussed in Remark 55 and Remark 56. However there is the subtler issue of the domain of definition that comes into play and is discussed in Remark 57.

In short, the conditions to iterate a function can be summarized as follows:

- f has domain of definition I;
- there are subsets I_n of I such that all the points of I_n can be iterated up to n times;
- the iterates of f on x over I_n grow at most polynomially in $\|x\|$ and n;
- each point x in I_n has an open neighborhood in I of radius at least $e^{-\operatorname{poly}(\|x\|)}$ and f has modulus of continuity of the form $\operatorname{poly}(\|x\|) + \mu$ over this set.

Formally:

Theorem 54 (Simulating Discrete by Continuous Time) *Let $I \subseteq \mathbb{R}^m$, $(f : I \to \mathbb{R}^m) \in \mathrm{ALP}$, $\eta \in [0, 1/2[$ and assume that there exists a family of subsets $I_n \subseteq I$, for all $n \in \mathbb{N}$ and polynomials $\mho : \mathbb{R}_+ \to \mathbb{R}_+$ and $\Pi : \mathbb{R}_+^2 \to \mathbb{R}_+$ such that for all $n \in \mathbb{N}$:*

- *$I_{n+1} \subseteq I_n$ and $f(I_{n+1}) \subseteq I_n$*
- *$\forall x \in I_n$, $\left\|f^{[n]}(x)\right\| \leqslant \Pi(\|x\|, n)$*
- *$\forall x \in I_n$, $y \in \mathbb{R}^m, \mu \in \mathbb{R}_+$, if $\|x - y\| \leqslant e^{-\mho(\|x\|)-\mu}$ then $y \in I$ and $\|f(x) - f(y)\| \leqslant e^{-\mu}$*

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

This result is far from trivial, and the whole Section 9.1 is devoted to its proof.

Remark 55 (Optimality of growth constraint) *It is easy to see that without any restriction, the iterates can produce an exponential function. Pick* $f(x) = 2x$ *then* $f \in \mathrm{ALP}$ *and* $f^{[n]}(x) = 2^n x$ *which is clearly not polynomial in* x *and* n. *More generally, it is necessary that* f^* *be polynomially bounded so clearly* $f^{[n]}(x)$ *must be polynomially bounded in* $\|x\|$ *and* n.

Remark 56 (Optimality of modulus constraint) *Without any constraint (specifically the constraint of the 3rd item in Theorem 54), it is easy to build an iterated function with exponential modulus of continuity. Define* $f(x) = \sqrt{x}$ *then* f *can be shown to be in* ALP *and* $f^{[n]}(x) = x^{\frac{1}{2^n}}$. *For any* $\mu \in \mathbb{R}$, $f^{[n]}(e^{-2^n\mu}) - f^{[n]}(0) = (e^{-2^n\mu})^{\frac{1}{2^n}} = e^{-\mu}$. *Thus* f^* *has exponential modulus of continuity in* n.

Remark 57 (Domain of definition) *Intuitively we would have written the theorem differently, only requesting that* $f(I) \subseteq I$, *however this has some problems. First if* I *is discrete, the iterated modulus of continuity becomes useless and the theorem is false. Indeed, define* $f(x,k) = (\sqrt{x}, k+1)$ *and* $I = \{(\sqrt[2^n]{e}, n), n \in \mathbb{N}\}$: $f\restriction_I$ *has polynomial modulus of continuity* $\mho$ *because* I *is discrete, yet* $f^*\restriction_I \notin \mathrm{ALP}$ *as we saw in Remark 56. But in reality, the problem is more subtle than that because if* I *is open but the neighborhood of each point is too small, a polynomial system cannot take advantage of it. To illustrate this issue, define* $I_n =]0, \sqrt[2^n]{e}[\times]n - \frac{1}{4}, n + \frac{1}{4}[$ *and* $I = \cup_{n\in\mathbb{N}} I_n$. *Clearly* $f(I_n) = I_{n+1}$ *so* I *is* f*-stable but* $f^*\restriction_I \notin \mathrm{ALP}$ *for the same reason as before.*

Remark 58 (Classical error bound) *The third condition in Theorem 54 is usually far more subtle than necessary. In practice, is it useful to note this condition is satisfied if* f *verifies for some constants* $\varepsilon, K > 0$ *that*

$$\text{for all } x \in I_n \text{ and } y \in \mathbb{R}^m,\ \text{ if } \|x-y\| \leqslant \varepsilon \text{ then } y \in I \text{ and } \|f(x) - f(y)\| \leqslant K \|x-y\|\,.$$

Remark 59 (Dependency of $\mho$ in n) *In the statement of the theorem,* $\mho$ *is only allowed to depend on* $\|x\|$ *whereas it might be useful to also make it depend on* n. *In fact the theorem is still true if the last condition is modified to be* $\|x-y\| \leqslant e^{-\mho(\|x\|,n)-\mu}$. *One way of showing this is to explicitly add* n *to the domain of definition by taking* $h(x,k) = (f(x), k-1)$ *and to take* $I'_n = I_n \times [n, +\infty[$ *for example.*

6.3.2 Proof of Direct direction of Theorem 52

Let $f \in \mathrm{FP}$, then there exists a Turing machine $\mathcal{M} = (Q, \Sigma, b, \delta, q_0, F)$ where $\Sigma = [\![0, k-2]\!]$ and $\gamma(\Gamma) \subset \Sigma \setminus \{b\}$, and a polynomial $p_\mathcal{M}$ such that for any word $w \in \Gamma^*$, $\mathcal{M}$ halts in at most $p_\mathcal{M}(|w|)$ steps, that is $\mathcal{M}^{[p_\mathcal{M}(|w|)]}(c_0(\gamma(w))) = c_\infty(\gamma(f(w)))$. Note that we assume that $p_\mathcal{M}(\mathbb{N}) \subseteq \mathbb{N}$. Also note that $\psi_k(w) = (0.\gamma(w), |w|)$ for any word $w \in \Gamma^*$.

Define $\mu = \ln(4k^2)$ and $h(c) = \langle\mathcal{M}\rangle(c, \mu)$ for all $c \in \mathbb{R}^4$. Define $I_\infty = \langle\mathcal{C}_\mathcal{M}\rangle$ and $I_n = I_\infty + [-\varepsilon_n, \varepsilon_n]^4$ where $\varepsilon_n = \frac{1}{4k^{2+n}}$ for all $n \in \mathbb{N}$. Note that $\varepsilon_{n+1} \leqslant \frac{\varepsilon_n}{k}$ and that $\varepsilon_0 \leqslant \frac{1}{2k^2} - e^{-\mu}$. By Theorem 49 we have $h \in \mathrm{ALP}$ and $h(I_{n+1}) \subseteq I_n$. In particular $\left\|h^{[n]}(\bar{c}) - h^{[n]}(c)\right\| \leqslant k^n \|c - \bar{c}\|$ for all $c \in I_\infty$ and $\bar{c} \in I_n$, for all $n \in \mathbb{N}$. Let $\delta \in \left[0, \frac{1}{2}\right[$ and define $J = \cup_{n\in\mathbb{N}} I_n \times [n-\delta, n+\delta]$. Apply Theorem 54 to get $(h_\delta^* : J \to I_0) \in \mathrm{ALP}$ such that for all $c \in I_\infty$ and $n \in \mathbb{N}$ and $h_\delta^*(c, n) = h^{[n]}(c)$.

Let π_i denote the i^{th} projection, that is $\pi_i(x) = x_i$, then $\pi_i \in \mathrm{ALP}$. Define

$$g(y, \ell) = \pi_3(h_\delta^*(0, b, \pi_1(y), q_0, p_{\mathcal{M}}(\ell)))$$

for $y \in \psi_k(\Gamma^*)$ and $\ell \in \mathbb{N}$. Note that $g \in \mathrm{ALP}$ and is well-defined. Indeed, if $\ell \in \mathbb{N}$ then $p_{\mathcal{M}}(\ell) \in \mathbb{N}$ and if $y = \psi_k(w)$ then $\pi_1(y) = 0.\gamma(w)$ then $(0, b, \pi_1(y), q_0) = \langle(\lambda, b, w, q_0)\rangle = \langle c_0(w)\rangle \in I_\infty$. Furthermore, by construction, for any word $w \in \Gamma^*$ we have:

$$\begin{aligned}
g(\psi_k(w), |w|) &= \pi_3\left(h_\delta^*(\langle c_0(w)\rangle, p_{\mathcal{M}}(|w|))\right) \\
&= \pi_3\left(h^{[p_{\mathcal{M}}(|w|)]}(c_0(w))\right) \\
&= \pi_3\left(\left\langle \mathcal{C}_{\mathcal{M}}^{[p_{\mathcal{M}}(|w|)]}(c_0(w))\right\rangle\right) \\
&= \pi_3\left(\langle c_\infty(\gamma(f(w)))\rangle\right) \\
&= 0.\gamma(f(w)) = \pi_1(\psi_k(f(w))).
\end{aligned}$$

Recall that to show emulation, we need to compute $\psi_k(f(w))$ and so far we only have the first component: the output tape encoding, but we miss the second component: its length. Since the length of the tape cannot be greater than the initial length plus the number of steps, we have that $|f(w)| \leqslant |w| + p_{\mathcal{M}}(|w|)$. Apply Corollary 63 (this corollary will appear only on the next section. But its proof does not depend on this result and therefore this does not pose a problem) to get that tape length $\mathrm{tlength}_{\mathcal{M}}(g(\psi_k(w), |w|), |w| + p_{\mathcal{M}}(|w|)) = |f(w)|$ since $f(w)$ does not contain any blank character (this is true because $\gamma(\Gamma) \subset \Sigma \setminus \{b\}$). This proves that f is emulable because $g \in \mathrm{ALP}$ and $\mathrm{tlength}_{\mathcal{M}} \in \mathrm{ALP}$.

6.4 On the robustness of previous characterization

An interesting question arises when looking at this theorem: does the choice of k in Definition 50 matters, especially for the equivalence with FP ? Fortunately not, as long as k is large enough, as shown in the next lemma.

Actually in several cases, we will need to either decode words from noisy encodings, or re-encode a word in a different basis. This is not a trivial operation because small changes in the input can result in big changes in the output. Furthermore, continuity forbids us from being able to decode all inputs. The following theorem is a very general tool. Its proof is detailed page 61. The following Corollary 61 is a simpler version when one only needs to re-encode a word.

Theorem 60 (Word decoding) *Let $k_1, k_2 \in \mathbb{N}^*$ and $\kappa : [\![0, k_1-1]\!] \to [\![0, k_2-1]\!]$. There exists a function $(\mathrm{decode}_\kappa :\subseteq \mathbb{R} \times \mathbb{N} \times \mathbb{R} \to \mathbb{R}) \in \mathrm{ALP}$ such that for any word $w \in [\![0, k_1-1]\!]^*$ and $\mu, \varepsilon \geqslant 0$:*

$$\text{if } \varepsilon \leqslant k_1^{-|w|}(1-e^{-\mu}) \text{ then } \mathrm{decode}_\kappa\left(\sum_{i=1}^{|w|} w_i k_1^{-i} + \varepsilon, |w|, \mu\right) = \left(\sum_{i=1}^{|w|} \kappa(w_i) k_2^{-i}, \#\{i | w_i \neq 0\}\right)$$

Corollary 61 (Re-encoding) *Let $k_1, k_2 \in \mathbb{N}^*$ and $\kappa : [\![1, k_1-2]\!] \to [\![0, k_2-1]\!]$. There exists a function* $(\operatorname{reenc}_\kappa :\subseteq \mathbb{R} \times \mathbb{N} \to \mathbb{R} \times \mathbb{N}) \in \mathrm{ALP}$ *such that for any word $w \in [\![1, k_1 - 2]\!]^*$ and $n \geqslant |w|$ we have:*

$$\operatorname{reenc}_\kappa\left(\sum_{i=1}^{|w|} w_i k_1^{-i}, n\right) = \left(\sum_{i=1}^{|w|} \kappa(w_i) k_2^{-i}, |w|\right)$$

Proof. The proof is immediate: extend κ with $\kappa(0) = 0$ and define

$$\operatorname{reenc}_\kappa(x, n) = \operatorname{decode}_\kappa(x, n, 0).$$

Since $n \geqslant |w|$, we can apply Theorem 60 with $\varepsilon = 0$ to get the result. Note that strictly speaking, we are not applying the theorem to w but rather to w padded with as many 0 symbols as necessary, ie $w0^{n-|w|}$. Since w does not contain the symbol 0, its length is the same as the number of non-blank symbols it contains. ■

Remark 62 (Nonreversible re-encoding) *Note that the previous theorem and corollary allows from nonreversible re-encoding when $\kappa(\alpha) = 0$ or $\kappa(\alpha) = k_2 - 1$ for some $\alpha \neq 0$. For example, it allows one to re-encode a word over $\{0, 1, 2\}$ with $k_1 = 4$ to a word over $\{0, 1\}$ with $k_2 = 2$ with $\kappa(1) = 0$ and $\kappa(2) = 1$ but the resulting number cannot be decoded in general (for continuity reasons). In some cases, only the more general Theorem 60 provides a way to recover the encoding.*

A typically application of this function is to recover the length of the tape after a computation. Indeed way to do this is to keep track of the tape length during the computation, but this usually requires a modified machine and some delimiters on the tape. Instead, we will use the previous theorem to recover the length from the encoding, assuming it does not contain any blank character. The only limitation is that to recover the lenth of w from its encoding $0.w$, we need to have an upper bound on the length of w.

Corollary 63 (Length recovery) *For any machine $\mathcal{M}$, there exists a function* $(\operatorname{tlength}_\mathcal{M} : \langle \mathcal{C}_\mathcal{M} \rangle \times \mathbb{N} \to \mathbb{N}) \in \mathrm{ALP}$ *such that for any word $w \in (\Sigma \setminus \{b\})^*$ and any $n \geqslant |w|$, $\operatorname{tlength}_\mathcal{M}(0.w, n) = |w|$.*

Proof. It is an immediate consequence of Corollary 61 with $k_1 = k_2 = k$ and $\kappa = \mathrm{id}$ where we throw away the re-encoding. ■

The previous tools are also precisely what is needed to prove that our notion of emulation is independant of k.

Lemma 64 (Emulation re-encoding) *Assume that $g \in \mathrm{ALP}$ emulates f with $k \in \mathbb{N}$. Then for any $k' \geqslant k$, there exists $h \in \mathrm{ALP}$ that emulates f with k'.*

Proof. The proof follows from Corollary 61 by a standard game playing with encoding/reencoding.

More precisely, let $k' \geqslant k$ and define $\kappa : [\![1, k']\!] \to [\![1, k]\!]$ and $\kappa^{-1} : [\![1, k]\!] \to [\![1, k']\!]$ as follows:

$$\kappa(w) = \begin{cases} w & \text{if } w \in \gamma(\Gamma) \\ 1 & \text{otherwise} \end{cases} \qquad \kappa^{-1}(w) = w.$$

In the following, $0.w$ (resp. $0'.w$) denotes the rational encoding in basis k (resp. k'). Apply Corollary 61 twice to get that $\mathrm{reenc}_\kappa, \mathrm{reenc}_{\kappa^{-1}} \in \mathrm{ALP}$. Define:

$$h = \mathrm{reenc}_{\kappa^{-1}} \circ g \circ \mathrm{reenc}_\kappa\,.$$

Note that $\gamma(\Gamma) \subseteq [\![1, k-1]\!]^* \subseteq [\![1, k'-1]\!]^*$ since γ never maps letters to 0 and $k \geqslant 1 + \max(\gamma(\Gamma))$ by definition. Consequently for $w \in \Gamma^*$:

$$\begin{aligned}
h(\psi_{k'}(w)) &= h(0'.\gamma(w), |w|) && \text{By definition of } \psi_{k'}\\
&= \mathrm{reenc}_{\kappa^{-1}}(g(\mathrm{reenc}_\kappa(0'.\gamma(w), |w|))) &&\\
&= \mathrm{reenc}_{\kappa^{-1}}(g(0.\kappa(\gamma(w)), |w|)) && \text{Because } \gamma(w) \in [\![1, k']\!]^*\\
&= \mathrm{reenc}_{\kappa^{-1}}(g(0.\gamma(w), |w|)) && \text{Because } \gamma(w) \in \gamma(\Gamma)^*\\
&= \mathrm{reenc}_{\kappa^{-1}}(g(\psi_k(w))) && \text{By definition of } \psi_k\\
&= \mathrm{reenc}_{\kappa^{-1}}(\psi_k(f(w))) && \text{Because } g \text{ emulates } f\\
&= \mathrm{reenc}_{\kappa^{-1}}(0.\gamma(f(w)), |f(w)|) && \text{By definition of } \psi_k\\
&= (0'.\kappa^{-1}(\gamma(f(w))), |f(w)|) && \text{Because } \gamma(f(w)) \in \gamma(\Gamma)^*\\
&= (0'.\gamma(f(w)), |f(w)|) && \text{By definition of } \kappa^{-1}\\
&= \psi_{k'}(f(w)). && \text{By definition of } \psi_{k'}
\end{aligned}$$

■

The previous notion of emulation was for single input functions, which is sufficient in theory because we can always encode tuples of words using a single word or give Turing machines several input/output tapes. But for the next results of this section, it will be useful to have functions with multiple inputs/outputs without going through an encoding. We extend the notion of discrete encoding in the natural way to handle this case.

Definition 65 (emulation) *$f : (\Gamma^*)^n \to (\Gamma^*)^m$ is called* emulable *if there exists $g \in \mathrm{ALP}$ and $k \in \mathbb{N}$ such that for any word $\vec{w} \in (\Gamma^*)^n$:*

$$g(\psi_k(\vec{w})) = \psi_k(f(\vec{w})) \qquad \textit{where} \quad \psi_k(x_1, \dots, x_\ell) = (\psi(x_1), \dots, \psi(x_\ell))$$

and ψ_k is defined as in Definition 50.

It is trivial that Definition 65 matches Definition 50 in the case of unidimensional functions, thus the two definitions are consistent with each other.

Theorem 52 then generalizes to the multidimensional case naturally as follows. Proof is in page 64.

Theorem 66 (Multidimensional FP equivalence) *For any $f : (\Gamma^*)^n \to (\Gamma^*)^m$, $f \in \mathrm{FP}$ if and only if f is emulable.*

7 A Characterization of P

We will now use this characterization of FP to give a characterization of P: Our purpose is now to prove that a decision problem (language) $\mathcal{L}$ belongs to the class P if and only if it is poly-length-analog-recognizable.

The following definition is a generalization (to general field $\mathbb{K}$) of Definition 1:

Definition 67 (Discrete recognizability) *A language $\mathcal{L} \subseteq \Gamma^*$ is called* $\mathbb{K}$-poly-length-analog-recognizable *if there exists a vector q of bivariate polynomials and a vector p of polynomials with d variables, both with coefficients in $\mathbb{K}$, and a polynomial $\amalg : \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $w \in \Gamma^*$, there is a (unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ such that for all $t \in \mathbb{R}_+$:*

- *$y(0) = q(\psi_k(w))$ and $y'(t) = p(y(t))$* ▶ *y satisfies a differential equation*
- *if $|y_1(t)| \geqslant 1$ then $|y_1(u)| \geqslant 1$ for all $u \geqslant t$* ▶ *decision is stable*
- *if $w \in \mathcal{L}$ (resp. $\notin \mathcal{L}$) and $\operatorname{len}_y(0, t) \geqslant \amalg(|w|)$ then $y_1(t) \geqslant 1$ (resp. $\leqslant -1$)* ▶ *decision*
- $\operatorname{len}_y(0, t) \geqslant t$ ▶ *technical condition*[13]

Using Theorem 52 on the characterization of FP, we can show that this class corresponds exactly to P. The proof is not complicated but because of the difference in the output format, we need to be careful. Indeed, in our characterization of FP, we simply show that after a polynomial length, the output value is exactly the encoding of the output string. In this characterization of P, we have a much more relaxed notion of signalling whether the computation is still in progress or done.

Theorem 68 (P **equivalence**) *Let $\mathbb{K}$ be a generable field such that $\mathbb{R}_G \subseteq \mathbb{K} \subseteq \mathbb{R}_P$. For any language $\mathcal{L} \subseteq \Gamma^*$, $\mathcal{L} \in$ P if and only if $\mathcal{L}$ is $\mathbb{K}$-poly-length-analog-recognizable.*

Proof. The direct direction will build on the equivalence with FP, except that a technical point is to make sure that the decision of the system is irreversible.

Let $\mathcal{L} \in$ P. Then there exist $f \in$ FP and two distinct symbols $\bar{0}, \bar{1} \in \Gamma$ such that for any $w \in \Gamma^*$, $f(w) = \bar{1}$ if $w \in \mathcal{M}$ and $f(w) = \bar{0}$ otherwise. Let dec be defined by $\operatorname{dec}(k^{-1}\gamma(\bar{0})) = -2$ and $\operatorname{dec}(k^{-1}\gamma(\bar{1})) = 2$. Recall that $\mathbb{1}_{\text{dec}} \in$ ALP by Lemma 41. Apply Theorem 52 to get g and k that emulate f. Note in particular that for any $w \in \Gamma^*$, $f(w) \in \{\bar{0}, \bar{1}\}$ so $\psi(f(w)) = (\gamma(\bar{0})k^{-1}, 1)$ or $(\gamma(\bar{1})k^{-1}, 1)$. Define $g^*(x) = \mathbb{1}_{\text{dec}}(g_1(x))$ and check that $g^* \in$ ALP. Furthermore, $g^*(\psi_k(w)) = 2$ if $w \in \mathcal{L}$ and $g^*(\psi_k(w)) = -2$ otherwise, by definition of the emulation and the interpolation.

We have $g^* \in \operatorname{ATSC}(\Upsilon, \amalg)$ for some polynomials $\amalg$ and Υ be polynomials with corresponding d, p, q. Assume, without loss of generality, that $\amalg$ and Υ are increasing functions. Let $w \in \Gamma^*$ and consider the following system:

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

$$\tau^* = \amalg(v_2(t), \ln 2)$$

In this system, 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 and z is the

[13] This could be replaced by only assuming that we have somewhere the additional ordinary differential equation $y_0' = 1$.

decision variable. Let $t \in [0, \tau^*]$, then by Lemma 34, $\|z'(t)\| \leqslant e^{-1-t}$ thus $\|z(t)\| \leqslant e^{-1} < 1$. In other words, at time τ^* the system has still not decided if $w \in \mathcal{L}$ or not. Let $t \geqslant \tau^*$, then by definition of Π and since $v_2(t) = \psi_{k,2}(w) = |w| = \|\psi_k(w)\|$, $\|y_1(t) - g^*(\psi_k(w))\| \leqslant e^{-\ln 2}$. Recall that $g^*(\psi_k(w)) \in \{-2, 2\}$ and let $\varepsilon \in \{-1, 1\}$ such that $g^*(\psi_k(w)) = \varepsilon 2$. Then $\|y_1(t) - \varepsilon 2\| \leqslant \frac{1}{2}$ which means that $y_1(t) = \varepsilon\lambda(t)$ where $\lambda(t) \geqslant \frac{3}{2}$. Apply Lemma 34 to conclude that z satisfies for $t \geqslant \tau^*$:

$$z(\tau^*) \in [-e^{-1}, e^{-1}] \qquad z'(t) = \phi(t)(\varepsilon\lambda(t) - z(t))$$

where $\phi(t) \geqslant 0$ and $\phi(t) \geqslant 1 - e^{-1}$ for $t \geqslant \tau^* + 1$. Let $z_\varepsilon(t) = \varepsilon z(t)$ and check that z_ε satisfies:

$$z_\varepsilon(\tau^*) \in [-e^{-1}, e^{-1}] \qquad z_\varepsilon'(t) \geqslant \phi(t)(\tfrac{3}{2} - z_\varepsilon(t))$$

It follows that z_ε is an increasing function and from a classical argument about differential inequalities that:

$$z_\varepsilon(t) \geqslant \frac{3}{2} - \left(\frac{3}{2} - z_\varepsilon(\tau^*)\right) e^{-\int_{\tau^*}^{t} \phi(u)du}$$

In particular for $t^* = \tau^* + 1 + 2\ln 4$ we have:

$$z_\varepsilon(t) \geqslant \frac{3}{2} - (\tfrac{3}{2} - z_\varepsilon(\tau^*))e^{-2\ln 4(1-e^{-1})} \geqslant \frac{3}{2} - 2e^{-\ln 4} \geqslant 1.$$

This proves that $|z(t)| = z_\varepsilon(t)$ is an increasing function, so in particular once it has reached 1, it stays greater than 1. Furthermore, if $w \in \mathcal{L}$ then $z(t^*) \geqslant 1$ and if $w \notin \mathcal{L}$ then $z(t^*) \leqslant 1$. Note that $\|(y, v, z, w)'(t)\| \geqslant 1$ for all $t \geqslant 1$ so the technical condition is satisfied. Also note that z is bounded by a constant, by a very similar reasoning. This shows that if $Y = (y, v, z, \tau)$, then $\|Y(t)\| \leqslant \operatorname{poly}(\|\psi_k(w)\|, t)$ because $\|y(t)\| \leqslant \Upsilon(\|\psi_k(w)\|, t)$. Consequently, there is a polynomial Υ^* such that $\|Y'(t)\| \leqslant \Upsilon^*$ (this is immediate from the expression of the system), and without loss of generality, we can assume that Υ^* is an increasing function. And since $\|Y'(t)\| \geqslant 1$, we have that $t \leqslant \operatorname{len}_Y(0, t) \leqslant t \sup_{u \in [0,t]} \|Y'(u)\| \leqslant t\Upsilon^*(\|\psi_k(w)\|, t)$. Define $\Pi^*(\alpha) = t^*\Upsilon^*(\alpha, t^*)$ which is a polynomial because t^* is polynomially bounded in $\|\psi_k(w)\| = |w|$. Let t such that $\operatorname{len}_Y(0, t) \geqslant \Pi^*(|w|)$, then by the above reasoning, $t\Upsilon^*(|w|, t) \geqslant \Pi^*(|w|)$ and thus $t \geqslant t^*$ so $|z(t)| \geqslant 1$, i.e. the system has decided.

The reverse direction of the proof is the following: assume that $\mathcal{L}$ is $\mathbb{K}$-poly-length-analog-recognizable. Apply Definition 67 to get d, q, p and Π. Let $w \in \Gamma^*$ and consider the following system:

$$y(0) = q(\psi_k(w)) \qquad y'(t) = p(y(t))$$

We will show that we can decide in time polynomial in $|w|$ whether $w \in \mathcal{L}$ or not. Note that q is a polynomial with coefficients in $\mathbb{R}_P$ (since we consider $\mathbb{K} \subset \mathbb{R}_P$) and $\psi_k(w)$ is a rational number so $q(\psi_k(w)) \in \mathbb{R}_P^d$. Similarly, p has coefficients in $\mathbb{R}_P$. Finally, note that[14]:

$$\operatorname{PsLen}_{y,p}(0, t) = \int_0^t \Sigma p \max(1, \|y(u)\|)^k du$$

[14] See Section 6.2.1 for the expression PsLen.

$$
\begin{aligned}
&\leqslant t\Sigma p \max\left(1, \sup_{u\in[0,t]} \|y(u)\|^k\right)\\
&\leqslant t\Sigma p \max\left(1, \sup_{u\in[0,t]} (\|y(0)\| + \operatorname{len}_y(0,t))^k\right)\\
&\leqslant t \operatorname{poly}(\operatorname{len}_y(0,t))\\
&\leqslant \operatorname{poly}(\operatorname{len}_y(0,t))
\end{aligned}
$$

where the last inequality holds because $\operatorname{len}_y(0,t) \geqslant t$ thanks to the technical condition. We can now apply Theorem 53 to conclude that we are able to compute $y(t) \pm e^{-\mu}$ in time polynomial in t, μ and $\operatorname{len}_y(0,t)$.

At this point, there is a slight subtlety: intuitively we would like to evaluate y at time $\Pi(|w|)$ but it could be that the length of the curve is exponential at this time.

Fortunately, the algorithm that solves the PIVP works by making small time steps, and at each step the length cannot increase by more than a constant[15]. This means that we can stop the algorithm as soon as the length is greater than $\Pi(|w|)$. Let t^* be the time at which the algorithm stops. Then the running time of the algorithm will be polynomial in t^*, μ and $\operatorname{len}_y(0,t^*) \leqslant \Pi(|w|) + \mathcal{O}(1)$. Finally, thanks to the technical condition, $t^* \leqslant \operatorname{len}_y(0,t^*)$ so this algorithm has running time polynomial in $|w|$ and μ. Take $\mu = \ln 2$ then we get $\tilde{y}$ such that $\|y(t^*) - \tilde{y}\| \leqslant \frac{1}{2}$. By definition of Π, $y_1(t) \geqslant 1$ or $y_1(t) \leqslant -1$ so we can decide from $\tilde{y}_1$ if $w \in \mathcal{L}$ or not. ■

8 A Characterization of Computable Analysis

8.1 Computable Analysis

There exist many equivalent definitions of polynomial-time computability in the framework of Computable Analysis. In this paper, we will use a particular characterization by [Ko91] in terms of computable rational approximation and modulus of continuity. In the next theorem (which can be found e.g. in [Wei00]), $\mathbb{D}$ denotes the set of dyadic rationals:

$$\mathbb{D} = \{m2^{-n}, m \in \mathbb{Z}, n \in \mathbb{N}\}.$$

Theorem 69 (Alternative definition of computable functions) *A real function $f : [a,b] \to \mathbb{R}$ is computable (resp. polynomial time computable) if and only if there exists a computable (resp. polynomial time computable[16]) function $\psi : (\mathbb{D}\cap[a,b])\times\mathbb{N} \to \mathbb{D}$ and a computable (resp. polynomial) function $m : \mathbb{N} \to \mathbb{N}$ such that:*

- *m is a modulus of continuity for f*
- *for any $n \in \mathbb{N}$ and $d \in [a,b] \cap \mathbb{D}$, $|\psi(d,n) - f(d)| \leqslant 2^{-n}$*

[15] For the unconvinced reader, it is still possible to write this argument formally by running the algorithm for increasing values of t, starting from a very small value and making sure that at each step the increase in the length of the curve is at most constant. This is very similar to how Theorem 53 is proved.

[16] The second argument of g must be in unary.

This characterization is very useful for us because it does not involved the notion of oracle, which would be difficult to formalize with differential equation. However, in one direction of the proofs, it will be useful to have the following variation of the previous theorem:

Theorem 70 (Alternative characterization of computable functions) *A real function $f : [a,b] \to \mathbb{R}$ is polynomial time computable if and only if there exists a polynomial $q : \mathbb{N} \to \mathbb{N}$, a polynomial time computable*[17] *function $\psi : X_q \to \mathbb{D}$ such that*

$$\text{for all } x \in [a,b] \text{ and } (r,n) \in X_q(x), |\psi(r,n) - f(x)| \leqslant 2^{-n}$$

where

$$X_q = \bigcup_{x\in[a,b]} X_q(x),$$

$$X_q(x) = \{(r,n) \in \mathbb{D} \times \mathbb{N} : |r-x| \leqslant 2^{-q(n)}\}.$$

Proof. This is a folklore result that directly follows from the oracle definition. ■

8.2 Mixing functions

Suppose that we have two continuous functions f_0 and f_1 that partially cover $\mathbb{R}$ but such that $\operatorname{dom} f_0 \cup \operatorname{dom} f_1 = \mathbb{R}$. We would like to build a new continuous function defined over $\mathbb{R}$ out of them. One way of doing this is to build a function f that equals f_0 over $\operatorname{dom} f_0 \setminus \operatorname{dom} f_1$, f_1 over $\operatorname{dom} f_1 \setminus \operatorname{dom} f_0$ and a linear combination of both in between. For example consider $f_0(x) = x^2$ defined over $]-\infty, 1]$ and $f_1(x) = x$ over $[0, \infty[$. This approach may work from a mathematical point of view, but it raises severe computational issues: how do we describe the two domains ? How do we compute a linear interpolation between arbitrary sets ? What is the complexity of this operation ? This would require to discuss the complexity of real sets, which is a whole subject by itself.

A more elementary solution to this problem is what we call *mixing*. We assume that we are given an indicator function i that covers the domain of both functions. Such an example would be $i(x) = x$ in the previous example. The intuition is that i describes both the domains and the interpolation. Precisely, the resulting function should be $f_0(x)$ if $i(x) \leqslant 0$, $f_1(x)$ if $i(x) \geqslant 1$ and a *mix* of $f_0(x)$ and $f_1(x)$ inbetween. The consequence of this choice is that the domain of f_0 and f_1 must overlap on the region $\{x : 0 < i(x) < 1\}$. In the previous example, we need to define f_0 over $]-\infty, 1[= \{x : i(x) < 1\}$ and f_1 over $]0, \infty] = \{x : i(x) > 0\}$. Several types of mixing are possible, the simplest being linear interpolation: $(1-i(x))f_0(x) + i(x)f_1(x)$. Formally, we would build the following continuous function:

Definition 71 (Mixing function) *Let $f_0 :\subseteq \mathbb{R}^n \to \mathbb{R}^d$, $f_1 :\subseteq \mathbb{R}^n \to \mathbb{R}^d$ and $i :\subseteq \mathbb{R}^n \to \mathbb{R}$. Assume that $\{x : i(x) < 1\} \subseteq \operatorname{dom} f_0$ and $\{x : i(x) > 0\} \subseteq \operatorname{dom} f_1$, and define the function $\operatorname{mix}(i, f_0, f_1) :\subseteq \mathbb{R}^n \to \mathbb{R}^d$ by:*

$$\operatorname{mix}(i, f_0, f_1)(x) = \begin{cases} f_0(x) & \text{if } i(x) \leqslant 0 \\ (1-i(x))f_0(x) + i(x)f_1(x) & \text{if } 0 < i(x) < 1 \\ f_1(x) & \text{if } i(x) \geqslant 1 \end{cases}.$$

[17] The second argument of g must be in unary.

where for $x \in \operatorname{dom} i$*.*

From closure properties, we get immediately:

Theorem 72 (Closure by mixing) *Let* $f_0 :\subseteq \mathbb{R}^n \to \mathbb{R}^d$*,* $f_1 :\subseteq \mathbb{R}^n \to \mathbb{R}^d$ *and* $i :\subseteq \mathbb{R}^n \to \mathbb{R}$*. Assume that* $f_0, f_1, i \in \mathrm{ALP}$ *, that* $\{x : i(x) < 1\} \subseteq \operatorname{dom} f_0$ *and that* $\{x : i(x) > 0\} \subseteq \operatorname{dom} f_1$*. Then* $\operatorname{mix}(i, f_0, f_1) \in \mathrm{ALP}$ *.*

Proof. By taking $\min(\max(0, i(x)), 1)$, which belongs to ALP, we can assume that $i(x) \in [0, 1]$. Furthermore, it is not hard to see that

$$\operatorname{mix}(i, f_0, f_1)(x) = \operatorname{mix}(i, 0, f_1)(x) + \operatorname{mix}(1 - i, 0, f_0)(x).$$

Thus we only need prove the result for the case where $f_0 \equiv 0$, that is

$$g(x) = \begin{cases} 0 & \text{if } \alpha(x) = 0 \\ \alpha(x)f(x) & \text{if } \alpha(x) > 0 \end{cases}.$$

Recall that by assumption, $f(x)$ is defined for $\alpha(x) > 0$ but may not be defined for $\alpha(x) = 0$. The idea is use Item (4) of Proposition 12 (online-computability): let δ, d, p, y_0 and d', q, z_0 that correspond to f and α respectively. Consider the following system for all $x \in \operatorname{dom} \alpha$:

$$\begin{aligned} y(0) &= y_0, \qquad y'(t) = p(y(t), x), \\ z(0) &= z_0, \qquad z'(t) = q(y(t), x), \\ &w(t) = y(t)z(t). \end{aligned}$$

There are two cases:

- If $\alpha(x) > 0$ then $x \in \operatorname{dom} f$ thus $y(t) \to f(x)$ and $z(t) \to \alpha(x)$ as $t \to \infty$. It follows that $w(t) \to \alpha(x)f(x) = g(x)$ as $t \to \infty$. We leave the convergence speed analysis to the reader since it's standard.

- If $\alpha(x) = 0$ then we have no guarantee on the convergence of y. However we know that
$$\|y(t)\| \leqslant \Upsilon(\|x\|, t)$$
where and Υ is a polynomial, and
$$|z(t) - \alpha(x)| \leqslant e^{-\mu} \qquad \text{for all } t \geqslant \amalg(\|x\|, \mu).$$
Thus for all $\mu \in \mathbb{R}_+$,
$$\begin{aligned} \|w(\amalg(\|x\|, \mu))\| &= \|z(t)y(t)\| \\ &= \Upsilon(\|x\|, \amalg(\|x\|, \mu))e^{-\mu}. \end{aligned}$$
But since Υ and $\amalg$ are polynomials, the right-hand side converges exponentially fast (in μ) to 0 whereas the time $\amalg(\|x\|, \mu)$ only grows polynomially.

This shows that $g \in \mathrm{ALP}$. ■

8.3 Computing effective limits

Intuitively, our notion of computation already contains the notion of effective limit. More precisely, if f is computable and is such that $f(x,t) \to g(x)$ when $t \to \infty$ effectively uniformly on x, then g is computable. The result below extends this result to the case where the limit is restricted to $t \in \mathbb{N}$. The intuition behind this result is that if we have $f(x,n) \to g(x)$ as $n \in \mathbb{N} \to \infty$, we can consider $h(x,t) = g(x, \lceil t \rceil)$ and then $h(x,t) \to g(x)$ as $t \in \mathbb{R} \to \infty$. The problem is that $t \mapsto \lceil t \rceil$ is not computable over $\mathbb{R}$. We can solve this problem by mixing $g(x, \lfloor t \rfloor)$ and $g(x, \lceil t \rceil)$: the first is computable over $\bigcup_{n\in\mathbb{N}} \left[n - \frac{1}{3}, n + \frac{1}{3}\right]$, and the second over $\bigcup_{n\in\mathbb{N}} \left[n + \frac{1}{6}, n + \frac{5}{6}\right]$. Indeed by introducing gaps in the domain of definition, we avoid the continuity problem, and by mixing the two we can cover all of $\mathbb{R}$. Indeed, the domains of definition of the two function overlap over $\bigcup_{n\in\mathbb{N}} \left[n + \frac{1}{6}, n + \frac{1}{3}\right]$, which provides a smooth transition between the functions.

Theorem 73 (Closure by effective limit) *Let $I \subseteq \mathbb{R}^n$, $f :\subseteq I \times \mathbb{N} \to \mathbb{R}^m$, $g : I \to \mathbb{R}^m$ and $\mho : \mathbb{R}_+^2 \to \mathbb{R}_+$ be a nondecreasing polynomial. Assume that $f \in \mathrm{ALP}$ and that*

$$\{(x,n) \in I \times \mathbb{N} : n \geqslant \mho(\|x\|, 0)\} \subseteq \operatorname{dom} f.$$

Further assume that for all $(x,n) \in \operatorname{dom} f$ and $\mu \geqslant 0$,

$$\text{if } n \geqslant \mho(\|x\|, \mu) \text{ then } \|f(x,n) - g(x)\| \leqslant e^{-\mu}.$$

Then $g \in \mathrm{ALP}$.

Proof. First note that $\frac{1}{2} - e^{-2} \geqslant \frac{1}{3}$ and define for $x \in I$ and $n \geqslant \mho(\|x\|, 0)$:

$$\begin{array}{ll} f_0(x,\tau) = f(x, \operatorname{rnd}(\tau, 2)) & \tau \in \left[n - \frac{1}{3}, n + \frac{1}{3}\right], \\ f_1(x,\tau) = f(x, \operatorname{rnd}(\tau + \frac{1}{2}, 2)) & \tau \in \left[n + \frac{1}{6}, n + \frac{5}{6}\right]. \end{array}$$

By Definition 31 and hypothesis on f, both are well-defined because for all $n \geqslant \mho(\|x\|, 0)$ and $\tau \in \left[n - \frac{1}{3}, n + \frac{1}{3}\right]$,

$$(x, \operatorname{rnd}^*(\tau, 2)) = (x, n) \in \operatorname{dom} f$$

and similarly for f_1. Also note that their domain of definition overlap on $[n + \frac{1}{6}, n + \frac{1}{3}]$ and $[n + \frac{2}{3}, n + \frac{5}{6}]$. Apply Theorem 32 and Theorem 21 to get that $f_0, f_1 \in \mathrm{ALP}$. We also need to build the indicator function: this is where the choice of above values will prove convenient. Define for any $x \in I$ and $\tau \geqslant \mho(\|x\|, 0)$:

$$i(x,\tau) = \tfrac{1}{2} - \cos(2\pi\tau).$$

It is now easy to check that:

$$\{(x,\tau) : i(x) < 1\} = I \times \bigcup_{n \geqslant \mho(\|x\|,0)} \left]n - \tfrac{1}{3}, n + \tfrac{1}{3}\right[\subseteq \operatorname{dom} f_0.$$

$$\{(x,\tau) : i(x) > 0\} = I \times \bigcup_{n \geqslant \mho(\|x\|,0)} \left]n + \tfrac{1}{6}, n + \tfrac{5}{3}\right[\subseteq \operatorname{dom} f_1.$$

Define for any $x \in I$ and $\mu \in \mathbb{R}_+$:

$$f^*(x,\mu) = \operatorname{mix}(i, f_0, f_1)(x, \mho(\operatorname{norm}_{\infty,1}(x), \mu)).$$

Recall that $\operatorname{norm}_{\infty,1}$, defined in Lemma 33, belongs to ALPand satisfies $\operatorname{norm}_{\infty,1}(x) \geqslant \|x\|$. We can thus apply Theorem 72 to get that $f^* \in \mathrm{ALP}$. Note that f^* is defined over $I \times \mathbb{R}_+$ since for all $x \in I$ and $\mu \geqslant 0$, $\mho(\operatorname{norm}_{\infty,1}(x), \mu) \geqslant \mho(\|x\|, 0)$ since $\mho$ is nondecreasing. We now claim that for any $x \in I$ and $\mu \in \mathbb{R}_+$, if $\tau \geqslant 1 + \mho(\|x\|, \mu)$ then $\|f^*(x,\tau) - g(x)\| \leqslant 2e^{-\mu}$. There are three cases to consider, illustrated in Figure 11:

- If $\tau \in [n - \frac{1}{6}, n + \frac{1}{6}]$ for some $n \in \mathbb{N}$ then $i(x) \leqslant 0$ so $\operatorname{mix}(i, f_0, f_1)(x,\tau) = f_0(x,\tau) = f(x,n)$ and since $n \geqslant \tau - \frac{1}{6}$ then $n \geqslant \mho(\|x\|, \mu)$ thus $\|f^*(x,\tau) - g(x)\| \leqslant e^{-\mu}$.
- If $\tau \in [n + \frac{1}{3}, n + \frac{2}{3}]$ for some $n \in \mathbb{N}$ then $i(x) \geqslant 1$ so $\operatorname{mix}(i, f_0, f_1)(x,\tau) = f_1(x,\tau) = f(x, n+1)$ and since $n \geqslant \tau - \frac{2}{3}$ then $n + 1 \geqslant \mho(\|x\|, \mu)$ thus $\|f^*(x,\tau) - g(x)\| \leqslant e^{-\mu}$.
- If $\tau \in [n+\frac{1}{6}, n+\frac{1}{3}] \cup [n+\frac{2}{3}, n+\frac{5}{6}]$ for some $n \in \mathbb{N}$ then $\|f^*(x,\tau) - g(x)\| \leqslant e^{-\mu}$ from Theorem 72 since $i(x,\tau) \in [0,1]$ so $f^*(x,\tau) = (1 - i(x,\tau)) f_0(x,\tau) + i(x,\tau) f_1(x,\tau) = (1 - i(x,\tau)) f(x, \lfloor \tau \rceil) + i(x,\tau) f(x, \lfloor \tau + \frac{1}{2} \rceil)$. Since $\lfloor \tau \rceil, \lfloor \tau + \frac{1}{2} \rceil \geqslant \mho(\|x\|, \mu)$, we get that $\|f(x, \lfloor \tau \rceil) - g(x)\| \leqslant e^{-\mu}$ and $\left\|f(x, \lfloor \tau + \frac{1}{2} \rceil) - g(x)\right\| \leqslant e^{-\mu}$ thus $\|f^*(x,\tau) - g(x)\| \leqslant 2e^{-\mu}$ because $|i(x,\tau)| \leqslant 1$.

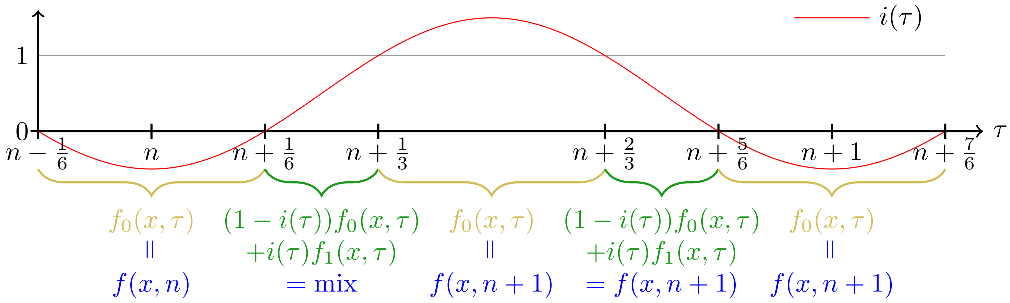


Figure 11: The various cases of the proof of Theorem 73: we use mixing to continuously choose between $f(x,n)$ and $f(x,n+1)$ as τ ranges over $[n, n+1]$. Note that $f_0(x,\tau) = f(x, \lfloor \tau + \frac{1}{2} \rfloor)$ and $f_1(x,\tau) = f(x, \lfloor \tau + 1 \rfloor)$ over some well-chosen intervals.

It follows that g is the effective limit of f^* and thus $g \in \mathrm{ALP}$ (see Remark 13). ∎

Remark 74 (Optimality) *The condition that $\mho$ is a polynomial is essentially optimal. Intuitively, if $f \in \mathrm{ALP}$ and satisfies $\|f(x,\tau) - g(x)\| \leqslant e^{-\mu}$ whenever $\tau \geqslant \mho(\|x\|, \mu)$ then $\mho$ is a modulus of continuity for g. By Theorem 22, if $g \in \mathrm{ALP}$ then it admits a polynomial modulus of continuity so $\mho$ must be a polynomial. For a formal proof of this intuition, see examples 75 and 76.*

Example 75 ($\mho$ must be polynomial in x) *Let $f(x,\tau)=\min(e^x,\tau)$ and $g(x)=e^x$. Trivially $f(x,\cdot)$ converges to g because $f(x,\tau)=g(x)$ for $\tau\geqslant e^x$. But $g\notin \mathrm{ALP}$ because it is not polynomially bounded. In this case $\mho(x,\mu)=e^x$ which is exponential and $f\in\mathrm{ALP}$ by Proposition 30.*

Example 76 ($\mho$ must be polynomial in μ) *Let $g(x)=\frac{-1}{\ln x}$ for $x\in[0,e]$ which is defined in 0 by continuity. Observe that $g\notin\mathrm{ALP}$, since its modulus of continuity is exponential around 0 because $g(e^{-e^\mu})=e^{-\mu}$ for all $\mu\geqslant 0$. However note that $g^*\in\mathrm{ALP}$ where $g^*(x)=g(e^{-x})=\frac{1}{x}$ for $x\in[1,+\infty[$. Let $f(x,\tau)=g^*(\min(-\ln x,\tau))$ and check, using that g is increasing and non-negative, that: $|f(x,\tau)-g(x)|=|g(\max(x,e^{-\tau}))-g(x)|\leqslant g(\max(x,e^{-\tau}))\leqslant\frac{1}{\tau}$. Thus $\mho(\|x\|,\mu)=e^\mu$ which is exponential and $f\in\mathrm{ALP}$ because $(x,\tau)\mapsto\min(-\ln x,\tau)\in\mathrm{ALP}$ by a proof similar to Proposition 30.*

8.4 Cauchy completion and complexity

We want to approach a function f defined over some domain $\mathcal{D}$ by some function g, where g is defined over

$$\left\{\left(\frac{p}{2^n},n\right),p\in\mathbb{Z}^d,n\in\mathbb{N}:\frac{p}{2^n}\in\mathcal{D}\right\},$$

the set of dyadic numbers in $\mathcal{D}$ (we need to include the precision n as argument for complexity reasons). Here, f is implicitely defined as $f(x)=\lim_{\frac{p}{2^n}\to x}g(\frac{p}{2^n},n)$. This is somewhat similar to Section 8.3 but with an extra difficulty since $\mathcal{D}$ can be arbitrary. The problem is that the shape of the domain $\mathcal{D}$ matters: if we want to compute $f(x)$, we will need to "approach" x from within the domain, since the above domain only allows dyadic numbers in $\mathcal{D}$. For example if f is defined over $[a,b]$ then to compute $f(a)$ we need to approach a *by above*, but for $f(b)$, we need to approach b *by below*. For more general domains, finding the right direction of approach might be (computationally) hard, if even possible, and depends on the shape of the domain. To avoid this problem, we requires that g be defined on a slightly larger domain so that this problem disappears. This notion is motivated by Theorem 70. Even with this assumption, the proof is nontrivial because of the difficulty to generate a converging dyadic sequence, see Section 9.2 for more details.

Theorem 77 *Let $d,e,\ell\in\mathbb{N}$, $\mathcal{D}\subseteq\mathbb{R}^{d+e}$, $k\geqslant 2$ and $f:\mathcal{D}\to\mathbb{R}^\ell$. Assume that there exists a polynomial $\mho:\mathbb{R}_+^2\to\mathbb{R}_+$ and $(g:\subseteq\mathbb{D}^d\times\mathbb{N}\times\mathbb{R}^e\to\mathbb{R}^\ell)\in\mathrm{ALP}$ such that for all $(x,y)\in\mathcal{D}$ and $n,m\in\mathbb{N},p\in\mathbb{Z}^d$,*

if $\left\|\frac{p}{2^m}-x\right\|\leqslant 2^{-m}$ and $m\geqslant\mho(\|(x,y)\|,n)$ then[18] $\left\|g(\frac{p}{2^m},m,y)-f(x,y)\right\|\leqslant 2^{-n}$.

Then $f\in\mathrm{ALP}$.

Section 9.2 is devoted to the proof of this theorem. We now show that this is sufficient to characterize Computable Analysis using continuous time systems.

[18]The domain of definition of g is exactly those points $(\frac{p}{2^m},m,y)$ that satisfy the previous "if".

8.5 From Computable Analysis to ALP

Theorem 78 (From Computable Analysis to ALP) *For any $a, b \in \mathbb{R}$, any generable field $\mathbb{K}$ such that $\mathbb{R}_G \subseteq \mathbb{K} \subseteq \mathbb{R}_P$, if $f \in C^0([a,b],\mathbb{R})$ is polynomial-time computable then $f \in \mathrm{ALP}$.*

Note that a and b need not be computable so we must take care not to use them in any computation!

Proof. Let $f \in C^0([a,b],\mathbb{R})$ and assume that f is polynomial-time computable. We will first reduce the general situation to a simpler case. Let $m, M \in \mathbb{Q}$ such that $m < f(x) < M$ for all $x \in [a,b]$. Let $l, r \in \mathbb{Q}$ such that $l \leqslant a < b \leqslant r$. Define

$$g(\alpha) = \frac{1}{4} + \frac{f(l + (r-l)(2\alpha - \frac{1}{2})) - m}{2(M-m)}$$

for all $\alpha \in [a', b'] = \left[\frac{1}{4} + \frac{a-l}{2(r-l)}, \frac{1}{4} + \frac{b-l}{2(r-l)}\right] \subseteq [\frac{1}{4}, \frac{3}{4}]$. It follows that $g \in C^0([a',b'],[\frac{1}{4},\frac{3}{4}])$ with $[a',b'] \subseteq [\frac{1}{4},\frac{3}{4}]$. Furthermore, by construction, for every $x \in [a,b]$ we have that

$$f(x) = 2(M-m)\left(g\left(\frac{1}{4} + \frac{x-l}{2(r-l)}\right) - \frac{1}{4}\right) + m.$$

Thus if $g \in \mathrm{ALP}$ then $f \in \mathrm{ALP}$ because of closure properties of ALP. Hence, in the remaining of the proof, we can thus assume that $f \in C^0([a,b], \frac{1}{4}, \frac{3}{4})$ with $[a,b] \subseteq [\frac{1}{4},\frac{3}{4}]$. This restriction is useful to simplify the encoding used later in the proof.

Let $f \in C^0([a,b],[\frac{1}{4},\frac{3}{4}])$ with $[a,b] \subseteq [\frac{1}{4},\frac{3}{4}]$ be a polynomial time computable function. Apply Theorem 70 to get g and $\mho$ (we renamed ψ to g and q to $\mho$ to avoid a name clash). Note that $g : X_\mho \to \mathbb{D}$ has its second argument written in unary. In order to apply the FP characterization, we need to discuss the encoding of rational numbers and unary integers. Let us choose a binary alphabet $\Gamma = \{0,1\}$ and its encoding function $\gamma(0) = 1$ and $\gamma(1) = 2$, and define for any $w, w' \in \Gamma^*$:

$$\psi_{\mathbb{N}}(w) = |w|, \qquad \psi_{\mathbb{D}}(w) = \sum_{i=1}^{|w|} w_i 2^{-i}.$$

Note that $\psi_{\mathbb{D}}$ is a surjection from Γ^* to $\mathbb{D} \cap [0,1[$, the dyadic part of $[0,1[$. Define for any relevant[19] $w, w' \in \Gamma^*$:

$$g_\Gamma(w,w') = \psi_{\mathbb{D}}^{-1}(g(\psi_{\mathbb{D}}(w), \psi_{\mathbb{N}}(w'))$$

where $\psi_{\mathbb{D}}^{-1}(x)$ is the smallest w such $\psi_{\mathbb{D}}(w) = x$ (it is unique). For $g_\Gamma(w,w')$ to be defined, we need that

- $(\psi_{\mathbb{D}}(w), \psi_{\mathbb{N}}(w')) \in \operatorname{dom} g = X_\mho$: in the case of interest, this is true if

$$\psi_{\mathbb{D}}(w) \in \left[a' - 2^{-\mho(|a'|,|w'|)}, b' + 2^{-\mho(|b'|,|w'|)}\right],$$

[19] We will discuss the domain of definition below.

- $g(\psi_{\mathbb{D}}(w), \psi_{\mathbb{N}}(w')) \in \operatorname{dom} \psi_{\mathbb{D}}^{-1} = \mathbb{D} \cap [0,1[$: since $|g(\psi_{\mathbb{D}}(w), \psi_{\mathbb{N}}(w')) - f(\psi_{\mathbb{D}}(w))| \leqslant 2^{-\psi_{\mathbb{N}}(w')}$ and $f(\psi_{\mathbb{D}}(w) \in [\frac{1}{4}, \frac{3}{4}]$, then it is true when $\psi(w') = |w'| \geqslant 3$ because

$$g(\psi_{\mathbb{D}}(w), \psi_{\mathbb{N}}(w')) \in f(\psi_{\mathbb{D}}(w) + [-2^{-3}, 2^{-3}] \subseteq [\tfrac{1}{4}, \tfrac{3}{4}] + [-\tfrac{1}{8}, \tfrac{1}{8}] \subset [0,1].$$

Since $\psi_{\mathbb{D}}$ is a polytime computable encoding, then $g_\Gamma \in \mathrm{FP}$ because it has running time polynomial in the length of $\psi_{\mathbb{D}}(w)$ and the (unary) value of $\psi_{\mathbb{N}}(w')$, which are the length of w and w' respectively, by definition of $\psi_{\mathbb{D}}$ and $\psi_{\mathbb{N}}$. Apply Theorem 66 to get that g_Γ is emulable. Thus there exist $h \in \mathrm{ALP}$ and $k \in \mathbb{N}$ such that for all $w, w' \in \operatorname{dom} g_\Gamma$:

$$h(\psi_k(w, w')) = \psi_k(g_\Gamma(w, w')).$$

where ψ_k is defined as in Definition 65. At this point, everything is encoded: the input and the output of h. Our next step is to get rid of the encoding by building a function that works the dyadic part of $[a, b]$ and returns a real number.

Define $\kappa : [\![0, k-2]\!] \to \{0,1\}$ by $\kappa(\gamma(0)) = 0$ and $\kappa(\gamma(1)) = 1$ and $\kappa(\alpha) = 0$ otherwise. Define $\iota : \{0,1\} \to [\![0, k-2]\!]$ by $\iota(0) = \gamma(0)$ and $\iota(1) = \gamma(1)$. For any relevant $q \in \mathbb{D}$ and $n, m \in \mathbb{N}$ define:

$$g^*(q, n, p) = \mathrm{reenc}_{\kappa,1}(h(\mathrm{reenc}_\iota(q, n), 0, p)).$$

We will see that this definition makes sense for some values. Let $n \in \mathbb{N}$, $p \geqslant 3$ and $m \in \mathbb{Z}$, write $q = m2^{-n}$ and assume that $m2^{-n} \in \left[a' - 2^{-\mho(|a'|,p)}, b' + 2^{-\mho(|b'|,p)}\right] \subseteq [0,1[$. Then there exists $w^q \in \{0,1\}^n$ such that $m2^{-n} = \sum_{i=1}^n w_i^q 2^{-i}$. Consequently,

$$\begin{aligned}
\mathrm{reenc}_\iota(q, n) &= \mathrm{reenc}_\iota\left(\sum_{i=1}^n w_i^q 2^{-i}, n\right) && \text{By Corollary 61} && (5)\\
&= \left(\sum_{i=1}^n \iota(w_i^q) k^{-i}, n\right) && \text{By definition of } \mathrm{reenc}_\iota && (6)\\
&= \left(\sum_{i=1}^n \gamma(w_i^q) k^{-i}, n\right) && \text{Because } \iota = \gamma && (7)\\
&= \psi_k(w^q). && && (8)
\end{aligned}$$

Furthermore, note that by definition of w^q:

$$\psi_{\mathbb{D}}(w^q) = \sum_{i=1}^{|w^q|} w_i^q 2^{-i} = q. \tag{9}$$

Similarly, note that

$$(0, p) = \left(\sum_{i=1}^p 0k^{-i}, p\right) = \psi_k(0^p) \tag{10}$$

and

$$\psi_{\mathbb{N}}(0^p) = |0^p| = p. \tag{11}$$

Additionally, for any $w \in \Gamma^*$ we have that

$$\begin{aligned}
\mathrm{reenc}_{\kappa,1}(\psi_k(w)) &= \mathrm{reenc}_{\kappa,1}\left(\sum_{i=1}^{|w|}\gamma(w_i)k^{-i}, |w|\right) && \text{By definition of } \psi_k\\
&= \sum_{i=1}^{|w|}\kappa(\gamma(w_i))2^{-i} && \text{By Corollary 61}\\
&= \sum_{i=1}^{|w|} w_i 2^{-i} && \text{Because } \kappa\circ\gamma = \mathrm{id}\\
&= \psi_{\mathbb{D}}(w). && (12)
\end{aligned}$$

Putting everything together, we get that

$$\begin{aligned}
g^*(q,n,p) &= \mathrm{reenc}_{\kappa,1}(h(\mathrm{reenc}_\iota(q,n),0,p))\\
&= \mathrm{reenc}_{\kappa,1}(h(\psi_k(w^q,0^p))) && \text{By (8) and (10)}\\
&= \mathrm{reenc}_{\kappa,1}(\psi_k(g_\Gamma(w^q,0^p))) && \text{By definition of } h\\
&= \mathrm{reenc}_{\kappa,1}(\psi_k(\psi_{\mathbb{D}}^{-1}(g(\psi_{\mathbb{D}}(w^q),\psi_{\mathbb{N}}(0^p))))) && \text{By definition of } g_\Gamma\\
&= \mathrm{reenc}_{\kappa,1}(\psi_k(\psi_{\mathbb{D}}^{-1}(g(q,p)))) && \text{By (9) and (11)}\\
&= \psi_{\mathbb{D}}(\psi_{\mathbb{D}}^{-1}(g(q,p))) && \text{By (12)}\\
&= g(q,p). && (13)
\end{aligned}$$

Finally, $g^* \in \mathrm{ALP}$ because $\mathrm{reenc}_\kappa, \mathrm{reenc}_\iota \in \mathrm{ALP}$ by Corollary 61. Finally for any relevant $n \geqslant 3$ and $q \in \mathbb{D}$, let

$$\tilde{g}(q,n) = g^*(q,n,n).$$

Clearly $\tilde{g} \in \mathrm{ALP}$. We will show that $\tilde{g}$ satisfies the assumption of Theorem 77. Let $x \in [a,b]$, $m,n \in \mathbb{N}$ and $p \in \mathbb{Z}$ such that

$$\left|x - \tfrac{p}{2^m}\right| \leqslant 2^{-m} \text{ and } m \geqslant \mho(|x|, n+2) + n + 2.$$

Then[20]

$$\begin{aligned}
|\tilde{g}(\tfrac{p}{2^m},m) - f(x)| &= |g^*(\tfrac{p}{2^m},m,m) - f(x)|\\
&= |g(\tfrac{p}{2^m},m) - f(x)| && \text{by (13)}\\
&\leqslant |g(\tfrac{p}{2^m},m) - f(\tfrac{p}{2^m})| + |f(\tfrac{p}{2^m}) - f(x)|
\end{aligned}$$

But for any rational q, $|g(q,n) - f(q)| \leqslant 2^{-n}$ for all $n \in \mathbb{N}$,

$$\begin{aligned}
&\leqslant 2^{-m} + |f(\tfrac{p}{2^m}) - f(x)|\\
&\leqslant 2^{-m} + |f(\tfrac{p}{2^m}) - g(\tfrac{p}{2^m},n+2)| + |g(\tfrac{p}{2^m},n+2) - f(x)|
\end{aligned}$$

But for any rational q, $|g(q,n) - f(q)| \leqslant 2^{-n}$ for all $n \in \mathbb{N}$,

$$\leqslant 2^{-m} + 2^{-n-2} + |g(\tfrac{p}{2^m},n+2) - f(x)|$$

[20] The proof is a bit involved because we naturally have $g(\frac{p}{2^m},m)$ with $m \geqslant \mho(|x|,n)$ but we want $g(\frac{p}{2^m},n)$ to apply Theorem 77.

But $|x - \frac{p}{2^m}| \leqslant 2^{-m} \leqslant 2^{-\mho(|x|,n+2)}$ so we can apply Theorem 70,

$$\begin{aligned}
&\leqslant 2^{-m} + 2^{-n-2} + 2^{-n-2} \\
&\leqslant 3 \cdot 2^{-n-2} && \text{since } m \geqslant n+2 \\
&\leqslant 2^{-n}.
\end{aligned}$$

Thus we can apply Theorem 77 to $\tilde{g}$ and get that $f \in \mathrm{ALP}$. ■

8.6 Equivalence with Computable Analysis

Note that the characterization works over $[a, b]$ where a and b can be arbitrary real numbers.

Theorem 79 (Equivalence with Computable Analysis) *For any $f \in C^0([a, b], \mathbb{R})$, f is polynomial-time computable if and only if $f \in \mathrm{ALP}$.*

Proof. The proof of the missing direction of the theorem is the following: Let $f \in \mathrm{ALP}$. Then $f \in \mathrm{ATSC}(\Upsilon, \amalg)$ where $\Upsilon, \amalg$ are polynomials which we can assume to be increasing functions, and corresponding d, p and q. Apply Theorem 22 to f to get $\mho$ and define

$$m(n) = \tfrac{1}{\ln 2}\mho(\max(|a|, |b|), n \ln 2).$$

It follows from the definition that m is a modulus of continuity of f since for any $n \in \mathbb{N}$ and $x, y \in [a, b]$ such that $|x - y| \leqslant 2^{-m(n)}$ we have:

$$|x - y| \leqslant 2^{-\frac{1}{\ln 2}\mho(\max(|a|,|b|),n \ln 2)} = e^{-\mho(\max(|a|,|b|),n \ln 2)} \leqslant e^{-\mho(|x|,n \ln 2)}.$$

Thus $|f(x) - f(y)| \leqslant e^{-n \ln 2} = 2^{-n}$. We will now see how to approximate f in polynomial time. Let $r \in \mathbb{Q}$ and $n \in \mathbb{N}$. We would like to compute $f(r) \pm 2^{-n}$. By definition of f, there exists a unique $y : \mathbb{R}_+ \to \mathbb{R}^d$ such that for all $t \in \mathbb{R}_+$:

$$y(0) = q(r) \qquad y'(t) = p(y(t).$$

Furthermore, $|y_1(\amalg(|r|, \mu)) - f(r)| \leqslant e^{-\mu}$ for any $\mu \in \mathbb{R}_+$ and $\|y(t)\| \leqslant \Upsilon(|r|, t)$ for all $t \in \mathbb{R}_+$. Note that since the coefficients of p and q belongs to $\mathbb{R}_P$, it follows that we can apply Theorem 53 to compute y. More concretely, one can compute a rational r' such that $|y(t) - r'| \leqslant 2^{-n}$ in time bounded by

$$\mathrm{poly}(\deg(p), \mathrm{PsLen}(0, t), \log \|y(0)\|, \log \Sigma p, -\log 2^{-n})^d.$$

Recall that in this case, all the parameters $d, \Sigma p, \deg(p)$ only depend on f and are thus fixed and that $|r|$ is bounded by a constant. Thus these are all considered constants. So in particular, we can compute r' such that $|y(\amalg(|r|, (n+1)\ln 2) - r'| \leqslant 2^{-n-1}$ in time:

$$\mathrm{poly}(\mathrm{PsLen}(0, \amalg(|r|, (n+1)\ln 2)), \log \|q(r)\|, (n+1)\ln 2).$$

Note that $|r| \leqslant \max(|a|, |b|)$ and since a and b are constants and q is a polynomial, $\|q(r)\|$ is bounded by a constant. Furthermore,

$$\mathrm{PsLen}(0, \amalg(|r|, (n+1)\ln 2)) = \int_0^{\amalg(|r|,(n+1)\ln 2)} \max(1, \|y(t)\|)^{\deg(p)} dt$$

$$
\begin{aligned}
&\leqslant \int_0^{\amalg(|r|,(n+1)\ln 2)} \mathrm{poly}(\Upsilon(\|r\|,t))dt \\
&\leqslant \amalg(|r|,(n+1)\ln 2)\,\mathrm{poly}(\Upsilon(|r|,\amalg(|r|,(n+1)\ln 2)))dt \\
&\leqslant \mathrm{poly}(|r|,n) \leqslant \mathrm{poly}(n).
\end{aligned}
$$

Thus r' can be computed in time:

$$\mathrm{poly}(n).$$

Which is indeed polynomial time since n is written in unary. Finally:

$$
\begin{aligned}
|f(r)-r'| &\leqslant |f(r)-y(\amalg(|r|,(n+1)\ln 2))| + |y(\amalg(|r|,(n+1)\ln 2)) - r'| \\
&\leqslant e{-}(n+1)\ln 2 + 2^{-n-1} \\
&\leqslant 2^{-n}.
\end{aligned}
$$

This shows that f is polytime computable. ■

Remark 80 (Domain of definition) *The equivalence holds over any interval $[a,b]$ but it can be extended in several ways. First it is possible to state an equivalence over $\mathbb{R}$. Indeed, classical real computability defines the complexity of $f(x)$ over $\mathbb{R}$ as polynomial in n and k where n is the precision and k the length of input, defined by $x \in [-2^k, 2^k]$. Secondly, the equivalence also holds for multidimensional domains of the form $I_1 \times I_2 \times \cdots \times I_n$ where $I_k = [a_k, b_k]$ or $I_k = \mathbb{R}$. However, extending this equivalence to partial functions requires some caution. Indeed, our definition does not specify the behavior of functions outside of the domain, whereas classical discrete computability and some authors in Computable Analysis mandate that the machine never terminates on such inputs. More work is needed in this direction to understand how to state the equivalence in this case, in particular how to translate the "never terminates" part. Of course, the equivalence holds for partial functions where the behavior outside of the domain is not defined.*

9 Missing Proofs

9.1 Proof of Theorem 54: Simulating Discrete by Continuous Time

9.1.1 A construction used elsewhere

Another very common pattern that we will use is known as "sample and hold". Typically, we have a variable signal and we would like to apply some process to it. Unfortunately, the device that processes the signal assumes (almost) constant input and does not work in real time (analog-to-digital converters would be a typical example). In this case, we cannot feed the signal directly to the processor so we need some black box that samples the signal to capture its value, and holds this value long enough for the processor to compute its output. This process is usually used in a τ-periodic fashion: the box samples for time δ and holds for time $\tau - \delta$. This is precisely what the sample function achieves. In fact, we show that it achieves much more: it is robust to noise and has good convergence properties when the input signal converges. The following result is from [BGP16c, Lemma 35]

Lemma 81 (Sample and hold) *Let $\tau \in \mathbb{R}_+$ and $I = [a, b] \subsetneq [0, \tau]$. Then there exists* $\mathrm{sample}_{I,\tau} \in \mathrm{GPVAL}$ *with the following properties. Let $y : \mathbb{R}_+ \to \mathbb{R}$, $y_0 \in \mathbb{R}$, $x, e \in C^0(\mathbb{R}_+, \mathbb{R})$ and $\mu : \mathbb{R}_+ \to \mathbb{R}_+$ be an increasing function. Suppose that for all $t \in \mathbb{R}_+$ we have*

$$y(0) = y_0, \qquad y'(t) = \mathrm{sample}_{I,\tau}(t, \mu(t), y(t), x(t)) + e(t).$$

Then:

$$|y(t)| \leqslant 2 + \int_{\max(0,t-\tau-|I|)}^{t} |e(u)|du + \max\left(|y(0)|\mathbb{1}_{[0,b]}(t), \sup_{\tau+|I|}|x|(t)\right)$$

Furthermore:

- *If $t \notin I \pmod{\tau}$ then $|y'(t)| \leqslant e^{-\mu(t)} + |e(t)|$.*
- *for $n \in \mathbb{N}$, if there exist $\bar{x} \in \mathbb{R}$ and $\nu, \nu' \in \mathbb{R}_+$ such that $|\bar{x} - x(t)| \leqslant e^{-\nu}$ and $\mu(t) \geqslant \nu'$ for all $t \in n\tau + I$ then*

  $$|y(n\tau + b) - \bar{x}| \leqslant \int_{n\tau+I} |e(u)|du + e^{-\nu} + e^{-\nu'}.$$

- *For $n \in \mathbb{N}$, if there exist $\check{x}, \hat{x} \in \mathbb{R}$ and $\nu \in \mathbb{R}_+$ such that $x(t) \in [\check{x}, \hat{x}]$ and $\mu(t) \geqslant \nu$ for all $t \in n\tau + I$ then*

  $$y(n\tau + b) \in [\check{x} - \varepsilon, \hat{x} + \varepsilon]$$

  where $\varepsilon = 2e^{-\nu} + \int_{n\tau+I} |e(u)|du$.

- *For any $J = [c, d] \subseteq \mathbb{R}_+$, if there exist $\nu, \nu' \in \mathbb{R}_+$ and $\bar{x} \in \mathbb{R}$ such that $\mu(t) \geqslant \nu'$ for all $t \in J$ and $|x(t) - \bar{x}| \leqslant e^{-\nu}$ for all $t \in J \cap (n\tau + I)$ for some $n \in \mathbb{N}$, then*

  $$|y(t) - \bar{x}| \leqslant e^{-\nu} + e^{-\nu'} + \int_{t-\tau-|I|}^{t} |e(u)|du$$

  for all $t \in [c + \tau + |I|, d]$.

- *If there exists $\Pi : \mathbb{R}_+ \to \mathbb{R}_+$ such that for any $J = [c, d]$ and $\bar{x} \in \mathbb{R}$ such that for all $\nu \in \mathbb{R}_+$, $n \in \mathbb{N}$ and $t \in (n\tau + I) \cap [c + \Pi(\nu), d]$, $|\bar{x} - x(t)| \leqslant e^{-\nu}$; then*

  $$|y(t) - \bar{x}| \leqslant e^{-\nu} + \int_{t-\tau-|I|}^{t} |e(u)|du$$

  for all $t \in [c + \Pi^*(\nu), d]$ where

  $$\Pi^*(\nu) = \max(\Pi(\nu + \ln(2 + \tau)), \mu^{-1}(\nu + \ln(2 + \tau))) + \tau + |I|.$$

Another tool is that of "digit extraction". In Theorem 60 we saw that we can decode a value, as long as we are close enough to a word. In essence, this theorem works around the continuity problem by creating gaps in the domain of the definition. This approach does not help on the rare occasions when we really want to extract some information about the encoding. How is it possible to achieve this without breaking the continuity requirement ? The compromise is to ask for *less information*. More precisely, write $x = \sum_{i=0}^{\infty} d_i 2^{-i}$, we call d_n is the n^{th} digit. The function that maps x to d_n is not continuous. Instead, we can compute $\cos(2\pi 2^n x) = \cos(\sum_{i\geqslant n} d_i 2^{-i})$. Intuitively, this is the next best thing we can hope for if we want a continuous map: it does not give us d_n but still gives us enough information.

Lemma 82 (Extraction) *For any $k \geqslant 2$, there exists* $\text{extract}_k \in \text{ALP}$ *such that for any $x \in \mathbb{R}$ and $n \in \mathbb{N}$:*

$$\text{extract}_k(x, n) = \cos(2\pi k^n x).$$

Proof. Let T_k be the k^{th} Tchebychev polynomial. It is a well-known fact that for every $\theta \in \mathbb{R}$,

$$\cos(k\theta) = T_k(\cos\theta).$$

For any $x \in [-1, 1]$, let

$$f(x) = T_k(x).$$

Then $f([-1, 1]) = [-1, 1]$ and $f \in \text{ALP}$ because T_k is a polynomial with integer coefficients. We can thus iterate f and get that for any $x \in \mathbb{R}$ and $n \in \mathbb{N}$,

$$\cos(2\pi k^n x) = f^{[n]}(\cos(2\pi x)).$$

In order to apply Theorem 54, we need to check some hypothesis. Since f is bounded by 1, clearly for all $x \in [-1, 1]$,

$$\left\| f^{[n]}(x) \right\| \leqslant 1.$$

Furthermore, f is C^1 on $[-1, 1]$ which is a compact set, thus f is a Lipschitz function. We hence conclude that Theorem 54 can be applied using Remark 58 and $f_0^* \in \text{ALP}$. For any $x \in \mathbb{R}$ and $n \in \mathbb{N}$, let

$$\text{extract}_k(x, n) = f_0^*(\cos(2\pi x), n).$$

Since $f_0^*, \cos \in \text{ALP}$ then $\text{extract}_k \in \text{ALP}$. And by construction,

$$\text{extract}_k(x, n) = f^{[n]}(\cos(2\pi x)) = \cos(2\pi x k^n).$$

■

9.1.2 Proof of Theorem 54

Proof. We use three variables y, z and w and build a cycle to be repeated n times. At all time, y is an online system computing $f(w)$. During the first stage of the cycle, w stays still and y converges to $f(w)$. During the second stage of the cycle, z copies y while w stays still. During the last stage, w copies z thus effectively computing one iterate.

A crucial point is in the error estimation, which we informally develop here. Denote the k^{th} iterate of x by $x^{[k]}$ and by $x^{(k)}$ the point computed after k cycles in the system. Because we are doing an approximation of f at each step step, the relationship between the two is that $x_0 = x^{[0]}$ and $\left\| x^{(k+1)} - f(x_k) \right\| \leqslant e^{-\nu_{k+1}}$ where ν_{k+1} is the precision of the approximation, that we control. Define μ_k

the precision we need to achieve at step k: $\left\|x^{(k)} - x^{[k]}\right\| \leqslant e^{-\mu_k}$ and $\mu_n = \mu$. The triangle inequality ensures that the following choice of parameters is safe:

$$\nu_k \geqslant \mu_k + \ln 2 \qquad \mu_{k-1} \geqslant \mho\left(\left\|x^{[k-1]}\right\|\right) + \mu_k + \ln 2$$

This is ensured by taking $\mu_k \geqslant \sum_{i=k}^{n-1} \mho(\Pi(\|x\|, i)) + \mu + (n-k)\ln 2$ which is indeed polynomial in k, μ and $\|x\|$. Finally a point worth mentioning is that the entire reasoning makes sense because the assumption ensures that $x^{(k)} \in I$ at each step.

Formally, apply Theorem 12 to get that $f \in \mathrm{AXC}(\Upsilon, \Pi, \Lambda, \Theta)$ where $\Upsilon, \Lambda, \Theta, \Pi$ are polynomials. Without loss of generability we assume that $\Upsilon, \Lambda, \Theta, \mho$ and Π are increasing functions. Apply Lemma 38 (AXP time rescaling) of [BGP16c] to get that Π can be assumed constant. Thus there exists $\omega \in [1, +\infty[$ such that for all $\alpha \in \mathbb{R}, \mu \in \mathbb{R}_+$

$$\Pi(\alpha, \mu) = \omega \geqslant 1.$$

Hence $f \in \mathrm{AXC}(\Upsilon, \Pi, \Lambda, \Theta)$ with corresponding δ, d and g. Define:

$$\tau = \omega + 2.$$

We will show that $f_0^* \in \mathrm{ALP}$.Let $n \in \mathbb{N}$, $x \in I_n$, $\mu \in \mathbb{R}_+$ and consider the following system:

$$\begin{cases} \ell(0) = \operatorname{norm}_{\infty,1}(x) \\ \mu(0) = \mu \\ n(0) = n \end{cases} \qquad \begin{cases} \ell'(t) = 0 \\ \mu'(t) = 0 \\ n'(t) = 0 \end{cases} \qquad \begin{cases} y(0) = 0 \\ z(0) = x \\ w(0) = x \end{cases}$$

$$\begin{cases} y'(t) = g(t, y(t), w(t), \nu(t)) \\ z'(t) = \operatorname{sample}_{[\omega,\omega+1],\tau}(t, \nu(t), z(t), y_{1..n}(t)) \\ w'(t) = \operatorname{hxl}_{[0,1]}(t - n\tau, \nu(t) + t, \operatorname{sample}_{[\omega+1,\omega+2],\tau}(t, \nu^*(t) + \ln(1+\omega), w(t), z(t))) \end{cases}$$

$$\ell^* = 1 + \Pi(\ell, n) \qquad \nu = n\mho(\ell^*) + n\ln 6 + \mu + \ln 3 \qquad \nu^* = \nu + \Lambda(\ell^*, \nu)$$

First notice that ℓ, μ and n are constant functions and we identify $\mu(t)$ with μ and $n(t)$ with n. Apply Lemma 33 to get that $\|x\| \leqslant \ell \leqslant \|x\| + 1$, so in particular ℓ^*, ν and ν^* are polynomially bounded in $\|x\|$ and n. We will need a few notations: for $i \in [\![0, n]\!]$, define $x^{[i]} = f^{[i]}(x)$ and $x^{(i)} = w(i\tau)$. Note that $x^{[0]} = x^{(0)} = x$. We will show by induction for $i \in [\![0, n]\!]$ that

$$\left\|x^{(i)} - x^{[i]}\right\| \leqslant e^{-(n-i)\mho(\ell^*) - (n-i)\ln 6 - \mu - \ln 3}.$$

Note that this is trivially true for $i = 0$. Let $i \in [\![0, n-1]\!]$ and assume that the result is true for i. We will show that it holds for $i+1$ by analyzing the behavior of the various variables in the system during period $[i\tau, (i+1)\tau]$.

- **For y and w, if $t \in [i\tau, i\tau + \omega + 1]$ then** apply Lemma 34 to get that $\mathrm{hxl} \in [0, 1]$ and Lemma 81 to get that $\|w'(t)\| \leqslant e^{-\nu^* - \ln(1+\omega)}$. Conclude that $\|w(i) - w(t)\| \leqslant e^{-\nu^*}$, in other words $\left\|w(t) - x^{(i)}\right\| \leqslant e^{-\Lambda(\left\|x^{(i)}\right\|, \nu)}$ since $\left\|x^{(i)}\right\| \leqslant \left\|x^{[i]}\right\| + 1 \leqslant 1 + \Pi(\|x\|, i) \leqslant \ell^*$ and $\nu^* \geqslant \Lambda(\ell^*, \nu)$. Thus, by definition of extreme computability, $\left\|f(x^{(i)}) - y_{1..n}(u)\right\| \leqslant e^{-\nu}$ if $u \in [i\tau + \omega, i\tau + \omega + 1]$ because $\Pi\left(\left\|x^{(i)}\right\|, \nu\right) = \omega$.

- **For** z**, if** $t \in [i\tau+\omega, i\tau+\omega+1]$ **then** apply Lemma 81 to get that

$$\left\|f(x^{(i)}) - z(i\tau+\omega+1)\right\| \leqslant 2e^{-\nu}.$$

  Notice that we ignore the behavior of z during $[i\tau, i\tau+\omega]$ in this part of the proof.

- **For** z **and** w**, if** $t \in [i\tau+\omega+1, i\tau+\omega+2]$ **then** apply Lemma 81 to get that $\|z'(t)\| \leqslant e^{-\nu}$ and thus $\|f(x^{(i)}) - z(t)\| \leqslant 3e^{-\nu}$. Apply Lemma 34 to get that

$$\left\|y'(t) - \text{sample}_{[\omega+1,\omega+2],\tau}(t, \nu^* + \ln(1+\omega), w(t), z(t))\right\| \leqslant e^{-\nu-t}.$$

  Apply Lemma 81 again to get that $\|f(x^{(i)}) - w(i\tau+\omega+2)\| \leqslant 4e^{-\nu} + e^{-\nu^*} \leqslant 5e^{-\nu}$.

Our analysis concluded that $\|f(x^{(i)}) - w((i+1)\tau)\| \leqslant 5e^{-\nu}$. Also, by hypothesis, $\|x^{(i)} - x^{[i]}\| \leqslant e^{-(n-i)\mho(\ell^*)-(n-i)\ln 6-\mu-\ln 3} \leqslant e^{-\mho(\|x^{[i]}\|)-\mu^*}$ where $\mu^* = (n-i-1)\mho(\ell^*) + (n-i)\ln 6 + \mu + \ln 3$ because $\|x^{[i]}\| \leqslant \ell^*$. Consequently, $\|f(x^{(i)}) - x^{[i+1]}\| \leqslant e^{-\mu^*}$ and thus:

$$\left\|x^{(i+1)} - x^{[i+1]}\right\| \leqslant 5e^{-\nu} + e^{-\mu^*} \leqslant 6e^{-\mu^*} \leqslant e^{-(n-1-i)\mho(\ell^*)-(n-1-i)\ln 6-\mu-\ln 3}.$$

From this induction we get that $\|x^{(n)} - x^{[n]}\| \leqslant e^{-\mu-\ln 3}$. We still have to analyze the behavior after time $n\tau$.

- **If** $t \in [n\tau, n\tau+1]$ **then** apply Lemma 81 and Lemma 34 to get that $\|w'(t)\| \leqslant e^{-\nu^*-\ln(1+\omega)}$ thus $\|w(t) - x^{(n)}\| \leqslant e^{-\nu^*-\ln(1+\omega)}$.
- **If** $t \geqslant n\tau+1$ **then** apply Lemma 34 to get that $\|w'(t)\| \leqslant e^{-\nu-t}$ thus $\|w(t) - w(n\tau+1)\| \leqslant e^{-\nu}$.

Putting everything together we get for $t \geqslant n\tau+1$ that:

$$\begin{aligned}\left\|w(t) - x^{[n]}\right\| &\leqslant e^{-\mu-\ln 3} + e^{-\nu^*-\ln(1+\omega)} + e^{-\nu}\\ &\leqslant 3e^{-\mu-\ln 3} \leqslant e^{-\mu}.\end{aligned}$$

We also have to show that the system does not grow too fast. The analysis during the time interval $[0, n\tau+1]$ has already been done (although we did not write all the details, it is an implicit consequence). For $t \geqslant n\tau+1$, have $\|w(t)\| \leqslant \|x^{[n]}\| + 1 \leqslant \Pi(\|x\|, n) + 1$ which is polynomially bounded. The bound on y comes from extreme computability:

$$\|y(t)\| \leqslant \Upsilon\left(\sup_\delta \|w\|(t), \nu, 0\right) \leqslant \Upsilon(\Pi(\|x\|, n), \nu, 0) \leqslant \text{poly}(\|x\|, n, \mu)$$

And finally, apply Lemma 81 to get that:

$$\|z(t)\| \leqslant 2 + \sup_{\tau+1} \|y_{1..n}\|(t) \leqslant \text{poly}(\|x\|, n, \mu)$$

This conclude the proof that $f_0^* \in \text{ALP}$.

We can now tackle the case of $\eta > 0$. Let $\eta \in]0, \frac{1}{2}[$ and $\mu_\eta \in \mathbb{Q}$ such that $\frac{1}{2} - e^{-\mu_\eta} < \eta$. Let $f_\eta^*(x, u) = f_0^*(x, \text{rnd}^*(u, \mu_\eta))$. Apply Theorem 21 to conclude that $f_\eta^* \in \text{ALP}$. By definition of rnd*, if $u \in]n-\eta, n+\eta[$ for some $n \in \mathbb{Z}$ then $\text{rnd}^*(x, \mu) = n$ and thus $f_\eta^*(x, u) = f_0^*(x, n) = x^{[n]}$. ■

9.2 Cauchy completion and complexity

The purpose of this section is to prove Theorem 77.

Given $x \in \mathcal{D}$ and $n \in \mathbb{N}$, we want to use g to compute an approximation of $f(x)$ within 2^{-n}. To do so, we use the "modulus of continuity" $\mho$ to find a dyadic rational (q, n) such that $\|x - q\| \leqslant 2^{-\mho(\|x\|,n)}$. We then compute $g(q, n)$ and get that $\|g(q, n) - f(x)\| \leqslant 2^{-n}$.

There are two problems with this approach. First, finding such a dyadic rational is not possible because it is not a continuous operation. Indeed, consider the mapping $(x, n) \mapsto (q, n)$ that satisfies the above condition: if it is computable, it must be continuous. But it cannot be continuous because its image is completed disconnected. This is where mixing comes into play: given x and n, we will compute two dyadic rationals (q, n) and (q', n') such that at least one of them satisfies the above criteria. We will then apply g on both of them and mix the result. The idea is that if both are valid, the outputs will be very close (because of the modulus of continuity) and thus the mixing will give the correct result. See Section 8.2 for more details on mixing. The case of multidimensional domains is similar except that we need to mix on all dimensions simultaneously, thus we need roughly 2^d mixes to ensure that at least one is correct, where d is the dimension.

Proof (of Theorem 77). We will show the result by induction on d. If $d = 0$ then $\|g(n, y) - f(y)\| \leqslant 2^{-n}$ for all $n \in \mathbb{N}, y \in \mathcal{D}$. We can thus apply Theorem 73 to get that $f \in \mathrm{ALP}$.

Assume that $d > 0$. Let $\kappa : \{0, 1\} \to \{0, 1\}, x \mapsto x$ and π_i denote the i^{th} projection. For any relevant[21] $u \in \mathbb{R}$ and $n \in \mathbb{N}$ and $\delta \in \{0, 1\}$, let

$$\begin{aligned}
v_\delta(u, n) &= v(u - \tfrac{\delta}{2}2^{-n}, n)\\
v(u, n) &= r(u, n) + v^*(u - r(u, n), n)\\
v^*(u, n) &= \pi_1(\mathrm{decode}_\kappa(u, n))\\
r(u, n) &= \mathrm{rnd}^*(u - \tfrac{1}{2} - e^{-\nu}, \nu) \text{ where } \nu = \ln 6 + n \ln 2.
\end{aligned}$$

We now discuss the domain of definition and properties of these functions. First $r \in \mathrm{ALP}$ since $\mathrm{rnd}^* \in \mathrm{ALP}$ by Theorem 32. Furthermore, by definition of rnd^* we have that

$$\text{if } u \in m + \left[0, 1 - \tfrac{1}{3}2^{-n}\right] \text{ for some } m \in \mathbb{Z} \text{ then } r(u, n) = m.$$

Indeed since $2e^{-\nu} = \frac{1}{3}2^{-n}$,

$$\begin{gathered}
m \leqslant u \leqslant m + 1 - 2e^{-\nu}\\
m - \tfrac{1}{2} + e^{-\nu} \leqslant u - \tfrac{1}{2} + e^{-\nu} \leqslant m + \tfrac{1}{2} - e^{-\nu}
\end{gathered}$$

thus $r(u, n) = \mathrm{rnd}^*(u - \frac{1}{2} + e^{-\nu}, \nu) = m$. We now claim that we have that

$$\text{if } u = \frac{p}{2^n} + \varepsilon \text{ for some } p \in \mathbb{Z} \text{ and } \varepsilon \in \left[0, 2^{-n}\tfrac{2}{3}\right] \text{ then } v(u, n) = \frac{p}{2^n}.$$

Indeed, write $p = m2^n + p'$ where $m \in \mathbb{Z}$ and $p' \in [\![0, 2^n - 1]\!]$. Then $u = m + \frac{p'}{2^n} + \varepsilon$ and

$$\frac{p'}{2^n} + \varepsilon \leqslant \frac{2^n - 1}{2^n} + \varepsilon \leqslant 1 - 2^{-n} + \frac{2}{3}2^{-n} \leqslant 1 - \frac{1}{3}2^{-n}.$$

[21] Domain of definition is discussed below.

Thus $r(u,n) = m$ and $u - r(u,n) = \frac{p'}{2^n} + \varepsilon$. Since $p' \in [\![0, 2^n - 1]\!]^d$, there exist $w_1, \dots, w_d \in \{0,1\}$ such that

$$\frac{p'}{2^n} = \sum_{j=1}^{n} w_j 2^{-j}.$$

It follows from Theorem 60 and the fact that $1 - e^{-2} \geqslant \frac{2}{3}$ that[22]

$$\operatorname{decode}_\kappa\left(\tfrac{p'}{2^n} + \varepsilon, n, 2\right) = \operatorname{decode}_\kappa\left(\sum_{j=1}^{n} w_j 2^{-j} + \varepsilon, n, 2\right) = \left(\sum_{j=1}^{n} w_j 2^{-j}, *\right) = \left(\tfrac{p'}{2^n}, *\right).$$

Consequently,

$$\begin{aligned}
v(u,n) &= r(u,n) + v^*\left(u - r(u,n), n\right)\\
&= m + v^*\left(\tfrac{p'}{2^n} + \varepsilon, n\right)\\
&= m + \pi_1\left(\operatorname{decode}_\kappa(\tfrac{p'}{2^n} + \varepsilon, n)\right)\\
&= m + \pi_1\left(\tfrac{p'}{2^n}, *\right)\\
&= m + \tfrac{p'}{2^n}\\
&= \tfrac{p}{2^n}.
\end{aligned}$$

To summarize, we have shown that

$$\text{if } u = \frac{p}{2^n} + \varepsilon \text{ for some } p \in \mathbb{Z} \text{ and } \varepsilon \in \left[0, \tfrac{2}{3}2^{-n}\right] \text{ then } v(u,n) = \frac{p}{2^n}.$$

and thus that for all $\delta \in \{0,1\}$,

$$\text{if } u = \frac{p}{2^n} + \frac{\delta}{2}2^{-n} + \varepsilon \text{ for some } p \in \mathbb{Z} \text{ and } \varepsilon \in \left[0, \tfrac{2}{3}2^{-n}\right] \text{ then } v_\delta(u,n) = \frac{p}{2^n}. \tag{14}$$

Before we proceed to mixing, we need an auxiliary function. For all $u \in \mathbb{R}$ and $n \in \mathbb{N}$, define

$$\operatorname{sel}(u,n) = \tfrac{1}{2} + \operatorname{extract}_2\left(u + \tfrac{1}{6}2^{-n}, n\right)$$

where $\operatorname{extract}_2$ is given by Lemma 82. We claim that for all $n \in \mathbb{N}$,

$$\{u \in \mathbb{R} : \operatorname{sel}(u,n) < 1\} \subseteq \left(2^{-n}\mathbb{Z} + \left[0, \tfrac{2}{3}2^{-n}\right]\right) \times \{n\} \tag{15}$$

and

$$\{u \in \mathbb{R} : \operatorname{sel}(u,n) > 0\} \subseteq \left(2^{-n}\mathbb{Z} + \left[-\tfrac{1}{2}2^{-n}, \tfrac{1}{6}2^{-n}\right]\right) \times \{n\}. \tag{16}$$

Indeed, by definition of $\operatorname{extract}_2$, if $u = \frac{p}{2^n} + \varepsilon$ with $\varepsilon \in [0, 2^{-n}[$, then

$$\begin{aligned}
\operatorname{sel}(u,n) &= \tfrac{1}{2} + \operatorname{extract}_2\left(\tfrac{p}{2^n} + \varepsilon + \tfrac{1}{6}2^{-n}, n\right)\\
&= \tfrac{1}{2} + \cos(2\pi 2^n(\tfrac{p}{2^n} + \varepsilon + \tfrac{1}{6}2^{-n}))\\
&= \tfrac{1}{2} + \cos(2\pi p + 2\pi 2^n\varepsilon + \tfrac{\pi}{3})\\
&= \tfrac{1}{2} + \cos(2\pi 2^n\varepsilon + \tfrac{\pi}{3})
\end{aligned}$$

[22]The $*$ denotes "anything" because we do not care about the actual value.

where $2\pi 2^n\varepsilon \in [0, 2\pi[$ and thus $2\pi 2^n\varepsilon + \frac{\pi}{3} \in [\frac{\pi}{3}, \frac{7\pi}{3}]$. Consequently,

$$\operatorname{sel}(u, n) < 1 \Leftrightarrow \varepsilon \in \left]0, \tfrac{2}{3}2^{-n}\right[.$$

And similarly,

$$\operatorname{sel}(u, n) > 0 \Leftrightarrow \varepsilon \in \left[0, \tfrac{1}{6}2^{-n}\right[\cup \left]\tfrac{1}{2}2^{-n}, 2^{-n}\right].$$

Now define for all relevant[23] $q \in \mathbb{Q}^{d-1}, n \in \mathbb{N}, z \in \mathbb{R}, y \in \mathbb{R}^e, \delta \in \{0, 1\}$,

$$\begin{aligned}
\tilde{g}_\delta(q, m, z, y) &= g(q, v_\delta(z, m), m, y), \\
\widetilde{\operatorname{sel}}(q, m, z, y) &= \operatorname{sel}(z, m), \\
\tilde{g}(q, m, z, y) &= \operatorname{mix}(\widetilde{\operatorname{sel}}, \tilde{g}_0, \tilde{g}_1)(q, m, z, y).
\end{aligned}$$

For any $\alpha \in \mathbb{R}_+$ and $n \in \mathbb{N}$, define

$$\mho^*(\alpha, n) = \mho(\alpha, n) + 1.$$

Let $(x, z, y) \in \mathcal{D}$ and $n, m \in \mathbb{N}, p \in \mathbb{Z}^{d-1}$, such that

$$\left\|\tfrac{p}{2^m} - x\right\| \leqslant 2^{-m} \text{ and } m \geqslant \mho^*(\|(x, z, y)\|, n). \tag{17}$$

Let $q = \frac{p}{2^m}$. There are three cases:

- **If $\widetilde{\operatorname{sel}}(\mathbf{q}, \mathbf{m}, \mathbf{z}, \mathbf{y}) = \mathbf{0}$:** then $\operatorname{sel}(z, m) = 0$. But then $\operatorname{sel}(z, m) < 1$ so by (15), $z \in 2^{-m}\mathbb{Z} + \left[0, \frac{2}{3}2^{-m}\right]$. Write $z = p'2^{-m} + \varepsilon$ where $p' \in \mathbb{Z}$ and $\varepsilon \in [0, \frac{2}{3}2^{-m}]$. Then $v_0(z, ,) = \frac{p'}{2^m}$ using (14). It follows that,

$$\begin{aligned}
\|(x, z) - (q, v_0(z, m))\| &= \max(\|x - q\|, |z - \tfrac{p'}{2^m}|) \\
&= \max(\|x - q\|, |\varepsilon|) && \text{since } z = p'2^{-m} + \varepsilon \\
&\leqslant \max(2^{-\mho^*(\|(x,z,y)\|,n)}, |\varepsilon|) && \text{by assumption on } q \\
&\leqslant \max(2^{-\mho^*(\|(x,z,y)\|,n)}, \tfrac{2}{3}2^{-m}) \\
&\leqslant 2^{-\mho^*(\|(x,z,y)\|,n)} && \text{since } m \geqslant \mho^*(\|(x, z, y)\|, n) \\
&\leqslant 2^{-\mho(\|(x,z,y)\|,n)-1} && \text{by definition of } \mho.
\end{aligned}$$

  It follows by assumption on g that $\|g(q, v_0(z, m), m, y) - f(x, z, y)\| \leqslant 2^{-n-1}$. But since $\widetilde{\operatorname{sel}}(q, m, z, y) = 0$, then

$$\tilde{g}(q, m, z, y) = \tilde{g}_0(q, m, z, y) = g(q, v_0(z, m), m, y),$$

  thus $\|\tilde{g}(q, m, z, y) - f(x, z, y)\| \leqslant 2^{-n-1} \leqslant 2^{-n}$.

- **If $\widetilde{\operatorname{sel}}(\mathbf{q}, \mathbf{m}, \mathbf{z}, \mathbf{y}) = \mathbf{1}$:** then $\operatorname{sel}(z, m) = 1$. But then $\operatorname{sel}(z, m) > 0$ so by (16), $z \in 2^{-m}\mathbb{Z} + \left[-\frac{1}{2}2^{-m}, \frac{1}{6}2^{-m}\right]$. Write $z = p'2^{-m} + \varepsilon$ where $p' \in \mathbb{Z}$ and $\varepsilon \in \left[-\frac{1}{2}2^{-m}, \frac{1}{6}2^{-m}\right]$. Then $v_1(z, m) = \frac{p'}{2^m}$ using (14). It follows that,

$$\|(x, z) - (q, v_1(z, m))\| = \max(\|x - q\|, |z - \tfrac{p'}{2^m}|)$$

[23] We will discuss the domain of definition below.

$$\leqslant 2^{-\mho(\|(x,z,y)\|,n)-1}$$

using the same chain of inequalities as in the previous case. It follows by assumption on g that $\|g(q, v_1(z,m), m, y) - f(x,z,y)\| \leqslant 2^{-n}$. But since $\widetilde{\text{sel}}(q,m,z,y) = 1$, then $\tilde{g}(q,m,z,y) = \tilde{g}_1(q,m,z,y) = g(q, v_1(z,m), m, y)$, thus $\|\tilde{g}(q,m,z,y) - f(x,z,y)\| \leqslant 2^{-n-1} \leqslant 2^{-n}$.

- **If $\mathbf{0} < \widetilde{\text{sel}}(\mathbf{q},\mathbf{m},\mathbf{z},\mathbf{y}) < \mathbf{1}$:** then

$$\tilde{g}(q,m,z,y) = (1-\alpha)\tilde{g}_0(q,m,z,y) + \alpha\tilde{g}_1(q,m,z,y)$$

where $\alpha = \text{sel}(z,m) \in]0,1[$. Using the same reasoning as in the previous two cases we get that

$$\|\tilde{g}_0(q,m,z,y) - f(x,z,y)\| \leqslant 2^{-n-1} \text{ and } \|\tilde{g}_1(q,m,z,y) - f(x,z,y)\| \leqslant 2^{-n-1}.$$

It easily follows that

$$\|\tilde{g}(q,m,z,y) - f(x,z,y)\| \leqslant 2\alpha \cdot 2^{-n-1} \leqslant 2^{-n}.$$

To summarize, we have shown that under assumption (17) we have that

$$\left\|g(\tfrac{p}{2^m}, m, z, y) - f(x,z,y)\right\| \leqslant 2^{-n}.$$

And since since $\tilde{g} \in \text{ALP}$, we can apply the result inductively to $\tilde{g}$ (which has only $d-1$ dyadic arguments) to conclude. ■

9.3 Proof of Theorem 60: Word decoding

Proof. We will iterate a function that works on tuple of the form (x, x', n, m, μ) where x is the remaining part to process, x' is the processed part, n the length of the processed part, m the number of nonzero symbols and μ will stay constant. The function will remove the "head" of x, re-encode it with κ and "queue" on x', increasing n and m if the head is not 0.

In the remaining of this proof, we write $\overline{0.x}^{k_i}$ to denote $0.x$ in basis k_i instead of k. Define for any $x, y \in \mathbb{R}$ and $n \in \mathbb{N}$:

$$g(x,y,n,m,\mu) = \left(\text{frac}^*(k_1x), y + k_2^{-n-1}\mathbb{1}_\kappa(\text{int}^*(k_1x)), n+1, m + \mathbb{D}_{\text{id}\neq 0}(\text{int}^*(k_1x)), \mu\right)$$

where

$$\text{int}^*(x) = \text{rnd}^*\left(x - \tfrac{1}{2} + \tfrac{3e^{-\mu}}{4}, \mu\right) \qquad \text{frac}^*(x) = x - \text{int}^*(x)$$

and rnd^* is defined in Definition 31. Apply Lemma 41 to get that $\mathbb{1}_\kappa \in \text{ALP}$ and Lemma 43 to get that $\mathbb{D}_{\text{id}\neq 0} \in \text{ALP}$. It follows that $g \in \text{ALP}$. We need a small result about int^* and frac^*. For any $w \in [\![0, k_1]\!]^*$ and $x \in \mathbb{R}$, define the following proposition:

$$A(x,w,\mu) : -k_1^{-|w|}\frac{e^{-\mu}}{2} \leqslant x - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1 - e^{-\mu}).$$

We will show that:

$$A(x,w,\mu) \Rightarrow \left\{ \begin{array}{l} \text{int}^*(k_1x) = \text{int}(k_1\overline{0.w}^{k_1}) \\ \left|\text{frac}^*(k_1x) - \text{frac}(k_1\overline{0.w}^{k_1})\right| \leqslant k_1\left|x - \overline{0.w}^{k_1}\right| \end{array} \right. . \tag{18}$$

Indeed, in this case, since $|w| \geqslant 1$, we have that

$$\begin{aligned}
-k_1^{1-|w|}\tfrac{e^{-\mu}}{2} \leqslant k_1x - k_1\overline{0.w}^{k_1} \leqslant k_1^{1-|w|}(1-e^{-\mu})\\
-k_1^{1-|w|}\tfrac{e^{-\mu}}{2} \leqslant k_1x - w_1 \leqslant k_1^{1-|w|}(1-e^{-\mu}) + \overline{0.w_{2..|w|}}^{k_1}\\
-\tfrac{e^{-\mu}}{2} \leqslant k_1x - w_1 \leqslant k_1^{1-|w|} - e^{-\mu} + \textstyle\sum_{i=1}^{|w|-1}(k_1-1)k_1^{-i}\\
-\tfrac{e^{-\mu}}{2} \leqslant k_1x - w_1 \leqslant k_1^{1-|w|} - e^{-\mu} + 1 - k_1^{1-|w|}\\
-\tfrac{1}{2} - \tfrac{e^{-\mu}}{2} \leqslant k_1x - \tfrac{1}{2} - w_1 \leqslant \tfrac{1}{2} - e^{-\mu}\\
-\tfrac{1}{2} + \tfrac{e^{-\mu}}{4} \leqslant k_1x - \tfrac{1}{2} + \tfrac{3e^{-\mu}}{4} - w_1 \leqslant \tfrac{1}{2} - \tfrac{e^{-\mu}}{4}
\end{aligned}$$

And conclude by applying Theorem 32 because $\operatorname{int}(k_1\overline{0.w}^{k_1}) = w_1$. The result on frac follows trivially. It is then not hard to derive from (18) applied twice that:

$$\begin{gathered}
A(x,w,\mu) \quad \wedge \quad A(x',w,\mu')\\
\Downarrow\\
\|g(x,y,n,m,\mu) - g(x',y',n',m',\nu)\| \leqslant 2k_1\,\|(x,y,n,m,\mu) - (x',y',n',m',\mu')\|\,.
\end{gathered} \tag{19}$$

It also follows that proposition A is preserved by applying g:

$$A(x,w,\mu) \quad\Rightarrow\quad A(\operatorname{frac}(k_1x), w_{2..|w|}, \mu). \tag{20}$$

Furthermore, A is stronger for longer words:

$$A(x,w,\mu) \quad\Rightarrow\quad A(x, w_{1..|w|-1}, \mu). \tag{21}$$

Indeed, if we have $A(x,w,\mu)$ then:

$$\begin{aligned}
-k_1^{-|w|}\tfrac{e^{-\mu}}{2} \leqslant x - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1-e^{-\mu})\\
-k_1^{-|w|}\tfrac{e^{-\mu}}{2} \leqslant x - \overline{0.w_{1..|w|-1}}^{k_1} \leqslant k_1^{-|w|}(1-e^{-\mu}) + w_{|w|}k_1^{-|w|}\\
-k_1^{1-|w|}\tfrac{e^{-\mu}}{2} \leqslant x - \overline{0.w_{1..|w|-1}}^{k_1} \leqslant k_1^{-|w|}(1-e^{-\mu}) + (k_1-1)k_1^{-|w|}\\
-k_1^{1-|w|}\tfrac{e^{-\mu}}{2} \leqslant x - \overline{0.w_{1..|w|-1}}^{k_1} \leqslant k_1^{-|w|}(k_1-e^{-\mu})\\
-k_1^{1-|w|}\tfrac{e^{-\mu}}{2} \leqslant x - \overline{0.w_{1..|w|-1}}^{k_1} \leqslant k_1^{1-|w|}(1-e^{-\mu})
\end{aligned}$$

It also follows from the definition of g that:

$$A(x,w,\mu) \quad\Rightarrow\quad \|g(x,y,n,m,\mu)\| \leqslant \max(k_1, 1 + \|x,y,n,m,\mu\|). \tag{22}$$

Indeed, if $A(x,w,\mu)$ then $\operatorname{int}^*(k_1x) \in [\![0,k_1-1]\!]$ thus $L_\kappa(\operatorname{int}^*(k_1x)) \in [\![0,k_2]\!]$ and $\mathbb{D}_{\mathrm{id}\neq 0}(\operatorname{int}^*(k_1x)) \in \{0,1\}$, the inequality follows easily. A crucial property of A is that it is open with respect to x:

$$A(x,w,\mu) \quad\wedge\quad |x-y| \leqslant e^{-|w|\ln k_1 - \mu - \nu} \quad\Rightarrow\quad A(y,w,\mu - \ln\tfrac{3}{2}). \tag{23}$$

Indeed, if $A(x,w,\mu)$ and $|x-y| \leqslant e^{-|w|\ln k_1 - \mu - \ln 4}$ we have:

$$\begin{aligned}
-k_1^{-|w|}\tfrac{e^{-\mu}}{2} \leqslant x - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1-e^{-\mu})\\
-k_1^{-|w|}\tfrac{e^{-\mu}}{2} + y - x \leqslant y - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1-e^{-\mu}) + y - x\\
-k_1^{-|w|}\tfrac{e^{-\mu}}{2} - |y-x| \leqslant y - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1-e^{-\mu}) + |y-x|\\
-k_1^{-|w|}\tfrac{e^{-\mu}}{2} - e^{-|w|\ln k_1 - \mu - \ln 4} \leqslant y - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1-e^{-\mu}) + e^{-|w|\ln k_1 - \mu - \ln 4}\\
-k_1^{-|w|}(e^{-\mu-\ln 4} + \tfrac{e^{-\mu}}{2}) \leqslant y - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1 - e^{-\mu} + e^{-\mu-\ln 4})\\
-k_1^{-|w|}\tfrac{3e^{-\mu}}{4} \leqslant y - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1 - \tfrac{3e^{-\mu}}{4})\\
-k_1^{-|w|}\tfrac{3e^{-\mu}}{4} \leqslant y - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1 - \tfrac{6e^{-\mu}}{4})\\
-k_1^{-|w|}\tfrac{e^{\ln\frac{3}{2}-\mu}}{2} \leqslant y - \overline{0.w}^{k_1} \leqslant k_1^{-|w|}(1 - e^{\ln\frac{3}{2}-\mu})
\end{aligned}$$

In order to formally apply Theorem 54, define for any $n \in \mathbb{N}$:

$$I_n = \left\{(x, y, \ell, m, \mu) \in \mathbb{R}^2 \times \mathbb{R}_+^3 : \exists w \in [\![0, k_1 - 1]\!]^n, A(x, w, \mu)\right\}.$$

It follows from (21) that $I_{n+1} \subseteq I_n$. It follows from (20) that $g(I_{n+1}) \subseteq I_n$. It follows from (22) that $\left\|g^{[n]}(x)\right\| \leqslant \max(k_1, \|x\| + n)$ for $x \in I_n$. Now assume that $X = (x, y, n, m, \mu) \in I_n$, $\nu \in \mathbb{R}_+$ and[24] $\|X - X'\| \leqslant e^{-\|X\| - n \ln k_1 - \nu}$ where $X' = (x', y', n', m, \mu')$ then by definition $A(x, w, \mu)$ for some $w \in [\![0, k_1 - 1]\!]^n$. It follows from (23) that $A(y, w, \mu - \ln \frac{3}{2})$ since $\|X\| + n \ln k_1 \geqslant |w| \ln k_1 + \mu$. Thus by (19) we have $\|g(X) - g(X')\| \leqslant 2k_1 \|X - X'\|$ which is enough by Remark 58. We are thus in good shape to apply Theorem 54 and get $g_0^* \in \mathrm{ALP}$. Define:

$$\mathrm{decode}_\kappa(x, n, \mu) = \pi_{2,4}(g_0^*(x, 0, 0, 0, \mu, n))$$

where $\pi_{2,4}(a, b, c, d, e, f, g) = (b, d)$. Clearly $\mathrm{decode}_\kappa \in \mathrm{ALP}$, it remains to see that it satisfies the theorem. We will prove this by induction on the length of $|w|$. More precisely we will prove that for $|w| \geqslant 0$:

$$\varepsilon \in [0, k_1^{-|w|}(1 - e^{-\mu})] \quad \Rightarrow \quad g^{[|w|]}(\overline{0.w}^{k_1} + \varepsilon, 0, 0, 0, \mu) = (k_1^{|w|}\varepsilon, \overline{0.\kappa(w)}^{k_2}, |w|, \#\{i | w_i \neq 0\}, \mu).$$

The case of $|w| = 0$ is trivial since it will act as the identity function:

$$\begin{aligned} g^{[|w|]}(\overline{0.w}^{k_1} + \varepsilon, 0, 0, 0, \mu) &= g^{[0]}(\varepsilon, 0, 0, 0, \mu) \\ &= (\varepsilon, 0, 0, 0, \mu) \\ &= (k_1^{|w|}\varepsilon, \overline{0.\kappa(w)}^{k_2}, |w|, \#\{i | w_i \neq 0\}, \mu). \end{aligned}$$

We can now show the induction step. Assume that $|w| \geqslant 1$ and define $w' = w_{1..|w|-1}$. Let $\varepsilon \in [0, k_1^{-|w|}(1 - e^{-\mu})]$ and define $\varepsilon' = k_1^{-|w|} w_{|w|} + \varepsilon$. It is clear that $\overline{0.w}^{k_1} + \varepsilon = \overline{0.w'}^{k_1} + \varepsilon'$. Then by definition $A(\overline{0.w'}^{k_1} + \varepsilon', |w|, \mu)$ so

$$\begin{aligned} g^{[|w|]}(\overline{0.w}^{k_1} + \varepsilon, 0, 0, 0, \mu) &= g(g^{[|w|-1]}(\overline{0.w'}^{k_1} + \varepsilon', 0, 0, 0, \mu)) \\ &= g(k_1^{|w|-1}\varepsilon', \overline{0.\kappa(w')}^{k_2}, |w'|, \#\{i | w'_i \neq 0\}, \mu) \\ &\qquad\qquad\qquad\qquad\qquad \text{By induction} \\ &= g(k_1^{-1} w_{|w|} + k_1^{|w|-1}\varepsilon, \overline{0.\kappa(w')}^{k_2}, |w'|, \#\{i | w'_i \neq 0\}, \mu) \\ &= (\mathrm{frac}^*(w_{|w|} + k_1^{|w|}\varepsilon), \quad \text{Where } k_1^{-|w|}\varepsilon \in [0, 1 - e^{-\mu}] \\ &\qquad \overline{0.\kappa(w')}^{k_2} + k_2^{-|w'|-1} \mathbb{1}_\kappa(\mathrm{int}^*(w_{|w|} + k_1^{|w|}\varepsilon)), \\ &\qquad |w'| + 1, \#\{i | w'_i \neq 0\} + \mathbb{D}_{\mathrm{id} \neq 0}(\mathrm{int}^*(w_{|w|} + k_1^{|w|}\varepsilon)), \mu) \\ &= (k_1^{|w|}\varepsilon, \overline{0.\kappa(w')}^{k_2} + k_2^{-|w|} \mathbb{1}_\kappa(w_{|w|}), \\ &\qquad |w|, \#\{i | w'_i \neq 0\} + \mathbb{D}_{\mathrm{id} \neq 0}(w_{|w|}), \mu) \\ &= (k_1^{|w|}\varepsilon, \overline{0.\kappa(w')}^{k_2}, |w|, \#\{i | w_i \neq 0\}, \mu). \end{aligned}$$

We can now conclude to the result. Let $\varepsilon \in [0, k_1^{-|w|}(1 - e^{-\mu})]$ then $A(\overline{0.0w}^{k_1} + \varepsilon, |w|, \mu)$ so in particular $(\overline{0.w}^{k_1} + \varepsilon, 0, 0, 0, \mu) \in I_{|w|}$ so:

$$\mathrm{decode}_\kappa(\overline{0.w}^{k_1} + \varepsilon, |w|, \mu) = \pi_{2,4}(g_0^*(\overline{0.w}^{k_1} + \varepsilon, 0, 0, 0, \mu))$$

[24] We use Remark 59 to allow a dependence of $\mho$ in n.

$$
\begin{aligned}
&= \pi_{2,4}(g^{[|w|]}(\overline{0.w}^{k_1} + \varepsilon, 0, 0, 0, \mu)) \\
&= \pi_{2,4}(\varepsilon, \overline{0.\kappa(w)}^{k_2}, |w|, \#\{i|w_i \neq 0\}, \mu) \\
&= (\overline{0.\kappa(w)}^{k_2}, \#\{i|w_i \neq 0\}).
\end{aligned}
$$

■

9.4 Proof of Theorem 66: Multidimensional FP equivalence

Proof. First note that we can always assume that $m = 1$ by applying the result componentwise. Similarly, we can always assume that $n = 2$ by applying the result repeatedly. Since FP is robust to the exact encoding used for pairs, we choose a particular encoding to prove the result. Let $\#$ be a fresh symbol not found in Γ and define $\Gamma^\# = \Gamma \cup \{\#\}$. We naturally extend γ to $\gamma^\#$ which maps $\Gamma^\#$ to $\mathbb{N}^*$ injectively. Let $h : {\Gamma^\#}^* \to \Gamma^*$ and define for any $w, w' \in \Gamma^*$:

$$h^\#(w, w') = h(w\#w').$$

It follows[25] that

$$f \in \text{FP} \text{ if and only if } \exists h \in \text{FP} \text{ such that } h^\# = f$$

Assume that $f \in$ FP. Then there exists $h \in$ FP such that $h^\# = f$. Note that h naturally induces a function (still called) $h : {\Gamma^\#}^* \to {\Gamma^\#}^*$ so we can apply Theorem 52 to get that h is emulable over alphabet $\Gamma^\#$. Apply Definition 50 to get $g \in$ ALP and $k \in \mathbb{N}$ that emulate h. In the remaining of the proof, ψ_k denotes encoding of Definition 50 for this particular k, in other words:

$$\psi_k(w) = \left(\sum_{i=1}^{|w|} \gamma^\#(w_i)k^{-i}, |w|\right)$$

Define for any $x, x' \in \mathbb{R}$ and $n, n' \in \mathbb{N}$:

$$\varphi(x, n, x', n) = \left(x + \left(\gamma^\#(\#) + x'\right) k^{-n-1}, n + m + 1\right).$$

We claim that $\varphi \in$ ALP and that for any $w, w' \in \Gamma^*$, $\varphi(\psi_k(w), \psi_k(w')) = \psi_k(w\#w')$. The fact that $\varphi \in$ ALP is immediate using Theorem 20 and the fact that $n \mapsto k^{-n-1}$ is analog-polytime-computable[26]. The second fact is follows from a calculation:

$$
\begin{aligned}
\varphi(\psi_k(w), \psi_k(w')) &= \varphi\left(\sum_{i=1}^{|w|} \gamma^\#(w_i)k^{-i}, |w|, \sum_{i=1}^{|w'|} \gamma^\#(w'_i)k^{-i}, |w'|\right) \\
&= \left(\sum_{i=1}^{|w|} \gamma^\#(w_i)k^{-i} + \left(\gamma^\#(\#) + \sum_{i=1}^{|w'|} \gamma^\#(w'_i)k^{-i}\right) k^{-|w|-1}, |w| + |w'| + 1\right)
\end{aligned}
$$

[25] This is folklore, but mostly because this particular encoding of pairs is polytime computable.

[26] Note that it works only because $n \geqslant 0$.

$$= \left(\sum_{i=1}^{|w\#w'|} \gamma^{\#}((w\#w')_i)k^{-i}, |w\#w'| \right)$$
$$= \psi_k(w\#w').$$

Define $G = g \circ \varphi$. We claim that G emulates f with k. First $G \in \text{ALP}$ thanks to Theorem 21. Second, for any $w, w' \in \Gamma^*$, we have:

$$\begin{aligned}
G(\psi_k(w, w')) &= g(\varphi(\psi_k(w), \psi_k(w'))) && \text{By definition of } G \text{ and } \psi_k \\
&= g(\psi_k(w\#w')) && \text{By the above equality} \\
&= \psi_k(h(w\#w')) && \text{Because } g \text{ emulates } h \\
&= \psi_k(h^{\#}(w, w')) && \text{By definition of } h^{\#} \\
&= \psi_k(f(w, w')). && \text{By the choice of } h
\end{aligned}$$

Conversely, assume that f is emulable. Define $F : \Gamma^{\#^*} \to \Gamma^{\#^*} \times \Gamma^{\#^*}$ as follows for any $w \in \Gamma^{\#^*}$:

$$F(w) = \begin{cases} (w', w'') & \text{if } w = w'\#w'' \text{ where } w', w'' \in \Gamma^* \\ (\lambda, \lambda) & \text{otherwise} \end{cases}.$$

Clearly $F_1, F_2 \in \text{FP}$ so apply Theorem 52 to get that they are emulable. Thanks to Lemma 64, there exists h, g_1, g_2 that emulate f, F_1, f_2 respectively with the same k. Define:

$$H = h \circ (g_1, g_2).$$

Clearly $H \in \text{ALP}$ because $g_1, g_2, h \in \text{ALP}$. Furthermore, H emulates $f \circ F$ because for any $w \in \Gamma^{\#^*}$:

$$\begin{aligned}
H(\psi_k(w)) &= h(g_1(\psi_k(w)), g_2(\psi_k(w))) \\
&= h(\psi_k(g_1(w)), \psi_k(g_2(w))) && \text{Because } g_i \text{ emulates } F_i \\
&= h(\psi_k(F(w))) && \text{By definition of } \psi_k \\
&= \psi_k(f(F(w))). && \text{Because } h \text{ emulates } f
\end{aligned}$$

Since $f \circ F : \Gamma^{\#^*} \to \Gamma^{\#^*}$ is emulable, we can apply Theorem 52 to get that $f \circ F \in \text{FP}$. It is now trivial so see that $f \in \text{FP}$ because for any $w, w' \in \Gamma^*$:

$$f(w, w') = (f \circ F)(w\#w')$$

and $((w, w') \mapsto w\#w') \in \text{FP}$. ■

10 How to only use rational coefficients

This section is devoted to prove that non-rational coefficients can be eliminated. In other words, we prove that Definitions 1 and 67 are defining the same class, and that Definitions 3 and 11 are defining the same class.

Our main Theorems 2 and 4 then clearly follow.

To do so, we introduce the following class. We write $\text{ATSP}_{\mathbb{Q}}$ (resp. $\text{ATSP}_{\mathbb{R}_G}$) for the class of functions f satisfying item (2) of Proposition 12 considering that $\mathbb{K} = \mathbb{Q}$ (resp. $\mathbb{K} = \mathbb{R}_G$) for some polynomials Υ and Π. Recall that $\mathbb{R}_G$ denotes the smallest generable field $\mathbb{R}_G$ lying somewhere between $\mathbb{Q}$ and $\mathbb{R}_P$. We write $\mathrm{AWP}_{\mathbb{Q}}$ (resp. $\mathrm{AWP}_{\mathbb{R}_G}$) for the class of functions f satisfying item (3) of Proposition 12 considering that $\mathbb{K} = \mathbb{Q}$ (resp. $\mathbb{K} = \mathbb{R}_G$) for some polynomials Υ and Π.

We actually show in this section that $\mathrm{AWP}_{\mathbb{R}_G} = \mathrm{ATSP}_{\mathbb{Q}}$. As clearly $\mathrm{ALP}_{\mathbb{K}} = \mathrm{ATSP}_{\mathbb{K}}$ over any field $\mathbb{K}$ [BGP16c], it follows that $\mathrm{ALP} = \mathrm{AWP}_{\mathbb{R}_G} = \mathrm{ATSP}_{\mathbb{Q}} = \mathrm{ALP}_{\mathbb{Q}}$ and hence all results follow.

A particular difficulty in the proof is that none of the previous theorems applies to $\mathrm{AWP}_{\mathbb{Q}}$ and $\mathrm{ATSP}_{\mathbb{Q}}$ because $\mathbb{Q}$ is not a generable field. We thus have to reprove some theorems for the case of rational numbers. In particular, when using rational numbers, we cannot use, in general, the fact that $y' = g(y)$ rewrites to a PIVP if g is generable, because it may introduce some non-rational coefficients.

10.1 Composition in $\mathrm{AWP}_{\mathbb{Q}}$

The first step is to show that $\mathrm{AWP}_{\mathbb{Q}}$ is stable under composition. This is not immediate since $\mathbb{Q}$ is not a generable field and we do not have access to any generable function. The only solution is to manually write a polynomial system with rational coefficients and show that it works. This fact will be crucial for the remainder of the proof.

In order to compose functions, it will be useful always assume $\Pi \equiv 1$ when considering functions in $\mathrm{AWC}_{\mathbb{Q}}(\Upsilon, \Pi)$.

Lemma 83 *If $f \in \mathrm{AWP}_{\mathbb{Q}}$ then there exists Υ a polynomial such that $f \in \mathrm{AWC}_{\mathbb{Q}}(\Upsilon, \Pi)$ where $\Pi(\alpha, \mu) = 1$ for all α and μ.*

Proof. Let $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{AWP}_{\mathbb{Q}}$. By definition, there exists Π and Υ polynomials such that $f \in \mathrm{AWC}(\Upsilon, \Pi)$ with corresponding d, p, q. Without loss of generality, we can assume that Υ and Π are increasing and have rational coefficients. Let $x \in \operatorname{dom} f$ and $\mu \geqslant 0$. Then there exists y such that for all $t \in \mathbb{R}_+$,

$$y(0) = q(x, \mu), \qquad y'(t) = p(y(t)).$$

Consider (z, ψ) the solution to

$$\begin{cases} z(0) = q(x, \mu) \\ \psi(0) = \Pi(1 + x_1^2 + \cdots + x_n^2, \mu) \end{cases} \qquad \begin{cases} z' = p(z) \\ \psi' = 0 \end{cases}$$

Note that the system is polynomial with rational coefficients since Π is a polynomial with rational coefficients. It is easy to see that z and ψ must exist over $\mathbb{R}_+$ and satisfy:

$$\psi(t) = \Pi(\alpha, \mu), \qquad z(t) = y(\psi(t)t)$$

where $\alpha = 1 + x_1^2 + \cdots + x_n^2$. But then for $t \geqslant 1$,

$$\Pi(\alpha, \mu)t \geqslant \Pi(\|x\|, \mu)$$

since Π is increasing and $\alpha = 1 + x_1^2 + \cdots + x_n^2 \geqslant \|x\|$. It follows by definition that,

$$\|z_{1..m}(t) - f(x)\| = \|y_{1..m}(\Pi(\alpha, \mu)t) - f(x)\| \leqslant e^{-\mu}$$

for any $t \geqslant 1$, by definition of y. Finally, since $\alpha \leqslant \operatorname{poly}(\|x\|)$,

$$\begin{aligned}\|(z, \psi)(t)\| &= \max(\|y(\psi(t)t)\|, \psi(t))\\ &\leqslant \max(\Upsilon(\|x\|, \mu, \Pi(\alpha, \mu)t), \Pi(\alpha, \mu)\\ &\leqslant \operatorname{poly}(\|x\|, \mu, t).\end{aligned}$$

This proves that $f \in \mathrm{AWC}(\operatorname{poly}, (\alpha, \mu) \mapsto 1)$ with rational coefficients only. ■

Lemma 84 *If* $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{AWP}_{\mathbb{Q}}$ *and* $r \in \mathbb{Q}^{\ell}[\mathbb{R}^m]$ *then* $r \circ f \in \mathrm{AWP}_{\mathbb{Q}}$.

Proof. Let Π, Υ be polynomials such that $f \in \mathrm{AWC}_{\mathbb{Q}}(\Upsilon, \Pi)$ with corresponding d, p, q. Using Lemma 83, we can assume that $\Pi \equiv 1$. Without loss of generality we also assume that Υ has rational coefficients and is non-decreasing in all variables. Let $x \in \operatorname{dom} g$ and $\mu \geqslant 0$. Let $\hat{q}$ be a polynomial with rational coefficients, to be defined later. Consider the system

$$y(0) = q(x, \hat{q}(x, \mu)), \qquad y' = p(y). \tag{24}$$

Note that by definition $\|f(x) - y_{1..m}(t)\| \leqslant e^{-\hat{q}(x,\mu)}$ for all $t \geqslant 1$. Using a similar proof to Proposition 23, one can see that for any $t \geqslant 1$.

$$\max(\|f(x)\|, \|y_{1..m}(t)\|) \leqslant 2 + \Upsilon(\|x\|, 0, 1). \tag{25}$$

Let $z(t) = r(y_{1..m(t)}(t))$ and observe that

$$z(0) = r(q(x, \hat{q}(x, \mu)), \qquad z'(t) = J_r(y_{1..m}(t))p_{1..m}(y(t)). \tag{26}$$

Note that since r, p and $\hat{q}$ are polynomials with rational coefficients, the system (24),(26) is of the form $w(0) = \operatorname{poly}(x, \mu)$, $w' = \operatorname{poly}(w)$ with rational coefficients, where $w = (y, z)$. Let $k = \deg(r)$, then

$$\begin{aligned}\|r(f(x)) - z(t)\| &= \|r(f(x)) - r(y_{1..m}(t))\|\\ &\leqslant k\Sigma r \max(\|f(x)\|, \|y_{1..m}(t)\|)^{k-1} \|f(x) - y_{1..m}(t)\|\\ &\leqslant k\Sigma r \left(2 + \Upsilon(\|x\|, 0, 1)\right)^{k-1} \|f(x) - y_{1..m}(t)\| && \text{using (25)}\\ &\leqslant k\Sigma r \left(2 + \Upsilon(\|x\|, 0, 1)\right)^{k-1} e^{-\hat{q}(x,\mu)} && \text{by definition of } y.\end{aligned}$$

We now define $\hat{q}(x, \mu) = \mu + k\Sigma r \left(2 + \Upsilon(1 + x_1^2 + \cdots + x_n^2, 0, 1)\right)^{k-1}$. Since Υ has rational coefficients, $\hat{q}$ is indeed a polynomial with rational coefficients. Furthermore, $\|x\| \leqslant 1 + \|x\|_2^2$ and Υ is non-decreasing, thus

$$\hat{q}(x, \mu) = \mu + k\Sigma r \geqslant \mu + k\Sigma r \left(2 + \Upsilon(\|x\|, 0, 1)\right)^{k-1}$$

and we get that

$$\|r(f(x)) - z(t)\| \leqslant k\Sigma r \left(2 + \Upsilon(\|x\|, 0, 1)\right)^{k-1} e^{-\mu + k\Sigma r \left(2+\Upsilon(\|x\|,0,1)\right)^{k-1}} \leqslant e^{-\mu}$$

using that $ue^{-u} \leqslant 1$ for any u. Finally, by construction we have

$$\|y(t)\| \leqslant \Upsilon(\|x\|, \hat{q}(x, \mu), t) \leqslant \operatorname{poly}(\|x\|, \mu, t)$$

and

$$\|z(t)\| = \|r(y_{1..m}(t))\| \leqslant \operatorname{poly}(\|y_{1..m}(t)\|) \leqslant \operatorname{poly}(\|y(t)\|) \leqslant \operatorname{poly}(\|x\|, \mu, t).$$

Thus $r \circ f \in \mathrm{AWP}_{\mathbb{Q}}$. ∎

We also need a technical lemma to provide us with a simplified version of a periodic switching function: a function that is periodically very small then very high (like a clock). Figure 12 gives the graphical intuition behind these functions.

Lemma 85 *Let $\nu \in C^1(\mathbb{R}, \mathbb{R}_+)$* ***with*** *$\nu(0) = 0$ and define for all $t \in \mathbb{Z}$,*

$$\theta_\nu(t) = \tfrac{1}{2} + \tfrac{1}{2}\tanh(2\nu(t)(\sin(2t) - \tfrac{1}{2})).$$

Then

$$\theta_\nu(0) = 0, \qquad \theta_\nu'(t) = p^\theta(\theta_\nu(t), \nu(t), \nu'(t), t, \sin(2t), \cos(2t))$$

where p^θ is a polynomial with rational coefficients. Furthermore, for all $n \in \mathbb{Z}$,

- *if $(n + \frac{1}{2})\pi \leqslant t \leqslant (n+1)\pi$ then $|\theta_\nu(t)| \leqslant e^{-\nu(t)}$,*
- *if $n\pi + \frac{\pi}{12} \leqslant t \leqslant (n + \frac{1}{2})\pi$ then $\theta_\nu(t) \geqslant \frac{1}{2}$.*

Proof. Check that

$$\theta_\nu'(t) = \big(\nu'(t)(\sin(2t) - \tfrac{1}{2}) + 2\nu(t)\cos(2t)\big)(1 - (2\theta_\nu(t) - 1)^2).$$

Recall that for all $x \in \mathbb{R}$, $|\operatorname{sgn}(x) - \tanh(x)| \leqslant e^{-x}$.

- If $t \in [(n + \frac{1}{2})\pi, (n+1)\pi]$, then $\sin(2t) \leqslant 0$ and since tanh is increasing,

$$\theta_\nu(t) \leqslant \tfrac{1}{2} + \tfrac{1}{2}\tanh(-\nu(t)) \leqslant e^{-\nu(t)}.$$

- If $t \in [n\pi + \frac{\pi}{12}, (n + \frac{1}{2})\pi - \frac{\pi}{12}]$ then $\sin(2t) \geqslant \frac{1}{2}$ and $\theta_\nu(t) \geqslant \frac{1}{2}$.

∎

Lemma 86 *Let $\nu \in C^1(\mathbb{R}, \mathbb{R}_+)$* ***with*** *$\nu(0) = 0$ and define for all $t \in \mathbb{Z}$,*

$$\psi_{0,\nu}(t) = \theta_\nu(2t)\theta_\nu(t), \qquad \psi_{1,\nu}(t) = \theta_\nu(-2t)\theta_\nu(t),$$

$$\psi_{2,\nu}(t) = \theta_\nu(2t)\theta_\nu(-t), \qquad \psi_{3,\nu}(t) = \theta_\nu(-2t)\theta_\nu(-t).$$

Then

$$\psi_{i,\nu}(0) = 0, \qquad \theta_{i,\nu}'(t) = p^{i,\psi}(\theta_\nu(t), \theta_\nu'(t), \theta_\nu(2t), \theta_\nu'(2t))$$

where $p^{i,\psi}$ is a polynomial with rational coefficients. Furthermore, for all $i \in \{0, 1, 2, 3\}$ and $n \in \mathbb{Z}$,

- *if $(t \mod \pi) \notin [\frac{i\pi}{4}, \frac{(i+1)\pi}{4}]$ then $|\psi_{i,\nu(t)}| \leqslant e^{-\nu(t)}$,*
- *$m_\psi \leqslant \int_{n\pi + \frac{i\pi}{4}}^{n\pi + \frac{(i+1)\pi}{4}} \psi_{i,\nu(t)} dt \leqslant M_\psi$ for some constants m_ψ, M_ψ that do not depend on ν,*

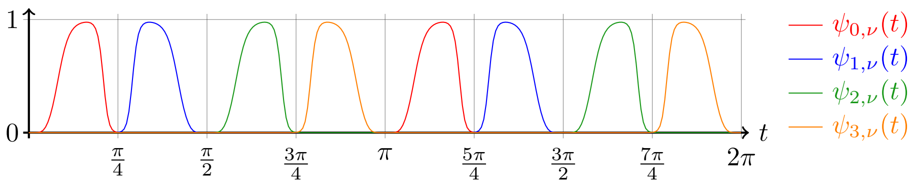


Figure 12: Graph of $\psi_{i,\nu}(t)$ for $\nu(t) = 3$.

- *for any* $\nu, \bar{\nu}$, $i \neq j$ *and* $(t \mod \pi) \in [\frac{i\pi}{4}, \frac{(i+1)\pi}{4}]$, *if* $\nu(t) \leqslant \bar{\nu}(t)$ *then* $\psi_{i,\nu}(t) \geqslant \psi_{j,\bar{\nu}}(t)$.

Proof. Note that $\psi_{i,\nu}(t) \in [0, 1]$ for all $t \in \mathbb{R}$. The first point is direct consequence of Lemma 85 and the fact that $\theta_\nu(-t) = \theta_\nu(t + \frac{\pi}{2})$. The second point requires more work. We only show it for $\psi_{0,\nu}$ since the other cases are similar. Let $n \in \mathbb{Z}$, if $t \in [n\pi + \frac{\pi}{12}, n\pi + \frac{5\pi}{24}]$ then $t \in [n\pi + \frac{\pi}{12}, n\pi + \frac{5\pi}{12}]$ thus $\gamma_\nu(t) \geqslant \frac{1}{2}$, and $2t \in [2n\pi + \frac{\pi}{12}, 2n\pi + \frac{5\pi}{12}]$ thus $\gamma_\nu(2t) \geqslant \frac{1}{2}$. It follows that $\psi_{0,\nu}(t) \geqslant \frac{1}{4}$ and thus

$$\int_{n\pi}^{n\pi+\frac{\pi}{4}} \psi_{0,\nu}(t)dt \geqslant \int_{n\pi+\frac{\pi}{12}}^{n\pi+\frac{5\pi}{24}} \tfrac{1}{4}dt \geqslant \tfrac{3\pi}{96}.$$

On the other hand,

$$\int_{n\pi}^{n\pi+\frac{\pi}{4}} \psi_{0,\nu}(t)dt \leqslant \int_{n\pi}^{n\pi+\frac{\pi}{4}} 1dt \leqslant \frac{\pi}{4}.$$

■

Thanks to the switching functions defined above, the system will construct will often be of a special form that we call “reach”. The properties of this type of system will be crucial for our proof.

Lemma 87 *Let* $d \in \mathbb{N}$, $[a, b] \subset \mathbb{R}$, $z_0 \in \mathbb{R}^d$, $y \in C^1([a, b], \mathbb{R}^d)$ *and* $A, b \in C^0(\mathbb{R}^d \times [a, b], \mathbb{R}^d)$. *Assume that* $A_i(x, t) > |b(x, t)|$ *for all* $t \in [a, b]$ *and* $x \in \mathbb{R}^d$. *Then there exists a unique* $z \in C^1([a, b], \mathbb{R}^d)$ *such that*

$$z(a) = z_0, \qquad z_i'(t) = A_i(z(t), t)(y_i(t) - z_i(t)) + b_i(z(t), t)$$

Furthermore, it satisfies

$$|z_i(t) - y_i(t)| \leqslant \max(1, |z_i(a) - y_i(a)|) + \sup_{s \in [a,t]} |y_i(s) - y_i(a)|, \quad \forall t \in [a, b].$$

Proof. By the Cauchy-Lipschitz theorem, there exists a unique z that satisfies the equation over its maximum interval of life $[a, c)$ with $a < c$. Let $u(t) = z(t) - y(t)$, then

$$\begin{aligned}
u_i'(t) &= z_i'(t) - y_i'(t) \\
&= -A_i(z(t), t)u_i(t) + b_i(z(t), t) - y_i'(t) \\
&= -A_i(u(t) + y(t), t)u_i(t) + b_i(u(t) + y(t), t) - y_i'(t) \\
&= F_i(u(t), y(t), t)
\end{aligned}$$

where

$$F_i(x,t) = -A_i(y(t)+x,t)x_i + b_i(y(t)+x,t) - y_i'(t).$$

But now observe that for any $t \in [a,c]$, $i \in \{1,\dots,d\}$ and $x \in \mathbb{R}^d$,

- if $x_i \geqslant 1$ then $F_i(x,t) < -y_i'(t)$,
- if $x_i \leqslant -1$ then $F_i(x,t) > -y_i'(t)$.

Indeed, if $x_i \geqslant 1$ then

$$\begin{aligned}
F_i(x,t) &= A_i(y(t)+x,t)x_i + b_i(y(t)+x,t) - y_i'(t) \\
&\geqslant A_i(y(t)+x,t) + b_i(y(t)+x,t) - y_i'(t) && \text{using } x_i \geqslant 1 \\
&> |b_i(y(t)+x,t)| + b_i(y(t)+x,t) - y_i'(t) && \text{using } A_i(x,t) > |b_i(x,t)| \\
&\geqslant -y_i'(t)
\end{aligned}$$

and similarly for $x_i \leqslant |y_i'(t)|$. It follows that for all $t \in [a,c)$,

$$|u_i(t)| \leqslant \max(1, |u_i(a)|) + \sup_{s\in[a,t]} |y_i(s) - y_i(a)|. \tag{27}$$

Indeed let $X_t = \{s \in [a,t] : |u_i(s)| \leqslant 1\}$. If $X_t = \varnothing$ then let $t_0 = a$, otherwise let $t_0 = \max X_t$. Then for all $s \in (t_0,t]$, $|u_i(t)| > 1$ thus by continuity of u there are two cases:

- either $u_i(s) > 1$ for all $s \in (t_0,t]$, then $u_i'(s) = F_i(u(s),s) < -y_i'(s))$ thus
$$u_i(t) \leqslant u_i(t_0) - \int_{t_0}^{s} y_i'(u)du = u_i(t_0) + y_i(t) - y_i(t_0),$$
- either $u_i(s) < -1$ for all $s \in (t_0,t]$, then $u_i'(s) = F_i(u(s),s) > -y_i'(s))$ thus
$$u_i(t) \geqslant u_i(t_0) - \int_{t_0}^{s} y_i'(u)du = u_i(t_0) + y_i(t) - y_i(t_0).$$

Thus in all cases

$$|u_i(t)| \leqslant |u_i(t_0)| + |y_i(t) - y_i(t_0)|.$$

But now notice that if $X_t = \varnothing$ then $t_0 = a$ and $|u_i(t_0)| = |u_i(a)|$. And otherwise, $t_0 = \max X_t$ and $|u_i(t_0)| \leqslant 1$.

But note that the upper bound in (27) has a finite limit when $t \to c$ since y is continuous over $[a,b] \supset [a,c)$. This implies that $u(c)$ exists and thus that $c = b$ because if it was not the case, by Cauchy-Lipschitz, we could extend the solution to the right of c and contradict the maximality of $[a,c)$. ■

Lemma 88 *Let $d \in \mathbb{N}$, $z_0, \varepsilon \in \mathbb{R}^d$, $[a,b] \subset \mathbb{R}$, $y \in C^1([a,b],\mathbb{R}^d)$ and $A, b \in C^0(\mathbb{R}^d \times [a,b], \mathbb{R}^d)$. Assume that $A_i(x,t) \geqslant 0$ and $|b_i(x,t)| \leqslant \varepsilon_i$ for all $t \in [a,b]$, $x \in \mathbb{R}^d$ and $i \in \{1,\dots,d\}$. Then there exists a unique $z \in C^1([a,b],\mathbb{R}^d)$ such that*

$$z(a) = z_0, \qquad z_i'(t) = A_i(z(t),t)(y_i(t) - z_i(t)) + b_i(z(t),t)$$

Furthermore, it satisfies

$$|z_i(t) - y_i(t)| \leqslant |z_i(a) - y_i(a)| \exp\left(-\int_a^t A_i(z(s),s)ds\right) + |y_i(t) - y_i(a)| + (t-a)\varepsilon_i.$$

Proof. The existence of a solution over $[a,b]$ is almost immediate since b is bounded. Let $u(t)=z(t)-y(t)$, then

$$u_i'(t)=z_i'(t)-y_i'(t)=-A_i(z(t),t)u_i(t)-y_i'(t)+b_i(z(t),t)$$

and thus we have a the following closed-form expression for u_i:

$$u_i(t)=e^{-\phi(t)}\left(\int_a^t e^{\phi(u)}(b_i(z(u),u)-y_i'(u))du+u_i(0)\right)$$

where

$$\phi(t)=\int_a^t A_i(z(s),s)ds$$

Thus

$$|u_i(t)|\leqslant e^{-\phi(t)}|u_i(0)|+\int_a^t e^{\phi(u)-\phi(t)}|b_i(z(u),u)|du+\left|\int_a^t e^{\phi(u)-\phi(t)}y_i'(u)du\right|.$$

But by the Mean Value Theorem, there exists $c_t\in[a,t]$ such that

$$\int_a^t e^{\phi(u)-\phi(t)}y_i'(u)du=e^{\phi(c_t)-\phi(t)}\int_a^t y_i'(u)du=e^{\phi(c_t)-\phi(t)}(y_i(t)-y_i(a)).$$

Thus by using that ϕ is increasing,

$$\begin{aligned}|u_i(t)|&\leqslant e^{-\phi(t)}|u_i(0)|+\int_a^t|b_i(z(u),u)|du+e^{\phi(c_t)-\phi(t)}|y_i(t)-y_i(a)|\\&\leqslant e^{-\phi(t)}|u_i(0)|+\int_a^t\varepsilon_i du+|y_i(t)-y_i(a)|\\&\leqslant e^{-\phi(t)}|u_i(0)|+(t-a)\varepsilon_i+|y_i(t)-y_i(a)|.\end{aligned}$$

■

We can now show the major result of this subsection: the composition of two functions of $\mathrm{AWP}_\mathbb{Q}$ is in $\mathrm{ATSP}_\mathbb{Q}$, that is computable using only rational coefficients. Note that we are intuitively doing two things at once: showing that the composition is computable, and that weak-computability implies computability; none of which are obvious in the case of rational coefficients.

Theorem 89 *If $f,g\in\mathrm{AWP}_\mathbb{Q}$ then $f\circ g\in\mathrm{ATSP}_\mathbb{Q}$.*

Proof. Let $(f:\subseteq\mathbb{R}^m\to\mathbb{R}^\ell)\,\mathrm{AWP}_\mathbb{Q}$ with corresponding d,p^f,q^f. Let $(g:\subseteq\mathbb{R}^n\to\mathbb{R}^m)\in\mathrm{AWP}_\mathbb{Q}$. Since $g\in\mathrm{AWP}_\mathbb{Q}$, the function $(x,\mu)\mapsto(g(x),\mu)$ trivially belongs to $\mathrm{AWP}_\mathbb{Q}$. Let $h(x,\mu)=q^f(g(x),\mu)$, then $h\in\mathrm{AWP}_\mathbb{Q}$ by Lemma 84 since q^f has rational coefficients. Using Lemma 83, we can assume that $f,h\in\mathrm{AWC}_\mathbb{Q}(\Upsilon,\amalg)$ with $\amalg\equiv1$. Note that we can always make the assumption that Υ is the same for both f and h by taking the maximum. We have $h\in\mathrm{AWP}_\mathbb{Q}$ with corresponding d',p^h,q^h.

To avoid any confusion, note that q^h takes two "μ" as input: the input of h is (x,μ) but q^h adds a ν for the precision: $q^h((x,\mu),\nu)$.

To simplify notations, we will assume that $d=d'$, that is both systems have the same number of variables, by adding useless variables to either system.

Let $x \in \operatorname{dom} g = \operatorname{dom} h$ and $\mu \geqslant 0$. Let R, S and Q be polynomials with rational coefficients, to be defined later, but increasing in all variables. Let m_ψ, M_ψ be the constants from Lemma 86. Without loss of generality, we can assume that the are rational numbers. Consider the following system:

$$\begin{aligned} \mu(0)&= 1, & \mu'(t)&= \psi_{3,\nu_\mu}(t)\alpha, \\ y(0)&= 0, & y_i'(t)&= \psi_{0,\nu_0^i}(t)g_{0,i}(t) + \psi_{1,\nu_1^i}(t)g_{1,i}(t) + \psi_{2,\nu_2^i}(t)g_{2,i}(t) \end{aligned}$$

where

$$\begin{aligned} g_{0,i}(t) &= A_{0,i}(t)(r_i(t) - y_i(t)), \\ g_{1,i}(t) &= \alpha p_i^h(y), \\ g_{2,i}(t) &= \alpha p_i^f(y), \\ A_{0,i}(t) &= \alpha Q(x, \mu(t)) + 2 + g_{1,i}(t)^2 + g_{1,2}(t)^2 \\ \alpha &= \max(1, \tfrac{1}{m_\psi}) \\ r_i(t) &= q_i^h(x, R(x, \mu(t)), S(x, \mu(t))), \\ \nu_0^i(t) &= \big(1 + g_{0,i}(t)^2 + Q(x, \mu(t))\big)\beta t, \\ \nu_1^i(t) &= \nu_0^i(t) + \big(1 + g_{1,i}(t)^2 + Q(x, \mu(t))\big)\beta t, \\ \nu_2^i(t) &= \nu_0^i(t) + \big(1 + g_{2,i}(t)^2 + Q(x, \mu(t))\big)\beta t, \\ \nu_\mu(t) &= \big(\alpha + \pi + Q(x, \mu(t))\big)\beta t, \\ \beta &= 4. \end{aligned}$$

Notice that we took the $\nu_{\dots}(t)$ such that $\nu_{\dots}(0) = 0$ since it will be necessary for the $\psi_{j,\nu}$. This explains the unexpected product by t.

We start with the analysis of μ, which is simplest. First note that $\mu'(t) \geqslant 0$ thus μ is increasing. And since $\mu'(t) \leqslant \alpha$ is bounded, it is clear that μ must exist over $\mathbb{R}$. As a result, since Q is increasing in μ, ν_μ is also an increasing function.

Let $n \in \mathbb{N}$, then

$$\begin{aligned} \mu((n+1)\pi) &= \mu((n + \tfrac{3}{4})\pi) + \int_{(n+\frac{3}{4})\pi}^{(n+1)\pi} \mu'(t)dt \\ &\geqslant \mu(n\pi) + \alpha \int_{(n+\frac{3}{4})\pi}^{(n+1)\pi} \psi_{3,\nu_\mu}(t)dt && \text{since } \mu \text{ increasing} \\ &\geqslant \mu(n\pi) + \alpha m_\psi && \text{by Lemma 86} \\ &\geqslant \mu(n\pi) + 1 && \text{since } \alpha \geqslant m_\psi. \end{aligned}$$

It follows that for all $n \in \mathbb{N}$,

$$\mu(n\pi) \geqslant n + \mu(0) \geqslant n + 1. \tag{28}$$

But on the other hand,

$$\mu(t) = \mu(0) + \int_0^t \mu'(u)du$$

$$
\begin{aligned}
&= 1 + \alpha \int_0^{n\pi} \psi_{3,\nu_\mu}(u) du \\
&\leqslant \alpha \int_0^{n\pi} 1 du \\
&\leqslant 1 + \alpha\pi t. \qquad (29)
\end{aligned}
$$

Let $n \in \mathbb{N}$, then by Lemma 86, for all $t \in [n\pi, (n + \frac{3}{4})\pi]$, $|\mu'(t)| \leqslant \alpha e^{-\nu_\mu(t)}$. So in particular, if $t \geqslant \frac{1}{\beta}$ then $\mu_\nu(t) \geqslant \pi + \alpha + Q(x, \mu(t)) \geqslant \pi + \alpha + Q(x, \mu(n\pi))$. It follows that

$$
|\mu(t) - \mu(t')| \leqslant \tfrac{3}{4}\pi\alpha e^{-\pi-\alpha-Q(x,\mu(n\pi))} \leqslant e^{-Q(x,\mu(n\pi))}, \qquad \forall t, t' \in [n\pi + \tfrac{1}{\beta}, (n + \tfrac{3}{4})\pi]. \tag{30}
$$

We can now start to analyze y. Let $n \in \mathbb{N}$, we will split the analysis in several time intervals that correspond to different behaviors. Note that we chose β such that $\frac{1}{\beta} \leqslant \frac{\pi}{4}$. We use the following fact many times during the proof: $|u| \leqslant 1 + u^2$ for all $u \in \mathbb{R}$.

We will prove the following invariant by induction over $n \in \mathbb{N}$: there exists a polynomial M such that

$$
\|y(n\pi)\| \leqslant M(x, \mu(n\pi)). \tag{31}
$$

At this stage M is still unspecified, but it is a very important requirement that M is **not allowed** to depend Q. Note that (31) is trivially satisfiable for $n = 0$.

Over $[\mathbf{n}\pi, \mathbf{n}\pi + \frac{\mathbf{1}}{\beta}]$: this part is special for $n = 0$, the various $\nu_{\ldots}$ are still "bootstrapping" because of the product by t that we added to make $\mu_{\ldots}(0) = 0$. The only thing we show is that the solution exists, a non-trivial fact at this stage. First note that by constrution, $\nu_1^i(t) \geqslant \nu_0^i(t)$ and $\nu_2^i(t) \geqslant \nu_0^i(t)$. It follows for any $t \in [n\pi, n\pi + \frac{1}{\beta}]$, using Lemma 86 that

$$
\psi_{0,\nu_0^i}(t) \geqslant \psi_{1,\nu_1^i}(t) \quad \text{and} \quad \psi_{0,\nu_0^i}(t) \geqslant \psi_{2,\nu_1^i}(t). \tag{32}
$$

Furthermore, also by construction,

$$
A_{0,i}(t) \geqslant |g_{1,i}(t)| + |g_{2,i}(t)|. \tag{33}
$$

Putting (32) and (33) we get that

$$
A_{0,i}(t)\psi_{0,\nu_0^i}(t) \geqslant |\psi_{1,\nu_1^i}(t) g_{1,i}(t)| + |\psi_{2,\nu_2^i}(t) g_{2,i}(t)|. \tag{34}
$$

Since the system is of the form

$$
y_i'(t) = \psi_{0,\nu_0^i}(t) A_{0,i}(t)(r(t) - y_i(t)) + \psi_{1,\nu_1^i}(t) g_{1,i}(t) + \psi_{2,\nu_2^i}(t) g_{2,i}(t),
$$

we can use (34) to apply Lemma 87 to conclude that y exists over $[n\pi, n\pi + \frac{1}{\beta}]$ and that

$$
|y_i(t) - r_i(t)| \leqslant \max(1, |y_i(n\pi) - r_i(n\pi)|) + \sup_{s \in [n\pi, t]} |r_i(s) - r_i(n\pi)|. \tag{35}
$$

Recall that $r_i(t) = q_i^h(x, R(x, \mu(t)), S(x, \mu(t)))$. So in particular, using (29),

$$
|r_i(t)| \leqslant q_i^h(x, R(x, 1 + \alpha\pi t), S(x, 1 + \alpha\pi t)). \tag{36}
$$

It follows that forall $t \in [n\pi, n\pi + \frac{1}{\beta}]$,

$$\begin{aligned}
|y_i(t) - r_i(t)| &\leqslant \max(1, |y_i(n\pi) - r_i(n\pi)|) + \sup_{s\in[n\pi,t]} |r_i(s) - r_i(n\pi)| && \text{using (35)}\\
&\leqslant 1 + |y_i(n\pi)| + |r_i(n\pi)| + 2 \sup_{s\in[n\pi,t]} |r_i(s)|\\
&\leqslant 1 + |y_i(n\pi)| + 3 \sup_{s\in[n\pi,t]} q_i^h(x, R(x, 1 + \alpha\pi s), S(x, 1 + \alpha\pi s)) && \text{using (36)}\\
&\leqslant 1 + M(x, \mu(n\pi)) + 3 \sup_{s\in[n\pi,t]} q_i^h(x, R(x, 1 + \alpha\pi s), S(x, 1 + \alpha\pi s)) && \text{using (31)}\\
&\leqslant P_1(x, \mu(n\pi)) && (37)
\end{aligned}$$

for some polynomial[27] P_1.

Over $[\mathbf{n}\pi + \frac{\mathbf{1}}{\beta}, (\mathbf{n} + \frac{\mathbf{1}}{\mathbf{4}})\pi]$: it is important to note that in this case, and all remaining cases, $\beta t \geqslant 1$. Indeed by construction we get for all $t \in [n\pi + \frac{1}{\beta}, (n+\frac{1}{4})\pi]$ that

$$\nu_1^i(t) \geqslant |g_{1,i}(t)| + Q(x, \mu(t)) \quad \text{and} \quad \nu_2^i(t) \geqslant |g_{2,i}(t)| + Q(x, \mu(t)).$$

It follows from Lemma 86 and the fact that μ is increasing that

$$|\psi_{1,\nu_1^i}(t) g_{1,i}(t)| \leqslant e^{-\nu_1^i(t)} |g_{1,i}(t)| \leqslant e^{-Q(x,\mu(t))} \leqslant e^{-Q(x,\mu(n\pi))} \tag{38}$$

and

$$|\psi_{2,\nu_1^i}(t) g_{2,i}(t)| \leqslant e^{-\nu_2^i(t)} |g_{2,i}(t)| \leqslant e^{-Q(x,\mu(t))} \leqslant e^{-Q(x,\mu(n\pi))}. \tag{39}$$

Thus we can apply Lemma 88 and get that

$$|y_i(t)-r_i(t)| \leqslant |y_i(n\pi+\tfrac{1}{\beta})-r_i(n\pi+\tfrac{1}{\beta})|e^{-B(t)}+2e^{-Q(x,\mu(n\pi))}+|r_i(t)-r_i(n\pi+\tfrac{1}{\beta})| \tag{40}$$

where

$$\begin{aligned}
B(t) &= \int_{n\pi+\frac{1}{\beta}}^{(n+\frac{1}{4})\pi} \psi_{0,\nu_0^i}(u) A_{0,i}(u) du\\
&\geqslant \int_{n\pi+\frac{1}{\beta}}^{(n+\frac{1}{4})\pi} \psi_{0,\nu_0^i}(u) \alpha Q(x, \mu(u)) du\\
&\geqslant \alpha Q(x, \mu(n\pi)) \int_{n\pi+\frac{1}{\beta}}^{(n+\frac{1}{4})\pi} \psi_{0,\nu_0^i}(u) du && \text{since } Q \text{ and } \mu \text{ increasing}\\
&\geqslant \alpha Q(x, \mu(n\pi)) m_\psi && \text{using Lemma 86}\\
&\geqslant Q(x, \mu(n\pi)) && \text{since } \alpha m_\psi \geqslant 1. \qquad (41)
\end{aligned}$$

Recall that $r_i(t) = q_i^h(x, R(x, \mu(t)), S(x, \mu(t))$ where q_i^h and R are polynomials. It follows that there exists a polynomial[28] Δ_r such that for all $t, t' \geqslant 0$,

$$|r_i(t) - r_i(t')| \leqslant \Delta_r(x, \max(|\mu(t)|, |\mu(t')|)) |\mu(t) - \mu(t')|.$$

[27]Note for later that P_1 depends on q^h, M, R and S.

[28]Note for later that Δ_r depends on q^h, R and S.

And using (29), and (30) we get that

$$|r_i(t) - r_i(t')| \leqslant \Delta_r(x, 1 + \alpha\pi t)e^{-Q(x,\mu(n\pi))}. \tag{42}$$

It follows that Putting , (41) and (42) we get that

$$\begin{aligned}
|y_i(t) - r_i(t)| &\leqslant |y_i(n\pi + \tfrac{1}{\beta}) - r_i(n\pi + \tfrac{1}{\beta})|e^{-B(t)} && \text{using (40)}\\
&\quad + 2e^{-Q(x,\mu(n\pi))} + |r_i(t) - r_i(n\pi + \tfrac{1}{\beta})| \\
&\leqslant P_1(x, \mu(n\pi))e^{-B(t)} + 2e^{-Q(x,\mu(n\pi))} + |r_i(t) - r_i(n\pi + \tfrac{1}{\beta})| && \text{using (37)}\\
&\leqslant P_1(x, \mu(n\pi))e^{-Q(x,\mu(n\pi))} + 2e^{-Q(x,\mu(n\pi))} + |r_i(t) - r_i(n\pi + \tfrac{1}{\beta})| && \text{using (41)}\\
&\leqslant P_1(x, \mu(n\pi))e^{-Q(x,\mu(n\pi))} + 2e^{-Q(x,\mu(n\pi))} + \Delta_r(x, 1 + \alpha\pi t)e^{-Q(x,\mu(n\pi))} && \text{using (42)}\\
&\leqslant P_2(x, \mu(n\pi))e^{-Q(x,\mu(n\pi))} && (43)
\end{aligned}$$

for some polynomial[29] P_2.

Over $[(\mathbf{n} + \tfrac{\mathbf{1}}{\mathbf{4}})\pi, (\mathbf{n} + \tfrac{\mathbf{1}}{\mathbf{2}})\pi]$**:** for all t in this interval,

$$\nu_0^i(t) \geqslant |g_{0,i}(t)| + Q(x, \mu(t)) \quad \text{and} \quad \nu_2^i(t) \geqslant |g_{2,i}(t)| + Q(x, \mu(t)).$$

It follows from Lemma 86 and the fact that μ is increasing that

$$|\psi_{0,\nu_1^i}(t)g_{0,i}(t)| \leqslant e^{-\nu_0^i(t)}|g_{0,i}(t)| \leqslant e^{-Q(x,\mu(t))} \leqslant e^{-Q(x,\mu(n\pi))} \tag{44}$$

and

$$|\psi_{2,\nu_1^i}(t)g_{2,i}(t)| \leqslant e^{-\nu_2^i(t)}|g_{2,i}(t)| \leqslant e^{-Q(x,\mu(t))} \leqslant e^{-Q(x,\mu(n\pi))}. \tag{45}$$

Consequently, the system is of the form

$$y_i'(t) = \alpha\psi_{1,\nu_1^i}(t)p_i^h(y(t)) + \varepsilon_i(t) \quad \text{where} \quad |\varepsilon_i(t)| \leqslant 2e^{-Q(x,\mu(n\pi))}. \tag{46}$$

For any $t \in [(n + \frac{1}{4})\pi, (n + \frac{1}{2})\pi]$, let

$$\xi(t) = (n + \tfrac{1}{4})\pi + \int_{(n+\frac{1}{4})\pi}^{t} \alpha\psi_{1,\nu_1^i}(u)du.$$

Since $\psi_{1,\nu_1^i} > 0$, ξ is increasing and invertible. Now consider the following system:

$$z_i((n + \tfrac{1}{4})\pi) = y_i((n + \tfrac{1}{4})\pi), \qquad z_i'(u) = p_i^h(z(u)) + \varepsilon(\xi^{-1}(u)). \tag{47}$$

It follows that, on the interval of life,

$$y_i(t) = z_i(\xi(t)). \tag{48}$$

Note using Lemma 86 that

$$1 \leqslant \alpha m_\psi \leqslant \xi((n + \tfrac{1}{2})\pi) - \xi((n + \tfrac{1}{4})\pi) \leqslant \alpha M_\psi. \tag{49}$$

Now consider the following system:

$$w_i((n + \tfrac{1}{4})\pi) = q_i^h(x, R(x, \mu(n\pi)), S(x, \mu(n\pi))), \qquad w_i'(u) = p_i^h(z(u)). \tag{50}$$

[29]Note that P_2 depends on P_1 and Δ_r. In particular it does not depend, even indirectly, on Q.

By definition of q^h and p^h, the solution w exists over $\mathbb{R}$ and satisfies that

$$|w_i(u) - h_i(x, R(x, \mu(n\pi)))| \leqslant e^{-S(x,\mu(n\pi))} \quad \text{for all } u - (n + \tfrac{1}{4})\pi \geqslant 1 \tag{51}$$

since $h \in \mathrm{AWC}_{\mathbb{Q}}(\Upsilon, \mathrm{II})$ with $\mathrm{II} \equiv 1$, and

$$\begin{aligned} |w_i(u)| &\leqslant \Upsilon(\|(x, R(x, \mu(n\pi)))\|, S(x, \mu(n\pi)), u - (n + \tfrac{1}{4})\pi) \\ &\leqslant P_3(x, \mu(n\pi), u - (n + \tfrac{1}{3})\pi) \qquad \text{for all } u \in \mathbb{R} \end{aligned} \tag{52}$$

for some polynomial[30] P_3. Following Theorem 16 of [BGP16c], let $\eta > 0$ and $a = (n + \frac{1}{4})\pi$ and let

$$\delta_\eta(u) = \left(\|z(a) - w(a)\| + \int_a^u \left\|\varepsilon(\xi^{-1}(s))\right\| ds\right) \exp\left(k\Sigma p^h \int_a^u (\|w(s)\| + \eta)^{k-1} ds\right) \tag{53}$$

where $k = \deg(p^h)$. Let $u \in [a, b]$ where $b = \xi((n + \frac{1}{2})\pi)$, then

$$\begin{aligned} \int_a^b \left\|\varepsilon(\xi^{-1}(s))\right\| ds &\leqslant 2(b - a)e^{-Q(x,\mu(n\pi))} && \text{using (46),} \\ \|z(a) - w(a)\| &= \left\|q^h(x, R(x, \mu(n\pi), S(x, \mu(n\pi)))) - y(a)\right\| \\ &= \|r(n\pi) - y(a)\| \\ &\leqslant P_2(x, \mu(n\pi))e^{-Q(x,\mu(n\pi))} && \text{using (43),} \\ k\Sigma p^h \int_a^b (\|w(s)\| + \eta)^{k-1} ds &\leqslant k\Sigma p^h (b - a)\big(\eta + P_3(x, \mu(n\pi), b)\big)^{k-1} && \text{using (52),} \\ b &\leqslant a + \alpha M_\psi && \text{using (49).} \end{aligned}$$

Plugging everything into (53) we get that for all $u \in [a, b]$,

$$\delta_1(u) \leqslant P_4(x, \mu(n\pi))e^{-Q(x,\mu(n\pi))}e^{P_5(x,\mu(n\pi))} \tag{54}$$

for some polynomials[31] P_4 and P_5. Since we have no chosen Q yet, we now let

$$Q(x, \nu) = P_5(x, \nu) + P_4(x, \nu) + Q^*(x, \nu) \tag{55}$$

where Q^* is some unspecified polynomial to be fixed later. Note that this definition makes sense because P_4 and P_5 do not (even indirectly) depend on Q. It then follows from (54) that

$$\delta_1(u) \leqslant e^{-Q^*(x,\mu(n\pi))} \leqslant 1$$

and thus we can apply Theorem 16 of [BGP16c] to get that

$$|z_i(u) - w_i(u)| \leqslant \delta_1(u) \leqslant e^{-Q^*(x,\mu(n\pi))} \qquad \text{for all } u \in [a, b]. \tag{56}$$

But in particular, (49) implies that $b - a \geqslant 1$ so by (51)

$$|z_i(b) - h_i(x, R(x, \mu(n\pi)))| \leqslant e^{-Q^*(x,\mu(n\pi))} + e^{-S(x,\mu(n\pi))}. \tag{57}$$

[30]Note that P_3 depends on Υ, R and S.
[31]Note that P_4 depends on P_2 and P_5 on Υ and p^h.

And finally, using (48) we get that

$$|y_i((n+\tfrac{1}{2})\pi) - h_i(x, R(x,\mu(n\pi)))| \leqslant e^{-Q^*(x,\mu(n\pi))} + e^{-S(x,\mu(n\pi))}. \tag{58}$$

At this stage, we let

$$Q^*(x,\nu) = S(x,\nu) + R(x,\nu) \tag{59}$$

so that

$$|y_i((n+\tfrac{1}{2})\pi) - h_i(x, R(x,\mu(n\pi)))| \leqslant 2e^{-S(x,\mu(n\pi))}. \tag{60}$$

Over $[(\mathbf{n}+\mathbf{\tfrac{1}{2}})\pi, (\mathbf{n}+\mathbf{\tfrac{3}{4}})\pi]$**:** the situation is very similar to the previous case so we omit some proof steps. The system is of the form

$$y_i'(t) = \alpha\psi_{2,\nu_1^i}(t)p_i^f(y(t)) + \varepsilon_i(t) \quad \text{where} \quad |\varepsilon_i(t)| \leqslant 2e^{-Q(x,\mu(n\pi))}. \tag{61}$$

We let

$$\xi(t) = (n+\tfrac{1}{2})\pi + \int_{(n+\frac{1}{2})\pi}^{t} \alpha\psi_{1,\nu_1^i}(u)du$$

and consider the following system:

$$z_i((n+\tfrac{1}{2})\pi) = y_i((n+\tfrac{1}{2})\pi), \qquad z_i'(u) = p_i^f(z(u)) + \varepsilon(\xi^{-1}(u)). \tag{62}$$

It follows that, on the interval of life,

$$y_i(t) = z_i(\xi(t)). \tag{63}$$

It is again the case that

$$1 \leqslant \alpha m_\psi \leqslant \xi((n+\tfrac{3}{4})\pi) - \xi((n+\tfrac{1}{2})\pi) \leqslant \alpha M_\psi. \tag{64}$$

We introduce the following system:

$$w_i((n+\tfrac{1}{2})\pi) = q_i^f(g(x), R(x,\mu(n\pi))), \qquad w_i'(u) = p_i^f(z(u)). \tag{65}$$

By definition of q^f and p^f, the solution w exists over $\mathbb{R}$ and satisfies that

$$|w_i(u) - f_i(g(x))| \leqslant e^{-R(x,\mu(n\pi))} \quad \text{for all } u - (n+\tfrac{1}{2})\pi \geqslant 1 \tag{66}$$

since $f \in \mathrm{AWC}_{\mathbb{Q}}(\Upsilon, \mathrm{II})$ with $\mathrm{II} \equiv 1$, and

$$\begin{aligned} |w_i(u)| &\leqslant \Upsilon(\|g(x)\|, R(x,\mu(n\pi)), u - (n+\tfrac{1}{2})\pi) \\ &\leqslant P_6(x, \mu(n\pi), u - (n+\tfrac{1}{2})\pi) \qquad \text{for all } u \in \mathbb{R} \end{aligned} \tag{67}$$

for some polynomial[32] P_6 since $\|g(x)\| \leqslant 1+\Upsilon(\|x\|, 0, 1)$. Following Theorem 16 of [BGP16c], let $\eta > 0$ and $a = (n+\frac{1}{2})\pi$ and let

$$\delta_\eta(u) = \left(\|z(a) - w(a)\| + \int_a^u \left\|\varepsilon(\xi^{-1}(s))\right\| ds\right) \exp\left(k\Sigma p^f \int_a^u (\|w(s)\| + \eta)^{k-1} ds\right) \tag{68}$$

where $k = \deg(p^f)$. Let $u \in [a, b]$ where $b = \xi((n+\frac{1}{2})\pi)$, then

$$\int_a^b \left\|\varepsilon(\xi^{-1}(s))\right\| ds \leqslant 2(b-a)e^{-Q(x,\mu(n\pi))} \qquad \text{using (61)},$$

[32] Note that P_6 depends on Υ and R.

$$
\begin{aligned}
&& \leqslant 2(b-a)e^{-S(x,\mu(n\pi))} && \text{using (55) and (59),}\\
\|z(a)-w(a)\| &&= \left\|q^f(g(x),R(x,\mu(n\pi)))-y(a)\right\| &&\\
&&= \|h(x,R(x,\mu(n\pi)))-y(a)\| &&\\
&&\leqslant 2e^{-S(x,\mu(n\pi))} && \text{using (60),}\\
k\Sigma p^f\int_a^b(\|w(s)\|+\eta)^{k-1}ds &&\leqslant k\Sigma p^h(b-a)\big(\eta+P_6(x,\mu(n\pi),b)\big)^{k-1} && \text{using (67),}\\
b &&\leqslant a+\alpha M_\psi && \text{using (64).}
\end{aligned}
$$

Plugging everything into (68) we get that for all $u\in[a,b]$,

$$\delta_1(u)\leqslant P_7(x,\mu(n\pi))e^{-S(x,\mu(n\pi))}e^{P_8(x,\mu(n\pi))} \tag{69}$$

for some polynomials[33] P_7 and P_8. Since we have no chosen S yet, we now let

$$S(x,\nu)=P_7(x,\nu)+P_8(x,\nu)+S^*(x,\nu) \tag{70}$$

where S^* is some unspecified polynomial to be fixed later. Note that this definition makes sense because P_7 and P_8 do not (even indirectly) depend on S. It then follows from (69) that

$$\delta_1(u)\leqslant e^{-S^*(x,\mu(n\pi))}\leqslant 1$$

and thus we can apply Theorem 16 of [BGP16c] to get that

$$|z_i(u)-w_i(u)|\leqslant\delta_1(u)\leqslant e^{-S^*(x,\mu(n\pi))}\qquad\text{for all } u\in[a,b]. \tag{71}$$

But in particular, (64) implies that $b-a\geqslant 1$ so by (66)

$$|z_i(b)-f_i(g(x))|\leqslant e^{-S^*(x,\mu(n\pi))}+e^{-R(x,\mu(n\pi))}. \tag{72}$$

And finally, using (63) we get that

$$|y_i((n+\tfrac{3}{4})\pi)-f_i(g(x))|\leqslant e^{-S^*(x,\mu(n\pi))}+e^{-R(x,\mu(n\pi))}. \tag{73}$$

Finally we let

$$S^*(x,\nu)=R(x,\nu) \tag{74}$$

so that

$$|y_i((n+\tfrac{3}{4})\pi)-f_i(g(x))|\leqslant 2e^{-R(x,\mu(n\pi))}. \tag{75}$$

Also note using (63), (67) and (71) that

$$|y_i(t)|\leqslant 1+P_6(x,\mu(n\pi),b-a)\leqslant P_9(x,\mu(n\pi)) \tag{76}$$

for some polynomial[34] P_9.

Over $[(\mathbf{n}+\tfrac{\mathbf{3}}{\mathbf{4}})\pi,(\mathbf{n}+\mathbf{1})\pi]$**:** for all $j\in\{0,1,2\}$, apply Lemma 86 to get that

$$|\psi_{j,\nu_j^i}(t)|\leqslant e^{-\nu_j^i(t)}\quad\text{and}\quad\nu_j^i(t)\geqslant|g_{j,i}(t)|+Q(x,\mu(t)). \tag{77}$$

It follows that

$$|y_i'(t)|\leqslant 3e^{-Q(x,\mu(t))}\leqslant 3e^{-Q(x,\mu(n\pi))} \tag{78}$$

[33]Note that P_7 depends on P_6 and P_8 on Υ and p^f.
[34]Note that P_9 depends on P_6.

and

$$|y_i(t) - y_i((n+\tfrac{3}{4})\pi)| \leqslant \int_{(n+\frac{3}{4})\pi}^{t} |y_i'(u)|du \leqslant 3e^{-Q(x,\mu(t))} \leqslant 5e^{-Q(x,\mu(n\pi))}. \quad (79)$$

And thus

$$\begin{aligned}
|y_i(t) - f_i(g(x))| &\leqslant |y_i(t) - y_i((n+\tfrac{3}{4})\pi)| + |y_i((n+\tfrac{3}{4})\pi) - f_i(g(x))| \\
&\leqslant 3e^{-Q(x,\mu(n\pi))} + |y_i((n+\tfrac{3}{4})\pi) - f_i(g(x))|| && \text{using (79)} \\
&\leqslant 3e^{-Q(x,\mu(n\pi))} + 2e^{-R(x,\mu(n\pi))} && \text{using (75)} \\
&\leqslant 5e^{-R(x,\mu(n\pi))} && \text{using (55) and (59).}
\end{aligned} \quad (80)$$

It follows using (79) and (76) that

$$\begin{aligned}
\|y((n+1)\pi)\| &\leqslant 1 + \left\|y((n+\tfrac{3}{4})\pi)\right\| \\
&\leqslant 1 + P_9(x,\mu(n\pi)).
\end{aligned}$$

We can thus let

$$M(x,\nu) = 1 + P_9(x,\nu) \quad (81)$$

to get the induction invariant. Note, as this is crucial for the proof, that M does not depend, even indirectly, on Q.

We are almost done: the system for y computes $f(g(x))$ with increasing precision in the time intervals $[(n+\frac{3}{4})\pi),(n+1)\pi]$ but the value could be anything during the rest of the time. To solve this issue, we create an extra system that "samples" y during those time intervals, and does nothing the rest of the time. Consider the system

$$z_i(0) = 0, \qquad z_i'(t) = \psi_{3,\nu_3^i}(t) g_{3,i}(t)$$

where

$$\begin{aligned}
g_{0,i}(t) &= A_{3,i}(t)(y_i(t) - z_i(t)), \\
A_{3,i}(t) &= \alpha R(x,\mu(t)) + \alpha N(x,\mu(t)) \\
\nu_3^i(t) &= \big(3 + g_{3,i}(t)^2 + R(x,\mu(t))\big)\beta t.
\end{aligned}$$

We will show the following invariant by induction n:

$$\|z(n\pi)\| \leqslant N(x,\mu(n\pi)) \quad (82)$$

for some polynomial N to be fixed later that **is not allowed to depend on** R. Note that since $z(0)=0$, it is trivially satisfiable for $n=0$.

Over $[\mathbf{0}.\frac{\mathbf{1}}{\beta}]$: similarly to y, the existence of z is not clear over this time interval because of the bootstrap time of ν_3^i. Since the argument is very similar to that of y (simpler in fact), we do not repeat it.

Over $[\mathbf{n}\pi,(\mathbf{n}+\frac{\mathbf{3}}{\mathbf{4}})\pi]$ for $\mathbf{n}\geqslant\mathbf{1}$: apply Lemma 86 to get that

$$|\psi_{3,\nu_3^i}(t)| \leqslant e^{-\nu_3^i(t)} \leqslant e^{-|g_{i,3}(t)|-Q(x,\mu(n\pi))-2}.$$

It follows that for all $t \in [n\pi, (n+\frac{3}{4})\pi]$,

$$|z_i(t) - z_i(n\pi)| \leqslant \tfrac{3}{4}\pi e^{-R(x,\mu(n\pi))-2} \leqslant e^{-R(x,\mu(n\pi))} \leqslant 1. \tag{83}$$

Over $[(\mathbf{n}+\frac{\mathbf{3}}{\mathbf{4}})\pi, (\mathbf{n}+\mathbf{1})\pi]$**:** apply Lemma 88 to get that

$$|z_i(t) - y_i(t)| \leqslant |z_i((n+\tfrac{3}{4})\pi) - y_i((n+\tfrac{3}{4})\pi)|e^{-B(t)} + |y_i(t) - y_i((n+\tfrac{3}{4})\pi)| \tag{84}$$

where

$$B(t) = \int_{(n+\frac{3}{4})\pi}^{t} A_{3,i}(u)\psi_{3,\nu_3^i}(u)du.$$

Let $b = (n+1)\pi$, then

$$\begin{aligned}
B(b) &= \int_{(n+\frac{3}{4})\pi}^{(n+1)\pi} A_{3,i}(u)\psi_{3,\nu_3^i}(u)du \\
&\geqslant \alpha(R(x,\mu(n\pi)) + N(x,\mu(n\pi))) \int_{(n+\frac{3}{4})\pi}^{(n+1)\pi} \psi_{3,\nu_3^i}(u)du \quad \text{using Lemma 86} \\
&\geqslant (R(x,\mu(n\pi)) + N(x,\mu(n\pi)))\alpha m_\psi \\
&\geqslant R(x,\mu(n\pi)) + N(x,\mu(n\pi)) \qquad \text{using } \alpha m_\psi \geqslant 1.
\end{aligned}$$

It follows that

$$\begin{aligned}
|z_i(b) - y_i(b)| &\leqslant |z_i((n+\tfrac{3}{4})\pi) - y_i((n+\tfrac{3}{4})\pi)|e^{-R(x,\mu(n\pi))-N(x,\mu(n\pi))} \\
&\quad + |y_i(t) - y_i((n+\tfrac{3}{4})\pi)| \\
&\leqslant \big(|z_i((n+\tfrac{3}{4})\pi)| + |y_i((n+\tfrac{3}{4})\pi)|\big)e^{-R(x,\mu(n\pi))-N(x,\mu(n\pi))} \\
&\quad + |y_i(t) - y_i((n+\tfrac{3}{4})\pi)| \\
&\leqslant \big(|z_i(n\pi)| + 1 + |y_i((n+\tfrac{3}{4})\pi)|\big)e^{-R(x,\mu(n\pi))-N(x,\mu(n\pi))} \\
&\quad + 5e^{-R(x,\mu(n\pi))} && \text{using (79)} \\
&\leqslant \big(N(x,\mu(n\pi)) + 1 + |y_i((n+\tfrac{3}{4})\pi)|\big)e^{-R(x,\mu(n\pi))-N(x,\mu(n\pi))} && \text{using (82)} \\
&\quad + 5e^{-R(x,\mu(n\pi))} \\
&\leqslant |y_i((n+\tfrac{3}{4})\pi)|e^{-R(x,\mu(n\pi))-N(x,\mu(n\pi))} + 7e^{-R(x,\mu(n\pi))} \\
&\leqslant (|y_i((n+1)\pi)| + 1)e^{-R(x,\mu(n\pi))-N(x,\mu(n\pi))} + 7e^{-R(x,\mu(n\pi))} && \text{using (79)} \\
&\leqslant (M(x,\mu(n\pi)) + 1)e^{-R(x,\mu(n\pi))-N(x,\mu(n\pi))} + 7e^{-R(x,\mu(n\pi))} && \text{using (31).}
\end{aligned}$$

Since we have not specified N yet, we can take

$$N(x,\nu) = M(x,\mu) \tag{85}$$

so that

$$|z_i(b) - y_i(b)| \leqslant 8e^{-R(x,\mu(n\pi))}. \tag{86}$$

It follows that

$$\begin{aligned}
|z_i(b) - f_i(g(x))| &\leqslant |z_i(t) - y_i(t)| + |y_i(t) - f_i(g(x))| \\
&\leqslant 8e^{-R(x,\mu(n\pi))} + |y_i(t) - f_i(g(x))| \qquad \text{using (86)}
\end{aligned}$$

$$
\begin{aligned}
&\leqslant 8e^{-R(x,\mu(n\pi))} + 5e^{-R(x,\mu(n\pi)} && \text{using (80)} \\
&\leqslant 13e^{-R(x,\mu(n\pi))}. && (87)
\end{aligned}
$$

Furthermore (84) gives that for all $t \in [(n+\frac{3}{4})\pi, (n+1)\pi]$,

$$
\begin{aligned}
|z_i(t) - y_i(t)| &\leqslant |z_i((n+\tfrac{3}{4})\pi) - y_i((n+\tfrac{3}{4})\pi)| + |y_i(t) - y_i((n+\tfrac{3}{4})\pi)| \\
&\leqslant |z_i((n+\tfrac{3}{4})\pi) - y_i((n+\tfrac{3}{4})\pi)| + 5e^{-Q(x,\mu(n\pi))} && \text{using (79)} \\
&\leqslant |z_i((n+\tfrac{3}{4})\pi) - z_i(n\pi)| + |z_i(n\pi) - f_i(g(x))| \\
&\quad + |f_i(g(x)) - y_i((n+\tfrac{3}{4})\pi)| + 5e^{-Q(x,\mu(n\pi))} \\
&\leqslant e^{-R(x,\mu(n\pi))} + |z_i(n\pi) - f_i(g(x))| && \text{using (83)} \\
&\quad + 2e^{-R(x,\mu(n\pi))} + 5e^{-Q(x,\mu(n\pi))} && \text{using (75)} \\
&\leqslant 8e^{-R(x,\mu(n\pi))} + |z_i(n\pi) - f_i(g(x))|. && (88)
\end{aligned}
$$

We can now leverage this analysis to conclude: putting (83) and (87) together we get that

$$
|z_i(t) - f_i(g(x))| \leqslant 14e^{-R(x,\mu(n\pi))} \qquad \text{for all } t \in [(n+1)\pi, (n+\tfrac{7}{4})\pi] \tag{89}
$$

and for all $t \in [(n+\frac{7}{4})\pi, (n+2)\pi]$,

$$
\begin{aligned}
|z_i(t) - f_i(g(x))| &\leqslant |z_i(t) - y_i(t)| + |y_i(t) - f_i(g(x))| \\
&\leqslant 8e^{-R(x,\mu(n\pi))} + |z_i(n\pi) - f_i(g(x))| + |y_i(t) - f_i(g(x))| && \text{using (88)} \\
&\leqslant 8e^{-R(x,\mu(n\pi))} + |z_i(n\pi) - f_i(g(x))| + 2e^{-R(x,\mu(n\pi))} && \text{using (75)} \\
&\leqslant 10e^{-R(x,\mu(n\pi))} + |z_i(n\pi) - f_i(g(x))|. && (90)
\end{aligned}
$$

And finally, putting (89) and (90) together, we get that

$$
|z_i(t) - f_i(g(x))| \leqslant 24e^{-R(x,\mu(n\pi))} \qquad \text{for all } t \in [(n+1)\pi, (n+2)\pi] \tag{91}
$$

Since we have not specified R yet, we can take

$$
R(x,\nu) = 24 + \nu \tag{92}
$$

so that

$$
|z_i(t) - f_i(g(x))| \leqslant e^{-\mu(n\pi)} \leqslant e^{-n} \qquad \text{for all } t \in [(n+1)\pi, (n+2)\pi]. \tag{93}
$$

This concludes the proof that $f \circ g \in \mathrm{ATSP}_{\mathbb{Q}}$ since we have proved that the system converges quickly, has bounded values and the entire system has a polynomial right-hand side using rational numbers only. ■

10.2 From $\mathrm{AWP}_{\mathbb{R}_G}$ to $\mathrm{AWP}_{\mathbb{Q}}$

The second step of the proof is to recast the problem entirely in the language of $\mathrm{AWP}_{\mathbb{Q}}$. The observation is that given a system, corresponding to $f \in \mathrm{AWP}_{\mathbb{R}_G}$, we can abstract away the coefficients and make them part of the input, so that $f(x) = g(x,\alpha)$ where $g \in \mathrm{AWP}_{\mathbb{Q}}$ and $\alpha \in \mathbb{R}_G^k$. We then show that we can see α

as the result of a computation itself: we build $h \in \mathrm{AWP}_{\mathbb{Q}}$ such that $\alpha = h(1)$. Now we are back to $x = g(x, h(1))$, in other words a composition of functions in $\mathrm{AWP}_{\mathbb{Q}}$.

First, let us recall the definition of $\mathbb{R}_G$ from [BGP16b]:

$$\mathbb{R}_G = \bigcup_{n \geqslant 0} G^{[n]}(\mathbb{Q})$$

where

$$G(X) = \{f(1) : (f : \mathbb{R} \to \mathbb{R}) \in \mathrm{GPVAL}_X\}.$$

Note that in [BGP16b], we defined G slightly differently using GVAL, the class of generable functions, instead of GPVAL. Those two definitions are equivalent because if $f \in \mathrm{GVAL}[X]$, we can define $h(t) = f(\frac{2t}{1+t^2})$ that is such that $h(0) = f(0)$, $h(1) = f(1)$ and belongs to GPVAL_X.

Lemma 90 *Let* $(f :\in \mathrm{AWP}_X$ *where* $\mathbb{Q} \subseteq X$*, then there exists* $\ell \in \mathbb{N}$*,* $\beta \in X^\ell$ *and* $h \in \mathrm{AWP}_{\mathbb{Q}}$ *with* $\operatorname{dom} h = \operatorname{dom} f$ *such that* $f = h \circ g$ *where* $g(x) = (x, \beta)$ *for all* $x \in \operatorname{dom} f$*.*

Proof. Let Π and Υ polynomials such that $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{AWC}(\Upsilon, \Pi)$ with corresponding d, q and p. Let $x \in \operatorname{dom} f$ and $\mu \geqslant 0$ and consider the following system:

$$y(0) = q(x, \mu), \qquad y'(t) = q(y(t)).$$

By definition, for any $t \geqslant \Pi(\|x\|, \mu)$,

$$\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$$

and for all $t \geqslant 0$,

$$\|y(t)\| \leqslant \Upsilon(\|x\|, \mu, t).$$

Let ℓ be the number of nonzero coefficients of p and q. Then there exists $\beta \in R^\ell$, $\hat{p} \in \mathbb{Q}^d[\mathbb{R}^{d+\ell}]$ and $\hat{q} \in \mathbb{Q}^d[\mathbb{R}^{n+1+\ell}]$ such that for all $x \in \mathbb{R}^n$, $\mu \geqslant 0$ and $u \in \mathbb{R}^d$,

$$q(x, \mu) = \hat{q}(x, \beta, \mu) \quad \text{and} \quad p(u) = \hat{p}(u, \beta).$$

Now consider the following system for any $(x, w) \in \operatorname{dom} f \times \{\beta\}$ and $\mu \geqslant 0$:

$$\begin{array}{ll} u(0) = w \qquad\quad , & u'(t) = 0, \\ z(0) = \hat{q}(x, w, \mu), & z'(t) = \hat{p}(z(t), u(t)). \end{array}$$

Note that this system only has rational coefficients because $\hat{q}$ and $\hat{p}$ have rational coefficients. Also $u(t)$ is the constant function equal to w and $w = \beta$ since $(x, w) \in \operatorname{dom} f \times \{\beta\}$. Thus $z'(t) = \hat{p}(z(t), \beta) = p(z(t))$, and $z(0) = \hat{q}(x, \beta, \mu) = q(x, \mu)$. It follows that $z \equiv y$ and thus this system weakly-computes $h(x, w) = f(x)$:

$$\|z_{1..m}(t) - f(x)\| = \|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$$

and

$$\|(u(t), z(t))\| \leqslant \|u(t)\| + \|z(t)\| \leqslant \|w\| + \Upsilon(\|x\|, \mu, t).$$

Thus $h \in \mathrm{AWP}_{\mathbb{Q}}$. It is clear that if $g(x) = (x, \beta)$ then $(h \circ g)(x) = h(x, \beta) = f(x)$.
■

Lemma 91 *For any $X \supseteq \mathbb{Q}$ and $(f : \mathbb{R} \to \mathbb{R}) \in \mathrm{GPVAL}_X$, $(x \in \mathbb{R} \mapsto f(1)) \in \mathrm{AWP}_X$.*

Proof. Expand the definition of f to get $d \in \mathbb{N}$, $y_0 \in X^d$ and $p \in X[\mathbb{R}^d]$ such that

$$y(0) = y_0, \qquad y'(t) = p(y(t))$$

satisfies for all $t \in \mathbb{R}$,

$$f(t) = y_1(t).$$

Now consider the following system for $x \in \mathbb{R}$ and $\mu \geqslant 0$:

$$\begin{array}{ll} \psi(0) = 1 \ , & \psi'(t) = -\psi(t), \\ z(0) = y_0, & z'(t) = \psi(t)p(z(t)). \end{array}$$

This system only has coefficients in X and it is not hard to see that

$$\psi(t) = e^{-t} \quad \text{and} \quad z(t) = y\left(\int_0^t \psi(s)ds\right) = y\left(1 - e^{-t}\right).$$

Furthermore, since $f(1) = y_1(1)$,

$$\begin{aligned} |f(1) - z_1(t)| &= |y_1(1) - y_1(1 - e^{-t})| \\ &= \left|\int_{1-e^{-t}}^1 y_1'(s)ds\right| \\ &\leqslant \int_{1-e^{-t}}^1 |p_1(y(s))|ds \\ &\leqslant e^{-t} \sup_{s\in[0,1]} |p_1(y(s))|. \end{aligned}$$

Let $A = \sup_{s\in[0,1]} |p_1(y(s))|$ which is finite because y is continuous and $[0, 1]$ is compact, and let $\Pi(x, \mu) = \mu + A$. Then for any $\mu \geqslant 0$ and $t \geqslant \Pi(\|x\|, \mu)$,

$$|f(1) - z_1(t)| \leqslant e^{-t}A \leqslant e^{-\Pi(\|x\|,\mu)}A \leqslant e^{-\mu-A}A \leqslant e^{-\mu}.$$

Furthermore,

$$\|z(t)\| = \left\|y(1 - e^t)\right\| \leqslant \sup_{s\in[0,1]} \|y(s)\|$$

where the right-hand is a finite constant because y is continuous and $[0, 1]$. This shows that $(x \in \mathbb{R} \mapsto f(1)) \in \mathrm{AWP}_X$. ∎

Proposition 92 *For all $n \in \mathbb{N}$, $\mathrm{AWP}_{G^{[n]}(\mathbb{Q})} \subseteq \mathrm{ATSP}_{\mathbb{Q}}$.*

Proof. When $n = 0$, the result is trivial because $G^{[0]}(\mathbb{Q}) = \mathbb{Q}$.

Assume the result is true for n and take $f \in \mathrm{AWP}_{G^{[n+1]}(\mathbb{Q})}$. Apply Lemma 90 to get $h \in \mathrm{AWP}_{\mathbb{Q}}$ such that $f = h \circ g$ where $g(x) = (x, \beta)$ where $\beta \in G^{[n+1]}(\mathbb{Q})^\ell$ for some $\ell \in \mathbb{N}$. Let $i \in \{1, \dots, \ell\}$, by definition of β_i, there exists $y_i \in \mathrm{GPVAL}_{G^{[n]}(\mathbb{Q})}$ such that $\beta_i = y_i(1)$. Apply Lemma 91 to get that $(x \in \mathbb{R} \mapsto y_i(1)) \in \mathrm{AWP}_{G^{[n]}(\mathbb{Q})}$. Now by induction, $(x \in \mathbb{R} \mapsto \beta_i) = (x \in \mathbb{R} \mapsto y_i(1)) \in \mathrm{AWP}_{\mathbb{Q}}$. Putting all those systems together, and adding variables to keep a copy of the input, it easily follows that $g \in \mathrm{AWP}_{\mathbb{Q}}$. Apply Theorem 89 to conclude that $f = h \circ g \in \mathrm{ATSP}_{\mathbb{Q}}$. ∎

We can now prove the main theorem of this section.

Theorem 93 $\mathrm{AWP}_{\mathbb{R}_G} = \mathrm{ATSP}_{\mathbb{Q}}$.

Proof. The inclusion $\mathrm{AWP}_{\mathbb{R}_G} \subseteq \mathrm{AWP}_{\mathbb{Q}}$ is trivial. Conversely, take $f \in \mathrm{AWP}_{\mathbb{R}_G}$. The system that computes f only has a finite number of coefficients, all in $\mathbb{R}_G$. Thus there exists $n \in \mathbb{N}$ such that all the coefficients belong to $G^{[n]}(\mathbb{Q})$ and then $f \in \mathrm{AWP}_{G^{[n]}(\mathbb{Q})}$. Apply Proposition 92 to conclude. ∎

As clearly $\mathrm{ALP}_{\mathbb{K}} = \mathrm{ATSP}_{\mathbb{K}}$ over any field $\mathbb{K}$ [BGP16c], it follows that $\mathrm{ALP} = \mathrm{AWP}_{\mathbb{R}_G} = \mathrm{ATSP}_{\mathbb{Q}} = \mathrm{ALP}_{\mathbb{Q}}$ and hence Definitions 3 and 11 are defining the same class. Similarly, and consequently, Definitions 1 and 67 are also defining the same class.

Acknowledgments. Daniel Graça was partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER POCTI/POCI via SQIG - Instituto de Telecomunicações through the FCT project UID/EEA/50008/2013. Olivier Bournez and Amaury Pouly were partially supported by *DGA Project CALCULS* and *French National Research Agency (ANR)* Project ANR-15-CE040-0016-01.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143.

EU disclaimer: the results presented in this paper reflect only the author's view and the EU is not responsible for any use that may be made of the information it contains.

APPENDIX

A Notations

Sets

Concept	Notation	Comment
Real interval	$[a,b]$	$\{x \in \mathbb{R} \mid a \leqslant x \leqslant b\}$
	$[a,b[$	$\{x \in \mathbb{R} \mid a \leqslant x < b\}$
	$]a,b]$	$\{x \in \mathbb{R} \mid a < x \leqslant b\}$
	$]a,b[$	$\{x \in \mathbb{R} \mid a < x < b\}$
Line segment	$[x,y]$	$\{(1-\alpha)x + \alpha y \in \mathbb{R}^n, \alpha \in [0,1]\}$
	$[x,y[$	$\{(1-\alpha)x + \alpha y \in \mathbb{R}^n, \alpha \in [0,1[\}$
	$]x,y]$	$\{(1-\alpha)x + \alpha y \in \mathbb{R}^n, \alpha \in]0,1]\}$
	$]x,y[$	$\{(1-\alpha)x + \alpha y \in \mathbb{R}^n, \alpha \in]0,1[\}$
Integer interval	$[\![a,b]\!]$	$\{a, a+1, \ldots, b\}$
Natural numbers	$\mathbb{N}$	$\{0,1,2,\ldots\}$
Integers	$\mathbb{Z}$	$\{\ldots,-2,-1,0,1,2,\ldots\}$
Rational numbers	$\mathbb{Q}$	
Dyadic rationnals	$\mathbb{D}$	$\{m2^{-n}, m \in \mathbb{Z}, n \in \mathbb{N}\}$
Real numbers	$\mathbb{R}$	
Non-negative numbers	$\mathbb{R}_+$	$\mathbb{R}_+ = [0,+\infty[$
Non-zero numbers	$\mathbb{R}^*$	$\mathbb{R}^* = \mathbb{R} \setminus \{0\}$
Positive numbers	$\mathbb{R}_+^*$	$\mathbb{R}_+^* =]0,+\infty[$
Set shifting	$x+Y$	$\{x+y, y \in Y\}$
Set addition	$X+Y$	$\{x+y, x \in X, y \in Y\}$
Matrices	$M_{n,m}(\mathbb{K})$	Set of $n \times m$ matrices over field $\mathbb{K}$
	$M_n(\mathbb{K})$	Shorthand for $M_{n,n}(\mathbb{K})$
	$M_{n,m}$	Set of $n \times m$ matrices over a field is deduced from the context
Polynomials	$\mathbb{K}[X_1,\ldots,X_n]$	Ring of polynomials with variables $X_1,\ldots,X_n$ and coefficients in $\mathbb{K}$
	$\mathbb{K}[\mathbb{A}^n]$	Polynomial functions with n variables, coefficients in $\mathbb{K}$ and domain of definition $\mathbb{A}^n$
Fractions	$\mathbb{K}(X)$	Field of rational fractions with coefficients in $\mathbb{K}$
Power set	$\mathcal{P}(X)$	The set of all subsets of X
Domain of definition	$\operatorname{dom} f$	If $f : I \to J$ then $\operatorname{dom} f = I$
Cardinal	$\#X$	Number of elements
Polynomial vector	$\mathbb{K}^n[\mathbb{A}^d]$	Polynomial in d variables with coefficients in $\mathbb{K}^n$
	$\mathbb{K}[\mathbb{A}^d]^n$	Isomorphic $\mathbb{K}^n[\mathbb{A}^d]$

Concept	Notation	Comment
Polynomial matrix	$M_{n,m}(\mathbb{K})[\mathbb{A}^n]$	Polynomial in n variables with matrix coefficients
	$M_{n,m}(\mathbb{K}[\mathbb{A}^n])$	Isomorphic $M_{n,m}(\mathbb{K})[\mathbb{A}^n]$
Smooth functions	C^k	Partial derivatives of order k exist and are continuous
	C^∞	Partial derivatives exist at all orders

Complexity classes

Concept	Notation	Comment
Polynomial Time	P	Class of decidable languages
	FP	Class of computable functions
Polytime computable numbers	$\mathbb{R}_P$	
Polytime computable real functions	$\mathrm{P}_{C[a,b]}$	Over compact interval $[a,b]$
Generable reals	$\mathbb{R}_G$	See [BGP16b]
Poly-length-computability	ALP	See Definition 3
	$\mathrm{ATSC}(\Upsilon,\Pi)$	Notation defined page 19
	$\mathrm{AOC}(\Upsilon,\Pi,\Lambda)$	Notation defined page 19
	$\mathrm{AXC}(\Upsilon,\Pi,\Lambda,\Theta)$	Notation defined page 19

Metric spaces and topology

Concept	Notation	Comment
p-norm	$\|x\|_p$	$\left(\sum_{i=1}^{n}\|x_i\|^p\right)^{\frac{1}{p}}$
Infinity norm	$\|x\|$	$\max(\|x_1\|,\dots,\|x_n\|)$

Polynomials

Concept	Notation	Comment
Univariate polynomial	$\sum_{i=0}^{d} a_i X^i$	
Multi-index	α	$(\alpha_1,\dots,\alpha_k)\in\mathbb{N}^k$
	$\|\alpha\|$	$\alpha_1+\cdots+\alpha_k$
	$\alpha!$	$\alpha_1!\alpha_2!\cdots\alpha_k!$
Multivariate polynomial	$\sum_{\|\alpha\|\leqslant d} a_\alpha X^\alpha$	where $X^\alpha = X_1^{\alpha_1}\cdots X_k^{\alpha_k}$

Concept	Notation	Comment
Degree	$\deg(P)$	Maximum degree of a monomial, X^α is of degree $\lvert\alpha\rvert$, conventionally $\deg(0) = -\infty$
	$\deg(P)$	$\max(\deg(P_i))$ if $P = (P_1, \dots, P_n)$
	$\deg(P)$	$\max(\deg(P_{ij}))$ if $P = (P_{ij})_{i\in[\![1,n]\!],j\in[\![1,m]\!]}$
Sum of coefficients	ΣP	$\Sigma P = \sum_\alpha \lvert a_\alpha\rvert$
	ΣP	$\max(\Sigma P_1, \dots, \Sigma P_n)$ if $P = (P_1, \dots, P_n)$
	ΣP	$\max(\Sigma P_{ij})$ if $P = (P_{ij})_{i\in[\![1,n]\!],j\in[\![1,m]\!]}$
A polynomial	poly	An unspecified polynomial

Miscellaneous functions

Concept	Notation	Comment
Sign function	$\operatorname{sgn}(x)$	Conventionally $\operatorname{sgn}(0) = 0$
Ceiling function	$\lceil x\rceil$	$\min\{n \in \mathbb{Z} : x \leqslant n\}$
Rounding function	$\lfloor x\rceil$	$\operatorname{argmin}_{n\in\mathbb{Z}} \lvert n - x\rvert$, undefined for $x = n + \frac{1}{2}$
Integer part function	$\operatorname{int}(x)$	$\max(0, \lfloor x\rfloor)$
	$\operatorname{int}_n(x)$	$\min(n, \operatorname{int}(x))$
Fractional part function	$\operatorname{frac}(x)$	$x - \operatorname{int} x$
	$\operatorname{frac}_n(x)$	$x - \operatorname{int}_n(x)$
Composition operator	$f \circ g$	$(f \circ g)(x) = f(g(x))$
Identity function	id	$\operatorname{id}(x) = x$
Indicator function	$\mathbb{1}_X$	$\mathbb{1}_X(x) = 1$ if $x \in X$ and $\mathbb{1}_X(x) = 0$ otherwise
n^{th} iterate	$f^{[n]}$	$f^{[0]} = \operatorname{id}$ and $f^{[n+1]} = f^{[n]} \circ f$

Calculus

Concept	Notation	Comment
Derivative	f'	
n^{th} derivative	$f^{(n)}$	$f^{(0)} = f$ and $f^{(n+1)} = {f^{(n)}}'$
Partial derivative	$\partial_i f$, $\frac{\partial f}{\partial x_i}$	with respect to the i^{th} variable
Scalar product	$x \cdot y$	$\sum_{i=1}^n x_i y_i$ in $\mathbb{R}^n$
Gradient	$\nabla f(x)$	$(\partial_1 f(x), \dots, \partial_n f(x))$
Jacobian matrix	$J_f(x)$	$(\partial_j f_i(x))_{i\in[\![1,n]\!],j\in[\![1,m]\!]}$
Taylor approximation	$T_a^n f(t)$	$\displaystyle\sum_{k=0}^{n-1} \frac{f^{(k)}(a)}{k!}(t-a)^k$
Big O notation	$f(x) = \mathcal{O}\left(g(x)\right)$	$\exists M, x_0 \in \mathbb{R}$, $\lvert f(x)\rvert \leqslant M\lvert g(x)\rvert$ for all $x \geqslant x_0$
Soft O notation	$f(x) = \tilde{\mathcal{O}}\left(g(x)\right)$	Means $f(x) = \mathcal{O}\left(g(x)\log^k g(x)\right)$ for some k

Concept	Notation	Comment
Subvector	$x_{i..j}$	$(x_i, x_{i+1}, \dots, x_j)$
Matrix transpose	M^T	
Past supremum	$\sup_\delta f(t)$	$\sup_{u\in[t,t-\delta]\cap\mathbb{R}_+} f(t)$
Partial function	$f :\subseteq X \to Y$	$\operatorname{dom} f \subseteq X$
Restriction	$f\restriction_I$	$f\restriction_I(x) = f(x)$ for all $x \in \operatorname{dom} f \cap I$

Words

Concept	Notation	Comment
Alphabet	Σ, Γ	A finite set
Words	Σ^*	$\bigcup_{n\geqslant 0} \Sigma^n$
Empty word	λ	
Letter	w_i	i^{th} letter, starting from one
Subword	$w_{i..j}$	$w_i w_{i+1} \cdots w_j$
Length	$\lvert w\rvert$	
Repetition	w^k	$\underbrace{ww\cdots w}_{k\text{ times}}$

References

[AD90] Rajeev Alur and David L. Dill. Automata for modeling real-time systems. In Mike Paterson, editor, *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, July 16-20, 1990, Proceedings*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 1990.

[BC08] Olivier Bournez and Manuel L. Campagnolo. *New Computational Paradigms. Changing Conceptions of What is Computable*, chapter A Survey on Continuous Time Computations, pages 383–423. Springer-Verlag, New York, 2008.

[BCdNM05] Olivier Bournez, Felipe Cucker, Paulin Jacobé de Naurois, and Jean-Yves Marion. Implicit complexity over an arbitrary structure: Sequential and parallel polynomial time. *Journal of Logic and Computation*, 15(1):41–58, 2005.

[BCGH06] Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, and Emmanuel 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, *Theory and Applications of Models of Computation TAMC'06*, LNCS 3959, pages 631–643. Springer-Verlag, 2006.

[BCGH07] Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, and Emmanuel Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. *J. Complexity*, 23(3):317–335, 2007.

[BCSS98] Lenore Blum, Felipe Cucker, Mike Shub, and Steve Smale. *Complexity and Real Computation.* Springer, 1998.

[BGP16a] Olivier Bournez, Daniel Graça, and Amaury Pouly. On the Functions Generated by the General Purpose Analog Computer. Technical report, January 2016. Under review for Information and Computation (current status: accepted for publicaton under minor revision).

[BGP16b] Olivier Bournez, Daniel S. Graça, and Amaury Pouly. On the functions generated by the general purpose analog computer. *CoRR*, abs/1602.00546, 2016.

[BGP16c] Olivier Bournez, Daniel Graça, and Amaury Pouly. Computing with polynomial ordinary differential equations. *Journal of Complexity*, pages –, 2016.

[BHFFS03] Asa Ben-Hur, Joshua Feinberg, Shmuel Fishman, and Hava T. Siegelmann. Probabilistic analysis of a differential equation for linear programming. *Journal of Complexity*, 19(4):474–510, 2003.

[BHSF02] Asa Ben-Hur, Hava T. Siegelmann, and Shmuel Fishman. A theory of complexity for continuous time systems. *J. Complexity*, 18(1):51–86, 2002.

[Bou97] Olivier Bournez. Some bounds on the computational power of piecewise constant derivative systems (extended abstract). In *ICALP*, pages 143–153, 1997.

[Bou99] Olivier Bournez. Achilles and the Tortoise climbing up the hyper-arithmetical hierarchy. *Theoret. Comput. Sci.*, 210(1):21–71, 1999.

[Bus31] Vannevar Bush. The differential analyzer. A new machine for solving differential equations. *J. Franklin Inst.*, 212:447–488, 1931.

[Cop98] B. Jack Copeland. Even Turing machines can compute uncomputable functions. In C.S. Calude, J. Casti, and M.J. Dinneen, editors, *Unconventional Models of Computations.* Springer-Verlag, 1998.

[Cop02] B. Jack Copeland. Accelerating Turing machines. *Minds and Machines*, 12:281–301, 2002.

[CP02] Cristian S. Calude and Boris. Pavlov. Coins, quantum measurements, and Turing's barrier. *Quantum Information Processing*, 1(1-2):107–127, April 2002.

[Dav01] Edward B. Davies. Building infinite machines. *The British Journal for the Philosophy of Science*, 52:671–682, 2001.

[Fay91] Leonid Faybusovich. Dynamical systems which solve optimization problems with linear constraints. *IMA Journal of Mathematical Control and Information*, 8:135–149, 1991.

[Fey82] Richard P. Feynman. Simulating physics with computers. *Internat. J. Theoret. Phys.*, 21(6/7):467–488, 1982.

[GBC07] Daniel S. Graça, Jorge Buescu, and Manuel L. Campagnolo. Boundedness of the domain of definition is undecidable for polynomial ODEs. In R. Dillhage, T. Grubba, A. Sorbi, K. Weihrauch, and N. Zhong, editors, *4th International Conference on Computability and Complexity in Analysis (CCA 2007)*, volume 202 of *Electron. Notes Theor. Comput. Sci.*, pages 49–57. Elsevier, 2007.

[GBC09] Daniel S. Graça, Jorge Buescu, and Manuel L. Campagnolo. Computational bounds on polynomial differential equations. *Appl. Math. Comput.*, 215(4):1375–1385, 2009.

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

[GM95] Erich Grädel and Klaus Meer. Descriptive complexity theory over the real numbers. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 315–324, Las Vegas, Nevada, 29May–1June 1995. ACM Press.

[GM02] Marco Gori and Klaus Meer. A step towards a complexity theory for analog systems. *Mathematical Logic Quarterly*, 48(Suppl. 1):45–58, 2002.

[Gra04] Daniel S. Graça. Some recent developments on Shannon's General Purpose Analog Computer. *Math. Log. Quart.*, 50(4-5):473–485, 2004.

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM, 1984.

[Kaw10] Akitoshi Kawamura. Lipschitz continuous ordinary differential equations are polynomial-space complete. *Computational Complexity*, 19(2):305–332, 2010.

[KMNY91] Masakazu Kojima, Nimrod Megiddo, Toshihito Noma, and Akiko Yoshise. *A unified approach to interior point algorithms for linear complementarity problems*, volume 538. Springer Science & Business Media, 1991.

[Ko91] Ker-I Ko. *Complexity Theory of Real Functions*. Progress in Theoretical Computer Science. Birkhaüser, Boston, 1991.

[Mac09] Bruce J MacLennan. Analog computation. In *Encyclopedia of complexity and systems science*, pages 271–294. Springer, 2009.

[MC06] Jerzy Mycka and José Felix Costa. The $p \neq np$ conjecture in the context of real and complex analysis. *J. Complexity*, 22(2):287–303, 2006.

[MM93] Norbert Müller and Bernd Moiske. Solving initial value problems in polynomial time. In *Proc. 22 JAIIO - PANEL '93, Part 2*, pages 283–293, 1993.

[Moo96] Cristopher Moore. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162(1):23–44, 5 August 1996.

[PE74] Marian B. Pour-El. Abstract computability and its relations to the general purpose analog computer. *Trans. Amer. Math. Soc.*, 199:1–28, 1974.

[PG16] Amaury Pouly and Daniel S. Graça. Computational complexity of solving polynomial differential equations over unbounded domains. *Theor. Comput. Sci.*, 626:67–82, 2016.

[Pou16] Amaury Pouly. Computational complexity of solving polynomial differential equations over unbounded domains with non-rational coefficients. *CoRR*, abs/1608.00135, 2016.

[Ruo93] Keijo Ruohonen. Undecidability of event detection for ODEs. *Journal of Information Processing and Cybernetics*, 29:101–113, 1993.

[Ruo94] Keijo Ruohonen. Event detection for ODEs and nonrecursive hierarchies. In *Proceedings of the Colloquium in Honor of Arto Salomaa. Results and Trends in Theoretical Computer Science (Graz, Austria, June 10-11, 1994)*, volume 812 of *Lecture Notes in Computer Science*, pages 358–371. Springer-Verlag, Berlin, 1994.

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

[Ulm13] Bernd Ulmann. *Analog computing*. Walter de Gruyter, 2013.

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