Quantifiying the Robustness of Dynamical Systems. Relating Time and Space to Length and Precision

Manon Blanc
Institut Polytechnique de Paris, Ecole Polytechnique, LIX, Palaiseau, France
Université Paris-Saclay, LISN, Orsay, France

Olivier Bournez
Institut Polytechnique de Paris, Ecole Polytechnique, LIX, Palaiseau, France

Abstract

Reasoning about dynamical systems evolving over the reals is well-known to lead to undecidability. In particular, it is known that there cannot be reachability decision procedures for first-order theories over the reals extended with even very basic functions, or for logical theories that reason about real-valued functions, or decision procedures for state reachability. This mostly comes from the fact that reachability for dynamical systems over the reals is fundamentally undecidable, as Turing machines can be embedded into (even very simple) dynamical systems.

However, various results in the literature have shown that decision procedures exist when restricting to robust systems, with a suitably-chosen notion of robustness. In particular, it has been established in the field of verification that if the state reachability is not sensitive to infinitesimal perturbations, then decision procedures for state reachability exist. In the context of logical theories over the reals, it has been established that decision procedures exist if we focus on properties not sensitive to arbitrarily small perturbations. For example by considering properties that are either true or δ-far from being true, for some $\delta > 0$.

In this article, we first propose a unified theory explaining in a uniform framework these statements, that were established in different contexts.

More fundamentally, while all these statements are only about computability issues, we also consider complexity theory aspects. We prove that robustness to some precision is inherently related to the complexity of the decision procedure. When a system is robust, it makes sense to quantify at which level of perturbation it is. We prove that assuming robustness to a polynomial perturbation on precision leads to a characterisation of PSPACE. We prove that assuming robustness to polynomial perturbation on time or length leads to similar statements for PTIME.

In other words, precision on computations is inherently related to space complexity, while length or time of trajectories, is intrinsically related to time complexity. These statements can also be interpreted in relation to several recent results about the computational power of analogue models of computation.

2012 ACM Subject Classification Theory of computation → Models of computation; Theory of computation → Computability; Theory of computation → Complexity classes; Theory of computation → Complexity theory and logic; Computer systems organization → Analog computers

Keywords and phrases Computability, Complexity theory, Computable analysis, Verification, Decision, Robustness, Dynamical Systems, Models of computation, Analogue Computations

Digital Object Identifier 10.4230/LIPIcs.CSL.2024.17

Funding This work was partially supported by Labex Digicosme and ANR δIFFERENCE.

1 Introduction

The relations between dynamical systems over the reals and computations have been the source of many works, with sometimes very different motivations [23, 22, 2, 4, 11]. Let us focus on some of them.

The context of the verification of continuous and hybrid systems. As many systems in our world are naturally modelled by dynamical systems over the reals (or by so-called hybrid systems, mixing continuous and discrete aspects), verification of safety properties on these systems is inherently related to state reachability for dynamical systems. Roughly speaking, a system is safe if the subset of "*bad states*" (i.e. those not satisfying some property) cannot be reached from the subset of the initial states of the system. Unfortunately, it is well-known that such questions are undecidable, even for very simple dynamical systems. For example, even for piecewise affine functions [22, 23] over the compact domain $[0, 1]^2$.

Such a statement is proved by showing that a Turing machine, which is a particular discrete-time dynamical system, evolving with time over configurations, can be embedded into such systems. This requires mapping the infinite set of possible configurations of a Turing machine into a real domain. Hence, it requires infinite precision encoding when the system is defined over a compact domain. This establishes that verification is undecidable for systems with infinite precisions. However this does not seem to prove that undecidability holds for systems that would not be based on infinite precision computations.

In that spirit, while several undecidability results were stated for hybrid systems, such as Linear Hybrid Automata [18], Piecewise Constant Derivative systems [2], an informal conjecture (that we will call the *"robustness conjecture"*) appeared in the field of verification of hybrid and continuous systems by various authors. It states that *undecidability is due to non-stability, non-robustness and sensitivity to the initial values of the systems.* There were several attempts to formalise and prove this, including [14, 1].

▶ Remark 1. Actually, the *"robustness conjecture"* is known to be an informal conjecture, as it is related to the considered mathematical notion of robustness. It holds for some mathematical concepts of robustness, such as the ones considered in this article. It is provably false for some other mathematical concepts of robustness: see e.g. [19].

▶ Remark 2. The mathematical formalisation of robustness, or the question of the various "natural" concepts of robustness, is related to philosophical questions about the limits of mathematical models, or of computability theory. A Turing machine is an ideal model, as well as dynamical systems over the reals are often idealisations of models. We do not aim at going to these kinds of discussions[1].

▶ Remark 3. Our point in this article is to understand how various notions of robustness lead to decidability, namely the ones of [1, 15, 28]. We argue that the *"robustness conjecture"* can be unified in a general theory, in these various approaches. Furthermore, we prove that this holds at the decidability level and also at the complexity level: quantifying the accepted level of robustness corresponds naturally to the complexity of the associated questions.

Here, we are starting from the approach of [1], where classes of dynamical systems are considered. A notion of perturbed dynamics by a small ϵ is associated with each of them. A perturbed reachability relation is defined as the intersection of all reachability relations obtained by ϵ-perturbations. The authors of [1] showed that, for many models, the perturbed reachability relation is co-computably enumerable (co-c.e., Π_1) and any co-c.e. relation can be defined as the perturbed reachability relation of such models. Consequently, it follows from basic computability arguments, namely that a computably enumerable and co-computably enumerable set is decidable that *if robustness is defined as the stability of the reachability relation under infinitesimal perturbation*, then robust systems have a decidable reachability relation and hence a decidable verification (i.e. the *robusness conjecture* holds).

[1] Even if our results shed some light on these questions, such as the fact that complexity theory is essentially quantifying the accepted level of robustness.

The context of the decision procedures for logical theories over the reals. In the context of decision procedures for logic over the reals, the authors of [15] observed that it is well-known that some logics, such as real arithmetic, are decidable. However, decidability does not hold for simple extensions of real arithmetic. Indeed, even the set of Σ_1-sentences in a language extending real arithmetic with the sine function is already undecidable. But if a relaxed and more "robust" notion of correctness is considered (one asks to answer true when a given formula ϕ is true and to return false when it is δ-robustly wrong) the truth of a formula becomes algorithmically solvable. In other words, undecidability intrinsically comes from the fact that the truth of a sentence might depend on infinitesimally small variations of its interpretation.

Recently, the author of [28] proposed a first-order predicate language for reasoning about multi-dimensional smooth real-valued functions. They proved the specification of an algorithm solving formulas robustly satisfiable with respect to some metrics. The proof of decidability can also be interpreted using an argument similar to [1, 15].

Our contributions

We extend and relate these approaches to very general settings and provide a general framework for explaining these observations.

Namely, using various arguments from computability and computable analysis, we establish the following:

- We consider various classes of dynamical systems: in turn, Turing machines, then discrete-time dynamical systems preserving the rationals, then general discrete-time dynamical systems and eventually continuous-time dynamical systems. For all of them, we define robustness as non-sensitivity to infinitesimal perturbations of the associated reachability relation, in the spirit of [1].
  - We prove that for this natural concept of robustness, the *"robustness conjecture"* holds: verification or reachability relation (and hence safety verification) is decidable (Corollary 14, Corollary 25, Corollary 42, Corollary 49, Corollary 54) .
  - We characterise and relate robustness to the question of decidability by proving that the converse holds if some property is added (Corollary 32). This means that there is a form of completeness of the above statement: when decidability holds, establishing the robustness of the corresponding system can be done, for a suitable perturbation or metric.
- Furthermore, we relate this approach, inspired by [1], in the context of verification, to the concept of δ-decision of [15], in the context of decision procedures in logic[2]. A system is robust iff its reachability relation is either true or ϵ-far from being true (Proposition 27).
- We also prove that robustness can be seen as having a reachability relation that can be represented as a pixelated image. It is a simple and elegant geometric property (Corollary 51 and 52).
- More fundamentally, while the above results are about decidability, we also discuss *complexity* issues. Indeed, when a system is robust, it is natural to quantify the allowed level of perturbation.

[2] We mix the notation δ and ϵ when talking about precision. They are indeed the same. Our problem is that the framework considered in [15] uses the terminology δ-decidability, whereas [1] is in a context of real-analysis and uses ϵ to quantify error bounds. We decided to keep both δ and ϵ. Otherwise, this would conflict with their usual meaning in the two contexts.

- We show that considering a perturbation polynomially small relates to a very intuitive way to the complexity of the associated verification or decision problem (Theorem 18, Theorem 36).
- More precisely, polynomial space computability is related to precision (Theorem 18, Theorem 36) while polynomial time computability is related to the time or length of the trajectory (Theorem 64).

More on related work. The approach of considering dynamical systems with respect to infinitesimal perturbations of dynamics is an old idea, sometimes with concepts reinvented later with other names. We can mention the concept of "chain reachability" studied by Conley in the 1970's. In the field of verification, the idea of infinitely perturbed dynamics has been considered to provide alternative semantics of some models: see e.g. [27] for timed automata. The approaches considered in [1] and [15] belong to the line of investigation considering general dynamical systems and aiming at studying the frontier between decidability and undecidability.

Up to our knowledge, such a unifying framework has never been established. For the computability aspects, with respect to some of the existing works: Compared to [1], we allow more general discrete-time and continuous-time dynamical systems, such as those with unbounded domains. Some generalisations have also been obtained in [9], but focusing on dynamical systems as language recognisers and mainly focusing on generalisations of [1, Theorem 4]. The logic considered in [15] allows to talk about finite-time reachability properties, but not reachable sets. As far as we know, complexity aspects have never been discussed.

Motivation and interpretation related to models of computation. An orthogonal field of research is about understanding how analogue (possibly continuous-time) models of computation behave compared to more classical discrete models such as Turing machines. This includes models based on ordinary differential equations like the GPAC [29], or algebraic models based on ordinary differential equations inspired from computability theory [10], or from computer algebra [6]. A long-standing open problem was how to measure time complexity in continuous time models. It was recently proved [8] that the length of the solution curves provides a measure equivalent to time for digital models. The question of a natural measure for space complexity remains open, despite some very recent characterisations of $(F)PSPACE$ using ODEs [7].

▶ Remark 4. The theory developed in the current article comes from an attempt to get to a simpler characterisation of $(F)PSPACE$, with continuous ODEs. We obtained this theory initially with the idea that getting to $(F)PSPACE$ requires a way to forbid undecidability. This led us to develop this theory based on these notions of robustness, guaranteeing computability.

The theory developed here provides arguments to state that, over a compact domain, space corresponds to the precision of the computations, while it corresponds to the logarithm of the size of some graphs for systems over more general domains. Meanwhile, this idea has been used to provide a simple characterisation of $FPSPACE$ with discrete ordinary differential equations in [3]. The question of whether a simple characterisation with continuous ODEs can be obtained remains open.

Preliminaries. $d(\cdot,\cdot)$ is norm-sup (also called uniform) distance. An (open) (resp. close) *rational ball* is a subset of real numbers of the form $B(\mathbf{x},\delta)=\{\mathbf{y}\in\mathbb{R}^d : d(\mathbf{x},\mathbf{y})<\delta\}$ (resp. $\overline{B}(\mathbf{x},\delta)=\{\mathbf{y}\in\mathbb{R}^d : d(\mathbf{x},\mathbf{y})\leq\delta\}$) for some rational $\mathbf{x}$ and δ, and some integer d. We could use the Euclidean distance, but this distance has the advantage that its balls correspond directly to rounding at a precision. A set of reals of the form $\prod_{i=1}^{d}[a_i, b_i]$, for rational (a_i), (b_i), will be called a *rational closed box*. An open rational box is obtained by considering open intervals in the previous definition. The least closed set containing X is denoted by $cls(X)$. We write $\ell(\cdot)$ for the function that measures the binary size of its argument. We say that a function $\mathbf{f} : \mathbb{Q}^d \to \mathbb{Q}^d$ or $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz when there exists some constant K such that $d(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y})) \leq K d(\mathbf{x}, \mathbf{y})$. We basically have in mind in all this article, dynamical systems over $\mathbb{R}^d$, even if in some of the subsections we consider that they might preserve rationals.

2 On graphs reachability and perturbated TMs

Our theory relies on some well-known observations from complexity theory. We start by recalling some facts and a few basic concepts.

2.1 Some considerations from complexity theory

First, we recall some complexity results about the following decision problem $\mathrm{PATH}(G, u, v)$: Given a directed graph $G = (V, \to)$ and some vertices $u, v \in V$, determine whether there is some path between u and v in G, denoted by $u \xrightarrow{*} v$.

▶ **Lemma 5** (Reachability for graphs, [30]). $\mathrm{PATH}(G, u, v) \in \mathrm{NLOGSPACE}$.

▶ **Lemma 6** (Immerman–Szelepcsényi's theorem [20, 31]). $\mathrm{NLOGSPACE} = \mathrm{coNLOGSPACE}$.

We mainly focus on its complement:

▶ **Corollary 7.** *Consider the following decision problem* $\mathrm{NOPATH}(G, u, v)$*: given a directed graph $G = (V, \to)$ and some vertices $u, v \in V$, determine whether there is no path between u and v in G.*

Then $\mathrm{NOPATH}(G, u, v) \in \mathrm{NLOGSPACE}$.

▶ **Theorem 8** (Savitch's theorem, [30]). *For any function $f : \mathbb{N} \to \mathbb{N}$ with $f(n) \geq \log n$, we have* $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$.

▶ **Corollary 9.** $\mathrm{PATH}(G, u, v) \in \mathrm{SPACE}(log^2(n))$ *and* $\mathrm{NOPATH}(G, u, v) \in \mathrm{SPACE}(log^2(n))$.

▶ Remark 10. Notice that detecting whether there is no path between u and v is equivalent to determining whether all paths starting from u "loop", i.e. remain disjoint from v. The above statement is established using a more subtle method than a simple depth-of-width search of the graph. One uses the trick of the proof of Savitch's theorem, i.e. a recursive procedure (expressing reachability in less than 2^t steps, called $\mathrm{CANYIELD(C_1, C_2, t)}$ in [30]) guaranteeing the wanted space complexity.

A (general) *discrete-time dynamical system* $\mathcal{P}$ is given by a set X, called *domain* and some (possibly partial) function f from X to X. A trajectory of $\mathcal{P}$ is a sequence (x_t) evolving according to f: that is $\forall t, x_{t+1} = f(x_t)$. We say x^* (or a set X^*) is reachable from x if there is a trajectory with $x_0 = x$ and $x_t = x^*$ (respectively $x_t \in X^*$) for some t.

▶ Remark 11. In other words, any discrete-time dynamical system $\mathcal{P}$ can be seen as a particular (deterministic) directed graph $G = (V, \to)$, where V is not necessarily finite: G corresponds to $V = X$ and $\to$ to the graph of the function f.

In particular, the following is a classical result (not following from the most obvious algorithm, but from Savitch theorem, i.e. from sometimes so-called arithmetisation techniques).

▶ **Lemma 12** (Reachability for finite graphs). *Let $s(n) \geq \log(n)$. Assume the vertices of $G = (V, \to)$ can be encoded in binary using words of length $s(n)$. Assume the relation $\to$ is decidable using a space polynomial in $s(n)$. Then, given the encoding of $u \in V$ and of $v \in V$, we can decide whether there is some path from u to v, in a space polynomial in $s(n)$.*

Our theory covers various dynamical systems. In particular, as a Turing machine is a particular type of discrete-time dynamical system, we think this helps, for pedagogical reasons, to discuss first the case of Turing machines. We follow, on this aspect, what was done in [1].

2.2 The case of Turing machines

We focus on the framework of Turing Machines (TMs). Let Σ be a finite alphabet and let $B \notin \Sigma$ be the blank symbol. A TM over Σ is a tuple (Q, q_0, F, R, Γ) where Q is a finite set of control states, $q_0 \in Q$ is the initial control state, $F \subseteq Q$ (respectively $R \subseteq Q$) is a set of accepting (respectively rejecting) states, with $F \cap R = \emptyset$ and Γ is a set of transitions of the form $(q, a) \to (q', b, \delta)$ where $q, q' \in Q$, $a, b \in \Sigma \cup \{B\}$ and $\delta \in \{-1, 0, 1\}$. When the machine has accepted or rejected, the decision remains unchanged: when $q \in F$, then $q' \in F$ and when $q \in R$ then $q' \in R$.

We write $\mathcal{C}_{\mathcal{M}}$ for the set of the configurations of a TM and write a configuration as a triple $(q, \cdots a_{-2}a_{-1}, a_0a_1a_2\cdots)$: q gives the internal state and a_0 the position of the head.

Given a transition $(q, a) \to (q', b, \delta)$ in Γ, if the control state is q and the symbol pointed by the head of the machine is equal to a, then the machine can change its configuration C to the configuration C' in the following manner: the control state is now q', the symbol pointed by the head is replaced by b and then the head is moved to the left or the right, or it stays at the same position according to whether δ is $-1, 1$, or 0, respectively. We write $C \vdash C'$ when this holds, i.e. C' is the one-step next configuration of the configuration C. Then $(\mathcal{C}_{\mathcal{M}}, \vdash)$ corresponds to a particular dynamical system.

Word $w = a_1 \cdots a_n \in \Sigma^*$ is accepted by $\mathcal{M}$ if, starting from the initial configuration $C_0 = C_0[w] = (q_0, \cdots BBB, a_1a_2\cdots a_nBBB\cdots)$ the machine eventually stops in an accepting control state: that is, if we write $\mathcal{F}$ for the configurations where $q \in F$, iff $C_0 \vdash^* C^*$ for some $C^* \in \mathcal{F}$. Let $L(\mathcal{M})$ denote the set of such words, i.e., the computably enumerable (c.e) language semi-recognised by $\mathcal{M}$. We say that w is rejected by $\mathcal{M}$ if, starting from the configuration C_0 the machine $\mathcal{M}$ eventually stops in a rejecting state. $\mathcal{M}$ is said to always halt if for all w, either w is accepted or w is rejected.

Article [1] introduces the concept of space-perturbed TM: given $n > 0$, the idea is that the n-perturbed version of the machine $\mathcal{M}$ is unable to remain correct at a distance more than n from the head of the machine. Formally, the n-perturbed version $\mathcal{M}_n$ of $\mathcal{M}$ is defined exactly as $\mathcal{M}$ except before any transition, all the symbols at a distance n or more from the head can be altered at every step. Hence $\mathcal{M}_n$ is nondeterministic. A word w is accepted by $\mathcal{M}_n$ iff there exists a run of this machine which stops in an accepting state. Let $L_n(\mathcal{M})$ be the n-perturbed language of $\mathcal{M}$. From definitions, if a word is accepted by $\mathcal{M}$, then it is also recognised by all the $\mathcal{M}_n$'s: perturbed machines have more behaviours. Moreover, $L_{n+1}(\mathcal{M}) \subseteq L_n(\mathcal{M})$. Let $L_\omega(\mathcal{M}) = \bigcap_n L_n(\mathcal{M})$: this is the set of words accepted by $\mathcal{M}$ when subject to arbitrarily "small" perturbations. We have $L(\mathcal{M}) \subseteq L_\omega(\mathcal{M}) \subseteq \cdots \subseteq L_2(\mathcal{M}) \subseteq L_1(\mathcal{M})$.

Here is a key observation: The ω-perturbed language of a TM is co-computably enumerable:

▶ **Theorem 13** (Perturbed reachability is co-c.e. [1])**.** $L_\omega(\mathcal{M}) \in \Pi_1^0$.

Since a set that is c.e. and co-c.e. is decidable, following [1], if we define robustness as $L_\omega(\mathcal{M}) = L(\mathcal{M})$, then robust languages are necessarily decidable (i.e. the *"robustness conjecture"* holds).

▶ **Corollary 14** (Robust $\approx$ decidable [1])**.** *If $L_\omega(\mathcal{M}) = L(\mathcal{M})$ then $L(\mathcal{M})$ is decidable. If $\mathcal{M}$ always halts, then $L(\mathcal{M})$ is decidable and $L_\omega(\mathcal{M}) = L(\mathcal{M})$.*

A simple point, but a key observation for coming discussions, is the following: one can talk about complexity and not only computability. Indeed, when a language is robust, it makes sense to measure what level of perturbation s can be tolerated. This is the purpose of Definition 16.

▶ Remark 15. Assume $L = L(M)$ is robust. By definition, $L = L(M) = L_\omega(\mathcal{M})$. This means that for any word w, there must exist some n (depending possibly on w) such that $w \in L(M)$ and $w \in L_n(w)$ have the same truth value. This n can be read as the associated tolerated level of perturbation. It quantifies the tolerated level of robustness. Now, we can always consider that this function that associates n to w depends only on its length (as there are finitely many words of a given length, and as we can always replace n by a bigger n).

Formally: assuming a language $L = L(M)$ is robust means $L = L(M) = L_\omega(\mathcal{M})$. Let us consider some length ℓ, and reason about words of length ℓ (i.e. about words of Σ^ℓ where Σ is the alphabet of the Turing machine). We must have $L \cap \Sigma^\ell = L(\mathcal{M}) \cap \Sigma^\ell = L_\omega(\mathcal{M}) \cap \Sigma^\ell$. Now, by definition of $L_\omega(M)$, $L_\omega(M) \cap \Sigma^\ell$ is necessarily $L_n(M) \cap \Sigma^\ell$ for some n. Consequently, we must have $L(\mathcal{M}) \cap \Sigma^\ell = L_n(M) \cap \Sigma^\ell$ for some $n = s(\ell)$ for a robust language.

In other words, for a robust language, we have necessarily

$$L = L(\mathcal{M}) = L_{\{s\}}(\mathcal{M})$$

for some function s, for the coming definition. This function $n = s(\ell)$ quantifies the tolerated level of robustness. If one prefers, a robust language is necessarily s-robust for some s, according to Definition 17.

▶ **Definition 16** (Level of robustness n given by s)**.** *Given a function $s : \mathbb{N} \to \mathbb{N}$, we write $L_{\{s\}}(\mathcal{M})$ for the set of words accepted by $\mathcal{M}$ with space perturbation s: $L_{\{s\}}(\mathcal{M}) = \{w | \ w \in L_{s(\ell(w))}(\mathcal{M})\}$.*

▶ **Definition 17** (s-robust language)**.** *We say that a robust language is s-robust, when $L = L(\mathcal{M}) = L_{\{s\}}(\mathcal{M})$.*

It is natural to consider the case where the function s is a polynomial. It turns out that this corresponds to (and is even a characterisation of) PSPACE:

▶ **Theorem 18** (Polynomial robustness $\Leftrightarrow$ PSPACE)**.** *$L \in$ PSPACE iff for some $\mathcal{M}$ and some polynomial p, $L = L(\mathcal{M}) = L_{\{p\}}(\mathcal{M})$.*

Proof. ($\Rightarrow$) If M always terminates and works in polynomial space, then there exists a polynomial $q(\cdot)$ that bounds the size of the used part of the tape of M. Considering a polynomial $p \geq q + 2$, we have for $n \in N$ $L_{p(\ell(w))}(\mathcal{M}) \subseteq L(M)$. We always have the other inclusion.

($\Leftarrow$) $L_{p(n)}(\mathcal{M}) \in PSPACE$ is direct from the definitions. ◀

3 Embedding TMs into dynamical systems

Many authors have embedded TMs in various classes of dynamical systems to get undecidability or various hardness results. We present in this section how this is usually done, to point out where (intuitive) (non-)robustness issues appear.

Generally speaking, the trick is the following. If we forget about blanks, assuming alphabet $\Sigma = \{0, 1\}$, we can consider that $\mathcal{C}_{\mathcal{M}} \subseteq \mathcal{C} = \mathbb{N} \times \Sigma^* \times \Sigma^*$. Fix some encoding function of configurations into a vector of real numbers: $\Upsilon : \mathcal{C} \to \mathbb{R}^d$, with $d \in \mathbb{N}$. For example, $\Upsilon(q, w_1, w_2) = (q, \gamma(w_1), \gamma(w_2))$ with $\gamma : \Sigma^* \to \mathbb{R}$ taken as:

- the encoding $\gamma_{\mathbb{N}}$ mapping the word $w = a_1 \dots a_n$ to the integer whose binary expression is w,
- or $\gamma_{[0,1]}$ mapping w to the real number of $[0, 1]$ whose binary expansion is w,
- or more generally, $\gamma^k_{[0,1]}$ or $\gamma^k_{\mathbb{N}}$, using base k instead of base 2 for $k \geq 2$,
- or $\gamma'^k_{[0,1]}$ mapping w to $(\gamma^k_{[0,1]}(w), \ell(w))$.

Consider a function $\mathbf{f} : X \subseteq \mathbb{R}^d \to X$ such that for any configuration C, denoting by C' the next configuration, $\mathbf{f}(\Upsilon(C)) = \Upsilon(C')$: one step of the TM corresponds to one step in the dynamical system $(X, \mathbf{f})$ with respect to γ. Then, the diagram of the execution commutes for any number of steps:

$$\begin{tikzcd}
C \arrow[r, "\vdash"] \arrow[d, "\Upsilon"'] & C' \arrow[r, "\vdash"] \arrow[d, "\Upsilon"] & C'' \arrow[r, "\vdash"] \arrow[d, "\Upsilon"] & C''' \arrow[r, dashed, "\vdash"] \arrow[d, "\Upsilon"] & {} \\
\Upsilon(C) \arrow[r, "\mathbf{f}"] & \Upsilon(C') \arrow[r, "\mathbf{f}"] & \Upsilon(C'') \arrow[r, "\mathbf{f}"] & \Upsilon(C''') \arrow[r, dashed, "\mathbf{f}"] & {}
\end{tikzcd}$$

The questions related to the existence of trajectories in the dynamical system $(X, \mathbf{f})$ associated with the TM correspond then to the questions about the existence of trajectories over $(X, \mathbf{f})$. Specifically, it provides c.e.-hardness of reachability for various classes of dynamical systems, as it is for TM. Call such a situation a *step-by-step* emulation. However, encodings such as $\gamma_{[0,1]}$, whose image is compact, map intrinsically distinct configurations to points arbitrarily close to each other (a sequence over a compact must have some accumulation point). Encodings like $\gamma_{\mathbb{N}}$ do not have a compact image but involve emulations with arbitrarily large integers, which is another issue. These observations led to the already mentioned (informal) conjecture that undecidabilty/hardness may hold only for non-robustness systems. This leads to discussing now formally robustness issues for general dynamical systems over $\mathbb{R}^d$ for some $d \in \mathbb{N}$.

4 Discrete-Time Dynamical Systems

We now study the case of general discrete-time systems. We aim at focusing on discrete-time systems of type $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$.

4.1 The case of rational systems

For clarity, as this general case requires to talk about computability issues on the reals (we do so later in Section 4.2) we first focus on the case of systems of type $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ such that $\mathbf{f}(\mathbb{Q}) \subseteq \mathbb{Q}$. In other words, we first focus on the case of rational systems, i.e. $\mathbf{f} : \mathbb{Q}^d \to \mathbb{Q}^d$ (possibly obtained as the restriction to the rationals of a function over the reals).

A rational discrete-time dynamical system will be called $\mathbb{Q}$-computable when the function (hence from the rationals to the rationals) is. A rational discrete-time dynamical system will be called Lipschitz when the function is: there exists some constant K such that $d(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y})) \leq Kd(\mathbf{x}, \mathbf{y})$, for all $\mathbf{x}$, $\mathbf{y}$. It will be called locally Lipschitz when for any $\mathbf{z}$ and $\epsilon > 0$ there exists some constant K such that $d(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y})) \leq Kd(\mathbf{x}, \mathbf{y})$, for all $\mathbf{x}$, $\mathbf{y}$ in $B(\mathbf{z}, \epsilon)$.

With each rational discrete-time dynamical system $\mathcal{P}$ is associated its reachability relation $R^{\mathcal{P}}(\cdot, \cdot)$ on $\mathbb{Q}^d \times \mathbb{Q}^d$. Namely, for two rational points $\mathbf{x}$ and $\mathbf{y}$, $R^{\mathcal{P}}(\mathbf{x}, \mathbf{y})$ holds iff there exists a trajectory of $\mathcal{P}$ from $\mathbf{x}$ to $\mathbf{y}$. The reachability relation of a $\mathbb{Q}$-computable system is computably enumerable: to enumerate, a Turing machine can just simulate the dynamics.

▶ Remark 19. Article [1] considers only the special case of Piecewise affine (PAM) maps, as representative of discrete-time systems, which are particular $\mathbb{Q}$-computable Lipschitz systems.

▶ Remark 20. Here is an example of $\mathbb{Q}$-computable Lipschitz systems. Take some recurrent neural network, with d neurons, with the ReLU activation function, defined as $\mathrm{ReLU}(x) = \max(0, x)$. Its dynamic can be written as $\mathbf{x}_{t+1} = \mathrm{ReLU}(\mathbf{A}\mathbf{x} + \mathbf{B})$ where $\mathbf{A}$ is some $d \times d$ matrix with rational entries and $\mathbf{B}$ is some vector of dimension d with rational entries, where the ReLU function is applied componentwise.

Reachability for $\mathbb{Q}$-computable systems is undecidable and c.e.-complete:

▶ **Theorem 21** (Computational power of PAMs [23, 22, 1])**.** *Any c.e. language is reducible to the reachability relation of a PAM.*

Let us discuss whether undecidability still holds for “robust systems”.

We apply the paradigm of small perturbations: consider a discrete-time dynamical system $\mathcal{P}$ with a function $\mathbf{f}$. For any $\epsilon > 0$ we consider the ϵ-perturbed system $\mathcal{P}_\epsilon$. Its trajectories are defined as sequences $\mathbf{x}_t$ satisfying $d(\mathbf{x}_{t+1}, \mathbf{f}(\mathbf{x}_t)) < \epsilon$ for all t. This non-deterministic system is considered as $\mathcal{P}$ submitted to a noise of magnitude ϵ. For convenience, we write $\mathbf{y} \in \mathbf{f}_\epsilon(\mathbf{x})$ as a synonym for $d(\mathbf{f}(\mathbf{x}), \mathbf{y}) < \epsilon$. We denote reachability in the system $\mathcal{P}_\epsilon$ by $R^{\mathcal{P}}_\epsilon(\cdot, \cdot)$.

All trajectories of a non-perturbed system $\mathcal{P}$ are also trajectories of the ϵ-perturbed system $\mathcal{P}_\epsilon$. If $\epsilon_1 < \epsilon_2$ then any trajectory of the ϵ_1-perturbed system is also a trajectory of the ϵ_2-perturbed PAM. Define $R^{\mathcal{P}}_\omega(\mathbf{x}, \mathbf{y})$ iff $\forall \epsilon > 0$ $R^{\mathcal{P}}_\epsilon(\mathbf{x}, \mathbf{y})$: this relation encodes reachability with arbitrarily small perturbing noise. From definitions:

▶ **Lemma 22** ([1])**.** *For any $0 < \epsilon_2 < \epsilon_1$ and any $\mathbf{x}$ and $\mathbf{y}$ the following implications hold:* $R^{\mathcal{P}}(\mathbf{x}, \mathbf{y}) \Rightarrow R^{\mathcal{P}}_\omega(\mathbf{x}, \mathbf{y}) \Rightarrow R^{\mathcal{P}}_{\epsilon_2}(\mathbf{x}, \mathbf{y}) \Rightarrow R^{\mathcal{P}}_{\epsilon_1}(\mathbf{x}, \mathbf{y})$.

▶ **Theorem 23** (Perturbed reachability is co-c.e.)**.** *Consider a locally Lipschitz $\mathbb{Q}$-computable system whose domain $X \subseteq \mathbb{R}^d$ is a closed rational box*[3]*. Then the relation $R^{\mathcal{P}}_\omega(\mathbf{x}, \mathbf{y}) \subseteq \mathbb{Q}^d \times \mathbb{Q}^d$ is in the class Π_1.*

Proof. This extends [1, Theorem 5], using an alternative proof. As $\mathbf{f}$ is locally Lipschitz and X is compact, we know that $\mathbf{f}$ is Lipschitz: there exists some $L > 0$ so that $d(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y})) \leq L \cdot d(\mathbf{x}, \mathbf{y})$. For every $\delta = 2^{-m}$, $m \in \mathbb{N}$, we associate some graph $G_m = (V_\delta, \to_\delta)$: its vertices, denoted by $(\mathcal{V}_i)_i$, correspond to some finite discretisation and covering of compact X by rational open balls $\mathcal{V}_i = B(\mathbf{x}_i, \delta_i)$ of radius $\delta_i < \delta$. There is an edge from $\mathcal{V}_i$ to $\mathcal{V}_j$ in this graph, that is to say $\mathcal{V}_i \to_\delta \mathcal{V}_j$, iff $B(\mathbf{f}(\mathbf{x}_i), (L+1)\delta) \cap \mathcal{V}_j \neq \emptyset$. With our hypothesis on the domain, such a graph can be effectively obtained from m, considering a suitable discretisation of the rational box X.

[3] Recall that a rational box X is a subset of real numbers, not of rational numbers. Consequently, such an X is compact.

▷ Claim 1. assume $R_\epsilon^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ with $\mathbf{x} \in \mathcal{V}_i$ for $\epsilon = 2^{-n}$. Then $\mathcal{V}_i \xrightarrow{*}_\epsilon \mathcal{V}_j$ for all $\mathcal{V}_j$ with $\mathbf{y} \in \mathcal{V}_j$. This holds as the graph for $\delta = \epsilon$ is made to always have more trajectories/behaviours than $R_\epsilon^{\mathcal{P}}$.

▷ Claim 2. for any $\epsilon = 2^{-n}$, there is some $\delta = 2^{-m}$ so that if we have $\mathcal{V}_i \xrightarrow{*}_\delta \mathcal{V}_j$ then $R_\epsilon^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ whenever $\mathbf{x} \in \mathcal{V}_i$, $\mathbf{y} \in \mathcal{V}_j$.

Claim 2 says that $\neg R_\epsilon^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ implies $\neg(\mathcal{V}_i \xrightarrow{*}_\delta \mathcal{V}_j)$ whenever $\mathbf{x} \in \mathcal{V}_i$, $\mathbf{y} \in \mathcal{V}_j$, for the corresponding δ.

From the two above items, $\neg R_\omega^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ holds iff for some $\delta = 2^{-m}$, $\neg(\mathcal{V}_i \xrightarrow{*}_\delta \mathcal{V}_j)$ for some $\mathcal{V}_i$, $\mathcal{V}_j$ with $\mathbf{x} \in \mathcal{V}_i$, $\mathbf{y} \in \mathcal{V}_j$. This holds iff for some integer m, $\mathrm{NOPATH}(G_m, \mathcal{V}_i, \mathcal{V}_j)$ for some $\mathcal{V}_i$, $\mathcal{V}_j$ with $\mathbf{x} \in \mathcal{V}_i$, $\mathbf{y} \in \mathcal{V}_j$. The latter property is c.e., as it is a union of decidable sets (uniform in m), as $\mathrm{NOPATH}(G_m, \mathcal{V}_i, \mathcal{V}_j)$ is a decidable property over finite graph G_m. ◀

▶ **Definition 24** (Robust reachability relation). *We say that the reachability relation is* robust *when $R^{\mathcal{P}} = R_\omega^{\mathcal{P}}$.*

We get the *"robustness conjecture"*:

▶ **Corollary 25** (Robust ⇒ decidable, [1, Corollary 5]). *Assume the hypotheses of Theorem 23. If the relation $R^{\mathcal{P}}$ is robust then it is decidable.*

4.1.1 Robustness versus decidability and δ-decidability

We now discuss how far the above statement is to a characterisation of decidability.

There is indeed a converse property if some condition is added. Before stating this in Corollary 32, we relate robustness to the concept of δ-decidability in [15] and also the existence of some witness of non-reachability.

Given $\mathbf{x}$, $R^{\mathcal{P}}(\mathbf{x})$ denotes the set of the points $\mathbf{y}$ reachable from $\mathbf{x}$: $R^{\mathcal{P}}(\mathbf{x}) = \{\mathbf{y} | R^{\mathcal{P}}(\mathbf{x},\mathbf{y})\}$. This is also the smallest set such that $\mathbf{x} \in R^{\mathcal{P}}(\mathbf{x})$ and $\mathbf{f}(R^{\mathcal{P}}(\mathbf{x})) \subseteq R^{\mathcal{P}}(\mathbf{x})$.

▶ **Definition 26.** *$R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is said to be ϵ-far from being true if there is $\mathcal{R}^* \subseteq X$ so that*
1. *$\mathbf{x} \in \mathcal{R}^*$,*
2. *$\mathbf{f}_\epsilon(\mathcal{R}^*) \subseteq \mathcal{R}^*$,*
3. *$\mathbf{y} \notin \mathcal{R}^*$.*

When this holds, we have $\neg R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$. Indeed, for all $\epsilon > 0$, $R_\epsilon^{\mathcal{P}}(\mathbf{x}) = \{\mathbf{y} | R_\epsilon^{\mathcal{P}}(\mathbf{x},\mathbf{y})\}$ is the smallest set satisfying $\mathbf{x} \in R_\epsilon^{\mathcal{P}}(\mathbf{x})$ and $\mathbf{f}_\epsilon(R_\epsilon^{\mathcal{P}}(\mathbf{x})) \subseteq R_\epsilon^{\mathcal{P}}(\mathbf{x})$. Thus, as $\mathcal{R}^*$ also satisfies these properties by the first two conditions, $R_\epsilon^{\mathcal{P}}(\mathbf{x}) \subseteq \mathcal{R}^*$ and hence $\mathbf{y} \notin R^{\mathcal{P}}(\mathbf{x})$ as $R^{\mathcal{P}}(\mathbf{x}) \subseteq R_\epsilon^{\mathcal{P}}(\mathbf{x}) \subseteq \mathcal{R}^*$. Then $\mathbf{y} \notin \mathcal{R}^*$ from the third condition.

In other words, $\mathcal{R}^*$ is a *witness* of the non-reachability of $\mathbf{y}$ from $\mathbf{x}$. We will say that it is *at level* ϵ. This provides a relation to δ-decidability considered in [15]:

▶ **Proposition 27** (Robust ⇔ Reachability relation is true or ϵ-far from being true). *We have $R_\omega^{\mathcal{P}} = R^{\mathcal{P}}$ if and only if for all $\mathbf{x},\mathbf{y} \in \mathbb{Q}^d$, either*
1. *$R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is true*
2. *or $R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is false and there exists $\epsilon > 0$ such that it is ϵ-far from being true.*

Proof.

(⇒): For all $\epsilon > 0$, $R_\epsilon^{\mathcal{P}}(\mathbf{x})$ satisfies $\mathbf{x} \in R_\epsilon^{\mathcal{P}}(\mathbf{x})$ and $\mathbf{f}_\epsilon(R_\epsilon^{\mathcal{P}}(\mathbf{x})) \subseteq R_\epsilon^{\mathcal{P}}(\mathbf{x})$ (this is even the smallest set such that this holds). Let $\mathbf{y} \in \mathbb{Q}^d$, let us assume that $R^{\mathcal{P}}(\mathbf{x},\mathbf{y}) = R_\omega^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is not true. Then, there exists ϵ such that $R_\epsilon^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is false, i.e. $\mathbf{y} \notin R_\epsilon^{\mathcal{P}}(\mathbf{x})$. Consider $\mathcal{R}^* = R_\epsilon^{\mathcal{P}}(\mathbf{x})$. Then, $\mathbf{x} \in \mathcal{R}^*$ and from the first paragraph $\mathbf{f}_\epsilon(\mathcal{R}^*) \subseteq \mathcal{R}^*$ and $\mathbf{y} \notin \mathcal{R}^*$.

($\Leftarrow$): When $R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is true, for all $\epsilon > 0$, $R^{\mathcal{P}}_{\epsilon}(\mathbf{x},\mathbf{y})$ is true, so $R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y})$ is. When $R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is false, by hypothesis, $R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ is ϵ-far from being true for some $\epsilon > 0$: there exists a set $\mathcal{R}^*$ satisfying $\mathbf{x} \in \mathcal{R}^*$ and $\mathbf{f}_\epsilon(\mathcal{R}^*) \subseteq \mathcal{R}^*$. As $R^{\mathcal{P}}_{\epsilon}(\mathbf{x})$ is the smallest such set, $R^{\mathcal{P}}_{\epsilon}(\mathbf{x}) \subseteq \mathcal{R}^*$. As $\mathbf{y} \notin \mathcal{R}^*$, $\mathbf{y} \notin R^{\mathcal{P}}_{\epsilon}(\mathbf{x})$. Hence $R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y})$ is false. ◀

We say that a subset R^* of X is ϵ-rejecting (with respect to $\mathbf{y}$) if it satisfies 2. and 3. of Definition 26: that is to say, $\mathbf{f}_\epsilon(R^*) \subseteq R^*$, and $\mathbf{y} \notin R^*$. A trajectory reaching such a R^* will never leave it.

▶ **Definition 28.** *A system is* eventually decisional *if for all* $\mathbf{x}$, $\mathbf{y}$, *there is some* R^* *ϵ-rejecting (with respect to* $\mathbf{y}$*) so that either the trajectory starting from* $\mathbf{x}$ *reaches* $\mathbf{y}$ *or, when not, it reaches* R^**.*

We come back to the converse of Corollary 25: from Proposition 27, a robust dynamical system (i.e. $R^{\mathcal{P}}_{\omega} = R^{\mathcal{P}}$) is eventually decisional, by considering $R^* = \mathcal{R}^*$ for the $\mathcal{R}^*$ given by item 2) there. Conversely:

▶ **Lemma 29.** *Take* $\mathbf{x}$ *and* $\mathbf{y}$ *with* $R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y})$ *but not* $R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$*. For* $\mathbf{f}$ *Lipschitz, the trajectory starting from* $\mathbf{x} \in \mathbb{Q}^d$ *can not reach any ϵ-rejecting subset.*

Proof. By contradiction, assume the trajectory starting from $\mathbf{x}$ reaches an ϵ-rejecting R^*. By considering one more step, we can assume that it reaches the interior of R^* for the first time at t, since, if it reaches the frontier at $\mathbf{x}^*$, $B(\mathbf{f}(\mathbf{x}^*),\epsilon) \subseteq R^*$ and $\mathbf{f}(\mathbf{x}^*)$ is in the interior of that ball. From $\mathbf{x}$ the position at time t remains at a positive distance of $\mathbf{y}$. As $\mathbf{f}$ is Lipschitz, the t-th iteration of $\mathbf{f}$ is. So, there exists $0 < \epsilon' < \epsilon$ taken sufficiently small so that $R^{\mathcal{P}}_{\epsilon'}$ intersects the interior of R^* and remains at a positive distance of $\mathbf{y}$. Once in R^*, ϵ'-perturbed trajectories stay in it ($\epsilon' < \epsilon$). We get $\mathbf{y} \notin R^{\mathcal{P}}_{\epsilon'}$. Thus $\neg R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y})$: contradiction. ◀

▶ **Corollary 30.** *Consider a Lipschitz rational dynamical system. It is robust iff it is eventually decisional.*

We can even compute the witnesses under the hypotheses of Theorem 23. A dynamical system is *effectively eventually decisional* when there is an algorithm such that, given $\mathbf{x}$ and $\mathbf{y}$, it outputs an R^* in the form of the union of rational balls. We can reinforce Corollary 25:

▶ **Proposition 31.** *Assume the hypotheses of Theorem 23. If* $R^{\mathcal{P}}_{\omega} = R^{\mathcal{P}}$ *then* $R^{\mathcal{P}}$ *is computable and the system is effectively eventually decisional.*

Proof. The proof of Theorem 23 shows that when $R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y})$ is false, then $R^{\mathcal{P}}_{\epsilon}(\mathbf{x},\mathbf{y})$ is false for some $\epsilon = 2^{-n}$ and there is a $\delta = 2^{-m}$ and some graph G_m with vertices $\mathcal{V}_i$ and $\mathcal{V}_j$, $\mathbf{x} \in \mathcal{V}_i$, $\mathbf{y} \in \mathcal{V}_j$ and $\neg(\mathcal{V}_i \xrightarrow{*}_{\delta} \mathcal{V}_j)$. Denote by R^{G_m} the union of the vertices $\mathcal{V}_k$ such that $\mathcal{V}_i \xrightarrow{*}_{\delta} \mathcal{V}_k$, $\mathbf{x} \in \mathcal{V}_i$ in G_m. Consider $\mathcal{R}^* = R^{G_m}$: this is a witness at level $\delta = 2^{-m}$ from the properties of the construction. Then m can be found by testing increasing m until a proper graph is found. The corresponding $\mathcal{R}^* = R^{G_m}$ of the first graph found will be a witness at level $\delta = 2^{-m}$. ◀

The reachability relation of an effectively eventually decisional system is necessarily decidable (given $\mathbf{x}$ and $\mathbf{y}$, compute the path until it reaches $\mathbf{y}$ (then accept), or R^* (then reject)):

▶ **Corollary 32** (Decidable $\Leftrightarrow$ Robust, for eventually decisional systems)**.** *Under the hypotheses of Theorem 23,* $R^{\mathcal{P}}$ *is robust iff* $R^{\mathcal{P}}$ *is decidable and* $R^{\mathcal{P}}$ *is effectively eventually decisional iff* $R^{\mathcal{P}}_{\omega}$ *is effectively eventually decisional.*

4.1.2 Complexity issues

Assume the dynamical system is robust. Hence, for all $\mathbf{x}, \mathbf{y} \in \mathbb{Q}$, there exists ϵ (depending on $\mathbf{x}$, $\mathbf{y}$) such that $R^{\mathcal{P}}(\mathbf{x}, \mathbf{y})$ and $R^{\mathcal{P}}_{\epsilon}(\mathbf{x}, \mathbf{y})$ have the same truth value (unchanged by smaller ϵ). It is then natural to quantify the level of required robustness according to $\mathbf{x}$ and $\mathbf{y}$, i.e. on the value ϵ. As we may always assume $\epsilon = 2^{-n}$ for some $n \in \mathbb{N}$, we write $R^{\mathcal{P}}_n$ for $R^{\mathcal{P}}_{\epsilon=2^{-n}}$ and we introduce:

▶ **Definition 33** (Level of robustness ϵ given by s)**.** *Given a function $s : \mathbb{N} \to \mathbb{N}$, we write ${R^{\mathcal{P}}}_{\{s\}}$ for the relation defined as: for any rational points $\mathbf{x}$ and $\mathbf{y}$ the relation holds iff $R^{\mathcal{P}}_{s(\ell(\mathbf{x})+\ell(\mathbf{y}))}(\mathbf{x}, \mathbf{y})$.*

A robust dynamical system is necessarily s-robust for some function s, according to the next definition: this follows from exactly the same arguments as the ones we used for the related concepts for Turing machines. This function s quantifies the tolerated level of robustness.

▶ **Definition 34** (s-robust language)**.** *We say that a dynamical system is s-robust, when $R^{\mathcal{P}} = {R^{\mathcal{P}}}_{\{s\}}$.*

We can then naturally consider the case where s is a polynomial: considering robustness to polynomial perturbations corresponds to PSPACE:

▶ **Theorem 35.** *Consider a locally Lipschitz $\mathbb{Q}$-computable system, with $\mathbf{f} : \mathbb{Q} \to \mathbb{Q}$ computable in polynomial time, whose domain X is a closed rational box. Given some polynomial p, ${R^{\mathcal{P}}}_{\{p\}} \in$ PSPACE.*

Proof. From Theorem 23, for all n there exists some m (depending on n), such that $R^{\mathcal{P}}_n(\mathbf{x}, \mathbf{y})$ and $R^{G_m}(\mathbf{x}, \mathbf{y})$ have the same truth value, where R^{G_m} denotes reachability in the graph G_m. With the hypotheses, given $\mathbf{x}$ and $\mathbf{y}$, we can determine whether ${R^{\mathcal{P}}}_{\{p\}}(\mathbf{x}, \mathbf{y})$, by determining the truth value of $R^{\mathcal{P}}_n(\mathbf{x}, \mathbf{y})$, taking n polynomial in $\ell(\mathbf{x}) + \ell(\mathbf{y})$. From Theorem 23, the corresponding m is linearly related to n. The analysis of Corollary 12 shows that the truth value of $R^{G_m}(\mathbf{x}, \mathbf{y})$ can be determined in space polynomial in m. ◀

▶ **Theorem 36** (Polynomially robust to precision $\Rightarrow$ PSPACE)**.** *With the same hypotheses, if $R^{\mathcal{P}} = {R^{\mathcal{P}}}_{\{p\}}$ for some polynomial p, then $R^{\mathcal{P}} \in$ PSPACE.*

This is even a characterisation of PSPACE:

▶ **Theorem 37** (Polynomially robust to precision $\Leftrightarrow$ PSPACE)**.** *Any PSPACE language is reducible to PAM's reachability relation: $R^{\mathcal{P}} = {R^{\mathcal{P}}}_{\{p\}}$, for some polynomial p.*

Assuming the hypotheses of Theorem 36, when $R^{\mathcal{P}} = {R^{\mathcal{P}}}_{\{p\}}$ for some polynomial p, we also see that we can determine a witness of $\neg R^{\mathcal{P}}(\mathbf{x}, \mathbf{y})$ in polynomial space (using a suitable representation of it).

4.2 The case of computable systems

We now consider the case of general discrete-time dynamical systems. Then $\mathbf{f}$ may take some non-rational values and we need the notion of computability of functions over the reals: this requires the model of computable analysis: see e.g. [32] or [12] for full presentations.

We review the most basic ideas of computable analysis in the next subsection.

4.2.1 Some basics of computable analysis

The idea behind classical computability and complexity is to fix some representations of objects (such as graphs, integers, etc, ...) using finite words over some finite alphabet, say $\Sigma = \{0,1\}$ and to say that such an object is computable when such a representation can be produced using a Turing machine. The computable analysis is designed to be able to also talk about objects such as real numbers, functions over the reals, closed subsets, compacts subsets, ..., which cannot be represented by finite words over Σ (a clear reason for it is that such words are countable while the set $\mathbb{R}$, for example, is not). However, they can be represented by some infinite words over Σ and the idea is to fix such representations for these various objects, called *names*, with suitable computable properties. In particular, in all the following proposed representations, it can be proved that an object is computable iff it has some computable representation.

▶ Remark 38. Here the notion of computability involved is one of Type 2 Turing machines, that is to say, computability over possibly infinite words: the idea is that such a machine has some read-only input tape(s), that contains the input(s), which can correspond to either a finite or infinite word(s), a read-write working tape and one (or several) write-only output tape(s). It evolves as a classical Turing machine, the only difference being that we consider it outputs an infinite word when it writes forever the symbols of that word on its (or one of its) write-only infinite output tape(s): see [32] for details.

A name for a point $\mathbf{x} \in \mathbb{R}^d$ is a sequence (I_n) of nested open rational balls with $I_{n+1} \subseteq I_n$ for all $n \in \mathbb{N}$ and $\{x\} = \bigcap_{n\in\mathbb{N}} I_n$. Such a name can be encoded as an infinite sequence of symbols.

We call a real function $f :\subseteq \mathbb{R} \to \mathbb{R}$ computable, iff some (Type 2 Turing) machine maps any name of any $x \in \mathrm{dom}(f)$ to a name of $f(x)$. For real functions $\mathbf{f} :\subseteq \mathbb{R}^n \to \mathbb{R}$ we consider machines reading n names in parallel.

It can be proved that a computable function is necessarily continuous. A name for a function $\mathbf{f}$ is a list of all pairs of open rational balls (I, J) such that $\mathbf{f}(cls(I)) \subseteq J$. Following the above remark, one can prove that a real function is computable iff it has some computable name.

A name for a closed set F is a sequence (I_n) of all open rational balls such that $cls(I_n) \cap F = \emptyset$ and a sequence (J_n) of all open rational balls such that $J_n \cap F \neq \emptyset$.

Given some closed set F, the distance function $d_F : \mathbb{R}^n \to \mathbb{R}$ is defined by $d_F(x) := \inf_{y\in F} d(x,y)$. Closed subset $F \subseteq \mathbb{R}^n$ is computable iff its distance function $d_A : \mathbb{R}^n \to \mathbb{R}$ is ([32, Corollary 5.1.8]). A name for a compact K is a name of F as a closed set and an integer L such that $K \subseteq B(0, L)$.

A closed set is called computably-enumerable closed if one can effectively enumerate the rational open balls intersecting it: $\{(q,\epsilon) \in \mathbb{Q}^n \times \mathbb{Q}_+ \mid \mathrm{B}(q,\epsilon) \cap A \neq \emptyset\}$ is computably enumerable ([12, Definition 5.13],[32, Definition 5.1.1]). A closed set is called co-computably-enumerable closed if one can effectively enumerate the rational closed balls in its complement: the set $\left\{(q,\epsilon) \in \mathbb{Q}^n \times \mathbb{Q}_+ \mid \overline{\mathrm{B}}(q,\epsilon) \subseteq U\right\}$ is computably enumerable ([12, Definition 5.10],[32, Definition 5.1.1]).

We need also the concept of polynomial time computable function in computable analysis: see [21]. In short, a quickly converging name of $\mathbf{x} \in \mathbb{R}^d$ is a name of $\mathbf{x}$, with I_n of radius $< 2^{-n}$. A function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^{d'}$ is said to be computable in polynomial time, if there is some oracle TM M, such that, for all $\mathbf{x}$, given any fast converging name of $\mathbf{x}$ as an oracle, given n, M produces some open rational ball of radius $< 2^{-n}$ containing $\mathbf{f}(\mathbf{x})$, in a time polynomial in n.

4.2.2 Computable systems

A system is said computable if the function $\mathbf{f}:\mathbb{R}^d \to \mathbb{R}^d$ is. From the model of computable analysis, given the name of $\mathbf{f}$, $\mathbf{x},\mathbf{y} \in \mathbb{Q}$, it is impossible in general to tell effectively if $\mathbf{f}(\mathbf{x}) = \mathbf{y}$. Thus, given some rational ball $B(\mathbf{y},\delta)$, we have to forbid "frontier reachability": $B(\mathbf{y},\delta)$ would not be reachable, but its frontier $\overline{B}(\mathbf{y},\delta) - B(\mathbf{y},\delta)$ would. A natural question arises: given some rational ball with the promise that either $B(\mathbf{y},\delta)$ is reachable (that case implies that $\overline{B}(\mathbf{y},\delta)$ is), or that $\overline{B}(\mathbf{y},\delta)$ is not, decide which possibility holds. We call this the *ball (decision) problem.* From definitions from CA, when $R^{\mathcal{P}}(\mathbf{x})$ is a closed set, $R^{\mathcal{P}}(\mathbf{x})$ is a computable closed set iff its associated ball problem is algorithmically solvable.

For computable systems, the ball decision problem is c.e: we mean, there is a Turing machine whose halting set intersected with the rational balls satisfying the promise is the set of positive instances. Indeed, just simulate the system's evolution, starting from $\mathbf{x}$ until step T, with increasing precision and T, until one finds the guarantee that $\mathbf{x}_T$ at time T remains in $B(\mathbf{y},\delta')$ for some $\delta' < \delta$. If the ball is reachable, it will terminate by computing a sufficient approximation of the corresponding $\mathbf{x}_T$. It cannot terminate without guaranteeing reachability. It is not co-c.e. in general.

▶ Remark 39. Our framework for discussing the computability of sets is similar to the concept of a maximally partially decidable set, formalised in [24, 25]. Similar ideas have also been implicitly used in many other articles considering various real problems, using computable analysis. A similar formalisation is also considered in [5].

To a discrete-time system, we can also associate its reachability relation $R^{\mathcal{P}}(\cdot,\cdot,\cdot)$ over $\mathbb{Q}^d \times \mathbb{Q}^d \times \mathbb{N}$. For two points $\mathbf{x},\mathbf{y} \in \mathbb{Q}$, $\eta = 2^{-p}$, encoded by $p \in \mathbb{N}$, $R^{\mathcal{P}}(\mathbf{x},\mathbf{y},p)$ iff there exists a trajectory of $\mathcal{P}$ from $\mathbf{x}$ to $\overline{B}(\mathbf{y},\eta)$. We define $R^{\mathcal{P}}_{\epsilon}$ similarly and $R^{\mathcal{P}}_{\omega} = \bigcap_{\epsilon} R^{\mathcal{P}}_{\epsilon}$. This relation encodes reachability with arbitrarily small perturbing noise to some closed ball.

▶ **Lemma 40.** *For any $0 < \epsilon_2 < \epsilon_1$ and any $\mathbf{x}$ and $\mathbf{y}$, η, the following implications hold: $R^{\mathcal{P}}(\mathbf{x},\mathbf{y},p) \Rightarrow R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y},p) \Rightarrow R^{\mathcal{P}}_{\epsilon_2}(\mathbf{x},\mathbf{y},p) \Rightarrow R^{\mathcal{P}}_{\epsilon_1}(\mathbf{x},\mathbf{y},p)$.*

Given $\mathbf{x}$ and $0 < \epsilon_2 < \epsilon_1$, $R^{\mathcal{P}}(\mathbf{x}) \subseteq R^{\mathcal{P}}_{\epsilon_2}(\mathbf{x}) \subseteq cls(R^{\mathcal{P}}_{\epsilon_2}(\mathbf{x})) \subseteq R^{\mathcal{P}}_{\epsilon_1}(\mathbf{x}) \subseteq cls(R^{\mathcal{P}}_{\epsilon_1}(\mathbf{x}))$. Hence, $R^{\mathcal{P}}_{\omega}(\mathbf{x}) = \bigcap_{\epsilon>0} R^{\mathcal{P}}_{\epsilon}(\mathbf{x}) = \bigcap_{\epsilon>0} cls(R^{\mathcal{P}}_{\epsilon}(\mathbf{x}))$ is a closed set.

▶ **Theorem 41** (Perturbed reachability is co-r.e.)**.** *Consider a locally Lipschitz computable system whose domain X is a computable compact. $R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y},p) \subseteq \mathbb{Q}^d \times \mathbb{Q}^d \times \mathbb{N}$ is in Π_1.*

This can be considered as extending [9, Theorem 13], established in a very simpler framework.

▶ **Corollary 42** (Robust $\Rightarrow$ decidable)**.** *Assume Theorem 41's hypotheses and that for all rational $\mathbf{x}$, $R^{\mathcal{P}}(\mathbf{x})$ is closed and $R^{\mathcal{P}}(\mathbf{x}) = R^{\mathcal{P}}_{\omega}(\mathbf{x})$. Then, the ball decision problem is decidable.*

Proof. Given some instance $B(\mathbf{y},\delta)$ of the ball problem, run in parallel the c.e. algorithm for it (and when its termination is detected, accepts) and the c.e. algorithm for $\left(R^{\mathcal{P}}(\mathbf{x})\right)^c = \left(R^{\mathcal{P}}_{\omega}(\mathbf{x})\right)^c$ (and when its termination is detected, rejects). ◀

4.2.3 Complexity issues

▶ **Definition 43** (Level of robustness ϵ given by s)**.** *Given some function $s:\mathbb{N}\to\mathbb{N}$, we write $R^{\mathcal{P}}{}_{\{s\}}$ as: for two rational points $\mathbf{x}$ and $\mathbf{y}$ and p, the relation holds iff $R^{\mathcal{P}}_{s(\ell(\mathbf{x})+\ell(\mathbf{y})+p)}(\mathbf{x},\mathbf{y},p)$.*

As before, a robust dynamical system is necessarily s-robust for some function s, according to the next definition. The function s quantifies the tolerated level of robustness.

▶ **Definition 44** (s-robust language). *We say that a dynamical system is s-robust, when* $R^{\mathcal{P}} = R^{\mathcal{P}}{}_{\{s\}}$.

▶ **Theorem 45.** *Take a locally Lipschitz system, with* $\mathbf{f}$ *polynomial time computable, whose domain* X *is a closed rational box. Then* $R^{\mathcal{P}}{}_{\{p\}} \subseteq \mathbb{Q}^d \times \mathbb{Q}^d \times \mathbb{N} \in \mathrm{PSPACE}$*, when* p *is a polynomial.*

Proof. The proof of Theorem 41 (like for Theorem 23) shows that when $R^{\mathcal{P}}_{\omega}(\mathbf{x}, \mathbf{y}, p)$ is false, then $R^{\mathcal{P}}_{\epsilon}(\mathbf{x}, \mathbf{y}, p)$ is false for some $\epsilon = 2^{-n}$. With the hypotheses, given $\mathbf{x}$, $\mathbf{y}$ and p, we take n polynomial in $\ell(\mathbf{x}) + \ell(\mathbf{y}) + p$. The corresponding m is polynomially related to n (linear in n). An analysis similar to Theorem 35, shows the truth value of $R^{G_m}(\mathbf{x}, \mathbf{y}, p)$ can be determined in space polynomial in m. ◀

Then, once again:

▶ **Theorem 46** (Polynomially robust to precision $\Rightarrow$ PSPACE). *Assuming Theorem 45's hypotheses, and that for all rational* $\mathbf{x}$*,* $R^{\mathcal{P}}(\mathbf{x})$ *is closed and* $R^{\mathcal{P}}(\mathbf{x}) = R^{\mathcal{P}}{}_{\{p\}}$ *for a polynomial* p*. Then the ball decision problem is in* PSPACE.

5 Relating robustness to drawability

We can go further and prove geometric properties: in the previous sections, we associated with every discrete-time dynamical system a reachability relation over the rationals. But we could also see it as a relation over the reals and use the framework of computable analysis, regarding subsets of $\mathbb{R}^d \times \mathbb{R}^d$. From the statements of [32], the following holds:

▶ **Theorem 47.** *Consider a computable discrete-time system* $\mathcal{P}$ *whose domain is a computable compact. For all computable* $\mathbf{x}$*,* $cls(R^{\mathcal{P}}(\mathbf{x})) \subseteq \mathbb{R}^d$ *is a c.e. closed subset.*

A closed set is called *co-c.e. closed* if we can effectively enumerate the rational closed balls in its complement. Using proofs similar to Theorems 41 and 23:

▶ **Theorem 48.** *Consider a computable locally Lipschitz discrete-time system whose domain* X *is a computable compact. For all computable* $\mathbf{x}$*,* $cls(R^{\mathcal{P}}_{\omega}(\mathbf{x})) \subseteq \mathbb{R}^d$ *is a co-c.e. closed subset.*

▶ **Corollary 49** (Robust $\Rightarrow$ computable)). *Assume the Theorem 48's hypotheses. If* $R^{\mathcal{P}}$ *is robust then for all computable* $\mathbf{x}$*,* $cls(R^{\mathcal{P}}(\mathbf{x})) \subseteq \mathbb{R}^d$ *is computable.*

For closed sets, the notion of computability can be interpreted as the possibility of being plotted with an arbitrarily chosen precision: $\mathbf{z}/2^n$ corresponds to a pixel at precision 2^{-n}, 1 is black (the pixel is plotted black), 0 is white (the pixel is plotted white).

▶ **Theorem 50** ([12]). *For a closed set* $A \subseteq \mathbb{R}^k$*,* A *is computable iff it can be plotted: there exists a computable function* $f : \mathbb{N} \times \mathbb{Z}^k \to \mathbb{N}$ *with* $range(f) \subseteq \{0, 1\}$ *and such that for all* $n \in \mathbb{N}$ *and* $\mathbf{z} \in \mathbb{Z}^k$

$$f(n, \mathbf{z}) = \begin{cases} 1 & \text{if } B(\frac{\mathbf{z}}{2^n}, 2^{-n}) \cap A \neq \emptyset, \\ 0 & \text{if } B(\frac{\mathbf{z}}{2^n}, 2.2^{-n}) \cap A = \emptyset, \\ 0 \text{ or } 1 & \text{otherwise.} \end{cases}$$

▶ **Corollary 51** (Robust $\Rightarrow$ drawable)). *Assume Theorem 48's hypotheses. If* $R^{\mathcal{P}}$ *is robust then for all computable* $\mathbf{x}$*,* $cls(R^{\mathcal{P}}(\mathbf{x})) \subseteq \mathbb{R}^d$ *can be plotted.*

This is even effective in the name of $\mathbf{x}$ and $\mathbf{f}$. The converse holds with additional topological properties.

▶ **Theorem 52.** *Assume $R^{\mathcal{P}}$ is closed and can be plotted effectively in the name of $\mathbf{x}$ and $\mathbf{f}$. Then the system is robust, i.e. $R^{\mathcal{P}}_{\omega} = R^{\mathcal{P}}$.*

We prove a stronger statement: if $cls(R^{\mathcal{P}})$ can be plotted effectively in a name of $\mathbf{x}$ and $\mathbf{f}$, then $R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y}) = R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ except maybe for $(\mathbf{x},\mathbf{y}) \in cls(R^{\mathcal{P}}) - R^{\mathcal{P}}$.

Proof. By Theorem 50, $cls(R^{\mathcal{P}})$ is computable which is equivalent to the computability of the distance function $d(\cdot, cls(R^{\mathcal{P}}))$ [32, Corollary 5.1.8]. It means that given a rational ball, a name for $\mathbf{x}$ and $\mathbf{y}$, with $\neg R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$, the following procedure terminates when $(\mathbf{x},\mathbf{y}) \notin cls(R^{\mathcal{P}}) - R^{\mathcal{P}}$: compute a name of $d((\mathbf{x},\mathbf{y}), cls(R^{\mathcal{P}}(\mathbf{x})))$ until a strictly positive proof is found: $d((\mathbf{x},\mathbf{y}), cls(R^{\mathcal{P}}(\mathbf{x}))) = 0$ would mean $(\mathbf{x},\mathbf{y}) \in cls(R^{\mathcal{P}})$, but not in $R^{\mathcal{P}}$.

It answers by reading $m \in \mathbb{N}$ cells of the names of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{f}$. It returns the same if the names are altered after m symbols. Thus, there exists a precision ϵ (related to m, usually 2^{-m} for exponentially fast convergence) so $\neg R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ remains true for an ϵ-neighborhood of $\mathbf{x}$ and $\mathbf{y}$ and unchanged by a small variation of $\mathbf{f}$. Hence, for all $\mathbf{x}$, $\mathbf{y}$, when $\neg R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$, there exists some ϵ such that $\neg R^{\mathcal{P}}_{\epsilon}(\mathbf{x},\mathbf{y})$ ($\neg R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y})$). When $R^{\mathcal{P}}(\mathbf{x},\mathbf{y})$ holds, $R^{\mathcal{P}}_{\omega}(\mathbf{x},\mathbf{y})$ holds. ◀

6 Continuous-time systems

The previous ideas can be extended to continuous-time or hybrid systems.

▶ **Definition 53.** *A* continuous-time dynamical system $\mathcal{P}$ *is given by a set $X \subseteq \mathbb{R}^d$ and some Ordinary Differential Equation of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ on X.*

The maximal interval of existence of solutions can be non-computable, even for computable Ordinary Differential Equations (ODEs) [16]. To simplify, we assume the ODEs have solutions[4] defined over all $\mathbb{R}$. A trajectory of $\mathcal{P}$ starting at $\mathbf{x}_0 \in X$ is a solution of the differential equation with initial condition $\mathbf{x}(0) = \mathbf{x}_0$, defined as a continuous right-derivable function $\xi : \mathbb{R}^+ \to X$ such that $\xi(0) = \mathbf{f}(\mathbf{x}_0)$ and for every t, $\mathbf{f}(\xi(t))$ is equal to the right-derivative of $\xi(t)$. To each continuous-time dynamical system $\mathcal{P}$ we associate its reachability relation $R^{\mathcal{P}}$ as before.

For any $\epsilon > 0$, the ϵ-perturbed system $\mathcal{P}_{\epsilon}$ is described by the differential inclusion $d(\dot{\mathbf{x}}, \mathbf{f}(\mathbf{x})) < \epsilon$. This non-deterministic system can be seen as $\mathcal{P}$ submitted to a noise of magnitude ϵ. We denote reachability in the system $\mathcal{P}_{\epsilon}$ by $R^{\mathcal{P}}_{\epsilon}$. The limit reachability relation $R^{\mathcal{P}}_{\omega}$ is introduced as before.

▶ **Theorem 54** (Perturbed reachability is co-r.e.). *Consider a continuous-time dynamical system, with $\mathbf{f}$ locally Lipschitz, computable, whose domain is a computable compact, then, for all computable $\mathbf{x}$, $cls(R^{\mathcal{P}}_{\omega}(\mathbf{x})) \subseteq \mathbb{R}^d$ is a co-c.e. closed subset.*

Proof. Its proof can be considered as the main technical result established in [26]. An alternative proof is similar to Theorems 41 and 23: adapt the construction of the involved graph G_m to cover the flow of the trajectory. With our hypotheses, the solutions are defined over all $\mathbb{R}$. It is proved in [16] that Lipschitz (and even effectively locally Lipschitz) homogeneous computable ODEs have computable solutions over their maximal domain. ◀

[4] A non-total solution must necessarily leave any compact, see e.g. [17], so X is compact is not a restriction.

▶ **Corollary 55** (Robust ⇒ decidable). *Assume the hypotheses of Theorem 54. If $R^{\mathcal{P}}$ is robust then for all computable* $\mathbf{x}$*, $cls(R^{\mathcal{P}}(\mathbf{x})) \subseteq \mathbb{R}^d$ is computable.*

7 Other perturbations

Inspired by analogue computations [8], when time has been related to the length of trajectories, we can also consider time or length-perturbations.

Time-perturbation. We can start by considering time-perturbed TM . The idea is that given $n > 0$, the n-perturbed version of $\mathcal{M}$ is unable to remain correct after a time n. Given $n > 0$, the n-perturbed version of $\mathcal{M}$, is defined exactly likewise, except after a time greater than n, its internal state q can change in a non-deterministic manner. The associated language is $L^n(\mathcal{M})$. From definitions: $L(\mathcal{M}) \subseteq L^{\omega}(\mathcal{M}) \subseteq \cdots \subseteq L^2(\mathcal{M}) \subseteq L^1(\mathcal{M})$.

▶ **Theorem 56** (Length robust ⇒ decidable). *$L^{\omega}(\mathcal{M})$ is in the class Π_1. Consequently, whenever $L^{\omega}(\mathcal{M}) = L(\mathcal{M})$, $L(\mathcal{M})$ is decidable.*

▶ **Theorem 57.** *When M always stops, $L^{\omega}(\mathcal{M}) = L(\mathcal{M})$.*

▶ **Definition 58** (Level of robustness n given by t). *Given $t : \mathbb{N} \to \mathbb{N}$, we write $L^{\{t\}}(\mathcal{M})$ for the set of words accepted by $\mathcal{M}$ with time perturbation t: $L^{\{t\}}(\mathcal{M}) = \{w |\ w \in L^{t(\ell(w))}(\mathcal{M})\}$.*

A robust dynamical system is necessarily t-robust for some function t, according to the next definition:

▶ **Definition 59** (t-robust to time language). *We say that a robust language is t-robust to time, when $L = L(\mathcal{M}) = L^{\{t\}}(\mathcal{M})$.*

▶ **Theorem 60** (Polynomially robust to time ⇔ PTIME). *A language L is in* PTIME *iff for some $\mathcal{M}$ and some polynomial p, $L = L(\mathcal{M}) = L^{\{p\}}(\mathcal{M})$.*

Furthermore, any PTIME *language is reducible to PAM's reachability: $R^{\mathcal{P}} = L^{\{p\}}(\mathcal{P})$ for some polynomial p.*

Length-perturbation. As we said, inspired by analogue computations [8], we can also consider length perturbations: Fix a distance $\delta(\cdot, \cdot)$ over the domain X. A finite trajectory of a discrete-time dynamical system $\mathcal{P}$ is a finite sequence $(x_t)_{t \in 0 \dots T}$ such that $x_{t+1} = f(x_t)$ for all $0 \leq t < T$. Its associated *length* is defined as $\mathcal{L} = \sum_{i=0}^{T-1} \delta(x_i, x_{i+1})$. We consider a *length-perturbed* discrete-time dynamical system: given $L > 0$, the L-perturbed version of the system is unable to remain correct after a length L. We define $R^{\mathcal{P},L}(\mathbf{x}, \mathbf{y})$ as there exists a finite trajectory of $\mathcal{P}$ from $\mathbf{x}$ to $\mathbf{y}$ of length $\mathcal{L} \leq L$. When considering TMs as dynamical systems, $\delta(\cdot, \cdot)$ is a distance over configurations of TMs. Word w is said to be accepted in length d if the trajectory starting from $C_0[w]$ to the accepting configuration has length $\leq d$.

▶ **Definition 61.** *Distance $\delta(C, C')$ is called* time-metric *iff whenever C' is the configuration following configuration C, we have $\delta(C, C') \leq p(\ell(C))$, and $\delta(C, C') \geq \frac{1}{p(\ell(C))}$ for some polynomial p.*

Write $\mathcal{L}(M, t)$ for the set of words accepted by M in length less than t.

▶ **Definition 62** (Tolerating some level of robustness L given by f). *Given $f : \mathbb{N} \to \mathbb{N}$, we write $L^{(f)}(\mathcal{M})$ for $L^{(f)}(\mathcal{M}) = \{w |\ w \in \mathcal{L}(\mathcal{M}, f(\ell(w))\}$.*

▶ **Theorem 63** (Length robust for some time-metric distance ⇔ PTIME). *Assume $\delta(\cdot,\cdot)$ is time metric. Then, a language L is in $PTIME$ iff for some TM $\mathcal{M}$ and some polynomial $p(n)$, $L = L(\mathcal{M}) = L^{(f)}(\mathcal{M})$.*

One way to obtain a distance $\delta(C, C')$ is to take the Euclidean distance between $\Upsilon(C)$ and $\Upsilon(C')$ for $\gamma = \gamma_{[0,1]}$, where $\gamma_{[0,1]}$ and Υ are the functions considered in Section 3. The obtained distance is time metric. Given $f : \mathbb{N} \to \mathbb{N}$, we write $R^{\mathcal{P},(f)}$ for the set of words accepted by $\mathcal{M}$ with length perturbation f: $R^{\mathcal{M},(f)} = \{w | \ w \in R^{\mathcal{M}, f(\ell(w))}\}$.

▶ **Theorem 64** (Polynomially length robust ⇔ PTIME). *Assume distance d is time metric and $R^{\mathcal{P}} = R^{\mathcal{P},(p)}$ for some polynomial p. Then $R^{\mathcal{P}} \in$ PTIME.*

8 Conclusion and future work

In this article, we have proposed a unified theory explaining in a uniform framework various statements relating robustness, defined as being non-sensitive to infinitesimal perturbations, to decidability. Most of the statements in the spirit of the *"robustness conjecture"* have been established using arguments from computability over the rationals or the reals, playing with variations on the statement that a semi-computable and co-semi-computable set is decidable.

More importantly, while existing statements of this form were only at the level of decidability, we showed that it is possible to also talk about complexity: robustness to polynomial perturbations on precision corresponds to PSPACE, robustness to polynomial perturbations on time or length corresponds to PTIME.

We also related the approach of [1] to the concept of δ-decidability of [15], as well as the drawability of the associated dynamics.

Notice that the proposed approach can also cover the so-called hybrid systems without difficulties. Various models have been considered in the literature for such systems, but one common point is that they all correspond to continuous-time dynamical systems, where the dynamics might be discontinuous, so not computable. In a very general view, a *hybrid system* $\mathcal{P}$ is given by a set $X \subseteq \mathbb{R}^d$, a semi-group T and a flow function $\phi : X \times T \to X$ satisfying $\phi(\mathbf{x}, 0) = \mathbf{x}$ and $\phi(\phi(\mathbf{x}, t), t') = \phi(\mathbf{x}, t + t')$. Previous proofs use the fact that reachability $R^{\mathcal{P}}$ is c.e. and perturbed reachability is co-c.e. The former is usually obvious in any of the considered models, as we expect to be able to simulate the model. The latter is usually less trivial. If we look at our proof methods, we only need to construct some computable abstraction satisfying Claims 1 and 2. One key remark is that we need these properties not about the function $\mathbf{f}$ but its graph. Assuming a function such that the closure of its graph is computable, is more general than assuming computability. For example, the characteristic function $\chi_{[0,\infty)}$ is not computable, as it is not continuous. But its graph, as well as its closure, is easy to draw: see discussions e.g. in [13]. In particular, this allowed us to talk about discontinuous functions in the current article.

Regarding analogue models of computation, a variation on our concept of robustness has already been used to provide a characterisation of PSPACE for discrete-time ordinary differential equations in [3].

We believe that the theory developed here might be used to prove formally that space complexity corresponds to precision for continuous-time models of computation, over some compact domains, providing a more natural measure than the conditions considered in [7].

References

1 Eugene Asarin and Ahmed Bouajjani. Perturbed Turing machines and hybrid systems. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS-01)*, pages 269–278, Los Alamitos, CA, June 16–19 2001. IEEE Computer Society Press.

2 Eugene Asarin, Oded Maler, and Amir Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65, February 1995.

3 Manon Blanc and Olivier Bournez. A characterisation of functions computable in polynomial time and space over the reals with discrete ordinary differential equations: Simulation of Turing machines with analytic discrete odes. In *Mathematical Foundations of Computer Science (MFCS'2023)*, 2023.

4 Vincent Blondel and John Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

5 Olivier Bournez, Johanne Cohen, and Valentin Dardilhac. On the δ-decidability of decision problems for neural network questions. In *Computability, Continuity, Constructivity - from Logic to Algorithms CCC'23*, 2023.

6 Olivier Bournez, Felipe Cucker, Paulin Jacobé de Naurois, and Jean-Yves Marion. Computability over an arbitrary structure. sequential and parallel polynomial time. In Andrew D. Gordon, editor, *Foundations of Software Science and Computational Structures, 6th International Conference (FOSSACS'2003)*, volume 2620 of *Lecture Notes in Computer Science*, pages 185–199, Warsaw, 2003. Springer.

7 Olivier Bournez, Riccardo Gozzi, Daniel S Graça, and Amaury Pouly. A continuous characterization of PSPACE using polynomial ordinary differential equations. *Journal of Complexity*, 77:101755, August 2023. URL: `https://www.sciencedirect.com/science/article/pii/S0885064X23000249?dgcid)=author`.

8 Olivier Bournez, Daniel S. Graça, and Amaury Pouly. Polynomial Time corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length. *Journal of the ACM*, 64(6):38:1–38:76, 2017. `doi:10.1145/3127496`.

9 Olivier Bournez, Daniel S. Graça, and Emmanuel Hainry. Robust computations with dynamical systems. In *Mathematical Foundations of Computer Science, MFCS'2010*, volume 6281 of *Lecture Notes in Computer Science*, pages 198–208. Springer, 2010. `doi:10.1007/978-3-642-15155-2_19`.

10 Olivier Bournez and Emmanuel Hainry. An analog characterization of elementary computable functions over the real numbers. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *International Colloquium on Automata, Languages and Programming (ICALP 2004)*, volume 3142 of *Lecture Notes in Computer Science*, pages 269–280, 2004.

11 Olivier Bournez and Amaury Pouly. A survey on analog models of computation. In *Handbook of Computability and Complexity in Analysis*, pages 173–226. Springer, 2021.

12 Vasco Brattka, Peter Hertling, and Klaus Weihrauch. A tutorial on computable analysis. In *New computational paradigms*, pages 425–491. Springer, 2008.

13 Mark Braverman. Computational complexity of Euclidean sets: Hyperbolic Julia sets are poly-time computable. Master's thesis, University of Toronto, 2004.

14 Martin Fränzle. Analysis of hybrid systems: An ounce of realism can save an infinity of states. In Jörg Flum and Mario Rodríguez-Artalejo, editors, *Computer Science Logic, 13th International Workshop, CSL '99, 8th Annual Conference of the EACSL, Madrid, Spain, September 20-25, 1999, Proceedings*, volume 1683 of *Lecture Notes in Computer Science*, pages 126–140. Springer, 1999.

15 Sicun Gao, Jeremy Avigad, and Edmund M Clarke. Delta-decidability over the reals. In *Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on*, pages 305–314. IEEE, 2012.

16 Daniel S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability and undecidability of maximal intervals of IVPs. *Transactions of the American Mathematical Society*, 2006. To appear.

17 Philip Hartman. *Ordinary Differential Equations.* John Wiley and Sons, 1964.

18 Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, August 1998.

19 Thomas A. Henzinger and Jean-François Raskin. Robust undecidability of timed and hybrid systems. In Nancy A. Lynch and Bruce H. Krogh, editors, *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000, Pittsburgh, PA, USA, March 23-25, 2000, Proceedings*, volume 1790 of *Lecture Notes in Computer Science*, pages 145–159. Springer, 2000.

20 N. Immerman. Nondeterministic space is closed under complementation. In *Structure in Complexity Theory Conference, 1988. Proceedings., Third Annual*, pages 112–115. IEEE, 1988.

21 Ker-I Ko. *Complexity Theory of Real Functions.* Progress in Theoretical Computer Science. Birkhaüser, Boston, 1991.

22 Pascal Koiran, Michel Cosnard, and Max Garzon. Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132(1-2):113–128, September 1994.

23 Cristopher Moore. Generalized shifts: unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4(3):199–230, 1991.

24 Eike Neumann. Decision problems for linear recurrences involving arbitrary real numbers. *Logical Methods in Computer Science*, 17, 2021.

25 Eike Neumann. On the complexity of robust eventual inequality testing for C-finite functions. In *International Conference on Reachability Problems*, pages 98–112. Springer, 2023.

26 A. Puri, V. Borkar, and P. Varaiya. Epsilon-approximation of differential inclusions. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 2892–2897, 1995.

27 Anuj Puri. Dynamical properties of timed automata. *Discrete Event Dynamic Systems*, 10:87–113, 2000.

28 Stefan Ratschan. Deciding predicate logical theories of real-valued functions. In *Symposium on Mathematical Foundations of Computer Science (MFCS'2023)*, 2023.

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

30 Michael Sipser. *Introduction to the Theory of Computation.* PWS Publishing Company, 1997.

31 R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta informatica*, 26(3):279–284, November 1988. doi:10.1007/BF00299636.

32 Klaus Weihrauch. *Computable Analysis: an Introduction.* Springer, 2000.