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

The General Purpose Analog Computer and Computable Analysis are two efficiently equivalent models of computations*

Olivier Bournez[1], Daniel S. Graça[2,3], and Amaury Pouly[1,2]

1 **Ecole Polytechnique, LIX, 91128 Palaiseau Cedex, France**
2 **CEDMES/FCT, Universidade do Algarve, C. Gambelas, 8005-139 Faro, Portugal**
3 **SQIG/Instituto de Telecomunicações, Lisbon, Portugal**

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 (time and space) elegant and simple characterization of P. We believe it is the first time such 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 model 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 at the computability and complexity level, 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 at a computability and at a computational complexity level.

1998 ACM Subject Classification F.1.1 Models of Computation. F.1.3 Complexity Measures and Classes. G.1.7 Ordinary Differential Equations

Keywords and phrases Analog Models of Computation, Continuous-Time Models of Computation, Computable Analysis, Implicit Complexity, Computational Complexity, Ordinary Differential Equations

Digital Object Identifier 10.4230/LIPIcs...

* 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.

1 Introduction

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

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, like time and space. Another approach is implicit computational complexity. The term "implicit" in "implicit computational complexity" can sometimes 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 characterization 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. that can be considered as being defined in a discrete space.

Models of computation working on a continuous space have also been considered: they include Blum Shub Smale machines [4], and in some sense Computable Analysis [40], or quantum computers [17] which usually feature discrete-time and continuous-space. Machine-independent characterizations of the corresponding complexity classes have also been devised: see e.g. [10, 24]. 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 P class. 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.

Analog Computers: Our results can also be interpreted in the context of analog models of computation and actually originate as a side effect from an attempt to understand continuous-time analog models of computation, and if they could solve some problem more efficiently than classical models. Refer to [39] for a very instructive historical account of the history of Analog computers. See also [29, 9] for other discussions.

Indeed, in 1941, Claude Shannon introduced in [38] the General Purpose Analog Computer (GPAC) model as a model for the Differential Analyzer [11], a mechanical programmable machine, on which he worked as an operator. The GPAC model was later refined in [35], [23]. 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).

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 Shannon's model if and only if it is a (component of the) solution of an ordinary differential equations (ODEs) with polynomial right-hand side [38], [23]. In this paper, we consider the refined version presented in [23].

We note that the original model of the GPAC presented in [38], [23] is not equivalent to Turing machine based models. However, the original GPAC model performs computations in real-time: at time t the output is $f(t)$, which different from the notion used by Turing machines. In [19] a new notion of computation for the GPAC, which uses "converging

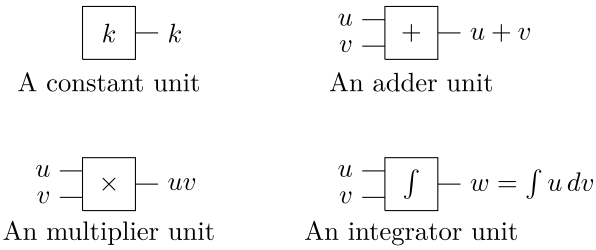
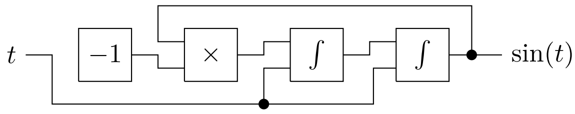


Figure 1 Circuit presentation of the GPAC: a circuit built from basic units



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

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

computations" as done by Turing machines was introduced and it was shown in [5],[6] that using this new notion of computation, the GPAC and computable analysis are two equivalent models of computation at a computability level.

In that sense, our paper extends this latter result and proves that the GPAC and computable analysis are two equivalent models of computation, both at the computability and at the complexity level. We also provide as a side effect a robust way to measure time in the GPAC, or more generally in computations performed by ordinary differential equations: basically, by considering the length of the curve.

This paper is organized as follows. Section 2 gives our main definitions and results. Section 3 discusses the related work and consequences of our results. Section 4 gives a very high-level overview of the proof. It also contains more definitions and results so that the reader can understand the big steps of the proof.

2 Our Results

We consider the following class of differential equations:

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

where $y : I \to \mathbb{R}^d$ for some interval $I \subset \mathbb{R}$ and where p is a vector of polynomials. Such systems are sometimes called PIVP, for polynomial initial value problems [21]. Observe that there is always a unique solution to the PIVP, which is analytic, defined on a maximum interval of life I containing y_0, which we refer to as "the solution".

Our crucial and key idea is that, when using PIVPs to compute a function f, the complexity should be measured as 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 infinite norm of y.

We assume the reader familiar with the notion of polynomial time computable function $f : [a, b] \to \mathbb{R}$ (see [40] for an introduction to computable analysis). We take $\mathbb{R}_+ = [0, +\infty[$ and denote by $\mathbb{R}_P$ the set of polynomial time computable reals. For any vector y, $y_{i...j}$ refers to the vector $(y_i, y_{i+1}, \dots, y_j)$. 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.

▶ Remark (The space $\mathbb{K}$ of the coefficients). In this paper, the coefficients of all considered polynomials will belong to $\mathbb{K}$. Formally, $\mathbb{K}$ needs to a be *generable field*, as introduced in [33]. However, without a significant loss of generality, the reader can consider that $\mathbb{K} = \mathbb{R}_P$ which is the set of polynomial time computable real numbers. All the reader needs to know about $\mathbb{K}$ is that it is a field and it is stable by generable functions (introduced in Section 4.2), meaning that if $\alpha \in \mathbb{K}$ and f is generable then $f(\alpha) \in \mathbb{K}$. It is shown in [33] that there exists a small generable field $\mathbb{R}_G$ lying somewhere between $\mathbb{Q}$ and $\mathbb{R}_P$, with expected strict inequality on both sides.

Our main results (the class AP is defined in Definition 3, and the notion of language recognized by a continuous system is given in Definition 4) are the following. Let us recall that $\mathrm{P}(\mathbb{R})$ is the class of polynomial time computable real functions, as defined in [27].

▶ **Theorem 1** (An implicit characterization of $\mathrm{P}(\mathbb{R})$). *Let $a, b \in \mathbb{R}_P$. A function $f : [a, b] \to \mathbb{R}$ is computable in polynomial time iff its belongs to the class* AP.

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

▶ **Definition 3** (Complexity Class AP). We say that $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ is in AP 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 $\mathbb{K}$, and a bivariate polynomial Ω 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 all $\mu \in \mathbb{R}_+$, if $\operatorname{len}_y(0, t) \geqslant \Omega(\|x\|, \mu)$ 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[1]

Intuitively, a function f belongs to AP if there is a PIVP that approximates f with a polynomial length to reach a given level of approximation.

In definition 3, the PIVP was given its input x as part of the initial condition: this is very natural because x was a real number. In the following, we will characterize languages with differential equations. Since a language is made up of words, we need to discuss how to represent (encode) a word with a real number. We fix a finite alphabet $\Gamma = \{0, .., k-2\}$ and define the encoding[2] $\psi(w) = \left(\sum_{i=1}^{|w|} w_i k^{-i}, |w|\right)$ for a word $w = w_1 w_2 \dots w_{|w|}$.

▶ **Definition 4** (Analog recognizability). A language $\mathcal{L} \subseteq \Gamma^*$ is called *analog-recognizable* if there exists a vector q of bivariate polynomials and a vector p of polynomials with d variables,

[1] This is a technical condition required for the proof. This can be weakened, for example to $\left\|y'(t)\right\| = \|p(y(t))\| \geqslant \frac{1}{\operatorname{poly}(t)}$. 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 "unatural" cases. This is satisfied by all examples of computations we know [39].

[2] 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) $\|\psi(w)\| \approx \operatorname{poly}(|w|)$ in other words, the norm of the encoding is roughly the size of the word.

both with coefficients in $\mathbb{K}$, and a polynomial $\Omega : \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $w \in \Gamma^*$ there is a (unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ such that for all $t \in \mathbb{R}_+$:

- $y(0) = q(\psi(w))$ 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 $u \geqslant t$ ▶ the decision is stable
- if $w \in \mathcal{L}$ (resp. $\notin \mathcal{L}$) and $\operatorname{len}_y(0,t) \geqslant \Omega(|w|)$ then $y_1(t) \geqslant 1$ (resp. $\leqslant -1$) ▶ decision
- $\operatorname{len}_y(0,t) \geqslant t$ ▶ technical condition

Intuitively this definition says that a language is 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 portion of *polynomial length* 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 will go: the value of y_1 determines if the word has been accepted or not, or if the computation is still in progress.

3 Discussion

Extensions: Our characterizations of the polynomial time can easily be extended to characterizations of deterministic complexity classes above polynomial time. For example, EXPTIME can be shown to correspond to the case where polynomial Ω is replaced by some exponential function(see Appendix C.1).

▶ **Theorem 5.** *Let a and b in $\mathbb{R}_P$. A function $f : [a,b] \to \mathbb{R}$ is computable in exponential time iff its belongs to the class $f \in$ AEXP.*

▶ **Definition 6** (Definition of the complexity class AEXP for continuous systems)**.** We say that $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ is in AEXP if and only if there exists a vector p of polynomial functions with d variables, a vector q of polynomial with n variables, both with coefficients in $\mathbb{K}$, an exponential function $\Omega : \mathbb{R}^2_+ \to \mathbb{R}_+$ 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 $y'(t) = p(y(t))$ for all $t \geqslant 0$ ▶ y satisfies a PIVP
- for any $\mu \in \mathbb{R}_+$, if $\operatorname{len}_y(0,t) \geqslant \Omega(\|x\|, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ ▶ $y_{1..m}$ converges
- $\|y'(t)\| \geqslant 1$ ▶ technical condition: The length grows at least linearly with time[3]

Applications to computational complexity: We believe these characterizations to really 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(see Appendix C.2), 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- (or EXPTIME-)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

[3] 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 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 "unatural" cases.

method do), but that for general (and even right-hand side polynomial) ODEs, no better method can work, unless some famous complexity questions do not hold. 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. [31]) 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.

Applications to continuous-time analog models: PIVPs are known to correspond to functions that can be generated by the GPAC of Claude Shannon [38].

Defining a robust (time) complexity notion for continuous time systems is a well known open problem [9] 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 measure of "time complexity" is problematic, since time can be arbitrarily contracted in a continuous system due to the "Zeno phenomena" (e.g. by using functions like arctan which contract the whole real line into a bounded set). It follows that all computable languages can then be computed by a continuous system in time $O(1)$ (see e.g. [36], [37], [30], [7], [8], [1], [12], [15], [13], [14]).

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. Notice that this is not sufficient to get robustness: the fact that we restrict to PIVP systems is crucial because more general ODEs are usually hard to simulate (e.g. see [26]). This explains why all previous attempts of a general complexity for general sytems failed in some sense [9]. Super-Turing "Zeno phenomena" can still happen with general ODEs, but not with PIVPs.

Applications to algorithms: We also 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 continuous but also 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 covers many interior-point methods or barrier methods where the problem can be transformed into the optimization of some continuous function (see e.g. [25, 16, 3, 28]).

Related work is mainly discussed section by section, with sometimes more details provided in Appendix B. We believe 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, such a result strengthens all existings 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 with a 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 continous-time systems include [18]. However, the theory developped 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 [2], also focussed 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 [3] for solving linear programming problems.

Both approaches are not at all intended to cover generic ODEs, and none of them is able to relate the obtained classes to classical classes from computational complexity.

Up to our knowledge, the most up to date survey about continuous time computation are [9, 29].

Relating computational complexity problems (like the P vs NP question) to problems of analysis has already been the motivation of series of works. In particular, Félix Costa and Jerzy Mycka have a series of work (see e.g. [32]) 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 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 $DAnalog$ e $NAnalog$ (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 operator (i.e. we only need to solve one ordinary differential equations with polynomial right-hand side).

We also mention that Friedman and Ko (see [27]) 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 (see also the Appendix B for related work). The complexity of solving Lipschitz continuous ordinary differential equation has been proved to be polynomial-space complete by Kawamura [26].

All the results of this paper are fully developped in the PhD thesis of Amaury Pouly [33]. For self-completeness, most proofs are in appendix. Please refer to [33] for missing details. Results mentioned in this paper have not yet been published, and are currently not submitted[4], with the exception of results on ODE solving (results of section 4.1) but in a slightly different and extended framework), which were very recently accepted [34].

4 Overview of the proof

To show our main results (Theorem 1 and Theorem 2), we need to show two implications: (i) if a function $f : [a, b] \to \mathbb{R}$ (resp. a language $\mathcal{L}$) is polynomial time computable, then it belongs to AP (resp. it is analog-recognizable) and (ii) if a function $f : [a, b] \to \mathbb{R}$ belongs to AP (resp. a language $\mathcal{L}$ is analog-recognizable) then it is polynomial time computable (resp. belongs to P).

The second implication (ii) is proved by computing the solution of a PIVP system using some numerical algorithm. If a function $f : [a, b] \to \mathbb{R}$ in AP can be computed (up to some given accuracy) by following the solution curve of its associated ODE up to a reasonable (polynomial) amount of the length of the curve, the numerical simulation of its associated ODE will use a reasonable (polynomial) amount of resources to simulate this bounded portion of the solution curve. Hence the function f will be computed (up to some given accuracy, as usual in Computable Analysis) by a Turing machine in polynomial time. A similar idea can be used for showing the implication (ii) for P and analog-recognizable languages.

The idea sketched above gives the intuition of the proof but the usual ODE solving algorithms cannot be used here since (1) they are only guaranteed to compute the solution of an ODE with a given accuracy over a bounded time domain, but here we need to compute

[4] Preliminary results were submitted in the past but not with the strength of the current statements.

this solution over an unbounded time domain[5] which introduce further complications and (2) we need polynomial complexity in the length of the curve, which is not a classical measure of complexity.

The first implication (i) is proved by simulating Turing machines with PIVPs and by showing that these simulations can be performed by using a reasonable (polynomial) amount of resources (length of the solution curve) if the Turing machine runs in polynomial time.

Some simulation of Turing machines with PIVPs was already performed e.g. in [6], [22]. Basically one has to simulate the behavior of a Turing machine with a continuous system. This is problematic since Turing machines behave discretely (e.g. "if x happens then do A, otherwise do B") and one only has access to continuous (analytic) functions. This can be solved by *approximating* discontinuous functions with continuous functions to obtain an approximation of the transition function of the Turing machine. Then, by using special techniques, one can iterate the new (now continuous) transition function to simulate the step-by-step evolution of the Turing machine. Here we have one new difficult problem to tackle (not covered in previous papers like [6] and [22]) because we must ensure that everything can be done using only a reasonable (polynomial) amount of the length of the solution curve of the PIVP. In particular, this constraint rules out particularly simple techniques like integer encodings of the tape and error correction, as used in the previously mentioned papers.

At a high level, our proof relies on considerations about (polynomial length) *ODE programming*: we prove that it is possible to "program" with polynomial length ODE systems that keep some variable fixed, do assignement, iterate some functions, compute limits, etc. We use those basic operations and basic functions with PIVPs (e.g. $\min, \max$, continuous approximation of rounding, etc.) to create more complex functions and operations that simulate the transition function of a given Turing machine and its iterations. To be sure that the more complex functions still satisfy all the properties we want (e.g. that they belong to AP), we prove several closure properties: in particular, we prove very strong and elegant equivalent definitions of class AP.

For reasons of lack of space, we do not detail all these operators and functions, but we sketch the proof of a few properties and some key ideas of our techniques. We use the following notation: when p is a polynomial, Σp is the sum of the absolute values of its coefficients and $\deg(p)$ its degree. If p is a vector of polynomials, we extend those notions by taking the maximum for each component.

4.1 Polytime analog computability implies polytime computability

We start by sketching the proof of the "only if" direction of Theorem 2, and then of Theorem 1. Recall that a real function is polynomial time computable if given arbitrary approximations of the input, we can produce arbitrary approximations of the output in polynomial time. As it is customary, we proceed in two steps. We first show that the function has a polynomial modulus of continuity. This allows us to restrict the problem to rational inputs of controlled size.

▶ **Theorem 7** (Modulus of continuity, Appendix E.2). *If $f \in \mathrm{AP}$, 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 $\mu \in \mathbb{R}_+$:*

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

---

[5] Note that while f has domain of definition $[a, b]$, from Definition 3 f is approximated by a PIVP whose solution is defined over the unbounded time domain $\mathbb{R}$

We then show that the solution of a such a PIVP can be approximated in polynomial time. For this, will need the following theorem to get the complexity of numerically solving this PIVP. The idea of the proof is detailled below.

▶ **Theorem 8** (Complexity of Solving PIVP[34]). *If* [6] $y : \mathbb{R} \to \mathbb{R}^d$ *satisfies for all* $t \geqslant 0$.

$$y(0) = y_0 \qquad y'(t) = p(y(t)). \tag{2}$$

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

$$\operatorname{poly}(\deg(p), \operatorname{len}_y(0,t), \log\|y_0\|, \log \Sigma p, \mu)^d. \tag{3}$$

More precisely, there exists a Turing machine $\mathcal{M}$ *such that for any oracle* $\mathcal{O}$ *representing*[7] (y_0, p, t) *and any* $\mu \in \mathbb{N}$, $\left\|\mathcal{M}^{\mathcal{O}}(\mu) - \operatorname{PIVP}(y_0, p, t)\right\| \leqslant 2^{-\mu}$ *if* $y(t)$ *exists, and the number of steps of the machine is bounded by* (3) *for all such oracles.*

General Idea: Assume that $\mathcal{L}$ is analog-recognizable in the sense of Definition 4, using corresponding notations d, q, p, Ω. Let $w \in \Gamma^*$ and consider the following system: $y(0) = q(\psi(w))$, $y'(t) = p(y(t))$. We show that we can decide in time polynomial in $|w|$ whether $w \in \mathcal{L}$ or not. Theorem 8 can be used to conclude that we can compute $y(t) \pm e^{-\mu}$ in time polynomial in $\log\|q(\psi(w))\|, \mu$ and $\operatorname{len}_y(0,t)$. Recall that $\|\psi(w)\| = |w|$ and that the system is guaranteed to give an answer as soon as $\operatorname{len}_y(0,t) \geqslant \Omega(|w|)$. This means that it is enough to compute $y(t^*)$, where t^* satisfies $\operatorname{len}_y(0,t^*) \geqslant \Omega(|w|)$, with precision $1/2$ to distinguish between $y_1(t) \geqslant 1$ and $y_1(t) \leqslant -1$. Since $\operatorname{len}_y(0,t) \geqslant t$, thanks to the technical condition of the definition, we know that we can find a $t^* \leqslant \Omega(|w|)$. Note that $\operatorname{len}_y(0, \Omega(|w|))$ might not be polynomial in $|w|$ so we cannot simply compute $y(\Omega(|w|))$.

Fortunately, the proof of Theorem 8 provides us with an algorithm that solves the PIVP by making small time steps, and at each step the length cannot increase by more than a constant. This means that we can run algorithm to compute $y(\Omega(|w|))$ and stop it as soon as the length is greater than $\Omega(|w|)$. Let t^* be the time at which the algorithm stops. Then the running time of the algorithm will be polynomial in t^*, μ and $\operatorname{len}_y(0,t^*) \leqslant \Omega(|w|) + \mathcal{O}(1)$. Finally, thanks to the technical condition, $t^* \leqslant \operatorname{len}_y(0,t^*)$, this algorithm has running time polynomial in $|w|$.

The proof of Theorem 1 (Appendix E.1) is established using the same principle based on Theorem 8, observing in addition that functions in AP can easily be approximated by considering only their value on rationals, since they have a polynomial modulus of continuity, as shown by the following theorem.

It thus appears that the true remaining difficulty lies in proving Theorem 8. An important point is that none of the classical methods for solving ordinary differential equations are polynomial time over unbounded time domains. Indeed, no method of fixed order r is polynomial in variable t over the whole domain $\mathbb{R}$.[8] For more information, we refer the reader to [34].

▶ Remark. Observe that the solution of the following PIVP $y_1' = y_1, y_2' = y_1y_2, y_3' = y_2y_3, \dots, y_n' = y_{n-1}y_n$ is a tower of n exponentials. Its solution can be computed in polynomial time over any fixed compact $[a,b]$ [31]. However, the solution cannot be computed

[6] The existence of a solution y up to a given time is undecidable [20] so we have to assume existence.
[7] See [27] for more details. In short, the machine can ask arbitrary approximation of y_0, p and t to the oracle. The polynomial is represented by the finite list of coefficients.
[8] This is why most studies restricts to a compact domain.

in polynomial time over $\mathbb{R}$, as just writing this value in binary cannot ever been done in polynomial time. Hence, the solution of a PIVP cannot be computed in polynomial time, over $\mathbb{R}$, in the general case. A key feature of our method is that we are searching methods polynomial in the length of the curve, which is not a classical framework.

4.2 Polytime computability implies polytime analog computability

The idea of the proof of the "if" directions is to simulate a Turing machine using a PIVP. But this is far from trivial since we need to do it with a polynomial length.

About generable functions: The following concept can be attributed to [38]: 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 first component of the vector y defined in $\mathbb{R}^d$. We need in our proof to extend the concept to talk about (i) multivariable functions and (ii) the growth of these functions. The following class and closure properties can be seen as extensions of results from [21].

▶ **Definition 9** (Polynomially bounded generable function). Let $d, e \in \mathbb{N}$, I be an open and connected subset of $\mathbb{R}^d$ and $f : I \to \mathbb{R}^e$. We say that $f \in \text{GPVAL}$ 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 $\mathbb{K}$, $x_0 \in \mathbb{K}^d$, $y_0 \in \mathbb{K}^n$ 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[9]
- $f(x) = y_{1..e}(x)$ ▶ f is a component of y
- $\|y(x)\| \leqslant \mathtt{sp}(\|x\|)$ ▶ y is polynomially bounded

▶ **Lemma 10** (Closure properties of GPVAL, Appendix F.1). *Let $f :\subseteq \mathbb{R}^d \to \mathbb{R}^n \in \text{GPVAL}$ and $g :\subseteq \mathbb{R}^e \to \mathbb{R}^m \in \text{GPVAL}$. Then $f + g$, $f - g$, fg and $f \circ g$ are in GPVAL.*

▶ **Lemma 11** (Generable functions are closed under ODE, Appendix F.2). *Let $d \in \mathbb{N}$, $J \subseteq \mathbb{R}$ an interval, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^d$ in GPVAL, $t_0 \in \mathbb{K} \cap J$ and $y_0 \in \mathbb{K}^d \cap \operatorname{dom} f$. Assume there exists $y : J \to \operatorname{dom} f$, and a polynomial $\overline{\mathtt{sp}} : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying for all $t \in J$:*

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

Then $y \in \text{GPVAL}$ and it is unique.

It follows that many polynomially bounded usual analytic[10] functions are in the class GPVAL. The inclusion GPVAL $\subset$ AP holds for functions whose domain is simple enough[11] (Appendix F.3). However, the inclusion GPVAL $\subset$ AP is strict[12], since functions like the inverse of the Gamma function $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ or Riemann's Zeta function $\zeta(x) = \sum_{k=0}^\infty \frac{1}{k^x}$ are not differentially algebraic [38] but belong to AP.

Robustness of AP: A very strong key argument of our proof is that the notion of computability given by Definition 3 is actually very robust and can be stated in many equivalent ways. A key point is that the definition can be weakened and strengthened. The following theorem shows that we weaken the definition without changing the class. Since it might not be obvious to the reader, we emphasize that this notion is a priori weaker (thus AP is a priori larger than AWP). Indeed, (i) the system accepts errors in the input (ii) the system

[9] J_y denotes the Jacobian matrix of y.
[10] Functions from GPVAL are necessarily analytic, as solutions of an analytic ODE are analytic.
[11] For example star domains with a rational vantage point.
[12] Even with functions with star domains with a vantage point.

does not even converge, but merely approximates the output, doing the best it can given the input error.

▶ **Theorem 12** (Weak Computability). AP = AWP *where* AWP *corresponds to the class of functions* $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ *such that there are some polynomials* $\Omega : \mathbb{R}_+^2 \to \mathbb{R}_+$ *and* $\Upsilon : \mathbb{R}_+^3 \to \mathbb{R}_+$*,* $d \in \mathbb{N}$*,* $p, q \in \mathrm{GPVAL}$*, 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*
- *if* $t \geqslant \Omega(\|x\|, \mu)$ *then* $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ ▶ *$y_{1..m}$ approximates $f(x)$ within $e^{-\mu}$*
- $\|y(t)\| \leqslant \Upsilon(\|x\|, \mu, t)$ ▶ *$y(t)$ is polynomially bounded*

The proof of Theorem 12, however, is quite involved: first p and q can be equivalently assumed to be polynomials instead of functions in GPVAL above, from Lemma 11. Then AP $\subset$ AWP, follows from the fact that this is possible to rescale the system using the length of the curve as a new variable to make sure it does not grow faster than a polynomial time, we get what is needed (Appendix F.4). The other direction (AWP $\subset$ AP) is really harder: the first step is to transform a computation into a computation that tolerates small perturbations of the dynamics (AWP $\subset$ ARP, Appendix F.5). The second problem is to avoid that the system explodes for inputs not in the domain of the function, or for too big perturbation of the dynamics perturbations on inputs (ARP $\subset$ ASP, Appendix F.6). As a third step, we allow the system to have its inputs (input and precision) changed during the computation and the system has a maximum delay to react to these changes (ASP $\subset$ AXP, Appendix F.7). Finally, as a fourth step, we add a mechanism that feeds the system with the input and some precision. By continuously increasing the precision with time, we ensure that the system will converge when the input is stable. The result of these 4 steps is the following lemma, yielding a nice notion of online-computation (AXP $\subset$ AOP, Appendix F.8). Equality AP = AWP = AOP follows because time and length are related for polynomially bounded systems. The notion of online computability is an example of a priori strengthening of our notion of computation; yet it still corresponds to the same class of function. Intuitively, a function is online computable if, on any (long enough) time interval where the input is almost constant, the system converges (after some delay) the output of the function. Of course, the output will have some error that is related to the input error (due to the input not being exactly constant).

▶ **Lemma 13** (Online computability). AWP $\subset$ AOP*, where* AOP *corresponds to the class of functions* $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ *such that for polynomials* $\Upsilon, \Omega, \Lambda : \mathbb{R}_+^2 \to \mathbb{R}_+$*, there exists* $\delta \geqslant 0$*,* $d \in \mathbb{N}$ *and* $p \in \mathbb{K}^d[\mathbb{R}^d \times \mathbb{R}^n]$ *and* $y_0 \in \mathbb{K}^d$ *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* $y'(t) = p(y(t), x(t))$
- $\|y(t)\| \leqslant \Upsilon\left(\sup_{u\in[t-\delta,t]} \|x(u)\|, t\right)$
- *For any* $I = [a, b]$*, if there exists* $\bar{x} \in \operatorname{dom} f$ *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* $a + \Omega(\|\bar{x}\|, \bar{\mu}) \leqslant u \leqslant b$*.*

ODE Programming: With the closure properties of AP, programming with (polynomial length) ODE becomes a rather pleasant exercise, once the logic is understood. For example, simulating the assignement $y := g_\infty$ corresponds to dynamics $y(0) = y_0$, $y'(t) = \mathrm{reach}(\phi(t), y(t), g(t)) + E(t)$, for a fixed function reach $\in$ GPVAL, tolerating bounded error $E(t)$ on dynamics, and g fluctuating around g_∞ (Lemma 46, Appendix F.8). Other example: from a AP system computing f, just adding the corresponding AOP-equations for g, yields a PIVP computing $g \circ f$(Lemma 55, Appendix F.9), by feeding output of the system computing f to the (online) input of g.

Turing machines: Consider a Turing machine $\mathcal{M} = (Q, \Sigma, b, \delta, q_0, q_\infty)$. A (instantaneous) configuration of M can be seen as 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. Let $\mathcal{C}_\mathcal{M}$ be the set of configurations of $\mathcal{M}$, and $\mathcal{M}$ denotes the function mapping a configuration to its next configuration. In order to simulate a machine, we encode configurations with real numbers as follows. Recall that $\Gamma = \{0, 1, \dots, k-2\}$ and let $\langle c \rangle = (0.x, \sigma, 0.y, q) \in \mathbb{Q} \times \Sigma \times \mathbb{Q} \times Q$ where $0.x = x_1 k^{-1} + x_2 k^{-2} + \cdots + x_{|x|} k^{-|x|} \in \mathbb{Q}$ with $x = x_1 x_2 \dots x_{|x|}$.

▶ **Theorem 14** (Robust Real Step, Appendix F.9)**.** *For any machine $\mathcal{M}$, there is some function $\langle \mathcal{M} \rangle \in \mathrm{AP}$ such that for all $c \in \mathcal{C}_\mathcal{M}$, $\mu \in \mathbb{R}_+$ and $\bar{c} \in \mathbb{R}^4$, if $\|\langle c \rangle - \bar{c}\| \leqslant \frac{1}{2k^2} - e^{-\mu}$ then $\|\langle \mathcal{M} \rangle (\bar{c}, \mu) - \langle \mathcal{M}(c) \rangle\| \leqslant k \|\langle c \rangle - \bar{c}\|$.*

The difficulty of the proof is that one step of Turing machine with our encoding naturally involves computing the integer and fractional parts of a number. These operations are discontinuous and thus cannot be done in AP in full generality. This is solved by proving that a continuous and good enough "fractional part" like-function is in AP (and avoids constructions from [21]).

Iterating Functions: A key point for proving the main result is to show that it is possible to iterate a function using a PIVP under some specific hypotheses. The proof consists in building by ODE programming an ordinary differential equation using three variables y, z and w updating in a cycle to be repeated n times. At all time, y is an online component of the 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. This computes all the iterates $f(x), f^{[2]}(x), \dots$. The crucial point of this process is the error estimation, to guarantee that the system does not diverge, while keeping polynomial length. One of the key assumption to ensure this is for f to admit a specific kind of modulus of continuity. The other key assumption is an effective "openness" of the iteration domain.

▶ **Theorem 15** (Closure by iteration, Appendix F.10)**.** *Let $I \subseteq \mathbb{R}^m$, $(f : I \to \mathbb{R}^m) \in \mathrm{AP}$, $\eta \in [0, 1/2[$ 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 $f(I_{n+1}) \subseteq I_n$*
- *for all $x \in I_n$, $\left\|f^{[n]}(x)\right\| \leqslant \Pi(\|x\|, n)$*
- *for all $x \in I_n$, $y \in \mathbb{R}^m, \mu \in \mathbb{R}_+$, 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 $f_\eta^* \in \mathrm{AP}$.

The iteration of the (transition) functions given by Theorem 14 leads to a way to emulate any function computable in polynomial time.

At a high level, the "if" direction of Theorem 2 then follows. Indeed (Appendix F.11), decidability can be seen as the computability of some particular function with boolean output.

For the "if" direction of Theorem 1 (Appendix F.10), there are further nontrivial obstacles to overcome. Given $x \in [a, b]$ and $\mu \in \mathbb{N}$, we want to compute an approximation of $f(x) \pm 2^{-\mu}$ and take the limit when $\mu \to \infty$. To compute f, we will use a polynomial time computable function g that computes f over rationals, and m a modulus of continuity. All we have to do is simulate g with input $\tilde{x}$ and μ, where $\tilde{x} = x \pm 2^{-m(\mu)}$ because we can only feed the machine with a finite input of course. The remaining nontrivial part of the proof is how to obtain the encoding of $\tilde{x}$ from x and μ. Indeed, the encoding is a discrete quantity whereas x is real number, so by a simple continuity argument, one can see that no such function can exist. The trick is the following: from x and μ, we can compute two encodings ψ_1 and ψ_2 such that at least one of them is valid, and we know which one it is. So we are going to simulate g on both inputs and then select the result. Again, the select operation cannot be done continuously unless we agree to "mix" both results, i.e. we will compute $\alpha g(\psi_1) + (1-\alpha)g(\psi_2)$. The trick is to ensure that $\alpha = 1$ or 0 when only one encoding is valid, $\alpha \in]0,1[$ when both are valid (by "when" we mean with respect to x). This way, a mixing of both will ensure continuity but in fact when both encodings are valid, the outputs are nearly the same so we are still computing f. Obtaining such encodings ψ_1 and ψ_2 is also nontrivial and requires more uses of the closure by iteration property.

References

1. 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.
2. A. Ben-Hur, H. T. Siegelmann, and S. Fishman. A theory of complexity for continuous time systems. *J. Complexity*, 18(1):51–86, 2002.
3. 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.
4. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer, 1998.
5. O. Bournez, M. L. Campagnolo, D. S. Graça, and E. Hainry. The General Purpose Analog Computer and Computable Analysis are two equivalent paradigms of analog computation. In J.-Y. Cai, S. B. Cooper, and A. Li, editors, *Theory and Applications of Models of Computation TAMC'06*, LNCS 3959, pages 631–643. Springer-Verlag, 2006.
6. O. Bournez, M. L. Campagnolo, D. S. Graça, and E. Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. *J. Complexity*, 23(3):317–335, 2007.
7. Olivier Bournez. Some bounds on the computational power of piecewise constant derivative systems (extended abstract). In *ICALP*, pages 143–153, 1997.
8. Olivier Bournez. Achilles and the Tortoise climbing up the hyper-arithmetical hierarchy. *Theoret. Comput. Sci.*, 210(1):21–71, 1999.
9. 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.
10. 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.
11. V. Bush. The differential analyzer. A new machine for solving differential equations. *J. Franklin Inst.*, 212:447–488, 1931.
12. C. S. Calude and B. Pavlov. Coins, quantum measurements, and Turing's barrier. *Quantum Information Processing*, 1(1-2):107–127, April 2002.
13. 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.
14. B. Jack Copeland. Accelerating Turing machines. *Minds and Machines*, 12:281–301, 2002.

15 E. B. Davies. Building infinite machines. *The British Journal for the Philosophy of Science*, 52:671–682, 2001.
16 Leonid Faybusovich. Dynamical systems which solve optimization problems with linear constraints. *IMA Journal of Mathematical Control and Information*, 8:135–149, 1991.
17 R. P. Feynman. Simulating physics with computers. *Internat. J. Theoret. Phys.*, 21(6/7):467–488, 1982.
18 Marco Gori and Klaus Meer. A step towards a complexity theory for analog systems. *Mathematical Logic Quarterly*, 48(Suppl. 1):45–58, 2002.
19 D. S. Graça. Some recent developments on Shannon's General Purpose Analog Computer. *Math. Log. Quart.*, 50(4-5):473–485, 2004.
20 D. S. Graça, J. Buescu, and M. 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.
21 D. S. Graça, J. Buescu, and M. L. Campagnolo. Computational bounds on polynomial differential equations. *Appl. Math. Comput.*, 215(4):1375–1385, 2009.
22 D. S. Graça, M. L. Campagnolo, and J. Buescu. Computability with polynomial differential equations. *Adv. Appl. Math.*, 40(3):330–349, 2008.
23 Daniel S. Graça and José Félix Costa. Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5):644–664, 2003.
24 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.
25 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.
26 A. Kawamura. Lipschitz continuous ordinary differential equations are polynomial-space complete. *Computational Complexity*, 19(2):305–332, 2010.
27 Ker-I Ko. *Complexity Theory of Real Functions.* Progress in Theoretical Computer Science. Birkhaüser, Boston, 1991.
28 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.
29 Bruce J MacLennan. Analog computation. In *Encyclopedia of complexity and systems science*, pages 271–294. Springer, 2009.
30 Cristopher Moore. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162(1):23–44, 5 August 1996.
31 N. Müller and B. Moiske. Solving initial value problems in polynomial time. In *Proc. 22 JAIIO - PANEL '93, Part 2*, pages 283–293, 1993.
32 J. Mycka and J. F. Costa. The $p \neq np$ conjecture in the context of real and complex analysis. *J. Complexity*, 22(2):287–303, 2006.
33 Amaury Pouly. *Continuous models of computation: from computability to complexity.* PhD thesis, Ecole Polytechnique and Unidersidade Do Algarve, Defended on July 6, 2015. 2015. https://pastel.archives-ouvertes.fr/tel-01223284.
34 Amaury Pouly and Daniel S. Graça. Computational complexity of solving polynomial differential equations over unbounded domains. *Theor. Comput. Sci.*, 626:67–82, 2016.
35 M. B. Pour-El. Abstract computability and its relations to the general purpose analog computer. *Trans. Amer. Math. Soc.*, 199:1–28, 1974.

36 Keijo Ruohonen. Undecidability of event detection for ODEs. *Journal of Information Processing and Cybernetics*, 29:101–113, 1993.

37 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.

38 C. E. Shannon. Mathematical theory of the differential analyser. *Journal of Mathematics and Physics MIT*, 20:337–354, 1941.

39 Bernd Ulmann. *Analog computing.* Walter de Gruyter, 2013.

40 K. Weihrauch. *Computable Analysis: an Introduction.* Springer, 2000.

A Table of Contents

Contents

B Complements on Related Works

Attempts to derive a complexity theory for continous-time systems include [18]: However, the theory developped 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 [2], also focussed 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 same authors to study a particular class of flow dynamics [3] for solving linear programming problems.

Both approaches are not at all intended to cover generic ODEs, and none of them is able to relate the obtained classes to classical classes from computational complexity.

Up to our knowledge, the most up to date survey about continuous time computation is [9].

Relating computational complexity problems (like the P vs NP question) to problems of analysis has already been the motivation of series of works: In particular, Felix Costa and Jerzy Mycka have a series of work (see e.g. [32]) relating the P vs NP question to questions in the context of real and complex analysis.

We give some arguments here, in case this is needed, to state that their approach is very different: they do so at the price of 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 $DAnalog$ e $NAnalog$ (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 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 [23], but this happens only in very specific cases, which do not seem to include the classes $DAnalog$ e $NAnalog$. 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.

C Some Formal Statements About Facts Mentioned in the Discussion

C.1 A Characterization of EXPTIME

▶ **Theorem 16.** *Let a and b in $\mathbb{R}_P$. A function $f : [a, b] \to \mathbb{R}$ is computable in exponential time iff its belongs to the class $f \in \mathrm{AEXP}$.*

▶ **Theorem 17** (An implicit characterization of P). *Let $\mathcal{L}$ be any decision problem (language). $\mathcal{L} \in \mathrm{P}$ if and only if $\mathcal{L}$ is exponential-length analog-recognizable.*

▶ **Definition 18** (Definition of the complexity class AEXP for continuous systems). We say that $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ is in AEXP if and only if there exists a vector p of polynomial functions with d variables, a vector q of polynomial with n variables, both with coefficients in $\mathbb{K}$, an exponential function $\Omega : \mathbb{R}_+^2 \to \mathbb{R}_+$ 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 $y'(t) = p(y(t))$ for all $t \geqslant 0$ ▶ y satisfies a PIVP
- for any $\mu \in \mathbb{R}_+$, if $\operatorname{len}_y(0,t) \geqslant \Omega(\|x\|, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ ▶ $y_{1..m}$ converges
- $\|y'(t)\| \geqslant 1$ ▶ technical condition: The length grows at least linearly with time[13]

▶ **Definition 19** (Discrete recognizability). A language $\mathcal{L} \subseteq \Gamma^*$ is called *exponential-length analog-recognizable* if there exists a vector q of polynomials with two variables, a vector p of polynomials with d variables, both with coefficients in $\mathbb{R}_P$, and an exponential function $\Omega : \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $w \in \Gamma^*$ there is a (unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ such that for all $t \in \mathbb{R}_+$:

- $y(0) = q(\psi(w))$ 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 $u \geqslant t$ ▶ the decision is stable
- if $w \in \mathcal{L}$ (resp. $\notin \mathcal{L}$) and $\operatorname{len}_y(0,t) \geqslant \Omega(|w|)$ then $y_1(t) \geqslant 1$ (resp. $\leqslant -1$) ▶ decision
- $\operatorname{len}_y(0,t) \geqslant t$ ▶ technical condition[14]

C.2 A (Too simple) Characterization of NP

Following the discussion page 5, here is a trivial way to get a characterization of NP.

▶ **Definition 20** (Discrete NP-recognizability). A language $\mathcal{L} \subseteq \Gamma^*$ is called NP-*analog-recognizable* if there exists a vector q of polynomials in with two variables, a vector p of polynomials with d variables, both with coefficients in $\mathbb{R}_P$, and a polynomial $\Omega : \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $w \in \Gamma^*$, for some $s \in \Gamma^*$ of size polynomial in $|w|$, there is a (unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ such that for all $t \in \mathbb{R}_+$,

- $y(0) = q(\psi(< w, s >))$ 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 $u \geqslant t$ ▶ the decision is stable
- if $w \in \mathcal{L}$ (resp. $\notin \mathcal{L}$) and $\operatorname{len}_y(0,t) \geqslant \Omega(|w|)$ then $y_1(t) \geqslant 1$ (resp. $\leqslant -1$) ▶ decision
- $\operatorname{len}_y(0,t) \geqslant t$ ▶ technical condition[15]

▶ **Theorem 21** (An implicit characterization of NP). *Let $\mathcal{L}$ be any decision problem (language). $\mathcal{L} \in$ NP if and only if $\mathcal{L}$ is NP-analog-recognizable.*

Following the discussion page 5, the purpose would be to get something more "natural", not involving logic like quantifiers (for e.g. a concept of analysis like ordinary differential inclusions instead of ordinary differential equations).

D Notations

Notations for sets

Concept	Notation	Comment
Real interval	$[a, b]$	$\{x \in \mathbb{R} \mid a \leqslant x \leqslant b\}$
	$[a, b[$	$\{x \in \mathbb{R} \mid a \leqslant x < b\}$

[13] 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 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 "unatural" cases.

[14] Same remarks as above.

[15] Same remarks as above.

Notations for sets

Concept	Notation	Comment
	$]a, b]$	$\{x \in \mathbb{R} \mid a < x \leqslant b\}$
	$]a, b[$	$\{x \in \mathbb{R} \mid a < x < b\}$
Line segment	$[x, y]$	$\{(1-\alpha)x + \alpha y, \alpha \in [0, 1]\}$
	$[x, y[$	$\{(1-\alpha)x + \alpha y, \alpha \in [0, 1[\}$
	$]x, y]$	$\{(1-\alpha)x + \alpha y, \alpha \in]0, 1]\}$
	$]x, y[$	$\{(1-\alpha)x + \alpha y, \alpha \in]0, 1[\}$
Integer interval	$[\![a, b]\!]$	$\{a, a+1, \ldots, b\}$
Natural numbers	$\mathbb{N}$	$\{0, 1, 2, \ldots\}$
	$\mathbb{N}^*$	$\mathbb{N} \setminus \{0\}$
Integers	$\mathbb{Z}$	$\{\ldots, -2, -1, 0, 1, 2, \ldots\}$
Rational numbers	$\mathbb{Q}$	
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 $\mathbb{K}^n[\mathbb{A}^d]$
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

Complexity classes

Concept	Notation	Comment
	FP	Class of computable functions
Polynomial time computable numbers	$\mathbb{R}_P$	
Polynomial time computable real functions	$P(\mathbb{R})$	

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	$\|x\|$	$\max(\|x_1\|, \dots, \|x_n\|)$

Notations for polynomials

Concept	Notation	Comment
Univariate polynomial	$\sum_{i=0}^{d} a_i X^i$	
Multi-index	α	$(\alpha_1, \dots, \alpha_k) \in \mathbb{N}^k$
	$\|\alpha\|$	$\alpha_1 + \dots + \alpha_k$
	$\alpha!$	$\alpha_1!\alpha_2! \cdots \alpha_k!$
Multivariate polynomial	$\sum_{\|\alpha\| \leqslant d} a_\alpha X^\alpha$	where $X^\alpha = X_1^{\alpha_1} \cdots X_k^{\alpha_k}$
Degree	$\deg(P)$	Maximum degree of a monomial, X^α is of degree $\|\alpha\|$, 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	$\Sigma P = \sum_\alpha \|a_\alpha\|$
	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	$\lceil x \rceil$	$\min\{n \in \mathbb{Z} \mid x \leqslant n\}$
Rounding function	$\lfloor x \rceil$	$\operatorname{argmin}_{n \in \mathbb{Z}} \|n - x\|$, undefined for $x = n + \frac{1}{2}$
Integer part function	$\operatorname{int}(x)$	$\max(0, \lfloor x \rfloor)$
	$\operatorname{int}_n(x)$	$\min(n, \operatorname{int}(x))$

Miscellaneous functions

Concept	Notation	Comment
Fractional part function	$\mathrm{frac}(x)$	$x - \mathrm{int}\, x$
	$\mathrm{frac}_n(x)$	$x - \mathrm{int}_n(x)$
Composition operator	$f \circ g$	$(f \circ g)(x) = f(g(x))$
Identity function	id	$\mathrm{id}(x) = x$
Indicator function	$\mathbb{1}_X$	$\mathbb{1}_X(x) = 1$ if $x \in X$ and $\mathbb{1}_X(x) = 0$ otherwise
n^{th} iterate	$f^{[n]}$	$f^{[0]} = \mathrm{id}$ and $f^{[n+1]} = f^{[n]} \circ f$

Calculus

Concept	Notation	Comment
Derivative	f'	
n^{th} derivative	$f^{(n)}$	$f^{(0)} = f$ and $f^{(n+1)} = f^{(n)\prime}$
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	$f(x) = \mathcal{O}(g(x))$	$\exists M, x_0 \in \mathbb{R}$, $\lvert f(x)\rvert \leqslant M \lvert g(x)\rvert$ for all $x \geqslant x_0$
Soft O notation	$f(x) = \tilde{\mathcal{O}}(g(x))$	Means $f(x) = \mathcal{O}\big(g(x) \log^k g(x)\big)$ for some k
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	$f :\subseteq X \to Y$	$\mathrm{dom}\, f \subseteq X$
Restriction	$f\restriction_I$	$f\restriction_I(x) = f(x)$ for all $x \in \mathrm{dom}\, f \cap I$

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	$\underbrace{ww\cdots w}_{k \text{ times}}$

E Polytime analog computability implies polytime computability

E.1 Proof of Theorem 1 (AP implies P)

Let us introduce the following definition.

▶ **Definition 22** (Analog computability). Let $n, m \in \mathbb{N}$, $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$ and $\Upsilon, \Omega : \mathbb{R}_+^2 \to \mathbb{R}_+$. We say that f is (Υ, Ω)-computable if and only if there exists $d \in \mathbb{N}$, 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

- $y(0) = q(x)$ and $y'(t) = p(y(t))$ for all $t \geqslant 0$ ▶ y satisfies a PIVP
- for all $\mu \in \mathbb{R}_+$, if $t \geqslant \Omega(\|x\|, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ ▶ $y_{1..m}$ converges to $f(x)$
- $\|y(t)\| \leqslant \Upsilon(\|x\|, t)$, for all $t \geqslant 0$ ▶ $y(t)$ is bounded

We denote by $\mathrm{AC}(\Upsilon, \Omega)$ the set of (Υ, Ω)-computable functions.

A function $f \in \mathrm{AP}$ must belong to $\mathrm{AC}(\Upsilon, \Omega)$ where Υ, Ω are polynomials which we can assume to be increasing function. This follows from the fact that any system can be rescaled using the length of the curve to make sure it does not grow faster than a polynomial. A formal proof of this fact can be found in Appendix F.4 (the proof that AP implies AWP implies exactly that).

Apply Definition 22 to get d, p and q. Apply Theorem 7 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:

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

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, $|y_1(\Omega(|r|, \mu)) - f(r)| \leqslant e^{-\mu}$ for any $\mu \in \mathbb{R}_+$ and $\|y(t)\| \leqslant \Upsilon(|r|, t)$ for all $t \in \mathbb{R}_+$. Note that the coefficients of p and q belongs to $\mathbb{R}_P$. One can compute a rational r' such that $|y(t) - r'| \leqslant 2^{-n}$ in time:

$$\operatorname{poly}(\deg(p), \operatorname{len}_y(0, t), \log\|y(0)\|, \log \Sigma p, -\log 2^{-n})^d$$

Recall that in this case, all the parameters $d, \Sigma p, \deg(p)$ only depend on f and 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 $|y(\Omega(|r|, (n+1)\ln 2) - r'| \leqslant 2^{-n-1}$ in time:

$$\operatorname{poly}(\operatorname{len}_y(0, \Omega(|r|, (n+1)\ln 2)), \log\|q(r)\|, (n+1)\ln 2)$$

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,

$$\begin{aligned}\operatorname{len}_y(0, \Omega(|r|, (n+1)\ln 2)) &= \int 0^{\Omega(|r|,(n+1)\ln 2)} \max(1, \|y(t)\|)^{\deg(p)} dt \\ &\leqslant \int 0^{\Omega(|r|,(n+1)\ln 2)} \operatorname{poly}(\Upsilon(\|r\|, t)) dt\end{aligned}$$

$$\leqslant \Omega(|r|,(n+1)\ln 2)\,\mathrm{poly}(\Upsilon(|r|,\Omega(|r|,(n+1)\ln 2)))dt$$
$$\leqslant \mathrm{poly}(|r|,n) \leqslant \mathrm{poly}(n)$$

Thus r' can be computed in time:

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

Finally:

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

This show that f is polytime computable.

E.2 Proof of Theorem 7

A way to get a short proof is to use Lemma 13: Let Ω, δ, d, p and y_0 be corresponding to the statement of Lemma 13. Without loss of generality, we assume Ω 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:

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

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

F Polytime computability implies polytime analog computability

F.1 Proof of Lemma 10

▶ Remark (Uniqueness). The uniqueness of y in Definition 9 can be seen as follows: consider $x \in I$ and γ a smooth curve[16] from x_0 to x with values in I and consider $z(t) = y(\gamma(t))$ for $t \in [0,1]$. It can be seen that $z'(t) = J_y(\gamma(t))\gamma'(t) = p(y(\gamma(t))\gamma'(t) = p(z(t))\gamma'(t)$, $z(0) = y(x_0) = y_0$ and $z(1) = y(x)$. The initial value problem $z(0) = y_0$ and $z'(t) = p(z(t))\gamma'(t)$ satisfies the hypothesis of the Cauchy-Lipschitz theorem and as such admits a unique solution. Since this IVP is independent of y, it shows that $y(x)$ must be unique. Note that the existence of y (and thus the domain of definition) is an hypothesis of the definition.

▶ Remark (Regularity). In the euclidean space $\mathbb{R}^n$, C^k smoothness is equivalent to the smoothness of the order k partial derivatives. Consequently, the equation $J_y = p(y)$ on the open set I immediately proves that y is C^∞. As solutions of analytic ODE are analytic, y is in fact real analytic.

[16] see Remark F.1

▶ Remark (Domain of definition). Definition 9 requires the domain of definition of f to be connected, otherwise it would not make sense. Indeed, we can only define the value of f at point u if there exists a path from x_0 to u in the domain of f. It could seem, at first sight, that the domain being "only" connected may be too weak to work with. This is not the case, because in the euclidean space $\mathbb{R}^d$, *open* connected subsets are always smoothly arc connected, that is any two points can be connected using a smooth C^1 (and even C^∞) arc.

▶ Remark (Multidimensional output). The following is true: $f :\subseteq \mathbb{R}^d \to \mathbb{R}^n$ is generable if and only if each of its component is generable (*i.e.* f_i is generable for all i).

▶ Remark (Definition consistency). Definition 9 for $d = e = 1$ corresponds to PIVP functions.

Lemma 10 follows clearly from the following more precise statement:

▶ **Lemma 23** (Arithmetic on generable functions). *Let $d, e, n, m \in \mathbb{N}$, $\mathtt{sp}, \overline{\mathtt{sp}} : \mathbb{R} \to \mathbb{R}_+$, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^n \in \mathrm{GVAL}$ and $g :\subseteq \mathbb{R}^e \to \mathbb{R}^m \in \mathrm{GVAL}$. Then:*

- *$f + g, f - g \in \mathrm{GVAL}$ over $\operatorname{dom} f \cap \operatorname{dom} g$ if $d = e$ and $n = m$*
- *$fg \in \mathrm{GVAL}$ if $d = e$ and $n = m$*
- *$f \circ g \in \mathrm{GVAL}$ if $m = d$ and $g(\operatorname{dom} g) \subseteq \operatorname{dom} f$*

Proof. We focus on the case of the composition, the other cases are very similar.

Apply Definition 9 to f and g to respectively get $l, \bar{l} \in \mathbb{N}$, $p \in M_{l,d}(\mathbb{K})[\mathbb{R}^l]$, $\bar{p} \in M_{\bar{l},e}(\mathbb{K})[\mathbb{R}^{\bar{l}}]$, $x_0 \in \operatorname{dom} f \cap \mathbb{K}^d$, $\bar{x}_0 \in \operatorname{dom} g \cap \mathbb{K}^e$, $y_0 \in \mathbb{K}^l$, $\bar{y}_0 \in \mathbb{K}^{\bar{l}}$, $y : \operatorname{dom} f \to \mathbb{R}^l$ and $\bar{y} : \operatorname{dom} g \to \mathbb{R}^{\bar{l}}$. Define $h = y \circ g$, then $J_h = J_y(g)J_g = p(h)\bar{p}_{1..m}(\bar{y})$ and $h(\bar{x_0}) = y(\bar{y_0}) \in \mathbb{K}^l$. In other words $(\bar{y}, h)$ satisfy:

$$\begin{cases} \bar{y}(\bar{x}_0) = y_0 \in \mathbb{K}^{\bar{l}} \\ h(\bar{x}_0) = y(\bar{y}_0) \in \mathbb{K}^l \end{cases} \qquad \begin{cases} \bar{y}' = \bar{p}(\bar{y}) \\ h' = p(h)\bar{p}_{1..m}(\bar{y}) \end{cases}$$

This shows that $f \circ g = z_{1..m} \in \mathrm{GVAL}$. Furthermore, $\|(\bar{y}(x), h(x))\| \leqslant \max(\|\bar{y}(x)\|, \|y(g(x))\|) \leqslant \max(\overline{\mathtt{sp}}(\|x\|), \mathtt{sp}(\|g(x)\|)) \leqslant \max(\overline{\mathtt{sp}}(\|x\|), \mathtt{sp}(\overline{\mathtt{sp}}(\|x\|)))$. ◀

F.2 Proof of Lemma 11

Lemma 11 follows from the following more general statement:

▶ **Theorem 24** (Generable ODE rewriting). *Let $d, n \in \mathbb{N}$, $I \subseteq \mathbb{R}^n$, $X \subseteq \mathbb{R}^d$, $\mathtt{sp} : \mathbb{R}_+ \to \mathbb{R}_+$ and $(f : I \times X \to \mathbb{R}^n) \in \mathrm{GVAL}$. Define $\overline{\mathtt{sp}} = \max(\mathrm{id}, \mathtt{sp})$. Then there exists $m \in \mathbb{N}$, $(g : I \times X \to \mathbb{R}^m) \in \mathrm{GVAL}$ and $p \in \mathbb{K}^m[\mathbb{R}^m \times \mathbb{R}^d]$ such that for any interval J, $t_0 \in \mathbb{K} \cap J$, $y_0 \in \mathbb{K}^n \cap J$, $y \in C^1(J, I)$ and $x \in C^1(J, X)$, if y satisfies:*

$$\begin{cases} y(t_0) = y_0 \\ y'(t) = f(y(t), x(t)) \end{cases} \qquad \forall t \in J$$

then there exists $z \in C^1(J, \mathbb{R}^m)$ such that:

$$\begin{cases} z(t_0) = g(y_0, x(t_0)) \\ z'(t) = p(z(t), x'(t)) \end{cases} \qquad \begin{cases} y(t) = z_{1..d}(t) \\ \|z(t)\| \leqslant \overline{\mathtt{sp}}(\|y(t), x(t)\|) \end{cases} \qquad \forall t \in J$$

Proof. Apply Definition 9 to f get $m \in \mathbb{N}$, $p \in M_{m,n+d}(\mathbb{K})[\mathbb{R}^m]$, $f_0 \in \operatorname{dom} f \cap \mathbb{K}^d$, $w_0 \in \mathbb{K}^m$ and $w : \operatorname{dom} f \to \mathbb{R}^m$ such that $w(f_0) = w_0$, $J_{w(v)} = p(w(v))$, $\|w(v)\| \leqslant \mathtt{sp}(\|v\|)$ and $w_{1..n}(v) = f(v)$ for all $v \in \operatorname{dom} f$. Define $u(t) = w(y(t), x(t))$, then:

$$u'(t) = J_w(y(t), x(t))(y'(t), x'(t))$$

$$
\begin{aligned}
&= p(w(y(t),x(t)))(f(y(t),x(t)),x'(t))\\
&= p(u(t))(u_{1..n}(t),x'(t))\\
&= q(u(t),x'(t))
\end{aligned}
$$

where $q \in \mathbb{K}^m[\mathbb{R}^{m+d}]$ and $u(t_0) = w(y(t_0)) = w(y_0, x(t_0))$. Note that w itself is a generable function and more precisely $w \in \mathrm{GPVAL}\,\mathtt{sp}$ by definition. Finally, note that $y'(t) = u_{1..d}(t)$ so that we get for all $t \in J$:

$$
\begin{cases} y(t_0) = y_0 \\ y'(t) = u_{1..d}(t) \end{cases} \qquad \begin{cases} u(t_0) = w(y_0, x(t_0)) \\ u'(t) = q(u(t), x'(t)) \end{cases}
$$

Define $z(t) = (y(t), u(t))$, then $z(t_0) = (y_0, w(y_0, x(t_0))) = g(y_0, x(t_0))$ where $y_0 \in \mathbb{K}^n$ and $w \in \mathrm{GVAL}$ so $g \in \mathrm{GVAL}$. And clearly $z'(t) = r(z(t), x'(t))$ where $r \in \mathbb{K}^{n+m}[\mathbb{R}^{n+m}]$. Finally, $\|z(t)\| = \|y(t), w(y(t), x(t))\| \leqslant \max(\|y(t)\|, \mathtt{sp}(\|y(t), x(t)\|)) \leqslant \overline{\mathtt{sp}}(\|y(t), x(t)\|)$. ◀

F.3 Proof of GPVAL $\subset$ AP under conditions on the domain

Generable functions are continuous and continuously differentiable, so locally Lipschitz continuous. We can give a precise expression for the modulus of continuity in the case where the domain of definition is simple enough.

▶ **Lemma 25** (Modulus of continuity). *Let* $\mathtt{sp} : \mathbb{R}_+ \to \mathbb{R}_+$, $f \in \mathrm{GVAL}$. *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* $f \in \mathrm{GPVAL}$ *then there exists* $q \in \mathbb{K}[\mathbb{R}]$ *such that if* $[x_1, x_2] \subseteq \operatorname{dom} f$ *then* $\|f(x_1) - f(x_2)\| \leqslant \|x_1 - x_2\| \, q(\max(\|x_1\|, \|x_2\|))$.

The following can be established: See [33].

▶ **Lemma 26** (Generable field stability, [33]). *Let* $(f :\subseteq \mathbb{R}^d \to \mathbb{R}^e) \in \mathrm{GVAL}$, *then* $f(\mathbb{K}^d \cap \operatorname{dom} f) \subseteq \mathbb{K}^e$.

We can go to the proof of GPVAL $\subset$ AP under conditions on the domain:

▶ **Definition 27** (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*.

▶ **Theorem 28** (GPVAL $\subseteq$ AP over star domains). *If* $f \in \mathrm{GPVAL}$ *has a star domain with a vantage point with coordinates in* $\mathbb{R}_P$ *then* $f \in \mathrm{AP}$.

Proof. Let $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{GPVAL}$ and $z_0 \in \operatorname{dom} f \cap \mathbb{K}^n$ a generable vantage point. Apply Definition 9 to get d, p, x_0, y_0 and y. Since y is generable and $z_0 \in \mathbb{K}^d$, apply Lemma 26 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 Lemma 25 to f to get q, and 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, $\|z(t)\| \leqslant \text{poly}(\|x\|)$ because $\|z(t)\|$ is polynomially bounded. This implies that the length of the curve is also polynomially bounded.

As a final remark, one can observe that the issue of the domain is in fact reduced to the problem of building γ. In the case of a star domain, this is trivial. In the general case, one would need to show that there is a "generic" such γ that given a point x goes from x_0 to x and stays in the domain of f. ◀

F.4 Proof that AP implies AWP

We start by a remark:

▶ **Lemma 29** (Norm function, [33]). *There is a family of function* $\text{norm}_{\infty,\delta} \in \text{GPVAL}$ *such that: For any* $x \in \mathbb{R}^n$ *and* $\delta \in]0,1]$ *we have:*

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

To prove AP $\subset$ AWP, the kea idea is to rescale the system using the length of the curve to make sure it does not grow faster than a polynomial. This is then ensured by the technical condition.

More precisely:

Let $f \in \text{ALP}$. Apply Definition 3 to get Ω, d, p, q. Also assume that polynomial Ω is an increasing function. Let $k = \deg(p)$. Apply Lemma 29 to get that $g(x) = \text{norm}_{\infty,1}(p(x))$ belongs to GPVAL. Apply Definition 9 to get corresponding m, r, x_0 and z_0. Let $x \in \operatorname{dom} f$. For the analysis, it will useful to consider the following systems:

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

Note that by definition $z_1(x) = g(x)$. Define $\psi(t) = g(y(t))$ and $\hat{\psi}(u) = \int_0^u \psi(t)dt$. Now define the following system:

$$\begin{cases} \hat{y}(0) = q(x) \\ \hat{z}(0) = z(q(x)) \\ \hat{w}(0) = \frac{1}{g(q(x))} \end{cases} \qquad \begin{cases} \hat{y}'(u) = \hat{w}(u)p(\hat{y}(u)) \\ \hat{z}'(u) = \hat{w}(u)r(\hat{z}(u))p(\hat{y}(u)) \\ \hat{w}'(u) = -\hat{w}(u)^3 r_1(\hat{z}(u))p(\hat{y}(u)) \end{cases}$$

where by r_1 we mean the first line of r. We will check that $\hat{y}(u) = y(\hat{\psi}^{-1}(u))$, $\hat{z}(u) = z(\hat{y}(u))$ and $\hat{w}(u) = (\hat{\psi}^{-1})'(u)$. We will use the fact that for any $h \in C^1$, $(g^{-1})' = \frac{1}{g' \circ g^{-1}}$. Also note that $\hat{\psi}' = \psi$.

- $\hat{y}(0) = y(\hat{\psi}^{-1}(0)) = y(0) = q(x)$
- $\hat{y}'(u) = (\hat{\psi}^{-1})'(u)y'(\hat{\psi}^{-1}(u)) = \hat{w}(u)p(y(\hat{\psi}^{-1}(u))) = \hat{w}(u)p(\hat{y}(u))$
- $\hat{z}(0) = z(\hat{y}(0)) = z(q(x))$
- $\hat{z}'(u) = J_z(\hat{y}(u))\hat{y}'(u) = \hat{w}(u)r(z(\hat{y}(u)))p(\hat{y}(u)) = \hat{w}(u)r(\hat{z}(u))p(\hat{y}(u))$
- $\hat{w}(0) = \frac{1}{\hat{\psi}'(\hat{\psi}^{-1}(0))} = \frac{1}{\psi(0)} = \frac{1}{g(q(x))}$
- $\hat{w}'(u) = \frac{-(\hat{\psi}^{-1})'(u)\hat{\psi}''(\hat{\psi}^{-1}(u))}{(\hat{\psi}'(\hat{\psi}^{-1}(u)))^2} = -\hat{w}(u)^3\psi'(\hat{\psi}^{-1}(u)) = \nabla g(y(\hat{\psi}^{-1}(u))) \cdot y'(\hat{\psi}^{-1})$ and since $\nabla g(x) = r_1(z(x))^T$ (transpose of the first line of the jaocibian matrix of z because $g = z_1$) then $\hat{w}'(u) = -\hat{w}(u)^3 r_1(z(y(\hat{\psi}^{-1}(u))))^T \cdot p(y(\hat{\psi}^{-1}(u))) = -\hat{w}(u)^3 r_1(\hat{z}(u))p(\hat{y}(u))$

We now claim that this system computes f quickly and has polynomial bound. First note that by Lemma 29, $\|y'(t)\| \leqslant g(y(t)) \leqslant \|y'(t)\| + 1$ thus $\text{len}_y(0,t) \leqslant \hat{\psi}(t) \leqslant \text{len}_y(0,t) + t$. Thus

$$\begin{aligned}\operatorname{len}_{\hat{y}}(0,u) &= \int_0^u \|\hat{y}'(\xi)\|\,d\xi = \int_0^{\hat{\psi}^{-1}(u)} \left\|\hat{w}(\hat{\psi}(t))p(\hat{y}(\hat{\psi}(t)))\right\| \hat{\psi}'(t)dt\\ &= \int_0^{\hat{\psi}^{-1}(u)} \left\|(\hat{\psi}^{-1})'(\hat{\psi}(t))\hat{\psi}'(t)p(y(t))\right\| dt = \int_0^{\hat{\psi}^{-1}(u)} \|p(y(t))\|\,dt = \operatorname{len}_y(0,\hat{\psi}^{-1}(u)) \leqslant \hat{\psi}(\hat{\psi}^{-1}(u)) \leqslant u.\end{aligned} \tag{4}$$

It follows that $\|\hat{y}(u)\| \leqslant \|\hat{y}(0)\| + u \leqslant \|q(x)\| + u \leqslant \operatorname{poly}(\|x\|, u)$. Similarly, $\|\hat{z}(u)\| = \|z(\hat{y}(u))\| \leqslant \operatorname{poly}(\|x\|, u)$ because $z \in \mathrm{GPVAL}$ and thus is polynomially bounded. Finally, $\|\hat{w}\| = \frac{1}{\psi(\hat{\psi}^{-1}(u)} = \frac{1}{g(\hat{y}(u))} \leqslant \frac{1}{\|y'(\hat{\psi}^{-1}(u))\|} \leqslant 1$ because by hypothesis, $\|y'(t)\| \geqslant 1$ for all $t \in \mathbb{R}_+$. This shows that indeed $\|(\hat{y},\hat{z},\hat{w})(u)\|$ is polynomially bounded in $\|x\|$ and u. Now let $\mu \in \mathbb{R}_+$ and $t \geqslant 1 + \Omega(\|x\|, \mu)$ then $\operatorname{len}_{\hat{y}}(0,t) = \operatorname{len}_y(0,\hat{\psi}^{-1}(t)) \geqslant \hat{\psi}(\hat{\psi}^{-1}(t)) - \hat{\psi}^{-1}(t) \geqslant t - \hat{\psi}^{-1}(t) \geqslant 1 + \Omega(\|x\|, \mu) - \frac{1}{\psi(\hat{\psi}^{-1}(t))} \geqslant \Omega(\|x\|, \mu)$ because, as we already saw, $\left\|\psi(\hat{\psi}^{-1}(t))\right\| \geqslant 1$. Thus by definition, $\|\hat{y}_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ because $\hat{y}(t) = y(\hat{\psi}^{-1}(t))$. This shows that $f \in \mathrm{AWP}$.

F.5 Proof that AWP implies ARP

The purpose is to state that one can tolerate small errors on the dynamic.

Formally:

▶ **Theorem 30** (Weak $\subseteq$ robust)**.** $\mathrm{AWP} \subseteq \mathrm{ARP}$.

where

▶ **Definition 31** (Analog robust computability)**.** Let $n, m \in \mathbb{N}$, $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$, $\Theta, \Omega : \mathbb{R}_+^2 \to \mathbb{R}_+$ and $\Upsilon : \mathbb{R}_+^3 \to \mathbb{R}_+$. We say that f is $(\Upsilon, \Omega, \Theta)$-robustly-computable if and only if there exists $d \in \mathbb{N}$, and $(h : \mathbb{R}^d \to \mathbb{R}^d), (g : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^d) \in \mathrm{GPVAL}$ such that for any $x \in \operatorname{dom} f$, $\mu \in \mathbb{R}_+$, $e_0 \in \mathbb{R}^d$ and $e \in C^0(\mathbb{R}_+, \mathbb{R}^d)$ satisfying $\|e_0\| + \int_0^\infty \|e(t)\|\,dt \leqslant e^{-\Theta(\|x\|,\mu)}$, there exists (a unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ satisfying for all $t \in \mathbb{R}_+$:

- $y(0) = g(x,\mu) + e_0$ and $y'(t) = h(y(t)) + e(t)$ ▶ y satisfies a generable IVP
- if $t \geqslant \Omega(\|x\|, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$ ▶ $y_{1..m}$ converges to $f(x)$
- $\|y(t)\| \leqslant \Upsilon(\|x\|, \mu, t)$ ▶ $y(t)$ is bounded

We denote by $\mathrm{AR}(\Upsilon, \Omega, \Theta)$ the set of $(\Upsilon, \Omega, \Theta)$-robustly-computable functions, and by ARP the set of $(\operatorname{poly}, \operatorname{poly}, \operatorname{poly})$-robustly-computable functions.

The following Lemma can be proved: See [33] for its motivation, and for an explanation of the proof of the next theorem on simple cases.

▶ **Lemma 32** (PIVP Slow-Stop,[33])**.** *Let $d \in \mathbb{N}$, $y_0 \in \mathbb{R}^d$, $T, \theta \in \mathbb{R}_+$, $(e_{0,y}, e_{0,A}) \in \mathbb{R}^{d+1}$, $(e_y, e_A) \in C^0(\mathbb{R}_+, \mathbb{R}^{d+1})$ and $p \in \mathbb{K}^d[\mathbb{R}^d]$. Assume that $\|e_0\| + \int_0^\infty \|e(t)\|\,dt \leqslant e^{-\theta}$ and consider the following system:*

$$\begin{cases} y(0) = y_0 + e_{0,y} \\ A(0) = T + 2 + e_{0,A} \end{cases} \qquad \begin{cases} y'(t) = \frac{1+\tanh(A(t))}{2} p(y(t)) + e_y(t) \\ A'(t) = -1 + e_A(t) \end{cases}$$

Then there exists an increasing function $\psi \in C^0(\mathbb{R}_+, \mathbb{R}_+)$ and $z : \psi(\mathbb{R}_+) \to \mathbb{R}^d$ such that:

$$\psi(0) = 0 \qquad z(0) = y_0 + e_{0,y} \qquad z'(t) = p(z(t)) + (\psi^{-1})'(t) e_y(\psi^{-1}(t))$$

and $y(t) = z(\psi(t))$. Furthermore $\psi(T+1) \geqslant T$ and $\psi(t) \leqslant T + 4$ for all $t \in \mathbb{R}_+$. Furthermore, $|A(t)| \leqslant T + 3$ for all $t \in \mathbb{R}_+$.

We will also need the following small theorem about PIVP.

▶ **Theorem 33** (Parameter dependency of PIVP,[34]). *Let $I=[a,b]$, $p\in\mathbb{R}^n[\mathbb{R}^{n+d}]$, $k=\deg(p)$, $e\in C^0(I,\mathbb{R}^d)$, $x,\delta\in C^0(I,\mathbb{R}^n)$ and $y_0,z_0\in\mathbb{R}^d$. Assume that $y,z:I\to\mathbb{R}^d$ satisfy:*

$$\begin{cases} y(a)= y_0\\ y'(t)= p(y(t),x(t))\end{cases}\qquad \begin{cases} z(a)= z_0\\ z'(t)= e(t)+p(z(t),x(t)+\delta(t))\end{cases}\qquad t\in I$$

Assume that there exists $\varepsilon>0$ such that for all $t\in I$,

$$\mu(t):=\left(\|z_0-y_0\|+\int_a^t\|e(u)\|+k\Sigma pM^{k-1}(u)\|\delta(u)\|\,du\right)\exp\left(k\Sigma p\int_a^t M^{k-1}(u)du\right)<\varepsilon$$

where $M(t)=\varepsilon+\|y(t)\|+\|x(t)\|+\|\delta(t)\|$. Then for all $t\in I$,

$$\|z(t)-y(t)\|\leqslant\mu(t)$$

Recall:

▶ **Definition 34** (Analog weak computability). Let $n,m\in\mathbb{N}$, $f:\subseteq\mathbb{R}^n\to\mathbb{R}^m$, $\Omega:\mathbb{R}_+^2\to\mathbb{R}_+$ and $\Upsilon:\mathbb{R}_+^3\to\mathbb{R}_+$. We say that f is (Υ,Ω)-weakly-computable if and only if there exists $d\in\mathbb{N}$, $p\in\mathbb{K}^d[\mathbb{R}^d],q\in\mathbb{K}^d[\mathbb{R}^{n+1}]$ 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
- if $t\geqslant\Omega(\|x\|,\mu)$ then $\|y_{1..m}(t)-f(x)\|\leqslant e^{-\mu}$ ▶ $y_{1..m}$ converges to $f(x)$
- $\|y(t)\|\leqslant\Upsilon(\|x\|,\mu,t)$ ▶ $y(t)$ is bounded

We denote by $\mathrm{AW}(\Upsilon,\Omega)$ the set of (Υ,Ω)-weakly-computable functions.

The proof of Theorem 30 is then the following.

Proof. Let Υ^*,Ω^* be polynomials such that $f\in\mathrm{AW}(\Upsilon^*,\Omega^*)$. Without loss of generality, we assume they are increasing functions of both arguments. Apply Definition 34 to get $d\in\mathbb{N}$, $p\in\mathbb{K}^d[\mathbb{R}^d]$, $q\in\mathbb{K}^d[\mathbb{R}^{n+1}]$ and let $k=\deg(p)$. Define:

$$T(\alpha,\mu)=\Omega^*(\alpha,\mu+\ln 2)$$
$$\Theta(\alpha,\mu)=k\Sigma p(T(\alpha+1,\mu)+4)(\Upsilon^*(\alpha,\mu,T(\alpha+1,\mu)+4)+1)^{k-1}+\mu+\ln 2$$
$$\Omega(\alpha,\mu)=T(\alpha+1,\mu)+1$$

Let $x\in\operatorname{dom}f$, $(e_{0,y},e_{0,A})\in\mathbb{R}^{d+1}$, $(e_y,e_A)\in C^0(\mathbb{R}_+,\mathbb{R}^{d+1})$ and $\mu\in\mathbb{R}_+$ such that $\|e_0\|+\int_0^\infty\|e(t)\|\,dt\leqslant e^{-\Theta(\|x\|,\mu)}$. Apply Lemma 32 and consider the following systems (where ψ is given by the lemma):

$$\begin{cases} y(0)= q(x,\mu)+e_{0,y}\\ A(0)= T(\mathrm{norm}_{\infty,1}(x),\mu)+2+e_{0,A}\end{cases}\qquad \begin{cases} y'(t)= \frac{1+\tanh(A(t))}{2}p(y(t))+e_y(t)\\ A'(t)= -1+e_A(t)\end{cases}$$

$$\begin{cases} z(0)= q(x,\mu)+e_{0,y}\\ z'(t)= p(z(t))+(\psi^{-1})'(t)e_y(\psi^{-1}(t))\end{cases}\qquad \begin{cases} w(0)= q(x,\mu)\\ w'(t)= p(w(t))\end{cases}$$

By definition of p and q, if $t\geqslant\Omega^*(\|x\|,\mu)$ then $\|w_{1..m}(t)-f(x)\|\leqslant e^{-\mu}$. Furthermore, $\|w(t)\|\leqslant\Upsilon^*(\|x\|,\mu,t)$ for all $t\in\mathbb{R}_+$. Define $T^*=T(\mathrm{norm}_{\infty,1}(x),\mu)$. Apply Lemma 29 to get that $\|x\| \leqslant \operatorname{norm}_{\infty,1}(x) \leqslant \|x\|+1$ and thus $T(\|x\|,\mu) \leqslant T^* \leqslant T(\|x\|+1,\mu)$. By construction, $\psi(t) \leqslant T^*+4$ for all $t \in \mathbb{R}_+$. Let $t \in \mathbb{R}_+$, apply Theorem 33 by checking that:

$$\begin{aligned}
&\left(\|e_{0,y}\| + \int_0^{\psi(t)} \left\|(\psi^{-1})'(u)e_y(\psi^{-1}(u))du\right\|\right) e^{k\Sigma p \int_0^{\psi(t)}(\|w(u)\|+1)^{k-1}du} \\
&\leqslant \left(\|e_{0,y}\| + \int_0^t \|e_y(u)\|\,du\right) e^{k\Sigma p \int_0^{\psi(t)}(\Upsilon^*(\|x\|,\mu,u)+1)^{k-1}du} && \text{by a change of variable}\\
&\leqslant e^{k\Sigma p\psi(t)(\Upsilon^*(\|x\|,\mu,\psi(t))+1)^{k-1}-\Theta(\|x\|,\mu)} && \text{by hypothesis on the error}\\
&\leqslant e^{k\Sigma p(T(\|x\|+1,\mu)+4)(\Upsilon^*(\|x\|,\mu,T(\|x\|+1,\mu)+4)+1)^{k-1}-\Theta(\|x\|,\mu)} && \text{because } \psi \text{ is bounded}\\
&\leqslant e^{-\mu-\ln 2} \leqslant 1 && \text{by definition of } \Theta
\end{aligned}$$

Thus $\|z(\psi(t))-w(\psi(t))\| \leqslant e^{-\mu-\ln 2}$ for all $t \in \mathbb{R}_+$. Furthermore, if $t \geqslant \Omega(\|x\|,\mu)$ then $\psi(t) \geqslant \psi(T(\|x\|+1,\mu)+1) \geqslant \psi(T^*+1) \geqslant T^*$. By construction $\psi(T^*) \geqslant T^*$ so $\psi(t) \geqslant T^* \geqslant T(\|x\|,\mu) = \Omega^*(\|x\|,\mu+\ln 2)$ thus $\|z(\psi(t))-f(x)\| \leqslant e^{-\mu-\ln 2}$. Consequently, $\|y(t)-f(x)\| \leqslant \|z(\psi(t))-w(\psi(t))\| + \|w(\psi(t))-f(x)\| \leqslant 2e^{-\mu-\ln 2} \leqslant e^{-\mu}$.

Let $t \in \mathbb{R}_+$, then $\|y(t)\| = \|z(\psi(t))\| \leqslant \|w(\psi(t))\| + e^{-\mu} \leqslant \Upsilon^*(\|x\|,\mu,\psi(t))+1 \leqslant \Upsilon^*(\|x\|,\mu,T(\|x\|+1,\mu)+4)+1 \leqslant \Upsilon^*(\|x\|,\mu,\Omega^*(\|x\|+1,\mu+\ln 2)+4)+1$ which is polynomially bounded in $\|x\|$ and μ. Furthermore $|A(t)| \leqslant T^*+4 \leqslant \Omega^*(\|x\|+1,\mu+\ln 2)+4$ which are both polynomially bounded in $\|x\|$, μ.

Finally, $(y,A)(0) = g(x,\mu)+e_0$ and $(y,A)'(t) = h(y(t),A(t))+e(t)$ where g and h belong to GPVAL because $\tanh, \operatorname{norm}_{\infty,1} \in$ GPVAL. ◀

▶ Remark (Polynomial versus generable). The proof of Theorem 30 also works if q is generable (i.e. $q \in$ GPVAL) instead of polynomial in Definition 22 or Definiinition 34.

F.6 Proof that ARP implies ASP

The purpose is to state that one can tolerate small errors on the dynamic + on inputs.

Formally:

▶ **Theorem 35** (Robust $\subseteq$ strong). ARP $\subseteq$ ASP.

where

▶ **Definition 36** (Analog strong computability). Let $n,m \in \mathbb{N}$, $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$, $\Theta,\Omega : \mathbb{R}_+^2 \to \mathbb{R}_+$ and $\Upsilon : \mathbb{R}_+^4 \to \mathbb{R}_+$. We say that f is (Υ,Ω,Θ)-strongly-computable if and only if there exists $d \in \mathbb{N}$, and $(h : \mathbb{R}^d \to \mathbb{R}^d), (g : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^d) \in$ GPVAL such that for any $x \in \mathbb{R}^n$, $\mu \in \mathbb{R}_+$, $e_0 \in \mathbb{R}^d$ and $e \in C^0(\mathbb{R}_+,\mathbb{R}^d)$, there is exists (a unique) $y : \mathbb{R}_+ \to \mathbb{R}^d$ satisfying for all $t \in \mathbb{R}_+$ and $\hat{e}(t) = \|e_0\| + \int_0^t \|e(u)\|\,du$:

- $y(0) = g(x,\mu)+e_0$ and $y'(t) = h(y(t))+e(t)$ ▶ y satisfies a generable IVP
- if $x \in \operatorname{dom} f$, $t \geqslant \Omega(\|x\|,\mu)$ and $\hat{e}(t) \leqslant e^{-\Theta(\|x\|,\mu)}$ then $\|y_{1..m}(t)-f(x)\| \leqslant e^{-\mu}$
- $\|y(t)\| \leqslant \Upsilon(\|x\|,\mu,\hat{e}(t),t)$ ▶ $y(t)$ is bounded

We denote by AS(Υ,Ω,Θ) the set of (Υ,Ω,Θ)-strongly-computable functions, and by ASP the set of (poly, poly, poly)-strongly-computable functions.

The following Lemma can be proved by providing explicitely such a function:

▶ **Lemma 37** (Max function, [33]). *There is a family of functions* $\mathrm{mx}_\delta \in$ GPVAL *such that: For any* $x,y \in \mathbb{R}$ *and* $\delta \in]0,1]$ *we have:*

$$\max(x,y) \leqslant \mathrm{mx}_\delta(x,y) \leqslant \max(x,y)+\delta$$

For any $x \in \mathbb{R}^n$ and $\delta \in]0,1]$ we have:

$$\max(x_1,\ldots,x_n) \leqslant \mathrm{mx}_\delta(x) \leqslant \max(x_1,\ldots,x_n)+\delta$$

The following lemmas can also be established:

▶ **Lemma 38** (Bounds on tanh, [33]). $1-\operatorname{sgn}(t)\tanh(t) \leqslant e^{-|t|}$ *for all* $t \in \mathbb{R}$.

▶ **Lemma 39** (Perturbed time-scaling). *Let $d \in \mathbb{N}$, $x_0 \in \mathbb{R}^d$, $p \in \mathbb{R}^d[\mathbb{R}^d]$, $e \in C^0(\mathbb{R}_+,\mathbb{R}^d)$ and $\phi \in C^0(\mathbb{R}_+,\mathbb{R}_+)$. Let $\psi(t)=\int_0^t \phi(u)du$. Assume that ψ is an increasing function and that $y,z:\mathbb{R}_+\to\mathbb{R}^d$ satisfy for all $t\in\mathbb{R}_+$:*

$$\begin{cases} y(0)= x_0\\ y'(t)= p(y(t))+(\psi^{-1})'(t)e(\psi^{-1}(t))\end{cases} \qquad \begin{cases} z(0)= x_0\\ z'(t)= \phi(t)p(z(t))+e(t)\end{cases}$$

Then $z(t)=y(\psi(t))$ for all $t\in\mathbb{R}_+$. In particular, $\int_0^{\psi(t)}\left\|(\psi^{-1})'(u)e(\psi^{-1}(u))\right\|du = \int_0^t \|e(u)\|\,du$ and $\sup_{u\in[0,\psi(t)]}\left\|(\psi^{-1})'(u)e(\psi^{-1}(u))\right\| = \sup_{u\in[0,t]}\frac{\|e(u)\|}{\phi(u)}$.

Proof. Use that $\phi=\psi'$, $\psi'\cdot(\psi^{-1})'\circ\psi=1$ and that $\psi'\geqslant 0$. ◀

On a more technical side, we will need to "apply" Definition 31 over finite intervals and we need the following lemma to do so.

▶ **Lemma 40** (Finite time robustness). *Let $f\in \mathrm{AR}(\Upsilon,\Omega,\Theta)$, $I=[0,T]$, $x\in\operatorname{dom} f$, $\mu\in\mathbb{R}_+$, $e_0\in\mathbb{R}^d$ and $e\in C^0(I,\mathbb{R}^d)$ such that $\|e_0\|+\int_I\|e(t)\|\,dt < e^{-\Theta(\|x\|,\mu)}$. Assume that $y:I\to\mathbb{R}^d$ satisfies for all $t\in I$:*

$$y(0)=g(x,\mu)+e_0 \qquad y'(t)=h(y(t))+e(t)$$

where g,h come from Definition 31 applied to f. Then for all $t\in I$:
- $\|y(t)\| \leqslant \Upsilon(\|x\|,\mu,t)$
- *if* $t\geqslant\Omega(\|x\|,\mu)$ *then* $\|y_{1..m}-f(x)\|\leqslant e^{-\mu}$

Proof. The trick is simply to extend e so that it is defined over $\mathbb{R}_+$ and such that:

$$\|e_0\|+\int_0^\infty \|e(u)\|\,du \leqslant e^{-\Theta(\|x\|,\mu)}$$

This is always possible because the truncated integral is stricer smaller than the bound. Formally, define for $t\in\mathbb{R}_+$:

$$\bar{e}(t)=\begin{cases} e(t) & \text{if } t\leqslant T\\ e(T)e^{\frac{e(T)}{\varepsilon}(T-t)} & \text{otherwise}\end{cases} \qquad \text{where } \varepsilon = e^{-\Theta(\|x\|,\mu)}-\|e_0\|-\int_I\|e(t)\|>0$$

One easily checks that $\bar{e}\in C^0(\mathbb{R}_+,\mathbb{R}^d)$ and that:

$$\begin{aligned}\|e_0\|+\int_0^\infty\|\bar{e}(t)\|\,dt &= \|e_0\|+\int_0^T\|e(t)\|\,dt+\int_T^\infty e(T)e^{\frac{e(T)}{\varepsilon}(T-t)}dt\\ &= e^{-\Theta(\|x\|,\mu)}-\varepsilon+\left[-\varepsilon e(T)e^{\frac{e(T)}{\varepsilon}(T-t)}\right]_T^\infty\\ &= e^{-\Theta(\|x\|,\mu)}\end{aligned}$$

Assume that $z:\mathbb{R}_+\to\mathbb{R}^d$ satisfies for $t\in\mathbb{R}_+$:

$$z(0)=g(x,\mu) \qquad z'(t)=g(z(t))+\bar{e}(t)$$

Then z satisfies Definition 31 so $\|z\|(t)\leqslant\Upsilon(\|x\|,\mu)$ and if $t\geqslant\Omega(\|x\|,\mu)$ then $\|z_{1..m}(t)-f(x)\|\leqslant e^{-\mu}$. Conclude by noting that $z(t)=y(t)$ for all $t\in[0,T]$ since $e(t)=\bar{e}(t)$. ◀

The proof of Theorem 35 is then the following.

Proof. Let Ω, Θ, Υ be polynomials and $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{AR}(\Upsilon, \Omega, \Theta)$. Without loss of generality, we assume that Ω, Θ, Υ are increasing functions of their arguments. Apply Definition 31 to get d, h and g. Let $x \in \mathbb{R}^n$, $\mu \in \mathbb{R}_+$, $(e_{0,y}, e_{0,\ell}) \in \mathbb{R}^{d+1}$ and $(e_y, e_\ell) \in C^0(\mathbb{R}_+, \mathbb{R}^{d+1})$. Define $\hat{e}(t) = \|e_0\| + \int_0^t \|e(u)\|\, du$, and consider the following system for $t \in \mathbb{R}_+$:

$$\begin{cases} y(0) = g(x, \mu) + e_{0,y} \\ y'(t) = \psi(t) h(y(t)) + e_y(t) \\ \ell(0) = \mathrm{mx}_1(\mathrm{norm}_{\infty,1}(x), \mu) + 1 + e_{0,\ell} \\ \ell'(t) = 1 + e_\ell(t) \end{cases}$$

$$\psi(t) = \frac{1 + \tanh(\Delta(t))}{2} \qquad \Delta(t) = \Upsilon(\ell(t), \ell(t), \ell(t)) + 1 - \mathrm{norm}_{\infty,1}(y(t))$$

We will first show that the system remains polynomially bounded. Apply Lemma 37 and Lemma 29 to get that:

$$\begin{aligned} \|\ell(0)\| &\leqslant \max(\|x\| + 1, \mu) + 1 + \|e_{0,\ell}\| \\ &\leqslant \mathrm{poly}(\|x\|, \mu) + \|e_{0,\ell}\| \end{aligned}$$

Consequently:

$$\begin{aligned} \|\ell(t)\| &\leqslant \|\ell(0)\| + \int_0^t 1 + \|e_\ell(u)\|\, du \\ &\leqslant \mathrm{poly}(\|x\|, \mu) + t + \|e_{0,\ell}\| + \int_0^t \|e_\ell(u)\|\, du \\ &\leqslant \mathrm{poly}(\|x\|, \mu) + t + \hat{e}(t) \\ &\leqslant \mathrm{poly}(\|x\|, \mu, t, \hat{e}(t)) \end{aligned} \tag{5}$$

Since $g, h \in \mathrm{GPVAL}$, there exists $\mathtt{sp}$ and $\overline{\mathtt{sp}}$ polynomials such that $\|g(x)\| \leqslant \mathtt{sp}(\|x\|)$ and $\|h(x)\| \leqslant \overline{\mathtt{sp}}(\|x\|)$ for all $x \in \mathbb{R}^d$ and without loss of generability, we assume that $\mathtt{sp}$ and $\overline{\mathtt{sp}}$ are increasing functions. Let $t \in \mathbb{R}_+$, there are two possibilities:

- If $\Delta(t) \geqslant 0$ then $\mathrm{norm}_{\infty,1}(y(t)) \leqslant 1 + \Upsilon(\ell(t), \ell(t), \ell(t))$ so apply Lemma 29 and use (5) to conclude that $\|y(t)\| \leqslant \mathrm{poly}(\|x\|, \mu, t, \hat{e}(t))$ and thus:

$$\begin{aligned} \|\psi(t) h(y(t))\| &\leqslant \overline{\mathtt{sp}}(\|y(t)\|) && \text{use that } \tanh < 1 \\ &\leqslant \mathrm{poly}(\|x\|, \mu, t, \hat{e}(t)) \end{aligned} \tag{6}$$

- If $\Delta(t) < 0$ then apply Lemma 38 to get that $\psi(t) \leqslant \frac{1}{2} e^{\Delta(t)} \leqslant e^{\Delta(t)}$. Apply Lemma 29 to get that $\Delta(t) \leqslant \Upsilon(\ell(t), \ell(t), \ell(t)) + 1 - \|y(t)\|$ and thus $\|y(t)\| \leqslant \Upsilon(\ell(t), \ell(t), \ell(t)) + 1 - \Delta(t)$ and thus:

$$\begin{aligned} \|\psi(t) h(y(t))\| &\leqslant e^{\Delta(t)} \overline{\mathtt{sp}}(\|y(t)\|) && \text{use the bound on } \psi \\ &\leqslant e^{\Delta(t)} \overline{\mathtt{sp}}(\Upsilon(\ell(t), \ell(t), \ell(t)) + 1 - \Delta(t)) && \text{use the bound on } \|y(t)\| \\ &\leqslant \mathrm{poly}(\ell(t)) e^{\Delta(t)} \mathrm{poly}(-\Delta(t)) && \text{use that } \Upsilon \text{ is polynomial} \\ &\leqslant \mathrm{poly}(\ell(t)) && \text{use that } e^{-x}\mathrm{poly}(x) = \mathcal{O}(1) \text{ for } x \geqslant 0 \text{ and fixed poly} \\ &\leqslant \mathrm{poly}(\|x\|, \mu, t, \hat{e}(t)) \end{aligned} \tag{7}$$

Putting (6) and (7) together, we get that:

$$\begin{aligned}\|y(t)\| &\leqslant \|g(x,\mu)\| + \|e_{0,y}\| + \int_0^t \|\psi(u)h(y(u))\| + \|e_y(u)\|\,du\\ &\leqslant \mathtt{sp}(\|x,\mu\|) + \int_0^t \operatorname{poly}(\|x\|,\mu,u,\hat{e}(u))du + \hat{e}(t)\\ &\leqslant \operatorname{poly}(\|x\|,\mu,t,\hat{e}(t))\end{aligned}$$

We will now analyze the behavior of the system when the error is bounded. Define $\Theta^*(\alpha,\mu) = \Theta(\alpha,\mu)+1$. Define $\hat{\psi}(t) = \int_0^t \psi(u)du$ and note that it is a diffeomorphism since $\psi > 0$. Apply Lemma 39 to get that $y(t) = z(\hat{\psi}(t))$ for all $t \in \mathbb{R}_+$, where z satisfies for $\xi \in \hat{\psi}(\mathbb{R}_+)$:

$$z(0) = g(x,\mu) + e_{0,y} \qquad z'(\xi) = h(z(\xi)) + \tilde{e}(\xi) \qquad \text{where } \int_0^{\hat{\psi}(t)} \|\tilde{e}(\xi)\|\,d\xi = \int_0^t \|e_y(u)\|\,du$$

Assume that $x \in \operatorname{dom} f$ and let $T \in \mathbb{R}_+$ such that $\hat{e}(T) \leqslant e^{-\Theta^*(\|x\|,\mu)}$. Then $\hat{e}(T) < e^{-\Theta(\|x\|,\mu)}$ and for all $t \in [0,T]$:

$$\begin{aligned}\|e_{0,y}\| + \int_0^{\hat{\psi}(t)} \|\tilde{e}\|(u)du &= \|e_{0,y}\| + \int_0^t \|e_y(u)\|\,du\\ &\leqslant \hat{e}(t) \leqslant e^{-\Theta(\|x\|,\mu)}\end{aligned}$$

Apply Lemma 40 to get for all $u \in [0,\hat{\psi}(T)]$:

$$\|z(u)\| \leqslant \Upsilon(\|x\|,\mu,u) \tag{8}$$

$$\text{if } u \geqslant \Omega(\|x\|,\mu) \text{ then } \|z_{1..m}(u) - f(x)\| \leqslant e^{-\mu} \tag{9}$$

Apply Lemmas 37 and 29 to get for all $t \in [0,T]$:

$$\begin{aligned}\ell(t) &\geqslant \operatorname{mx}_1(\operatorname{norm}_{\infty,1}(\|x\|,\mu)) + 1 - \|e_{0,\ell}\| + t - \int_0^t \|e_\ell(u)\|\,du\\ &\geqslant \max(\|x\|,\mu) + 1 + t - \hat{e}(t)\\ &\geqslant \max(\|x\|,\mu,t) && \text{using that } \hat{e}(t) \leqslant 1\end{aligned}$$

Consequently, using Lemma 29, for all $t \in [0,T]$:

$$\begin{aligned}\Delta(t) &\geqslant \Upsilon(\ell(t),\ell(t),\ell(t)) - \|y(t)\|\\ &\geqslant \Upsilon(\|x\|,\mu,t) - \|y(t)\| && \text{using that } \ell(t) \geqslant \max(\|x\|,\mu,t)\\ &= \Upsilon(\|x\|,\mu,t) - \left\|z(\hat{\psi}(t))\right\| && \text{using that } y(t) = z(\hat{\psi}(t))\\ &\geqslant 0 && \text{because } \hat{\psi}(t) \in [0,\hat{\psi}(T)]\end{aligned}$$

Consequently for all $t \in [0,T]$:

$$\hat{\psi}(t) = \int_0^t \psi(u)du = \int_0^t \frac{1+\tanh(\Delta(u))}{2}du \geqslant \frac{t}{2}$$

Define $\Omega^*(\alpha,\mu) = 2\Omega(\alpha,\mu)$. Assume that $T \geqslant \Omega^*(\|x\|,\mu)$ then $\hat{\psi}(T) \geqslant \Omega(\|x\|,\mu)$ and thus $\|y_{1..m}(T) - f(x)\| = \left\|z(\hat{\psi}(T)) - f(x)\right\| \leqslant e^{-\mu}$.

Finally, $(y,\ell)(0) = g^*(x,\mu) + e_0$ where $g^* \in \mathrm{GPVAL}$. Similarly $(y,\ell)'(t) = h^*((y,\ell)(t)) + e(t)$ where $h^* \in \mathrm{GPVAL}$. Note again that both h^* and g^* are defined over the entire space. This concludes the proof that $f \in \mathrm{AS}(\Omega^*, \operatorname{poly}, \Theta^*)$. ◀

F.7 Proof that ASP implies AXP

The purpose is to deal with the fact a system could explode (i.e. behave uncorrectly) for inputs not in the domain of the function, or for too big perturbation of the dynamics, by adding a mechanism to forbid explosions in these cases

Formally:

▶ **Theorem 41** (Strong $\subseteq$ extreme)**.** ASP $\subseteq$ AXP. *If $f \in$ ASP then there exists polynomials $\Upsilon, \Lambda, \Theta$ and a constant polynomial Ω such that $f \in \mathrm{AX}(\Upsilon, \Omega, \Lambda, \Theta)$.*

where

▶ **Definition 42** (Extreme computability)**.** Let $n, m \in \mathbb{N}$, $f :\subseteq \mathbb{R}^n \to \mathbb{R}^m$, $\Upsilon : \mathbb{R}_+^3 \to \mathbb{R}_+$ and $\Omega, \Lambda, \Theta : \mathbb{R}_+^2 \to \mathbb{R}_+$. We say that f is $(\Upsilon, \Omega, \Lambda, \Theta)$-extremely-computable if and only if there exists $\delta \geqslant 0$, $d \in \mathbb{N}$ and $(g : \mathbb{R}^d \times \mathbb{R}^{n+1} \to \mathbb{R}^d) \in$ GPVAL 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 exists $\bar{x} \in \operatorname{dom} f$ and $\check{\mu}, \hat{\mu} \geqslant 0$ such that for all $t \in I$, $\mu(t) \in [\check{\mu}, \hat{\mu}]$, $\|x(t) - \bar{x}\| \leqslant e^{-\Lambda(\|\bar{x}\|, \hat{\mu})}$ and $\int_a^b \|e(u)\| du \leqslant e^{-\Theta(\|\bar{x}\|, \hat{\mu})}$ then $\|y_{1..m}(u) - f(\bar{x})\| \leqslant e^{-\check{\mu}}$ whenever $a + \Omega(\|\bar{x}\|, \hat{\mu}) \leqslant u \leqslant b$.

We denote by $\mathrm{AX}(\Upsilon, \Omega, \Lambda, \Theta)$ the set of $(\Upsilon, \Omega, \Lambda, \Theta)$-extremely-computable functions and by AXP the set of (poly, poly, poly, poly)-extremely-computable functions.

A very common pattern in signal processing is known as "sample and hold", where we have a variable signal and we would like to apply some process to it. Unfortunately, the processor often assumes (almost) constant input and does not work in real time (analog-to-digital converters are 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 hold 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 $\tau - \delta$.

The following is proved in [33]

▶ **Lemma 43** (Sample and hold, [33])**.** *There is a family of functions* $\mathrm{sample}_{I,\tau}(t, \mu, x, g) \in$ GPVAL*, where $t \in \mathbb{R}, \mu, \tau \in \mathbb{R}_+, x, g \in \mathbb{R}, I = [a, b] \subsetneq [0, \tau]$, such that: Let $\tau \in \mathbb{R}_+$, $I = [a, b] \subsetneq [0, \tau]$, $y : \mathbb{R}_+ \to \mathbb{R}$, $y_0 \in \mathbb{R}$, $x, e \in C^0(\mathbb{R}_+, \mathbb{R})$ and $\mu : \mathbb{R}_+ \to \mathbb{R}_+$ an increasing function. Suppose that for all $t \in \mathbb{R}_+$:*

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

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 exists $\bar{x} \in \mathbb{R}$ 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 exists $\check{x}, \hat{x} \in \mathbb{R}$ 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 exists* $\nu, \nu' \in \mathbb{R}_+$ *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* $\Omega : \mathbb{R}_+ \to \mathbb{R}_+$ *such that for any* $J = [a, b]$ *and* $\bar{x} \in \mathbb{R}$ *such that for all* $\nu \in \mathbb{R}_+$, $n \in \mathbb{N}$ *and* $t \in (n\tau + I) \cap [a + \Omega(\nu), b]$, $|\bar{x} - x(t)| \leqslant e^{-\nu}$; *then* $|y(t) - \bar{x}| \leqslant e^{-\nu}$ *for all* $t \in [a + \Omega^*(\nu), b]$ *where* $\Omega^*(\nu) = \max(\Omega(\nu + \ln 3), \mu^{-1}(\nu + \ln 3)) + \tau + |I|$

▶ **Lemma 44** ("periodic low-integral-low"). *There is a family of functions* $\mathrm{plil}_{I,\tau} \in \mathrm{GPVAL}$ *where* $\mu, \tau \in \mathbb{R}_+$, $I = [a, b] \subsetneq [0, \tau]$ *and* $x \in \mathbb{R}$ *such that: there exists a constant* K *and* ϕ *such that* $\mathrm{plil}_{I,\tau}(t, \mu, x) = \phi(t, \mu, x)x$ *and:*

- $\mathrm{plil}_{I,\tau}(\cdot, \mu, x)$ *is* τ*-periodic*
- $\forall t \notin I$, $|\,\mathrm{plil}_{I,\tau}(t, \mu, x)| < e^{-\mu}$
- *for any* $\alpha : I \to \mathbb{R}_+, \beta : I \to \mathbb{R}$:

$$1 \leqslant \int_a^b \phi(t, \alpha(t), \beta(t))dt \leqslant K$$

We then get to the proof of Theorem 41

Proof. Let $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{AS}(\Upsilon, \Omega, \Theta)$ where Υ, Ω Θ are polynomials which we assume, without loss of generability, to be increasing functions of theirs inputs. Apply Definition 36 to get d, h and g.

Let $e = 1 + d + m$, $x \in C^0(\mathbb{R}_+, \mathbb{R}^n)$, $\mu \in C^0(\mathbb{R}_+, \mathbb{R}_+)$, $(\nu_0, y_0, z_0) \in \mathbb{R}^e$, $(e_\nu, e_y, e_z) \in C^0(\mathbb{R}_+, \mathbb{R}^e)$ and consider the following system:

$$\begin{cases} \nu(0) = \nu_0 \\ y(0) = y_0 \\ z(0) = z_0 \end{cases} \qquad \begin{cases} \nu'(t) = \mathrm{sample}_{[0,1],4}(t, \mu^*(t), \nu(t), \mu(t) + \ln\Delta + 7) + e_\nu(t) \\ y'(t) = \mathrm{sample}_{[1,2],4}(t, \mu^*(t), y(t), g(x(t), \nu(t))) \\ \qquad\qquad + \mathrm{plil}_{[2,3],4}(t, \mu^*(t), A(t)h(y(t))) + e_y(t) \\ z'(t) = \mathrm{sample}_{[3,4],4}(t, \mu^*(t), z(t), y_{1..m}(t)) + e_z(t) \end{cases}$$

where

$$\Delta = 5 \qquad \Delta' = \ln\Delta + 10$$

$$\mu^*(t) = \mho^*(1 + \mathrm{norm}_{\infty,1}(x(t)), \nu(t) + 4)$$

$$A(t) = 1 + \Omega(1 + \mathrm{norm}_{\infty,1}(x(t)), \nu(t))$$

$$\Lambda^*(\alpha, \mu) = \Theta^*(\alpha, \mu) = \mho^*(\alpha, \mu + \Delta')$$

$$\mho^*(\alpha, \mu) = \mu + \ln\Delta + \Theta(\alpha, \mu) + \ln q(\alpha + \mu)$$

Let $I = [a, b]$ and assume there exists $\bar{x} \in \operatorname{dom} f$ and $\check{\mu}, \hat{\mu} \in \mathbb{R}_+$ such that for all $t \in I$, $\mu(t) \in [\check{\mu}, \hat{\mu}]$, $\|x(t) - \bar{x}\| \leqslant e^{-\Lambda^*(\|\bar{x}\|, \hat{\mu})}$ and $\int_a^b \|e(u)\|\, du \leqslant e^{-\Theta^*(\|\bar{x}\|, \hat{\mu})}$. Apply Theorem 25 to g to get $q \in \mathbb{K}[\mathbb{R}]$, without loss of generality we can assume that q is an increasing function and $q \geqslant 1$. We will use Lemma 29 to get that $\mathrm{norm}_{\infty,1}(x(t)) + 1 \geqslant \|\bar{x}\|$ because $\|x(t) - \bar{x}\| \leqslant 1$. Also note that $\mu^*, \Theta^*, \Lambda^*$ are increasing functions of their arguments. Let $n \in \mathbb{N}$ such that $[4n, 4n + 4] \subseteq I$ and $t \in [4n, 4n + 4]$. We will first analyse the variable ν, note that the analysis is extremely rough to simplify the proof.

- **if** $t \in [4n, 4n + 1]$ **then** $\mu^*(t) \geqslant 0$ so apply Lemma 43 to get that $\nu(4n + 1) \in [\check{\mu} + \ln\Delta + 7 - \varepsilon, \hat{\mu} + \ln\Delta + 7 + \varepsilon]$ where $\varepsilon \leqslant 2e^{-0} + \int_{4n}^{4n+1} |e_\nu(u)|du \leqslant 3$ because $\int_a^b \|e(t)\| \leqslant 1$. Define $\bar{\nu} = \nu(4n + 1)$, then $\bar{\nu} \in [\check{\mu} + \ln\Delta + 4, \hat{\mu} + \underbrace{\ln\Delta + 10}_{=\Delta'}]$

- **if** $t \in [4n+1, 4n+4]$ **then** $\mu^*(t) \geqslant 0$ so apply Lemma 43 to get that $|\nu'(t)| \leqslant e^{-0} + \int_{4n+1}^{t} |e_\nu(u)| du$ and thus $|\nu(t)-\bar{\nu}| \leqslant (t-4n-1)+\int_{4n+1}^{t} \|e(u)\| \, du \leqslant 4$ because $\int_a^b \|e(t)\| \leqslant 1$. In other words $\nu(t) \in [\bar{\nu}-4, \bar{\nu}+4]$.

Furthermore for $t \in [4n+1, 4n+4]$ we have:

$$\mu^*(t) \geqslant \Theta^*(1 + \operatorname{norm}_{\infty,1}(x(t)), \nu(t)+4) \geqslant \mho^*(\|\bar{x}\|, \bar{\nu})$$

It will also be useful to note that:

$$\begin{aligned}\Lambda^*(\|\bar{x}\|, \hat{\mu}) = \Theta^*(\|\bar{x}\|, \hat{\mu}) &\geqslant \mho^*(\|\bar{x}\|, \hat{\mu}+\Delta') \\ &\geqslant \mho^*(\|\bar{x}\|, \bar{\nu})\end{aligned}$$

We can now analyze y using this property:

- **if** $t \in [4n+1, 4n+2]$ **then** $|\nu'(t)| \leqslant e^{-\mu^*(t)} + |e_\nu(t)|$ thus $|\nu(t)-\bar{\nu}| \leqslant e^{-\mho^*(\|\bar{x}\|,\bar{\nu})} + \int_{4n+1}^{4n+2} |e_\nu(u)| du$. Furthermore $\sup_{[4n+1,4n+2]} \|x\| \leqslant \|\bar{x}\| + 1$, thus:

$$\begin{aligned}\|g(\bar{x},\bar{\nu}) - g(x(t),\nu(t))\| &\leqslant \max(|\nu(t)-\bar{\nu}|, \|x(t)-\bar{x}\|) q(\max(\|\bar{x}\|, |\bar{\nu}|)) \\ &\leqslant \max\left(e^{-\Theta^*(\|\bar{x}\|,\hat{\mu})} + e^{-\mho^*(\|\bar{x}\|,\bar{\nu})}, e^{-\Lambda^*(\|\bar{x}\|,\hat{\mu})}\right) q(\|\bar{x}\| + \bar{\nu}) \\ &\leqslant 2e^{-\Theta(\|\bar{x}\|,\bar{\nu})-\ln\Delta}\end{aligned}$$

  Also note that $\left\|y'(t) - \operatorname{sample}_{[1,2],4}(t, \mu^*(t), y(t), g(x(t),\nu(t)))\right\| \leqslant e^{-\mu^*(t)}$ by Lemma 44. So we can apply Lemma 43 to get that $\|y(4n+2) - g(\bar{x},\bar{\nu})\| \leqslant 2e^{-\Theta(\|\bar{x}\|,\bar{\nu})-\ln\Delta} + e^{-\mho^*(\|\bar{x}\|,\bar{\nu})} + \int_{4n+1}^{4n+2} \|e(u)\| \, du \leqslant 4e^{-\Theta(\|\bar{x}\|,\bar{\nu})-\ln\Delta}$.
- **if** $t \in [4n+2, 4n+3]$ **then** apply Lemmas 43 and 44 to get ϕ such that $\int_{4n+2}^{4n+3} \phi(u) du \geqslant 1$ and $\|y'(t) - \phi(t)A(t)h(y(t))\| \leqslant e^{-\mu^*(t)} + \|e_y(t)\|$. Define $\psi(t) = \int_{4n+2}^{t} \phi(u)A(u)du$ then $\psi(4n+3) \geqslant \Omega(\|\bar{x}\|, \bar{\nu})$ since $A(u) \geqslant \Omega(\|\bar{x}\|, \bar{\nu})$ for $u \in [4n+2, 4n+3]$. Apply Lemma 39 over $[4n+2, 4n+3]$ to get that $y(t) = w(\psi(t))$ where w satisfies $w(0) = y(4n+2)$ and $w'(\xi) = h(w(\xi)) + \tilde{e}(\xi)$ where $\tilde{e} \in C^0(\mathbb{R}_+, \mathbb{R}^d)$ satisfies $\int_0^{\psi(t)} \|\tilde{e}(\xi)\| \, d\xi = \int_{4n+2}^{t} \|e_y(u)\| \, du \leqslant e^{-\Theta^*(\|\bar{x}\|,\hat{\mu})} \leqslant e^{-\Theta(\|\bar{x}\|,\bar{\nu})-\ln\Delta}$. Furthermore, $\|w(0) - g(\bar{x},\bar{\nu})\| \leqslant 4e^{-\Theta(\|\bar{x}\|,\bar{\nu})-\ln\Delta}$ from the result above. In other words, $w(0) = g(\bar{x},\bar{\nu}) + \tilde{e}_0$ and $w'(t) = g(w(t))+\tilde{e}(t)$ where $\|\tilde{e}_0\|+\int_0^{\psi(t)} \|e(u)\| \, du \leqslant 5e^{-\Theta(\|\bar{x}\|,\bar{\nu})-\ln\Delta} \leqslant e^{-\Theta(\|\bar{x}\|,\bar{\nu})}$ because $\Delta \geqslant 5$. Apply Definition 36 to get that $\|w_{1..m}(\psi(4n+3)) - f(\bar{x})\| \leqslant e^{-\bar{\nu}}$ since $\psi(4n+3) \geqslant \Omega(\|\bar{x}\|, \bar{\nu})$.
- **if** $t \in [4n+3, 4n+4]$ **then** $\|y'(t)\| \leqslant e^{-\mu^*(t)} + \|e_y(t)\|$ thus $\|y(t) - y(4n+3)\| \leqslant e^{-\mho^*(\|\bar{x}\|,\bar{\nu})} + \int_{4n+3}^{t} \|e_y(u)\| \, du \leqslant 2e^{-\bar{\nu}}$ so $\|y_{1..m}(t) - f(\bar{x})\| \leqslant 3e^{-\bar{\nu}}$.

Note that the above reasoning is also true for the last segment $[4n, b] \subseteq I$ in which case the result only applies up to time b of course. In other words, the results apply as long as $t \in [4n, 4+4] \cap I$ and $4n \geqslant a$. From this we conclude that if $t \in [a+4, b] \cap [4n+3, 4n+3]$ for some $n \in \mathbb{N}$ then $\|y_{1..m}(t) - f(\bar{x})\| \leqslant 3e^{-\bar{\nu}}$. Apply Lemma 43 to get, using that $\bar{\nu} \geqslant \check{\mu} + \ln\Delta$ and $\Delta \geqslant 5$, that for all $t \in [a+5, b]$:

$$\begin{aligned}\|z(t) - f(\bar{x})\| &\leqslant 3e^{-\bar{\nu}} + e^{-\mho^*(\|\bar{x}\|,\bar{\nu})} + \int_{t-5}^{t} \|e(u)\| \, du \leqslant 5e^{-\bar{\nu}} \\ &\leqslant e^{-\check{\mu}}\end{aligned}$$

To complete the proof, we must also analyze the norm of the system. As a shorthand, we introduce the following notation:

$$\operatorname{int}_\delta^+ \alpha(t) = \int_{\max(0,t-\delta)}^{t} \alpha(u) du$$

Apply Lemma 43 to get that:

$$
\begin{aligned}
|\nu(t)| &\leqslant 2 + \int\limits_{\max(0,t-5)}^{t} |e_\nu(u)|du + \max\left(|\nu_0|\mathbb{1}_{[0,4]}(t), \sup\nolimits_5|\mu + \ln\Delta + 7|(t)\right)\\
&\leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_5^+ |e_\nu|(t), \sup\nolimits_5\mu(t)\right)
\end{aligned}
$$

The analysis of y is a bit more painful, as it uses both results about the sampling function and the strongly-robust system we are simulating. Let $n \in \mathbb{N}$, and $t \in [4n, 4n+4]$:

- **if** $t \in [4n, 4n+1]$ **then** apply Lemmas 43 and 44 to get, using that $\mu(t) \geqslant 0$, that $\|y'(t)\| \leqslant 2 + \|e(t)\|$ and thus $\|y(t) - y(4n)\| \leqslant 2 + \int_{4n}^t \|e(u)\|\,du$.
- **if** $t \in [4n+1, 4n+2]$ **then** using the result on ν, $\|g(x(t), \nu(t))\| \leqslant \sup_{[4n+1,t]} \operatorname{poly}(\|x\|, \nu) \leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_6^+ \|e\|(t), \sup_6\mu(t), \sup_1 \|x\|(t)\right)$. Apply Lemmas 43 and 44 to get, using that $\mu(t) \geqslant 0$ and the result on ν, that:

  $$
  \begin{aligned}
  \|y(4n+2)\| &\leqslant \sup_{[4n+1,4n+2]} \|g(x,\nu)\| + 2 + \int_{4n+1}^{4n+2} \|e(u)\|\,du\\
  &\leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(4n+2) + \operatorname{int}_6^+ \|e\|(4n+2), \sup\nolimits_6\mu(4n+2), \sup\nolimits_1 \|x\|(4n+2)\right)
  \end{aligned}
  $$

  and also that:

  $$
  \begin{aligned}
  \|y(t)\| &\leqslant \max\left(\sup_{[4n+1,t]} \|g(x,\nu)\| + 2, \|y(4n+1)\|\right) + \int_{4n+1}^{t} \|e(u)\|\,du\\
  &\leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_6^+ \|e\|(t), \sup\nolimits_6\mu(t), \sup\nolimits_1 \|x\|(t), \|y(4n)\|\right)
  \end{aligned}
  $$

- **if** $t \in [4n+2, 4n+3]$ **then** apply Lemma 43, Lemmas 44, 39 and 36 to get that $\|y(t)\| \leqslant \Upsilon(0, 0, \hat{e}(\hat{A}(t)), \hat{A}(t))$ where $\hat{A}(t) = \int_{4n+2}^t A(u)du$ and $\hat{e}(\hat{A}(t)) = \|y(4n+2) - g(0,0)\| + \int_{4n+2}^t 1 + \|e(u)\|\,du$. Since Ω is a polynomial, and using the result on ν, we get that:

  $$
  \begin{aligned}
  \hat{A}(t) &\leqslant \sup_{[4n+2,t]} \operatorname{poly}(\|x\|, |\nu|)\\
  &\leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_6^+ \|e\|, \sup\nolimits_6\mu(t), \sup\nolimits_1 \|x\|(t)\right)
  \end{aligned}
  $$

  and using that $4n+2 \leqslant t \leqslant 4n+3$:

  $$
  \begin{aligned}
  \|y(4n+2) - g(0,0)\| &\leqslant \|y(4n+2)\| + \|g(0,0)\|\\
  &\leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_6^+ \|e\|, \sup\nolimits_7\mu(t), \sup\nolimits_2 \|x\|(t)\right)
  \end{aligned}
  $$

  And since Υ is a polynomial, we conclude that:

  $$
  \|y(t)\| \leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_6^+ \|e\|(t), \sup\nolimits_7\mu(t), \sup\nolimits_2 \|x\|(t)\right)
  $$

- **if** $t \in [4n+3, 4n+4]$ **then** apply Lemmas 43 and 44 to get, using that $\mu(t) \geqslant 0$, that $\|y'(t)\| \leqslant 2 + \|e(t)\|$ and thus $\|y(t) - y(4n+3)\| \leqslant 2 + \int_{4n+3}^t \|e(u)\|\,du$.

From this analysis we can conclude that for all $t \in [0, 2]$:

$$
\begin{aligned}
\|y(t)\| &\leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_6^+ \|e\|(t), \sup\nolimits_6\mu(t), \sup\nolimits_1 \|x\|(t), \|y(0)\|\right)\\
&\leqslant \operatorname{poly}\left(|\nu_0| + \operatorname{int}_6^+ \|e\|(t), \sup\nolimits_6\mu(t), \sup\nolimits_1 \|x\|(t), \|y_0\|\right)
\end{aligned}
$$

and for all $n \in \mathbb{N}$ and $t \in [4n+2, 4n+6]$:

$$
\|y(t)\| \leqslant \operatorname{poly}\left(|\nu_0|\mathbb{1}_{[0,5]}(t) + \operatorname{int}_9^+ \|e\|(t), \sup\nolimits_9\mu(t), \sup\nolimits_4 \|x\|(t)\right)
$$

Putting everything together, we get for all $t \in \mathbb{R}_+$:

$$
\|y(t)\| \leqslant \operatorname{poly}\left(\|y_0, \nu_0\|\,\mathbb{1}_{[0,5]}(t) + \operatorname{int}_9^+ \|e\|(t), \sup\nolimits_9\mu(t), \sup\nolimits_4 \|x\|(t)\right)
$$

Finally apply Lemma 43 to get the a similar bound on z and thus on the entire system. ◀

F.8 Proof that AXP implies AOP

The purpose is now to go to a notion of online computation, i.e. to Lemma 13

We start by the following lemmas:

▶ **Lemma 45** (AXP time rescaling). *If $f \in \mathrm{AXP}$ then there exists polynomials $\Upsilon, \Lambda, \Theta$ and a constant polynomial Ω such that $f \in \mathrm{AX}(\Upsilon, \Omega, \Lambda, \Theta)$.*

Proof. We go for the shortest proof: we will show that AXP $\subseteq$ AWP and use Theorem 30 then Theorem 35 followed by Theorem 41 which proves exactly our statement.

The proof that AXP $\subseteq$ AWP is next to trivial because the extreme system and some given input and precision, we can simply store the input and precision into some variables and feed them into the system. We make the system autonomous by using a variable to store the time.

Let $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{AX}(\Upsilon, \Omega, \Lambda, \Theta)$, apply Definition 42 to get δ, d and g. Let $x \in \operatorname{dom} f$ and $\mu \in \mathbb{R}_+$, and consider the following system:

$$\begin{cases} x(0)= x \\ \mu(0)= \mu \\ \tau(0)= 0 \\ y(0)= 0 \end{cases} \qquad \begin{cases} x'(t)= 0 \\ \mu'(t)= 0 \\ \tau'(t)= 1 \\ y'(t)= g(t, y(t), x(t), \mu(t)) \end{cases}$$

Clearly he system of the form $z(0) = h(x, \mu)$ and $z'(t) = H(z(t))$ where h and H belong to GPVAL (and are defined over the entire space). Apply the definition to get that:

$$\|y(t)\| \leqslant \Upsilon(\|x\|, \mu, 0)$$

And thus the entire system in bounded by a polynomial in $\|x\|, \mu$ and t. Furthermore, if $t \geqslant \Omega(\|x\|, \mu)$ then $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\mu}$. To conclude the proof, we need to rewrite the system as a PIVP using Theorem 24. ◀

The following is established in [33]

▶ **Lemma 46** (Reach). *There exists* reach $\in$ GPVAL *such that: For any $I = [a, b]$, any $\phi \in C^0(I, \mathbb{R}_+)$, any $g, E \in C^0(I, \mathbb{R})$, any $y_0, g_\infty \in \mathbb{R}$ and $\eta > 0$ such that for all $t \in I$, $|g(t) - g_\infty| \leqslant \eta$. Assume that $y : I \to \mathbb{R}$ satisfies*

$$\begin{cases} y(0)= y_0 \\ y'(t)= \operatorname{reach}(\phi(t), y(t), g(t)) + E(t) \end{cases}$$

Then for any $t \in I$,

$$|y(t) - g_\infty| \leqslant \eta + \int_a^t |E(u)| du + \exp\left(-\int_a^t \phi(u) du\right) \qquad \text{whenever } \int_a^t \phi(u) du \geqslant 1$$

And for any $t \in I$,

$$|y(t) - g_\infty| \leqslant \max(\eta, |y(0) - g_\infty|) + \int_0^t |E(u)| du$$

We then get to the proof of Lemma 13.

Proof. Apart from the issue of the input, the system is quite intuitive: we constantly feed the extreme system with the (smoothed) input and some precision. By increasing the precision with time, we ensure that the system will converge when the input is stable. However there is a small catch: over a time interval I, if we change the precision within a range $[\check{\mu},\hat{\mu}]$ then we must provide the extreme system with precision based on $\hat{\mu}$ in order to get precision $\check{\mu}$. Since the extreme system takes time $\Omega(\|x\|,\hat{\mu})$ to compute, we need arrange so that the requested precision doesn't change too much over periods of this duration to make things simpler. We will use to our advantage that Ω can always be assumed to be a constant.

Let $(f :\subseteq \mathbb{R}^n \to \mathbb{R}^m) \in \mathrm{AX}(\Upsilon,\Omega,\Lambda,\Theta)$ where Υ,Ω,Λ and Θ are polynomials, which we can assume to be increasing functions of their arguments. Apply Lemma 45 to get $\omega > 0$ such that for all $\alpha \in \mathbb{R}^n, \mu \in \mathbb{R}_+$:

$$\Omega(\alpha,\mu) = \omega$$

Apply Definition 42 to get δ, d and g. Define:

$$\tau = \omega + 2 \qquad \delta' = \max(\delta, \tau + 1)$$

Let $x \in C^0(\mathbb{R}_+,\mathbb{R}^n)$ and consider the following systems:

$$\begin{cases} x^*(0) = 0 \\ y(0) = 0 \\ z(0) = 0 \end{cases} \qquad \begin{cases} x^{*\prime}(t) = \mathrm{reach}(\phi(t), x^*(t), x(t)) \\ y'(t) = g(t, y(t), x^*(t), \mu(t)) \\ z'(t) = \mathrm{sample}_{[\omega+1,\omega+2],\tau}(t, \mu(t), z(t), y_{1..m}(t)) \end{cases}$$

where

$$\phi(t) = \ln 2 + \mu(t) + \Lambda^*(2 + x_1(t)^2 + \cdots + x_n(t)^2, \mu(t)) \qquad \mu(t) = \frac{t}{\tau}$$

Let $t \geqslant 1$, since $\phi \geqslant 1$ then Lemma 46 gives:

$$\|x^*(t)\| \leqslant \sup_1 \|x\|(t) + e^{-\int_{t-1}^{t}\phi(u)du} \leqslant \sup_1 \|x\|(t) + 1$$

Also for $t \in [0,1]$ we get that:

$$\|x^*(t)\| \leqslant \sup_{[0,t]} \|x\|$$

This proves that $\|x^*(t)\| \leqslant \sup_1 \|x\|(t) + 1$ for all $t \in \mathbb{R}_+$. From this we deduce that:

$$\begin{aligned}\|y(t)\| &\leqslant \Upsilon(\sup_\delta \|x^*\|(t), \sup_\delta \mu(t), 0) \\ &\leqslant \mathrm{poly}(\sup_\delta \|x\|(t), t)\end{aligned}$$

Apply Lemma 43 to get that:

$$\begin{aligned}\|z(t)\| &\leqslant 2 + \sup_{\tau+1} \|y\|(t) \\ &\leqslant \mathrm{poly}(\sup_{\delta'} \|x\|(t), t)\end{aligned}$$

Let $I = [a,b]$ and assume there exists $\bar{x} \in \operatorname{dom} f$ and $\bar{\mu}$ such that for all $t \in I$, $\|x(t) - \bar{x}\| \leqslant e^{-\Lambda(\|\bar{x}\|,\bar{\mu})}$. Note that $2 + \sum_{i=1}^n x_i(t)^2 \geqslant 1 + \|x(t)\| \geqslant \|\bar{x}\|$ for all $t \in I$. Let $n \in \mathbb{N}$ such that $n \geqslant \bar{\mu} + \ln 2$ and $[n\tau, (n+1)\tau] \subseteq I$. Note that $\mu(t) \in [n, n+1]$ for all $t \in I_n$. Apply Lemma 46, using that $\phi \geqslant 1$, to get that for all $t \in [n\tau + 1, (n+1)\tau]$:

$$\|x^*(t) - \bar{x}\| \leqslant e^{-\Lambda^*(\|\bar{x}\|,n)} + e^{-\int_{n\tau}^{t}\phi(u)du} \leqslant 2e^{-\Lambda^*(\|\bar{x}\|,n)}$$

$$\leqslant e^{-\Lambda(\|\bar{x}\|,\bar{\mu}+\ln 2)}$$

Using the definition of extreme computability, we get that for all $t \in [n\tau+1+\omega,(n+1)\tau] = [n\tau+\omega+1, n\tau+\omega+2]$:

$$\|y_{1..m} - f(\bar{x})\| \leqslant e^{-\bar{\mu}+\ln 2}$$

Define $J = [a + (1+\bar{\mu}+\ln 2)\tau, b] \subseteq I$. Assume that $t \in J \cap [n\tau+1,(n+1)\tau]$ for some $n \in \mathbb{N}$, then we must have $(n+1)\tau \geqslant (1+\bar{\mu}+\ln 2)\tau$ and thus $n \geqslant \bar{\mu}+\ln 2$ so we can apply the above reasoning to get that $\|y_{1..m}(t) - f(x)\| \leqslant e^{-\bar{\mu}+\ln 2}$. Furthermore, we also have $\mu(t) \geqslant \frac{(1+\bar{\mu}+\ln 2)\tau}{\tau} \geqslant \bar{\mu}+\ln 2$ for all $t \in J$. Apply Lemma 43 to conclude that for any $t \in [a+\tau+\bar{\mu}+\ln 2+\tau+1, b]$, we have $\|z(t) - f(x)\| \leqslant 2e^{-\bar{\mu}+\ln 2} \leqslant e^{-\bar{\mu}}$.

To conclude the proof, we need to rewrite the system as a PIVP using Lemma 24. Note that this works because we only rewrite the variable y, and doing so we require that x^* be a C^1 function (which is the case) and the new initial variable will depend on $x^*(0) = 0$ which is constant. ◀

F.9 Proof of Theorem 14

We first precise some concepts.

F.9.1 More on Turing Machines

First the *step* function of a Turing machine $\mathcal{M}$ corresponds to the function 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 47** (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. 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.

F.9.2 Polynomial interpolation

In order to implement the transition function of the Turing Machine, we will use a polynomial interpolation scheme (Lagrange interpolation). But since our simulation may have to deal with some amount of error in inputs, we have to investigate how this error propagates through the interpolating polynomial.

▶ **Definition 48** (Lagrange polynomial). Let $d \in \mathbb{N}$ and $f : G \to \mathbb{R}$ where G is a finite subset of $\mathbb{R}^d$, we define

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

▶ **Lemma 49** (Lagrange interpolation). *For any finite $G \subseteq \mathbb{K}^d$ and $f : G \to \mathbb{K}$, $L_f \in \mathrm{AP}$ and $L_f \restriction_G = f$.*

Proof. The fact that L_f matches f on G is a classical calculation. Also L_f is a polynomial with coefficients in $\mathbb{K}$ so clearly it belongs to AP. ◀

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 50** (Characteristic interpolation). Let $d \in \mathbb{N}$, $f : G \to \mathbb{R}$ where G is a finite subset of $\mathbb{R}^d$, $\alpha \in \mathbb{R}$, and define:

$$D_{f=\alpha}(x) = L_{f_\alpha}(x) \qquad D_{f\neq\alpha}(x) = L_{1-f_\alpha}(x) \qquad f_\alpha(x) = \begin{cases} 1 & \text{if } f(x) = \alpha \\ 0 & \text{otherwise} \end{cases}$$

▶ **Lemma 51** (Characteristic interpolation). *For any finite $G \subseteq \mathbb{K}^d$, $f : G \to \mathbb{K}$ and $\alpha \in \mathbb{K}$, $D_{f=\alpha}, D_{f\neq\alpha} \in \mathrm{AP}$.*

Proof. Observe that $f_\alpha : G \to \{0, 1\}$ and $\{0, 1\} \subseteq \mathbb{K}$. Apply Lemma 49. ◀

F.9.3 Specific Functions and Operations

We need some specific adhoc functions:

▶ **Definition 52** (Round). Let $\mathrm{rnd}^* \in C^0(\mathbb{R}, \mathbb{R})$ be the unique function such that:

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

▶ **Theorem 53** (Round, [33]). $\mathrm{rnd}^* \in \mathrm{AP}$.

The idea of the proof of above theorem 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. In the details, building this function is not 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 $\sin(2\pi x)$.

▶ **Theorem 54** (Closure by arithmetic operations). *If $f, g \in \mathrm{AP}$ then $f \pm g, fg \in \mathrm{AP}$, with the obvious restrictions on the domains of definition.*

Proof. We do the proof in the case of $f + g$ in details. Let $\Omega, \Upsilon, \Omega', \Upsilon'$ polynomials such that $f \in \mathrm{AC}(\Upsilon, \Omega)$ and $g \in \mathrm{AC}(\Upsilon', \Omega')$. Apply Definition 22 to f and g to get d, p, q and d', p', q' respectively. Let $x \in \operatorname{dom} f \cap \operatorname{dom} g$ and consider the following system:

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

Let $\Omega^*(\alpha, \mu) = \max(\Omega(\alpha, \mu + \ln 2), \Omega'(\alpha, \mu + \ln 2))$ and $\Upsilon^*(\alpha, t) = \Upsilon(\alpha, t) + \Upsilon'(\alpha, t)$. Since, by construction, $w(t) = y(t) + z(t)$, if $t \geqslant \Omega^*(\alpha, \mu)$ 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 $\|w(t)\| \leqslant \Upsilon^*(\|x\|, t)$.

The case of $f-g$ is exactly the same. The case of fg is slightly more involved: one need to take $w'(t)=y_1'(t)z_1(t)+y_1(t)z_1'(t)=p_1(y(t))z_1(t)+y_1(t)p_1'(z(t))$ so that $w(t)=y(t)z(t)$. The error analysis is a bit more complicated. First note that $\|f(x)\|\leqslant 1+\Upsilon(\|x\|,\Omega(\|x\|,0))$ and $\|g(x)\|\leqslant 1+\Upsilon'(\|x\|,\Omega'(\|x\|,0))$, and denote by $\ell(\|x\|)$ and $\ell^*(\|x\|)$ those two bounds respectively. Let $t\geqslant\Omega(\|x\|,\mu+\ln 2\ell^*(\|x\|))$ then $\|y_1(t)-f(x)\|\leqslant e^{-\mu-\ln 2\|g(x)\|}$ and similarly if $t\geqslant\Omega'(\|x\|,\mu+\ln 2(1+\ell^*(\|x\|)))$ 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 $\|y_1(t)\|\leqslant 1+\|f(x)\|\leqslant 1+\ell(\|x\|)$. ◀

▶ **Theorem 55** (Closure by composition)**.** *If $f,g\in\mathrm{AP}$ and $f(\operatorname{dom} f)\subseteq\operatorname{dom} g$ then $g\circ f\in\mathrm{AP}$.*

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 both f and g are online-computable. We could show directly that $g\circ f$ is online-computable but this would only complicated the proof for no apparent gain.

Apply Lemma 13 to get that g is $(\Upsilon,\Omega,\Lambda)$-online-computable,
where

▶ **Definition 56** (Online computability)**.** Let $n,m\in\mathbb{N}$, $f:\subseteq\mathbb{R}^n\to\mathbb{R}^m$ and $\Upsilon,\Omega,\Lambda:\mathbb{R}_+^2\to\mathbb{R}_+$. We say that f is $(\Upsilon,\Omega,\Lambda)$-online-computable if and only if there exists $\delta\geqslant 0$, $d\in\mathbb{N}$ and $p\in\mathbb{K}^d[\mathbb{R}^d\times\mathbb{R}^n]$ and $y_0\in\mathbb{K}^d$ 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 $y'(t)=p(y(t),x(t))$
- $\|y(t)\|\leqslant\Upsilon\big(\sup_\delta\|x\|(t),t\big)$
- For any $I=[a,b]$, if there exists $\bar{x}\in\operatorname{dom} f$ 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 $a+\Omega(\|\bar{x}\|,\bar{\mu})\leqslant u\leqslant b$.

We denote by $\mathrm{AO}(\Upsilon,\Omega,\Lambda)$ the set of $(\Upsilon,\Omega,\Lambda)$-online-computable functions.

Apply Definition 56 to get e,Δ,z_0 for g. Assume that f is (Υ',Ω')-computable. Apply Definition 22 to get d,p,q for f. Let $x\in I$ and consider the following system:

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

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

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

Define $\bar{x}=f(x)$, $\Upsilon^*(\alpha)=1+\Upsilon'(\alpha,0)$ and $\Omega''(\alpha,\mu)=\Omega'(\alpha,\Lambda(\Upsilon^*(\alpha),\mu))+\Omega(\Upsilon^*(\alpha),\mu)$. By definition of Υ', $\|\bar{x}\|\leqslant 1+\Upsilon'(\|x\|,0)=\Upsilon^*(\|x\|)$. Let $\mu\geqslant 0$ then by definition of

Ω', if $t \geqslant \Omega'(\|x\|, \Lambda(\Upsilon^*(\|x\|), \mu))$ then $\|y_{1..m}(t) - \bar{x}\| \leqslant e^{-\Lambda(\Upsilon^*(\|x\|),\mu)} \leqslant e^{-\Lambda(\|\bar{x}\|,\mu)}$. Apply Defiintion 56 for $a = \Omega'(\|x\|, \Lambda(\Upsilon^*(\|x\|), \mu))$ to get that $\|z_{1..l}(t) - g(f(x))\| \leqslant e^{-\mu}$ for any $t \geqslant a + \Omega(\bar{x}, \mu)$. And since $t \geqslant a + \Omega(\bar{x}, \mu)$ whenever $t \geqslant \Omega''(\|x\|, \mu)$, we get that $g \circ f$ is computable. ◀

F.9.4 On Encoding and Ideal Step Function

Finaly, a remark on our selected encoding:

Recall:

▶ **Definition 57** (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 $0.x = x_1 k^{-1} + x_2 k^{-2} + \cdots + x_{|w|} k^{-|w|} \in \mathbb{Q}$.

We have:

▶ **Lemma 58** (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 considered 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 59** (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:

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

▶ **Lemma 60** ($\langle \mathcal{M}\rangle_\infty$ 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 58. ◀

F.9.5 Proof of Theorem 14

We consider the following function.

▶ **Definition 61** (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:

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

where:

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

where:

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

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

rnd* is defined in Definition 52.

We can now prove Theorem 14:

Proof Of Theorem 14. 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 58, $k0.x = n + \alpha$ 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 52 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, $L_{\delta_i}(\bar{q}, \bar{\sigma}) = \delta_i(q, \sigma)$ 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 60.
- **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 60.
- **If** $d' = R$ **then** $\text{choose}[l, s, r] = r$ and everything else is similar to the case of $d' = L$.

Finally apply Lemma 49, Theorem 53, and Theorem 54 and Theorem 55 to get that $\langle \mathcal{M} \rangle \in \text{AP}$. ◀

F.10 Proof of Theorem 15

F.10.1 Some facts

We first state some facts.

▶ **Lemma 62** (Round, [33]). *There exists* $\text{rnd} \in \text{GPVAL}$ *such that: For any* $n \in \mathbb{Z}$, $\lambda \geqslant 2$, $\mu \geqslant 0$, $|\text{rnd}(x, \mu, \lambda) - n| \leqslant \frac{1}{2}$ *for all* $x \in \left[n - \frac{1}{2}, n + \frac{1}{2}\right]$ *and* $|\text{rnd}(x, \mu, \lambda) - n| \leqslant e^{-\mu}$ *for all* $x \in \left[n - \frac{1}{2} + \frac{1}{\lambda}, n + \frac{1}{2} - \frac{1}{\lambda}\right]$.

▶ **Lemma 63** (Clamped exponential). *For any* $a, b, c, d, x \in \mathbb{R}$ *such that* $a \leqslant b$ *and* $\ell \in \mathbb{R}_+$, *define* h *as follows. Then* $h \in \text{AP}$*:*

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

F.10.2 Computing limits

Intuitively, this model of computation already contains the notion of limit. More precisely, if f is computable and is such that $f(x,t)\to g(x)$ when $t\to\infty$ then g is computable. This is just a reformulation of equivalence between computability and weak-computability. The result below extends this result to the case where the limit is restricted to $t\in\mathbb{N}$. The optimality of the assumptions is discussed in Remark F.10.2.

The idea of the proof is to show that g is weakly-computable and use the equivalence with computability. Given x and μ, we want to run f on $(x,\lceil\omega\rceil)\in I\times J$ where $\omega=\mho(\|x\|,\mu)$. Unfortunately we cannot compute the ceiling value in a continuous fashion. The trick is to run two systems in parallels: one on $(x,(\operatorname{rnd}\omega))$ and one on $(x,\operatorname{rnd}(\omega+\frac{1}{2}))$. This way one system will always have a correct input value but we must select which one. If rnd is a good rounding function around $[n-\frac{1}{3},n+\frac{1}{3}]$, we build the selecting function to pick the first system in $[n,n+\frac{1}{6}]$, a barycenter of both in $[n+\frac{1}{6},n+\frac{1}{3}]$ and the second system in $[n+\frac{1}{3},n+\frac{2}{3}]$ and so on. The crucial point is that in the region where we mix both system, both have correct inputs so the mixing process doesn't create any error. Furthermore, we can easily build such a continuous selecting function and the mixing process has already been studied in a previous section.

▶ **Theorem 64** (Closure by limit)**.** *Let $f:I\times J\subseteq\mathbb{R}^{n+1}\to\mathbb{R}^m$, $g:I\to\mathbb{R}^m$ and $\mho:\mathbb{R}_+^2\to\mathbb{R}_+$ a polynomial. Assume that $f\in\mathrm{AP}$ and that $J\supseteq\mathbb{N}$. Further assume that for all $(x,\tau)\in I\times J$ and $\mu\geqslant 0$, if $\tau\geqslant\mho(\|x\|,\mu)$ then $\|f(x,\tau)-g(x)\|\leqslant e^{-\mu}$. Then $g\in\mathrm{AP}$.*

Proof. First note that $\frac{1}{2}-e^{-2}\geqslant\frac{1}{3}$ and define for $x\in I$ and $n\in\mathbb{N}$:

$$\begin{aligned}
f_0(x,\tau)&=f(x,\operatorname{rnd}^*(\tau,2)) && \tau\in\left[n-\tfrac{1}{3},n+\tfrac{1}{3}\right]\\
f_1(x,\tau)&=f(x,\operatorname{rnd}^*(\tau+\tfrac{1}{2},2)) && \tau\in\left[n+\tfrac{1}{6},n+\tfrac{5}{6}\right]
\end{aligned}$$

By Definition 52 and hypothesis on f, both are well-defined because $\mathbb{N}\subseteq J$. Also note that their domain of definition overlap on $[n+\frac{1}{6},n+\frac{1}{3}]$ and $[n+\frac{2}{3},n+\frac{5}{6}]$ for all $n\in\mathbb{N}$. Apply Theorems 53 and 55 to get that $f_0,f_1\in\mathrm{AP}$. We also need to build the indicator function: this is where the choice of above values will prove convenient. Define for any $\tau\in\mathbb{R}_+$:

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

It is now easy to check that:

$$\begin{aligned}
\{(x,n)\,|\,i(x)<1\}&=\mathbb{R}_+\cap\cup_{n\in\mathbb{N}}\left]n-\tfrac{1}{3},n+\tfrac{1}{3}\right[\subseteq\operatorname{dom}f_0\\
\{(x,n)\,|\,i(x)>0\}&=\mathbb{R}_+\cap\cup_{n\in\mathbb{N}}\left]n+\tfrac{1}{6},n+\tfrac{5}{3}\right[\subseteq\operatorname{dom}f_1
\end{aligned}$$

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

$$f^*(x,\tau)=\operatorname{mix}(i,f_0,f_1)(x,\tau)$$

We can thus apply Theorem 77 to get that $f^*\in\mathrm{AP}$. Note that f^* is defined over $I\times\mathbb{R}_+$. We now claim that for any $x\in I$ and $\mu\in\mathbb{R}_+$, if $t\geqslant 1+\mho(\|x\|,\mu)$ then $\|f^*(x,\tau)-g(x)\|\leqslant 2e^{-\mu}$. There are three cases to consider:

- 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 $i(x) \in [0,1]$ 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 $\lfloor\tau\rceil, \lfloor\tau+\frac{1}{2}\rceil \geqslant \mho(\|x\|,\mu)$ then $\|f(x,\lfloor\tau\rceil)-g(x)\| \leqslant e^{-\mu}$ 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 $|i(x,\tau)| \leqslant 1$.

It follows that g is the limit of f^* and thus $g \in \mathrm{AWP}$ (see Remark F.12.2) and one concludes using that $\mathrm{AWP} = \mathrm{AP}$. ◀

▶ Remark (Optimality). The condition that $\mho$ be a polynomial is essentially optimal. Intuitively, if $f \in \mathrm{AP}$ and satisfies that $\|f(x,\tau)-g(x)\| \leqslant e^{-\mu}$ whenever $\tau \geqslant \mho(\|x\|,\mu)$ then $\mho$ is a modulus of continuity for g. By Theorem 7, if $g \in \mathrm{AP}$ then it admits a polynomial modulus of continuity so $\mho$ must be a polynomial. For a formal proof of this intuition, see examples 65 and 66

▶ **Example 65** ($\mho$ 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 $g \notin \mathrm{AP}$ because it is not polynomially bounded. In this case $\mho(x,\mu) = e^x$ which is exponential and $f \in \mathrm{AP}$ by Lemma 63.

▶ **Example 66** ($\mho$ 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 $g \notin \mathrm{AP}$, indeed its modulus of continuity is exponential around 0 because $g(e^{-e^{\mu}}) = e^{-\mu}$ for all $\mu \geqslant 0$. However note that $g^* \in \mathrm{AP}$ 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 $f \in \mathrm{AP}$ because $(x,\tau) \mapsto \min(-\ln x,\tau) \in \mathrm{AP}$ by a proof similar to Lemma 63.

F.10.3 Proof of Theorem 15

We now go to the Proof of Theorem 15.

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.

The 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 mentionning is that the entire reasoning makes sense because the assumption ensures that $x^{(k)} \in I$ at each step.

Formally, apply Lemma 13 to get that $f \in \mathrm{AX}(\Upsilon,\Omega,\Lambda,\Theta)$ where $\Upsilon,\Lambda,\Theta,\Omega$ are polynomials. Without loss of generability we assume that $\Upsilon,\Lambda,\Theta,\mho$ and Π are increasing functions. Apply Lemma 45 to get $\omega \geqslant 1$ such that for all $\alpha \in \mathbb{R}, \mu \in \mathbb{R}_+$:

$$\Omega(\alpha,\mu) = \omega \geqslant 1$$

Apply Definition 42 to get δ, d and g. Define:

$$\tau = \omega + 2$$

We will show that $f_0^* \in \mathrm{AWP} = \mathrm{AP}$: 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 29 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 sytem during period $[i\tau, (i+1)\tau]$.

- **For y and w, if** $t \in [i\tau, i\tau + \omega + 1]$ **then** apply Lemma 71 to get that $\operatorname{hxl} \in [0, 1]$ and Lemma 43 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 $\left\|x^{(i)}\right\| \leqslant \left\|x^{[i]}\right\| + 1 \leqslant 1 + \Pi(\|x\|, i) \leqslant \ell^*$. 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 $\Omega\left(\left\|x^{(i)}\right\|, \nu\right) = \omega$.
- **For z, if** $t \in [i\tau + \omega, i\tau + \omega + 1]$ **then** apply Lemma 43 to get that $\left\|f(x^{(i)}) - z(i\tau + \omega + 1)\right\| \leqslant 2e^{-\nu}$.
- **For z and w, if** $t \in [i\tau + \omega + 1, i\tau + \omega + 2]$ **then** apply Lemma 43 to get that $\|z'(t)\| \leqslant e^{-\nu}$ thus $\left\|f(x^{(i)}) - z(t)\right\| \leqslant 3e^{-\nu}$. Apply Lemma 71 to get that $\left\|y'(t) - \operatorname{sample}_{[\omega+1,\omega+2],\tau}(t, \nu^* + \ln(1+\omega), w(t), z(t))\right\|$ $e^{-\nu - t}$. Apply Lemma 43 again to get that $\left\|f(x^{(i)}) - w(i\tau + \omega + 2)\right\| \leqslant 4e^{-\nu} + e^{-\nu^*} \leqslant 5e^{-\nu}$.

Our analysis concluded that $\left\|f(x^{(i)}) - z((i+1)\tau)\right\| \leqslant 5e^{-\nu}$. Also, by hypothesis, $\left\|x^{(i)} - x^{[i]}\right\| \leqslant e^{-(n-i)\mho(\ell^*) - (n-i)\ln 6 - \mu - \ln 3} \leqslant e^{-\mho\left(\left\|x^{[i]}\right\|\right) - \mu^*}$ where $\mu^* = (n-i-1)\mho(\ell^*) + (n-i)\ln 6 + \mu + \ln 3$ because $\left\|x^{[i]}\right\| \leqslant \ell^*$. Consequently, $\left\|f(x^{(i)}) - x^{[i+1]}\right\| \leqslant e^{-\mu^*}$ 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 $\left\|x^{(n)} - x^{[n]}\right\| \leqslant e^{-\mu - \ln 3}$. We still have to analyze the behavior after time $n\tau$.

- **If** $t \in [n\tau, n\tau + 1]$ **then** apply Lemmas 43 and 71 to get that $\|w'(t)\| \leqslant e^{-\nu^* - \ln(1+\omega)}$ thus $\left\|w(t) - x^{(n)}\right\| \leqslant e^{-\nu^* - \ln(1+\omega)}$.

- **If** $t \geqslant n\tau+1$ **then** apply Lemma 71 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 to 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 $\|w(t)\| \leqslant \left\|x^{[n]}\right\|+1 \leqslant \Pi(\|x\|, n)+1$ which is polynomially bounded. The bound on y comes from Definition 42:

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

And finally, apply Lemma 43 to get that:

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

This conclude the proof that $f_0^* \in \mathrm{AWP}$.

We will now tackle the case of $\eta > 0$. Let $\eta \in]0, \frac{1}{2}[$ and define $g_\eta(x,\mu) = \operatorname{rnd}(x, \mu, \frac{1}{2}-\eta)$ for $x \in \mathbb{Z}+]-\eta, \eta[$. Apply Lemma 62 to get that rnd $\in$ GPVAL and Theorem 28 to get[17] that $g_\eta \in \mathrm{AP}$. By definition, $\|g_\eta(x,\mu)-n\| \leqslant e^{-\mu}$ if $x \in [n-\eta, n+\eta]$ thus we can apply Theorem 64 to get that $g_\eta^*(x) = \lim_{\mu\to\infty} g_\eta(x,\mu)$ belongs to AP and $g_\eta^*(x) = n$ for any $x \in [n-\eta, n+\eta]$. Now define $f_\eta^*(x,u) = f_0^*(x, g_\eta^*(u))$ and apply Theorem 55 to conclude. As a final remark, note that g_η^* is a pretty good rounding function but we can do much better: see Theorem 53 for more details.

▶ Remark (Optimality of growth constraint). It is easy to see that without any condition, the iterates can produce an exponential function. Pick $f(x) = 2x$ then $f \in \mathrm{AP}$ and $f^{[n]}(x) = 2^n x$ which is clearly not polynomial in x and n. More generally, by Lemma 74, it is necessary that f^* be polynomially bounded so clearly $f^{[n]}(x)$ must be polynomially bounded in $\|x\|$ and n.

▶ Remark (Optimality of modulus constraint). Without any constraint, it is easy to build an iterated function with exponential modulus of continuity. Define $f(x) = \sqrt{x}$ then $f \in \mathrm{AP}$ 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 (Domain of definition). Intuitively we could 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 $f^* \restriction_I \notin \mathrm{AP}$ as we saw in Remark F.10.3. But in reality, the problem is more subtle than that because if I is open but the neighbourhood 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 $f^* \restriction_I \notin \mathrm{AP}$ for the same reason as before.

[17] Although this is a forward reference, the proof does not relies on the iteration of functions

▶ Remark (Classical error bound). The third condition in Theorem 15 is usually far more subtle than necessary. In practice, is it useful to note this condition is satisfied in f verifies for some constants $\varepsilon, K > 0$ that

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

▶ Remark (Dependency of $\mho$ in n). In the statement of thereom, $\mho$ 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 $\|x-y\| \leqslant e^{-\mho(\|x\|,n)-\mu}$. The proof is straightfoward:

F.11 Proof of Theorem 2

F.11.1 FP iff emulable

In this subsection, we fix an alphabet Γ and all languages are considered over Γ, so in particular $P \subset \Gamma^*$. 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 $\gamma : \Gamma \to \mathbb{N}^*$, in other words every letter has an uniquely assigned positive number. By extension, γ applies letterwise over words.

We start by proving that functions of FP (i.e. computable in polynomial time) are emulable and conversely.

Before, we state that the following lemma can be proved:

▶ **Lemma 67** (Size recovery, [33]). *For any machine $\mathcal{M}$, there exists a function* ($\mathrm{tsize}_{\mathcal{M}} : \langle\mathcal{C}_{\mathcal{M}}\rangle \times \mathbb{N} \to \mathbb{N}) \in \mathrm{AP}$ *such that for any word $w \in (\Sigma \setminus \{b\})^*$ and any $n \geqslant |w|$, the size of the tape satisfies* $\mathrm{tsize}_{\mathcal{M}}(0.w, n) = |w|$.

▶ **Definition 68** (Discrete emulation). $f : \Gamma^* \to \Gamma^*$ is called *emulable* if there exists $g \in \mathrm{AP}$ and $k \geqslant 1 + \max(\gamma(\Gamma))$ such that for any word $w \in \Gamma^*$:

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

We say that g *emulates* f with k.

▶ **Theorem 69** (FP equivalence). *$f \in \mathrm{FP}$ if and only if f is emulable (with $k = 2 + \max(\gamma(\Gamma))$).*

Proof. 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 $\psi(w) = (0.\gamma(w), |w|)$ for any word $w \in \Gamma^*$.

Let $\mu = \ln(4k^2)$ and $h(c) = \mathcal{M}(c, \mu)$ 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 that $\varepsilon_0 \leqslant \frac{1}{2k^2} - e^{-\mu}$. Apply Theorem 14 to get that $h \in \mathrm{AP}$ 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 15 to get $(h^* : J \to I_0) \in \mathrm{AP}$ such that for all $c \in I_\infty$ and $n \in \mathbb{N}$ and $h^*(c, n) = h^{[n]}(c)$.

Let π_3 denote the third projection, that is $\pi_3(a, b, c, d) = c$, then $\pi_3 \in \mathrm{AP}$. Define $g(y, \ell) = \pi_3(h^*(0, b, y, q_0, p_{\mathcal{M}}(\ell)))$ for $y \in \psi(\Gamma^*)$ and $\ell \in \mathbb{N}$. Note that $g \in \mathrm{AP}$ and is well-defined. Indeed, if $\ell \in \mathbb{N}$ then $p_{\mathcal{M}}(\ell) \in \mathbb{N}$ and if $y = \psi(w) = 0.w$ then $(0, b, y, q_0) = \langle(\lambda, b, w, q_0)\rangle = \langle c_0(w)\rangle \in I_\infty$. Furthermore, by construction, for any word $w \in \Gamma^*$ we have:

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

Furthermore, the size of the tape cannot be greater than the initial size plus the number of steps, thus $|f(w)| \leqslant |w| + p_{\mathcal{M}}(|w|)$. Apply Lemma 67 to get that $\operatorname{tsize}_{\mathcal{M}}(g(\psi(w), |w|), |w| + p_{\mathcal{M}}(|w|)) = |f(w)|$ 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 $g \in \mathrm{AP}$ and the tape size $\operatorname{tsize}_{\mathcal{M}} \in \mathrm{AP}$.

Conversely, assume that f is emulable and apply Definition 68 to get $g \in \mathrm{AC}(\Upsilon, \Omega)$ where Υ, Ω are polynomials, and $k \in \mathbb{N}$. Let $w \in \Gamma^*$: we will describe an FP algorithm to compute $f(w)$. Apply Definition 22 to g to get d, p, q and consider the following system:

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

Note that by construction, y is defined over $\mathbb{R}_+$. Also note[18] that the coefficients of p, q belong to $\mathbb{R}_P$ which means that they are polynomial time computable. And since $\psi(w)$ is a pair of rational numbers with polynomial size (with respect to $|w|$), then $q(\psi(w)) \in \mathbb{R}_P^d$.

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

Let $t_w = \Omega(|w|, 2)$ for any $w \in \Sigma^*$, note that $t_w \in \mathbb{R}_P$ and that it is polynomially bounded in $|w|$ because Ω is a polynomial. Apply Theoorem 8 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 $\operatorname{len}_y(0, t_w) \leqslant \operatorname{poly}(t_w, \sup_{[0,t_w]}\|y\|)$ and by construction, $\|y(t)\| \leqslant \Upsilon(\|\psi(w)\|, t_w)$ for $t \in [0, t_w]$ where Υ is a polynomial. Furthermore, by definition $\|y(t_w) - g(\psi(w))\| \leqslant e^{-2}$ thus $\|\tilde{y} - \psi(f(w))\| \leqslant 2e^{-2} \leqslant \frac{1}{3}$. But since $\psi(f(w)) = (0.\gamma(f(w)), |f(w)|)$, from $\tilde{y}_2$ we can find $|f(w)|$ by rounding to the closest integer (which is unique because at 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 $t'_w = \Omega(|w|, 2 + |f(w)| \ln k)$ which is polynomial in $|w|$ because Ω is a polynomial and $|f(w)|$ is at most polynomial in $|w|$. We can use the same reasoning and apply Theorem 8 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, $\|\tilde{y}_1 - 0.\gamma(f(w))\| \leqslant 2e^{-2-|f(w)|\ln k} \leqslant \frac{1}{3}k^{-|f(w)|}$. 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. ◀

[18] and that is absolutely crucial

An question arises when looking at this theorem: does the choice of k in Definition 68 matters, especially for the equivalence with FP ? Fortunately not, as long as k is large enough, as shown in the next lemma.

▶ **Lemma 70** (Emulation reencoding, [33]). *Assume that $g \in \mathrm{AP}$ emulates f with $k \in \mathbb{N}$. Then for any $k' \geqslant k$, there exists $h \in \mathrm{AP}$ that emulates f with k'.*

F.11.2 Proof of Theorem 2

We start by some technical lemma.

▶ **Lemma 71** ("low-X-high" and "high-X-low", [33]). *There exists $\mathrm{lxh}_I, \mathrm{hxl}_I \in \mathrm{GPVAL}$ such that: Let $I = [a, b]$, $\mu \in \mathbb{R}_+$, then $\forall t, x \in \mathbb{R}$:*

- *$\exists \phi_1, \phi_2$ such that $\mathrm{lxh}_I(t, \mu, x) = \phi_1(t, \mu, x)x$ and $\mathrm{hxl}_I(t, \mu, x) = \phi_2(t, \mu, x)x$*
- *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 $|\mathrm{hxl}_I(t, \mu, x)| \leqslant |x|$*

The proof is based on the equivalence between AP and ALP, and the FP equivalence. Indeed, decidability can be seen as the computability of particular functions with boolean output. The only technical point is to make sure that the decision of the system is irreversible. To do that, we run the system from the FP equivalence (which will output 0 or 1) for long enough so that the output is approximate but good enough. Only then will another variable reach -1 or 1. The fact that the decision complexity is based on the length of the curve also makes the proof slightly more complicated because the system we build essentially takes a decision after a certain time (and not length).

Let $\mathcal{L} \in \mathrm{P}$, then there exists $f \in \mathrm{FP}$ 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 $\mathrm{dec}(k^{-1}\gamma(\bar{0})) = -2$ and $\mathrm{dec}(k^{-1}\gamma(\bar{1})) = 2$. Recall that $L_{\mathrm{dec}} \in \mathrm{AP}$ by Lemma 49. Apply Theorem 69 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 $g^*(x) = L_{\mathrm{dec}}(g_1(x))$ and check that $g^* \in \mathrm{AP}$. Furthermore, $g^*(\psi(w)) = 2$ if $w \in \mathcal{L}$ and $g^*(\psi(w)) = -2$ otherwise, by definition of the emulation and the interpolation. Let Ω and Υ be polynomials such that $g^* \in \mathrm{AC}(\Upsilon, \Omega)$ and assume, without loss of generality, that they are increasing functions. Apply Definition 22 to get d, p, q. Let $w \in \Gamma^*$ and consider the following system:

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

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

In this system, y computes g^* f, v is a constant variable used to store the input and in particular the input size ($v_2(t) = |w|$), $\tau(t) = t$ is used to keep the time and z is the decision variable. Let $t \in [0, \tau^*]$, then by Lemma 71, $\|z'(t)\| \leqslant e^{-1-t}$ 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 Ω and since $v_2(t) = \psi_2(w) = |w| = \|\psi(w)\|$, $\|y_1(t) - g^*(\psi(w))\| \leqslant e^{-\ln 2}$. Recall that $g^*(\psi(w)) \in \{-2, 2\}$ and let $\varepsilon \in \{-1, 1\}$ such that $g^*(\psi(w)) = \varepsilon 2$. 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 71 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 $|z(t)|$ 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 $z(t^*) \leqslant 1$. Also note that $\|(y, v, z, w)'(t)\| \geqslant 1$ for all $t \geqslant 1$. Also note that z is bounded by a constant, by a very similar reasoning. This shows that if $Y = (y, v, z, \tau)$, then $\|Y(t)\| \leqslant \text{poly}(\|\psi(w)\|, t)$ because $\|y(t)\| \leqslant \Upsilon(\|\psi(w)\|, t)$. 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 $t \leqslant \text{len}_Y(0, t) \leqslant t \sup_{u\in[0,t]} \|Y'(u)\| \leqslant t\Upsilon^*(\|\psi(w)\|, t)$. Define $\Omega^*(\alpha) = t^*\Upsilon^*(\alpha, t^*)$ which is a polynomial becase t^* is polynomially bounded in $\|\psi(w)\| = |w|$. Let t such that $\text{len}_Y(0, t) \geqslant \Omega^*(|w|)$, then by the above reasoning, $t\Upsilon^*(|w|, t) \geqslant \Omega^*(|w|)$ and thus $t \geqslant t^*$ so $|z(t)| \geqslant 1$, i.e. the system has decided.

F.12 Proof of Theorem 1

F.12.1 FP iff emulable: extension to multiple inputs/outputs

The equivalence between FP and the fact of beeing emulable has been proved in Theorem 69 for single input function, 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. For what follows, it will be useful to have function with multiple inputs/ouputs without going through an encoding. We extend the notion of discrete encoding in the natural way to handle this case.

▶ **Definition 72** (Discrete emulation). $f : (\Gamma^*)^n \to (\Gamma^*)^m$ is called *emulable* if there exists $g \in \text{AP}$ and $k \in \mathbb{N}$ such that for any word $\vec{w} \in (\Gamma^*)^n$:

$$g(\psi(\vec{w})) = \psi(f(\vec{w})) \qquad \text{where} \quad \psi(x_1, \ldots, x_\ell) = (\psi(x_1), \ldots, \psi(x_\ell))$$

and ψ is defined as in Definition 68.

▶ Remark (Consistency). It is trivial that Definition 72 matches Definition 68 in the case of unidimensional functions, thus the two definitions are consistent with each other.

▶ **Theorem 73** (Multidimensional FP equivalence). *Let $f : (\Gamma^*)^n \to (\Gamma^*)^m$. Then $f \in \mathrm{FP}$ if and only if f is emulable.*

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[19] that

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

Assume that $f \in \mathrm{FP}$, then there exists $h \in \mathrm{FP}$ such that $h^{\#} = f$. Note that h naturally induces a function (still called) $h : {\Gamma^{\#}}^* \to {\Gamma^{\#}}^*$ so we can apply Theorem 69 to get that h is emulable over alphabet $\Gamma^{\#}$. Apply Definition 68 to get $g \in \mathrm{AP}$ and $k \in \mathbb{N}$ that emulate h. In the remaining of the proof, ψ denotes encoding of Definition 68 for this particular k, in other words:

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

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 $\varphi \in \mathrm{AP}$ and that for any $w, w' \in \Gamma^*$, $\varphi(\psi(w), \psi(w')) = \psi(w\#w')$. The fact that $\varphi \in \mathrm{AP}$ is immediate using Theorem 54 and the fact that $n \mapsto k^{-n-1}$ is analog-polytime-computable[20]. The second fact is follows from a calculation:

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

Define $G = g \circ \varphi$, we claim that G emulates f with k. First $G \in \mathrm{AP}$ thanks to Theorem 55. Second, for any $w, w' \in \Gamma^*$, we have:

$$G(\psi(w, w')) = g(\varphi(\psi(w), \psi(w'))) \qquad \text{By definition of } G \text{ and } \psi$$

[19] This is folklore, but mostly because this particular encoding of pairs is polytime computable.
[20] Note that it works only because $n \geqslant 0$.

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

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 $F_1, F_2 \in \mathrm{FP}$ so apply Theorem 69 to get that they are emulable. Thanks to Lemma 70, 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 $H \in \mathrm{AP}$ because $g_1, g_2, h \in \mathrm{AP}$. Furthermore, H emulates $f \circ F$ because for any $w \in {\Gamma^{\#}}^*$:

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

Since $f \circ F : {\Gamma^{\#}}^* \to {\Gamma^{\#}}^*$ is emulable, we can apply Theorem 69 to get that $f \circ F \in \mathrm{FP}$. It is now trivial so see that $f \in \mathrm{FP}$ because for any $w, w' \in \Gamma^*$:

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

and $((w, w') \mapsto w\#w') \in \mathrm{FP}$ ◀

F.12.2 Some facts

We need the following facts.

▶ **Lemma 74.** *Let $f \in \mathrm{AP}$, there exists a polynomial P such that $\|f(x)\| \leqslant P(\|x\|)$ for all $x \in \operatorname{dom} f$.*

Proof. Assume that $f \in \mathrm{AC}(\Upsilon, \Omega)$ and apply Definition 22 to get d, p, q. Let $x \in \operatorname{dom} f$ and let y be the solution of $y(0) = q(x)$ and $y' = p(y)$. Apply the definition to get that $\|f(x) - y_{1..m}(\Omega(\|x\|, 0))\| \leqslant 1$ and $\|y(\Omega(\|x\|, 0))\| \leqslant \Upsilon(\|x\|, \Omega(\|x\|, 0)) \leqslant \operatorname{poly}(\|x\|)$ since Υ and Ω are polynomials. ◀

▶ **Theorem 75** (Extraction, [33]). *There exists* extract $\in \mathrm{AP}$ *such that for any $x \in \mathbb{R}$ and $n \in \mathbb{N}$:*

$$\operatorname{extract}(x, n) = \cos(2\pi 2^n x)$$

Notice that the proof of above theorem is obtained using an iteration: See [33].

▶ **Definition 76** (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 $\{x \mid i(x) < 1\} \subseteq \operatorname{dom} f_0$ and $\{x \mid i(x) > 0\} \subseteq \operatorname{dom} f_1$, and define for $x \in \operatorname{dom} i$:

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

▶ **Theorem 77** (Closure by mixing, [33]). *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 $f_0, f_1, i \in \mathrm{AP}$, that $\{x \mid i(x) < 1\} \subseteq \operatorname{dom} f_0$ and that $\{x \mid i(x) > 0\} \subseteq \operatorname{dom} f_1$. Then $\operatorname{mix}(i, f_0, f_1) \in \mathrm{AP}$.*

▶ Remark (Limit computability). A careful look at Definition 34 shows that analog weak computability is a form of 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 $f \in \mathrm{AP}$ and that for any $x \in I$ and $\tau \in \mathbb{R}_+^*$, if $\tau \geqslant \mho(\|x\|, \mu)$ then $\|f(x, \tau) - f(x)\| \leqslant e^{-\mu}$. Then $g \in \mathrm{AWP}$ because the analog system for f satisfies all the items of the definition.

▶ **Theorem 78** (Word decoding, [33]). *Let $k_1, k_2 \in \mathbb{N}^*$ and $\kappa : [\![0, k_1 - 1]\!] \to [\![0, k_2 - 1]\!]$. There exists a function $(\operatorname{decode}_\kappa :\subseteq \mathbb{R} \times \mathbb{N} \times \mathbb{R} \to \mathbb{R}) \in \mathrm{AP}$ such that for any word $w \in [\![0, k_1 - 1]\!]^*$ and $\mu, \varepsilon \geqslant 0$:*

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

▶ **Lemma 79** (Reencoding). *Let $k_1, k_2 \in \mathbb{N}^*$ and $\kappa : [\![1, k_1 - 2]\!] \to [\![0, k_2 - 1]\!]$. There exists a function $(\operatorname{reenc}_\kappa :\subseteq \mathbb{R} \times \mathbb{N} \to \mathbb{R} \times \mathbb{N}) \in \mathrm{AP}$ 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 78 with $\varepsilon = 0$ to get the result. Note that stricly 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 so its length is the same as the number of non-blank symbols it contains. ◀

F.12.3 Proof of Theorem 1

Assume that the theorem is true for functions in $C^0([0, 1/2])$, then we claim the theorem follows. Indeed, if $f \in C^0([a, b], \mathbb{R})$ is polynomial time computable, then there exists[21] $m, M \in \mathbb{R}_P$ such that $m < f(x) < M$ for all $x \in [a, b]$. Define for $\alpha \in [0, 1/2]$:

$$g(\alpha) = \frac{f(a + 2\alpha(b - a)) - m}{2(M - m)}$$

[21] To see that, observe that any polytime computable function is bounded by a polynomial.

then clearly $g \in C^0([0,1/2])$ and is polytime computable because $a, b, m, M \in \mathbb{R}_P$. It follows that $g \in \mathrm{AP}$ and then $f \in \mathrm{AP}$ by the closure properties of AP. Conversely, if $f \in C^0([0,1/2])$ belongs to AP then there also exists[22] $m, M \in \mathbb{R}_P$ as above and the reasoning is exactly the same. In the remaining of the proof, we assume that $f \in C^0([0,1/2])$. This restriction is useful to simplify the encoding used later in the proof.

Let $f \in C^0([0,1/2])$ be a polynomial time computable function. From classical recursive analysis arguments, there exists a computable (resp. polynomial time computable[23]) function $g : (\mathbb{Q} \cap [a,b]) \times \mathbb{N} \to \mathbb{Q}$ 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 $d \in [a,b] \cap \mathbb{Q}$, $|g(d,n) - f(d)| \leqslant 2^{-n}$

Note that $g : \mathbb{Q} \cap [0,1/2] \times \mathbb{N} \to \mathbb{Q} \cap [0,1/2]$ 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 $\Gamma = \{0,1\}$ with $\gamma(0) = 1$ and $\gamma(1) = 2$ and define for any $w, w' \in \Gamma^*$:

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

Note that $\psi_{\mathbb{Q}}$ is a bijection from Γ^* to $\mathbb{Q} \cap [0,1[$. Define for any $w, w' \in \Gamma^*$:

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

Since ψ_Q is a polytime computable encoding, then $g_\Gamma \in \mathrm{FP}$ because it has running time polynomial in the size of $\psi_Q(w)$ and the (unary) value of $\psi_{\mathbb{N}}(w')$, which are the size of w and w' respectively, by definition of $\psi_{\mathbb{Q}}$ and $\psi_{\mathbb{N}}$. Apply Theorem 73 to get that g_Γ is emulable. Thus there exits $h \in \mathrm{AP}$ and $k \in \mathbb{N}$ such that for all $w, w' \in \Gamma^*$:

$$h(\psi(w,w')) = \psi(g_\Gamma(w,w'))$$

where ψ is defined as in Definition 72, for this specific value of k. Define $\kappa : [\![0, k-2]\!] \to \{0,1\}$ by $\kappa(\gamma(0)) = 0$ and $\kappa(\gamma(1)) = 1$ and $\kappa(\alpha) = 0$ otherwise, and define:

$$\psi_{\mathbb{Q}}^*(x,n) = \mathrm{reenc}_{\kappa,1}(x,n)$$

It follows from Lemma 79 that $\psi_{\mathbb{Q}}^* \in \mathrm{AP}$ and:

$$\psi_{\mathbb{Q}}^*(\psi(w)) = \mathrm{reenc}_{\kappa,1}\left(\sum_{i=1}^{|w|} \gamma(w_i) k^{-i}, |w|\right) = \sum_{i=1}^{|w|} \kappa(\gamma(w_i)) 2^{-i} = \sum_{i=1}^{|w|} w_i 2^{-i} = \psi_{\mathbb{Q}}(w)$$

We can now define:

$$g_\Gamma^*(x,n,x',n') = \psi_{\mathbb{Q}}^*(h(x,n,x',n')$$

and get that for any $w, w' \in \Gamma^*$:

$$g_\Gamma^*(\psi(w,w')) = \psi_{\mathbb{Q}}^*(h(\psi(w,w'))) = \psi_{\mathbb{Q}}^*(\psi(g_\Gamma(w,w')) = \psi_{\mathbb{Q}}(g_\Gamma(w,w')) = g(\psi_{\mathbb{Q}}(w), \psi_{\mathbb{N}}(w'))$$

Let us summarize what we have done so far: we built $g_\Gamma^* \in \mathrm{AP}$ that, if provided with the encoding of w, w', compute $g(\psi_{\mathbb{Q}}(w), \psi_{\mathbb{N}}(w'))$. To use this function, we need to be able to

[22] By Lemma 74, functions in AP are bouned by a polynomial.
[23] The second argument of g must be in unary.

compute, from the input $x \in [0,1]$ and the requested precision $\mu \geqslant 0$, words w, w' such that $|w'| \geqslant \mu$ and $|x - \psi_{\mathbb{Q}}(w)| \leqslant 2^{-m(\psi_{\mathbb{N}}(w'))}$ so that we can run g_Γ^* and get an approximation of $f(x) \pm 2^{-\mu}$. The problem is that for continuity reason, it is impossible to compute such w, w' in general. This is where mixing comes into play: given x and μ, we will compute two pairs w, w' and u, u' such that at least one of them satisfies the above criteria. We will then apply g_Γ^* on both of them and mix the result.

Define[24] $\iota : \Gamma \to [\![0, k-1]\!]$ by $\iota = \gamma$. Apply Theorems 78 and 75 to get $\text{decode}_\iota, \text{extract} \in \text{AP}$. Define for any $n \in \mathbb{N}$:

$$u(n) = \left(\tfrac{1-k^{-n}}{k-1}, n\right)$$

Clearly $u \in \text{AP}$ and one checks that $u(n) = \psi(0^n)$ because:

$$\psi(0^n) = \left(\sum_{i=1}^{n} \gamma(0) k^{-i}, n\right) = \left(\tfrac{1}{k}\tfrac{1-k^n}{1-k}, n\right) = u(n)$$

Now define for any $n \in \mathbb{N}$ and relevant[25] $x \in [0,1]$:

$$v(x,n) = (\text{decode}_{\iota,1}(x, n, 2), n)$$

It follows from Theorem 78 and the fact that $1 - e^{-2} \geqslant \frac{2}{3}$ that:

$$\text{if } x = \psi_{\mathbb{Q}}(w) + \varepsilon \text{ for some } w \in \Gamma^n \text{ and } \varepsilon \in \left[0, 2^{-n}\tfrac{2}{3}\right] \text{ then } v(x,n) = \psi(w)$$

Now define for any $n \in \mathbb{N}$ and relevant $x \in [0,1]$:

$$\begin{aligned} f_0(x,n) &= g_\Gamma^*(v(x,n), u(n)) \\ f_1(x,n) &= g_\Gamma^*\left(v\left(x + 2^{-n-1}, n\right), u(n)\right) \\ i(x,n) &= \tfrac{1}{2} + \text{extract}\left(x + 2^{-n}\tfrac{1}{6}, n\right) \end{aligned}$$

From the domain of definition of v, it follows that:

$$\bigcup_{w \in \Gamma^n} \left[\psi_{\mathbb{Q}}(w), \psi_{\mathbb{Q}}(w) + 2^{-n}\tfrac{2}{3}\right] \subseteq \text{dom } f_0$$

$$\bigcup_{w \in \Gamma^n} \left[\psi_{\mathbb{Q}}(w) - 2^{-n-1}, \psi_{\mathbb{Q}}(w) + 2^{-n}\tfrac{1}{6}\right] \subseteq \text{dom } f_1$$

First off, check that for any $n \in \mathbb{N}$:

$$\bigcup_{w \in \times\Gamma^n} \left[\psi_{\mathbb{Q}}(w), \psi_{\mathbb{Q}}(w) + 2^{-n}\right[= [0, 1[$$

Check that for any $n \in \mathbb{N}$, $\varepsilon \in [0, 2^{-n}[$, $n \in \mathbb{N}$ and $w \in \Gamma^n$:

$$i(\psi_{\mathbb{Q}}(w) + \varepsilon, n) = \tfrac{1}{2} + \cos(2\pi 2^n \varepsilon + \tfrac{\pi}{3})$$

It follows that any $n \in \mathbb{N}$, $\varepsilon \in [0, 2^{-n}[$, $w \in \Gamma^n$ and $x = \psi_{\mathbb{Q}}(w) + \varepsilon$ we have:

$$\varepsilon \in \left[0, 2^{-n}\tfrac{1}{6}\right[\qquad \Rightarrow \qquad i(x,n) \in [0, 1[$$

[24] This is a technicality because decode_ι will encode the output in basis k if $\iota : \Gamma \to [\![0, k-1]\!]$.
[25] We will discuss the domain of definition of v right after.

$$\begin{aligned}\varepsilon \in \left[2^{-n}\tfrac{1}{6}, 2^{-n}\tfrac{1}{2}\right] &\quad\Rightarrow\quad i(x,n) \leqslant 0\\ \varepsilon \in \left]2^{-n}\tfrac{1}{2}, 2^{-n}\tfrac{2}{3}\right[&\quad\Rightarrow\quad i(x,n) \in]0,1[\\ \varepsilon \in \left[2^{-n}\tfrac{2}{3}, 2^{-n}\right[&\quad\Rightarrow\quad i(x,n) \geqslant 1\end{aligned}$$

Thus:

$$\{(x,n) \mid i(x,n) < 1\} \subseteq \operatorname{dom} f_0 \qquad\qquad \{(x,n) \mid i(x,n) > 0\} \subseteq \operatorname{dom} f_1$$

Define for any $x \in [0,1/2]$ and $n \in \mathbb{N}$:

$$g^*(x,n) = \operatorname{mix}(i, f_0, f_1)(x,n)$$

We can thus apply Theorem 77 to get that $g^* \in \mathrm{AP}$. Note that g^* is defined over $[0,1[\times\mathbb{N}$ which obviously contains $[0,1/2] \times \mathbb{N}$. We will now see that g^* approximates f and conclude that $f \in \mathrm{AWP}$. To do so, we will show the following statement by a case analysis, for all $x \in [0,1/2]$ and $n \in \mathbb{N}$:

$$\exists y,z \in \mathbb{Q}\cap[0,1/2], \alpha \in [0,1], |x-y|, |x-z| \leqslant 2^{-n} \text{ and } g^*(x,n) = \alpha g(y,n) + (1-\alpha) g(z,n)$$

To see that, first note that there exists[26] $w \in \Gamma^n$ such that[27] $x \in [\psi_{\mathbb{Q}}(w), \psi_{\mathbb{Q}}(w) + 2^{-n}]$. Furthermore, since $x \in [0,1/2]$, we can always assume that $w \in \{0\} \times \Gamma^{n-1}$. Write $\varepsilon = x - \psi_{\mathbb{Q}}(w)$, then there are four possible cases. It will be useful to keep in mind that $u(n) = \psi(0^n)$ as shown previously and that $\psi_{\mathbb{N}}(0^n) = n$. Also remember that we showed that $g^*_\Gamma(\psi(w,w')) = g(\psi_{\mathbb{Q}}(w), \psi_{\mathbb{N}}(w'))$. In almost all cases, we will define $y = \psi_{\mathbb{Q}}(w)$ and thus $|x-y| = \varepsilon \leqslant 2^{-n}$.

- **If** $\varepsilon \in \left[0, 2^{-n}\frac{1}{6}\right[$ **then** $i(x,n) \in [0,1[$ thus $g^*(x,n) = i(x,n) f_0(x,n) + (1-i(x,n)) f_1(x,n)$. By construction of v, $v(x,n) = \psi(w)$ thus $f_0(x,n) = g^*_\Gamma(\psi(w), \psi(0^n)) = g^*_\Gamma(\psi(w,0^n)) = g(\psi_{\mathbb{Q}}(w), \psi_{\mathbb{N}}(0^n)) = g(y,n)$. Since, $x + 2^{-n-1} - \psi_{\mathbb{Q}}(w) \in [0, 2^{-n}\frac{2}{3}]$ we similarly have $v(x + 2^{-n-1}, n) = \psi(w)$ and thus $f_1(x,n) = f_0(x,n) = g(y,n)$. It follows that $g^*(x,n) = g(y,n)$. So in this case, $z = y \in [0,1/2]$ and α can be anything.
- **If** $\varepsilon \in \left[2^{-n}\frac{1}{6}, 2^{-n}\frac{1}{2}\right]$ **then** $i(x,n) \leqslant 0$ thus $g^*(x,n) = f_0(x,n)$. By construction of v, $v(x,n) = \psi(w)$ thus $f_0(x,n) = g^*_\Gamma(\psi(w), \psi(0^n)) = g^*_\Gamma(\psi(w,0^n)) = g(\psi_{\mathbb{Q}}(w), \psi_{\mathbb{N}}(0^n)) = g(y,n)$. It follows that $g^*(x,n) = g(y,n)$. So in this case, $z = y \in [0,1/2]$ and α can be anything.
- **If** $\varepsilon \in \left]2^{-n}\frac{1}{2}, 2^{-n}\frac{2}{3}\right[$ **then** $i(x,n) \in [0,1[$ thus $g^*(x,n) = i(x,n) f_0(x,n) + (1-i(x,n)) f_1(x,n)$. By construction of v, $v(x,n) = \psi(w)$ thus $f_0(x,n) = g^*_\Gamma(\psi(w), \psi(0^n)) = g^*_\Gamma(\psi(w,0^n)) = g(\psi_{\mathbb{Q}}(w), \psi_{\mathbb{N}}(0^n)) = g(y,n)$. However, $x + 2^{-n-1} - \psi_{\mathbb{Q}}(w) \in [2^{-n}, 2^{-n}(1+\frac{1}{6})]$. Thus define $w' \in \Gamma^n$ such that[28] $\psi_{\mathbb{Q}}(w') = \psi_{\mathbb{Q}}(w) + 2^{-n}$ and define $z = \psi_{\mathbb{Q}}(w')$. It follows that $x - \psi_{\mathbb{Q}}(w') \in [0, 2^{-n}\frac{1}{6}]$ thus $v(x+2^{-n-1}, n) = \psi(w')$ and thus $f_1(x,n) = f_0(x,n) = g(z,n)$. It follows that $g^*(x,n) = \alpha g(y,n) + (1-\alpha) g(z,n)$ where $\alpha = i(x,n) \in [0,1]$. Furthermore, $|z - x| \leqslant 2^{-n}$ by construction of w'.
- **If** $\varepsilon \in \left]2^{-n}\frac{2}{3}, 2^{-n}\right]$ **then** $i(x,n) \geqslant 1$ thus $g^*(x,n) = f_1(x,n)$. Define $w' \in \Gamma^n$ such that[28] $\psi_{\mathbb{Q}}(w') = \psi_{\mathbb{Q}}(w) + 2^{-n}$ and define $z = \psi_{\mathbb{Q}}(w')$. It follows that $x - \psi_{\mathbb{Q}}(w') \in [0, 2^{-n}\frac{1}{2}]$ thus $v(x + 2^{-n-1}, n) = \psi(w')$ and thus $f_1(x,n) = f_0(x,n) = g(z,n)$. So in this case, $y = z \in [0,1/2]$ and α can be anything.

---

[26] It may not be unique since we closed the interval on both sides in order to get all of $[0,1/2]$ with words in $\{0\} \times \Gamma^n$. If we opened the interval on the right, we would only get $[0,1/2[$ with such words.

[27] The use of this assumption will become later on. Essentially, it is there to ensure that the y we construct belongs to $[0,1/2]$ so that we can apply the function g to it.

[28] This is always possible, formally if w is seen as a number, written in binary, then w' is $w+1$.

We are now in position to conclude thanks to the modulus of continuity of f. Recall that by definition, m is a polynomial such that for any $x, y \in [0, 1/2]$ and $k \in \mathbb{N}$, if $|x - y| \leqslant 2^{-m(k)}$ then $|f(x) - f(y)| \leqslant 2^{-k}$. Without loss of generality, we can assume[29] that $m(\mathbb{N}) \subseteq \mathbb{N}$ and $m(n) \geqslant n$. Now define for any $x \in [0, 1/2]$ and $n \in \mathbb{N}$:

$$g^{**}(x, n) = g^*(x, m(n+1))$$

Clearly $g \in \mathrm{AP}$ since $g \in \mathrm{AP}$ and m is a polynomial. Let $x \in \mathbb{N}$ and $n \in \mathbb{N}$. Then we have shown that there exists $y, z \in \mathbb{Q} \cap [0, 1/2]$ and $\alpha \in \mathbb{N}$ such that $|x - y|, |x - z| \leqslant 2^{-m(n+1)}$ and $g^*(x, m(n+1)) = \alpha g(y, m(n+1)) + (1-\alpha) g(z, m(n+1))$. By definition of g, $|g(y, m(n+1)) - f(y)| \leqslant 2^{-m(n+1)} \leqslant 2^{-n-1}$ since $m(n+1) \geqslant 2$. Similarly, $|g(z, m(n+1)) - f(z)| \leqslant 2^{-n-1}$. Furthermore, $|f(y) - f(x)|, |f(z) - f(x)| \leqslant 2^{-n-1}$. Thus:

$$\begin{aligned}
|g^{**}(x, n) - f(x)| &\leqslant \alpha |g^*(y, m(n+1)) - f(x)| + (1-\alpha)|g^*(z, m(n+1)) - f(x)| \\
&\leqslant \alpha(2^{-n-1} + |f(y) - f(x)|) + (1-\alpha)(2^{-n-1} + |f(z) - f(x)|) \\
&\leqslant \alpha 2^{-n} + (1-\alpha) 2^{-n} \\
&\leqslant 2^{-n}
\end{aligned}$$

Using Remark F.12.2, we have thus shown that $f \in \mathrm{AWP}$ and since $\mathrm{AWP} = \mathrm{AP}$, $f \in \mathrm{AP}$.

[29] Do do so, consider the same polynomial where each coefficient is the ceiling value of the absolute value of the corresponding coefficient of m, and add the monomial $x \mapsto x$.