Fundamenta Informaticae XX (2005) 1–25
IOS Press

Decidability and Universality in Symbolic Dynamical Systems

Jean-Charles Delvenne[1], **Petr Kůrka**[2], **and Vincent Blondel**[1]

[1] *Université Catholique de Louvain, Department of Mathematical Engineering,*
Avenue Georges Lemaitre 4, B-1348 Louvain-la-Neuve, Belgium
email: {delvenne,blondel}@inma.ucl.ac.be
[2]*Center for Theoretical Studies & Faculty of Mathematics and Physics,*
Charles University in Prague, Malostranské náměstí 25, CZ-11800 Praha 1, Czechia
email: kurka@ms.mff.cuni.cz

Abstract. Many different definitions of computational universality for various types of dynamical systems have flourished since Turing's work. We propose a general definition of universality that applies to arbitrary discrete time symbolic dynamical systems. Universality of a system is seen as a special case of undecidability of a model-checking problem. For Turing machines and tag systems, our definition coincides with the classical one. It yields, however, a new definition for cellular automata and subshifts. Our definition is robust with respect to initial condition. This is a desirable feature for physical realizability.

We derive necessary conditions for undecidability and universality. For instance, a universal system must have a sensitive point and a proper subsystem. We conjecture that universal systems have infinite number of subsystems. We also discuss the thesis that computation should occur at the 'edge of chaos' and we exhibit a universal chaotic system.

1. Introduction

Computability is usually defined via universal Turing machines. A Turing machine can be regarded as a dynamical system, i.e., a set of configurations together with a transformation acting on this set. A configuration consists of the state of the head and the content of the tape. Computation is done by observing the trajectory of an initial point under iterated transformation.

There is no reason why Turing machines should be the only dynamical systems capable of universal computation. Indeed, such capabilities have been also claimed for artificial neural networks [30], cellular automata [35], billiard balls on some pool tables, or a ray of light between a set of mirrors [24]. For all these systems, many particular definitions of universality have been advanced. Most of them mimic the definition of computation for Turing machines: an initial point is chosen, then we observe the trajectory of this point and see whether it reaches some 'halting' set; see for instance [31] and [3]. However, many variants of these definitions exist and lead to different classes of universal dynamical systems. In particular, there is no consensus for what it means for a cellular automaton to be universal. Moreover, in the presence of noise many of these systems loose their computational properties [1, 21, 10]. See [28, 27, 26] for some definitions of analog computation and issues relative to noise and physical realizability.

Another field of investigation is to make a link between the computational properties of a system and its dynamical properties. For instance, attempts have been made to relate 'universal' cellular automata to Wolfram's classification. It has also been suggested that a 'complex' system must be on the 'edge of chaos': this means that the dynamical behavior of such a system is neither simple (i.e., a globally attracting fixed point) nor chaotic; see [35, 23, 5, 20]. Other authors nevertheless argue that a universal system may be chaotic; see [30].

The basic questions we would like to address are the following:

- How to define computationally universality for dynamical systems?
- What are the dynamical properties of a universal system?

A long-term motivation is to answer these questions from the point of view of physics. What physical systems are universal? Is the gravitational N-body problem universal [24]? Is the Navier-Stokes equation universal [25]? However in this paper we focus on *symbolic effective* dynamical systems, i.e., systems defined on the Cantor set $\{0,1\}^{\mathbb{N}}$ or a subset of it, whose transformations are computable. Some motivating examples of such systems are Turing machines, cellular automata and subshifts.

Turing's machine was originally meant as a model of a computation performed by a human operator using paper and pencil [33]. We adapt Turing's reasoning to the case where the human operator does not compute by himself, but relies on a dynamical system to make the computation. The system is said to be computationally universal if the observations made by the human operator are enough to solve any problem that could also be solved by a universal Turing machine. We conclude that a system is universal if some property of its trajectories, such as reachability of a halting set, is r.e.-complete. This is an extension of Davis' definition of universal Turing machine.

However, rather than considering point-to-point or point-to-set properties, we consider set-to-set properties. Typically, given an initial set and a halting set, we want to know whether there is at least one configuration in the initial set whose trajectory eventually reaches the halting set. We require the initial and halting sets to be clopen (closed and open) sets of the Cantor state space. Clopen sets can be described in a natural way with a finite number of bits. Finally, we do not restrict ourselves to the property 'Is there a trajectory going from U to V?' alone. In a previous paper [7] we have considered properties expressible by temporal logic. In the present paper we consider the wider class of all properties that can be observed by some finite automaton. Checking whether such a property is verified is called a 'model-checking problem' in verification theory. Model-checking problems are usually defined for finite or countable systems; see [12] for instance. Finally, a universal system is a system for which a certain model-checking problem is r.e.-complete.

This definition addresses the two issues raised above. Firstly, it is a general definition directly applicable to any (effective) symbolic system. Secondly, dealing with clopen sets rather than points takes into account some constraints of physical realizability, such as robustness to noise. With this definition in mind, we prove necessary conditions for a symbolic system to be universal. In particular, we show that a universal symbolic system is not minimal, not equicontinuous and does not satisfy the shadowing property. We conjecture that a universal system must have infinitely many subsystems, and we show that there is a chaotic system that is universal, contradicting the idea that computation can only happen at the 'edge of chaos'.

Preliminaries are given in Sections 2, 3 and 4. Decidable and universal systems are defined in Section 5. In Section 6 necessary conditions for a system to be universal are given, related to minimality, equicontinuity and shadowing property; chaos and edge of chaos are considered in Section 7. The definition of universality is discussed in Section 8.

2. Effective symbolic spaces

A *symbolic space* is a compact metric space whose clopen (closed and open) sets form a countable basis: every open set is a union of clopen sets. The elements of a symbolic space are called *points* or *configurations*. A typical example is the Cantor set $\{0,1\}^{\mathbb{N}}$ endowed with the product topology. The topology is given by the metric $d(x,y) = 2^{-n}$, where n is the index of the first bit on which x and y differ. Note that this metric satisfies the *ultrametric inequality*: $d(x,z) \leq max(d(x,y), d(y,z))$.

If $w \in \{0,1\}^*$ is a finite binary word, then $[w]$ denotes the set of all infinite configurations with prefix w. In fact, sets of this form, usually called *cylinders*, are exactly the balls of the metric space. They are clopen sets and any clopen set of $\{0,1\}^{\mathbb{N}}$ is a finite union of cylinders. Similar distances are defined in the spaces $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$, $A^{\mathbb{N}}$, $Q \times A^{\mathbb{Z}}$, $A^{\mathbb{Z}^d}$ where Q and A are finite and d is a positive integer. Closed subsets of the Cantor space are symbolic spaces themselves. The converse is well known to hold as well: Every symbolic space is homeomorphic to a closed subset of the Cantor space and every perfect symbolic space is homeomorphic to the Cantor space. For instance, $\{0,1\}^{\mathbb{Z}}$ is homeomorphic to $\{0,1\}^{\mathbb{N}}$.

To define computational universality, we need effective symbolic spaces, in which we can perform boolean combinations on clopen sets effectively.

Definition 1. An *effective symbolic space* is a pair (X,P), where X is a symbolic space and $P : \mathbb{N} \to 2^X$ is an injective function whose range is the set of all clopen sets of X, such that the intersection and complementation of clopen sets are computable operations. This means that there exist computable functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $X \setminus P_n = P_{f(n)}$ and $P_n \cap P_m = P_{g(n,m)}$.

Of course, union of clopen sets is then also computable. Often we denote an effective symbolic space by X alone instead of (X,P) when no confusion is to be feared. In Cantor space $\{0,1\}^{\mathbb{N}}$, the lexicographic ordering yields a standard enumeration

$$[\lambda], [0], [1], [00], [01], [10], [11], [00] \cup [11], [01] \cup [10], [00] \cup [01] \cup [10], [00] \cup [01] \cup [11], \ldots$$

Other widely used symbolic spaces like $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$, $A^{\mathbb{N}^d}$, $A^{\mathbb{Z}^d}$, $Q \times A^{\mathbb{Z}}$, have also their standard enumerations. Note that we could require intersections and complements to be primitive recursive rather than computable, without altering the examples and results of the paper.

Definition 2. Let (X,P) and (Y,Q) be two effective symbolic spaces. An *effective continuous map* is a continuous map $h : X \to Y$ such that $h^{-1}(Q_n) = P_{k(n)}$, for some computable map $k : \mathbb{N} \to \mathbb{N}$. If h is bijective then it is an *effective homeomorphism*, and (X,P) is said to be *effectively homeomorphic* to (Y,Q).

Note that the composition of effective continuous maps is an effective continuous map, the identity is an effective continuous map and the inverse map of an effective homeomorphism is also an effective homeomorphism. In particular, being effectively homeomorphic is an equivalence relation for effective symbolic spaces.

Given an effective symbolic space (X, P), a closed subset Y is said to be *effective*, if the family of clopen sets intersecting Y is decidable. In particular any clopen set is effective. An effective set Y can be endowed with the relative topology, whose clopen sets are all intersections of clopen sets of X with Y. Thus, the enumeration $P_0, P_1, P_2, \ldots$ of clopen sets of X yields an enumeration of clopen sets of Y: $Y \cap P_0, Y \cap P_1, Y \cap P_2, \ldots$. This enumeration may contain empty sets and repetitions, but we can detect them in an effective way and renumber the sequence accordingly. Hence we get an effective topology for the effective closed set Y. Equivalently, the inclusion $i : Y \hookrightarrow X$ is an effective continuous map.

Proposition 1. Every effective symbolic space is effectively homeomorphic to an effective subset of the Cantor space. Every perfect effective symbolic space is effectively homeomorphic to the Cantor space.

Proof:
Let (X, P) be an effective symbolic space. For every point $x \in X$, construct the infinite configuration $g(x) \in \{0,1\}^{\mathbb{N}}$, where $g(x)_n = 1$ if and only if $x \in P_n$. Then the map $g : X \to \{0,1\}^{\mathbb{N}}$ is injective and continuous. As X is compact, $g(X)$ is closed. Moreover, as every step of the construction is effective, $g(X)$ is an effective closed set and the map g is effective.

If the space is perfect, then we construct another map $h : X \to \{0,1\}^{\mathbb{N}}$. We may write X as a partition of two clopen sets $X = A_0 \cup A_1$, where A_0 is the clopen set of smallest index to be different from X and $\emptyset$; this is always possible thanks to perfectness. Suppose that we have already constructed A_w, where w is a binary word. Let n be the first index such that $A_w \cap P_n$ differs from both A_w and $\emptyset$, and set $A_{w0} = A_w \cap P_n$, $A_{w1} = A_w \setminus P_n$. For $x \in X$ let $h(x) \in \{0,1\}^{\mathbb{N}}$ be the unique configuration such that $x \in A_w$ for all prefixes w of $h(x)$. Then $h : \{0,1\}^{\mathbb{N}} \to X$ is an effective homeomorphism. □

We see that there is no loss of generality in supposing that in any effective symbolic space, for any rational ϵ there exists a finite number of balls of radius ϵ and that we can compute all of them. Indeed, this is the case for all effective subsets of the Cantor space.

3. Effective symbolic systems

Definition 3. An *effective symbolic dynamical system* is an effective continuous map from an effective symbolic space to itself.

In other words, an effective symbolic system is a symbolic space with a continuous self-map in which intersections, complements, and inverse images of clopen sets are computable. This definition of effective function in a Cantor space is equivalent to classical definitions in computable analysis; see for instance [34]. We denote an effective symbolic system by a map $f : X \to X$ or simply f, when the enumeration P of X is implicit. Extending Definition 2, we define a relation of equivalence between effective systems.

Definition 4. The effective symbolic systems $f : X \to X$ and $g : Y \to Y$ are *effectively conjugated* if there exists an effective homeomorphism $h : X \to Y$ such that $h \circ f = g \circ h$. If $h : X \to Y$ is an effective surjective map (and not bijective), then the system $g : Y \to Y$ is said to be an *effective factor* of $f : X \to X$. The factor g can be seen as a 'simplification' of f.

The identity on any symbolic space is the simplest example of an effective symbolic system. A cellular automaton is an effective symbolic system acting on the space $A^{\mathbb{Z}^d}$, where A is the finite alphabet and d is the dimension.

Turing machines are usually described as working on finite configurations. A finite configuration is an element of $\{0,1\}^* \times Q \times \{0,1\}^*$, where Q denotes the set of states of the head, the first binary word is the content of the tape to the left of the head and the second binary word is the right part of the tape. However, $\{0,1\}^*$ cannot be naturally equipped with a compact topology, so we consider its compactification $W = \{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$, i.e., the set of finite and infinite binary words. Then the Turing machine function is also defined on $W \times Q \times W$, which is a compact space, whose isolated points are $\{0,1\}^* \times Q \times \{0,1\}^*$. An isolated point is clopen in $W \times Q \times W$. Hence a Turing machine with a blank symbol is an effective symbolic system on the space $W \times Q \times W$.

A Turing machine without blank symbol is an effective symbolic system as well. As we do not suppose that almost all cells are filled with a blank symbol, a configuration is given by an arbitrary element of $\{0,1\}^{\mathbb{N}} \times Q \times \{0,1\}^{\mathbb{N}}$ or, equivalently, $Q \times A^{\mathbb{Z}}$. This is a Turing machine with moving tape, as considered in [17]: the head is always in position zero, and the tape moves to the left or to the right.

3.1. Shifts and subshifts

A one-sided or two-sided *shift* is a dynamical system on $A^{\mathbb{N}}$ or $A^{\mathbb{Z}}$ (where A is a finite alphabet) with the map $\sigma : A^{\mathbb{N}} \to A^{\mathbb{N}}$ or $\sigma : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ defined by $\sigma(x)_i = x_{i+1}$. A shift is an effective system. A *subshift* is a subsystem of the shift, i.e., a closed subset that is invariant under the shift map. Most subshifts we consider in this article are one-sided subshifts. An *effective subsystem* of an effective symbolic system is an effective closed subset that is invariant under the map. With the relative topology, it is itself an effective symbolic system. In particular, a subshift that is an effective closed subset of $A^{\mathbb{N}}$ is again an effective symbolic system.

The set $\mathcal{L}(X)$ of all finite words appearing at least once in at least one point of the subshift X is called the *language* of the subshift. It is easy to see that a subshift is effective iff its language is recursive. In particular every *sofic* subshift (a subshift whose language is regular) is effective. Another widely studied class of subshifts are *substitutive* subshifts defined by substitutions $\vartheta : A \to A^+$. Since a substitution is a finitary object, every substitutive subshift is effective. A Sturmian subshift Σ_α associated to an irrational number α is a symbolic model of rotation of the circle $x \mapsto x + \alpha$; see e.g. [19]. A Sturmian subshift Σ_α is is effective iff α is a computable real number.

From any symbolic dynamical system (effective or not), we can generate one-sided subshifts in a natural way. A *clopen partition* of a symbolic space is a partition of the space into a finite number of disjoint clopen sets. A partition $\mathcal{A}$ is *finer* than $\mathcal{A}'$, or $\mathcal{A}'$ is *coarser* than $\mathcal{A}$, if every clopen set of $\mathcal{A}$ is included in some clopen set of $\mathcal{B}$. Given a clopen partition $\mathcal{A} = \{A_1, \dots, A_N\}$ of X, the subshift *induced* by this partition is the set of infinite words $a_0a_1a_2a_3\ldots \in \mathcal{A}^{\mathbb{N}}$, such that there is a point in a_0 whose trajectory goes successively through $a_1, a_2, \dots$. Note that here A_1, say, is both a subset of X and a symbol from a finite alphabet. Thus $A_1A_3A_1$ denotes a word of three symbols and not for instance a cartesian product. The language of the subshift is also said to be *induced* by the partition. An induced subshift is a factor of the system and conversely any factor subshift is induced by a clopen partition.

Following this observation, we can characterize effective symbolic systems in terms of their induced subshifts.

Proposition 2. A symbolic system is effective if and only if there is an algorithm deciding from any given clopen partition and any given finite word whether this word belongs to the language of the subshift induced by the partition.

Proof:
Let $\mathcal{A} = \{A_1, \ldots, A_N\}$ be a clopen partition. Then a word $a_0 a_1 \ldots a_{l-1} \in \mathcal{A}^*$ is in the language of the subshift induced by the partition if and only if $a_0 \cap f^{-1}(a_1) \cap \cdots \cap f^{-(l-1)}(a_{l-1})$ is not empty. But this can be checked algorithmically.

Conversely, suppose that all induced subshifts have decidable languages, and that we can compute a decision algorithm given the partition. Let P_n be a clopen set of X. There exists a clopen partition $\mathcal{A} = \{A_1, \ldots, A_N\}$ such that

- for every i, either $A_i \subseteq P_n$ or $A_i \subseteq X \setminus P_n$;
- if $A_i A_j$ and $A_i A_k$ belong to the language of the induced subshift, then A_j and A_k are either both parts of P_n or both parts of $X \setminus P_n$.

The first condition says that the partition is finer then P_n, the second condition says that the partition is finer than $f^{-1}(P_n)$. It can be checked algorithmically whether a clopen partition has these two properties. Thus a partition with these properties can be found algorithmically. Then we can compute $f^{-1}(P_n)$ as the union of all A_i such that there exists a word $A_i A_j$ in the language of the induced subshift and that $A_j \subseteq P_n$. □

If the subshifts have decidable languages, but decision algorithms are not computable with respect to the clopen partition, then the system may fail to be effective. This happens in the following example.

Example 1. Assume $k : \mathbb{N} \to \mathbb{N}$ is an uncomputable strictly increasing total function. We define a function f on the Cantor space $\{0,1\}^{\mathbb{N}}$ by $f(x) = f_0(x) f_1(x) f_2(x) \ldots$, where $f_i(x_0 x_1 x_2 \ldots) = \max\{x_0, x_1, x_2, \ldots, x_{k(i)}\}$. Then it is easy to see that for any point x either $f(x) = 0^\omega$ or $f^n(x) = 1^\omega$ for some $n \geq 0$ (here 0^ω is a shortcut for 000...). For any partition $\mathcal{A} = \{A_1, \ldots, A_N\}$, if $0^\omega \in A_1$ and $1^\omega \in A_2$ (say), then every point in $A_3 \cup \ldots \cup A_N$ reaches A_2 in bounded time, say t. Then every finite word of the language of the subshift induced by the partition is of the form $A_1^* S A_2^*$, where S is some subset of $\{A_1, \ldots, A_N\}^t$. This is certainly a decidable language. However f is not effective, for otherwise we could compute k.

In the rest of the paper, we use the terms 'symbolic system' or even 'system' to denote an effective symbolic dynamical system.

3.2. Products

Let $(f_n : X_n \to X_n)_{n \in \mathbb{N}}$ be a family of *uniformly effective* systems on the effective symbolic spaces (X_n, P_n) ; we mean that there exists an algorithm that, given n and two clopen sets of X_n, can compute their intersection, complements and inverse images. Then the *effective product* of $(f_n)_{n \in \mathbb{N}}$ is the system $f : X \to X$ on the effective symbolic space (X, P) such that

- the set X is the product of all sets X_n;
- the clopen sets of X are all products of clopen sets $\prod_{n\in\mathbb{N}} A_n$ such that only finitely many $A_n \subseteq X_n$ are different from X_n (this is the usual product topology);
- the clopen sets are indexed by finite sets of integers in a straightforward manner, and f is defined componentwise.

We see that this is indeed an effective symbolic dynamical system. The projections $\pi_n : X \to X_n$ are effective as well. Products are useful to build examples.

4. Finite automata

Consider an effective system $f : X \to X$ and two clopen sets $U, V \subseteq X$. We would like to know whether there is a point of U that eventually reaches V, that is, whether

$$\exists x \in U, \exists n \in \mathbb{N} : f^n(x) \in V, \tag{1}$$

We call *halting problem of* f, the problem of answering this question given U and V. We will see later that this is indeed a generalization of the halting problem traditionally defined for Turing machines or counter machines. We find another formulation of the halting problem. Suppose that the system f is only partially observable. All we can know about f is whether the system is currently in U, in V or in $W = X \setminus (U \cup V)$ (we suppose for simplicity that U and V are disjoint). The system is observed by a finite automaton (formally defined below) as illustrated in Figure 1 . At every time step, the automaton jumps to a new state, according to whether the system is in U, in V or in W. The halting problem amounts to deciding whether it is possible, for some initial point of the space X, that the automaton eventually reaches the final state from the initial state.

We would like also consider variants of the halting problem. For instance, given three clopen sets U, V and W, we would like to know whether there is a trajectory starting from U, that eventually reaches V but does not pass through W meanwhile. In other words, we want to check whether the following formula is satisfied for some $x \in U$.

$$\exists n : f^n(x) \in V \text{ and} \forall m < n : f^m(x) \notin W, \tag{2}$$

where n and m are nonnegative integers. A finite automaton which accepts exactly points with this property is constructed in Figure 2. However, not all interesting properties can be checked by automata working with finite words only. We can ask, for example, whether there exists a trajectory, which stays forever in a given clopen set U, i.e., whether the formula

$$\forall n : f^n(x) \in U \tag{3}$$

is satisfied for some $x \in X$?

This is the case if and only if the finite automaton in Figure 3, starting from the initial state and observing the system f, can possibly reach infinitely often the final state. This leads to the theory of ω-regular languages which can be recognized by Muller or Büchi automata. In general we are interested in all properties that can be observed by automata.

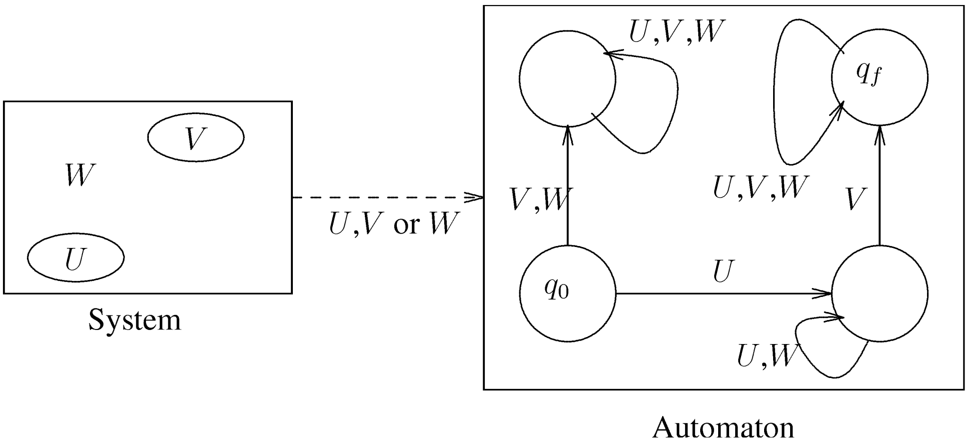

Figure 1. The symbolic system is partitioned into U, V and $W = X \setminus (U \cup V)$. At every time step, the finite automaton is fed with the symbol U, V or W and jumps to a new state. It is possible to reach the final state q_f from the initial state q_0 iff it is possible that q_f (and only q_f) is reached infinitely often from the initial state q_0 iff there is a point of U that eventually reaches V. Checking whether this is true given U and V, is the *halting problem* of f. The automaton can be considered as a finite automaton (the final state is q_f) or as a Muller automaton (for the family $\{\{q_f\}\}$).

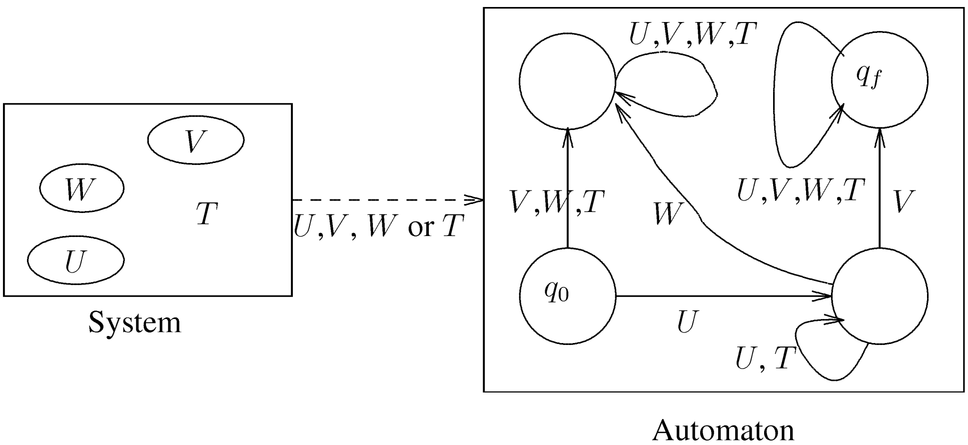

Figure 2. The symbolic system is partitioned into U, V, W and $T = X \setminus (U \cup V \cup W)$. There is a point of U that stays in $T \cup U$ until it eventually reaches V, iff it is possible that q_f (and only q_f) is reached infinitely often from the initial state q_0.

A (deterministic) *finite automaton* is given by a finite *set of states* Q, an *initial state* $q_0 \in Q$, a set of *final* states $Q_1 \subseteq Q$, a finite *input alphabet* A and a transition function $\Delta : Q \times A \to Q$. The transition function is extended to $\Delta : Q \times A^* \to Q$ by $\Delta(q, ua) = \Delta(\Delta(q, u), a)$. A language $L \subseteq A^*$ is *regular*, if there exists a finite automaton which accepts L, i.e., $u \in L$ iff $\Delta(q_0, u) \in Q_1$.

A *Muller automaton* consists of a finite set of states Q, a transition function $\Delta : Q \times A \to Q$, an initial state $q_0 \in Q$ and a family $\mathcal{F}$ of subsets of Q. A given infinite word $u \in A^{\mathbb{N}}$ is accepted by a Muller automaton, if the set of states that are visited infinitely often by a path generated by the given word is a

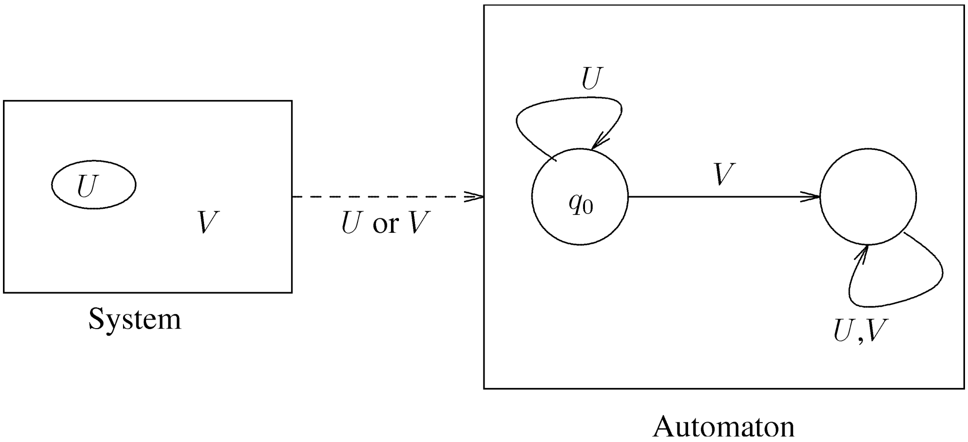

Figure 3. The system is partitioned into U and V. There is a point that never leaves U iff it is possible that q_0 (and only q_0) is reached infinitely often from the initial state q_0.

member of $\mathcal{F}$. A language $L \subseteq A^{\mathbb{N}}$ is *ω-regular*, if it is accepted by a Muller automaton, i.e.,

$$u \in L \text{ iff } \{q \in Q : \forall n, \exists m > n : \Delta(q_0, u_0 \ldots, u_{m-1}) = q\} \in \mathcal{F}$$

Alternatively, ω-regular languages can be defined by nondeterministic Büchi finite automata. An infinite word is accepted, if there is a trajectory passing infinitely often through a given set of final states. Although Büchi automata are simpler to define, Muller automata are deterministic, which is sometimes an advantage. In this paper we make little use of Büchi automata. Coming back to Figure 1, the halting problem for a symbolic system asks whether there is a finite word induced by the partition U,V, W that is accepted by the finite automaton. It is equivalent to ask whether there is an infinite word induced by the partition that is accepted by the automaton interpreted as a Muller automaton.

In general, given a clopen partition $\mathcal{A} = \{A_1, \ldots, A_N\}$ and a finite automaton over $\mathcal{A}$, we would like to know whether there is a non-empty intersection between the language associated to the partition and the regular language accepted by the automaton. In other words, the problem is to know whether there exists a point of the symbolic system whose trajectory, when observed through the partition, is accepted by the automaton. The same question can be asked for a Muller automaton instead of finite automaton.

The automaton may be interpreted as *observing* the system. This formalism includes all three properties described above, including the halting problem. These are examples of *model-checking* problems. Model-checking aims at finding decision algorithms to check whether the trajectories of a dynamical system satisfy a given property. But systems considered in the literature of model-checking are often non-deterministic and finite or countable, whereas we deal with deterministic systems with a possibly uncountable configuration space.

Note that Muller (or Büchi) automata are rather powerful to express properties on infinite words. They are equivalent to several logical formalisms, including the so-called μ-calculus and monadic second-order formulae. First-order formulae, including (1),(2), (3), are equivalent to linear temporal logic and strictly weaker than Muller automata. For precise definitions of all these formalisms, see for instance [29, 14, 12].

5. Decidable and universal systems

Definition 5. An effective symbolic system is *decidable* if there exists an algorithm which decides whether the subshift induced by a given clopen partition has a nonempty intersection with a given ω-regular language (described by a Muller automaton).

In short, we say that a system is decidable if the model-checking problem for Muller automata is decidable. Clearly, decidability is preserved by effective conjugacies and the factor of a decidable system is decidable. The identity map on any effective symbolic space is decidable. Indeed, for a partition $A_1, A_2, \dots, A_N$, the only words induced by the partition will be A_1^ω, A_2^ω, ... and A_N^ω. Given a finite automaton, it is enough to check whether one of these paths starting from an initial state of the automaton passes infinitely often through a final state. Alternatively it is a consequence of the forthcoming Proposition 9. The map $x \mapsto 0x$ on $\{0,1\}^{\mathbb{N}}$ with a unique attracting fixed point 0^ω is decidable. This follows from Proposition 8. The full shift on any finite alphabet is a decidable system by a corollary to Proposition 13.

If a system is not decidable, how undecidable can it be? We show that the problem of model-checking is at worse Σ_1^1-complete, which is rather high. As this question is not central to the paper we shall stay at an informal level. We can suppose that the space X of the system is an effective closed subset of the Cantor space $2^{\mathbb{N}}$. Let x be a sequence taking values in $\mathbb{N}$. Then the assertion '$x \in X$' is equivalent to the formula '$\forall t \in \mathbb{N} : x_0, x_1, \dots, x_t \in \{0,1\}$ and $[x_0 x_1 \dots x_t] \cap X \neq \emptyset$'. Let m be a natural integer encoding a Büchi automaton whose alphabet is a partition of X. Here Büchi automata are of easier use than Muller automata. A Büchi automaton is given by a finite set of states, an alphabet and a transition relation and a set of final states. If we suppose that $x \in X$, then call $R_f(x, m, t)$ the relation 'for the initial condition x, the system f drives the automaton m from its initial state to a final state at time t'. It is a recursive relation in the sense that it can be checked in finite time by a Turing machine with m and t as initial data and x as oracle.

Then the problem of model-checking can be expressed by the logical formula '$\exists x : x \in X$ and $\forall t, \exists t' : R_f(x, m, t)$', with m as free variable. Such a set of integers m defined by a formula involving an existential quantifier over sequences of natural integers, quantifiers over natural integers and recursive relations is called a Σ_1^1 *problem.* The class Σ_1^1 of problems belongs to the so-called analytical hierarchy, which is larger than the so-called arithmetical hierarchy; see [13] for more details.

We show that for some system f, the problem of model-checking is even Σ_1^1-complete, meaning that every Σ_1^1 problem is many-one reducible to it.

The set of natural integers k such that there exists a sequence of integers x for which the universal Turing machine with initial data k and oracle x does not halt is well-known to be Σ_1^1-complete; see [13]. An oracle Turing machine can be built in the following way. We take a one-tape universal Turing machine in the usual sense, to which we adjoin a tape that contains on its right part the oracle encoded in form $10^{x_0}10^{x_1}10^{x_2}1\dots$. The head has access to both tapes. Not every possible content of the second tape is a valid oracle; indeed the word 0^ω cannot appear on the tape. We can suppose without loss of generality that when the head wants to query x_n, it first checks that x_n is properly encoded by scanning the tape in some state q_{search} until it discovers a 1 and then jumps to the state q_{found}. This two-tape Turing machine is an effective dynamical system, similarly to the one-tape Turing machine discussed at the end of Section 3. Call Q the states of the head, q_0 the initial state and q_h the halting state. It can be supposed that to be impossible to leave q_h once we reach it. We want to know whether there is a initial configuration of this system, composed of a state of Q and the contents of both tapes, such that we start in the clopen set $\{q_0\} \times [k] \times [1]$ and the head reaches infinitely often $Q \setminus \{q_{\text{search}}, q_h\}$. This property can be observed by a Muller automaton in straightforward manner. Putting all together, we have constructed a reduction from a Σ_1^1 problem to a model-checking problem of some symbolic system; the latter is therefore Σ_1^1-complete as well.

We are ready now to state the main definition of computational universality. We define a universal symbolic system as a special kind of undecidable system, where Muller automata are replaced by finite automata. The universality of Turing machines is a particular example of this definition.

Definition 6. An effective dynamical system is *universal* if the problem whether the language induced by a given clopen partition has a nonempty intersection with a given regular language is recursively-enumerable complete.

An *r.e.-complete* problem, or Σ_1*-complete* problem, is a recursively enumerable problem, to which any recursively enumerable problem is many-one reducible. Note that the problem described in the Definition 6 is always recursively enumerable, because the language induced by a clopen partition is recursively enumerable and the language accepted by a finite automaton is recursive; if the intersection can be recursively enumerated and if it is nonempty then we can know it after a finite time. Universality is obviously preserved by effective conjugacies, and a system with a universal factor is also universal.

Proposition 3. A universal system is not decidable.

Proof:
If the model-checking for Muller automata is decidable then so is the model-checking problem for finite automata. Indeed, the latter is reducible to the former in the following way. Given a finite automaton, modify it in a such a way that the final states are fixed points of the transition function, whatever the input is; the resulting automaton is interpreted as a Muller automaton, for the family of all sets whose unique elements is a final state. □

Note that a non-deterministic scheme of computation underlies the definition of universality. The computation succeeds if and only if at least one trajectory exhibits a given behavior. For example, recall from Section 4 that the halting problem consists in determining, given the clopen sets U and V, whether there is a configuration in U that eventually reaches V. We may think of V as the halting set and of U as an initial configuration of which we know only the first digits. The unspecified digits of the initial configuration may be seen as encoding the non-deterministic choices occurring during the computation.

5.1. Examples

Turing machines with blank symbol.

A Turing machine with blank symbol that is universal in the sense of Turing, is also universal according to Definition 6, because the halting problem ‘Can we go from a clopen set U to a clopen set V?’ is r.e.-complete. Indeed the halting problem restricted to clopen sets that are isolated points is already r.e.-complete. Recall that isolated points are exactly finite configurations. Incidentally, we have shown that what we have called ‘halting problem’ for a general symbolic system is indeed a generalization of the usual halting problem for Turing machines.

Turing machines without blank symbol.

It is only slightly more complicated to build a universal Turing machine without blank symbol. In such a Turing machine, there is no obvious notion of 'finite configuration'. The trick is basically to encode the initial data in a self-delimiting way. Take a Turing machine that is universal in the sense given by Turing. Then add two new symbols L and R to the tape alphabet. On an initial configuration, put an L on the left end and an R on the right end of the encoded data. When the head encounters an L, it pushes it one cell to the left, leaving some more space available for computation. It acts similarly for an R symbol. The working space is always delimited by a L and a R; the symbols situated outside this zone are considered as noise, and do not influence the computation. For this modified universal Turing machine, the (clopen-set-to-clopen-set) halting problem is again undecidable.

Cellular automata.

Let us take a universal Turing machine with a blank symbol. We suppose that when the halting state is reached, then the head comes back to the cell of index 0. We can simulate it in a classic way with a one-dimensional cellular automaton. The alphabet of the automaton is $A \cup (A \times Q) \cup \{L, R, Error\}$, where A is the tape alphabet (including the blank symbol) and Q the set of states. Let us take a point in the cylinder $[L,$ initial data of the Turing machine, $R]$, and observe its trajectory. The symbol L moves to the left at the speed of light, leaving behind blank symbols. The symbol R moves to the right in a similar way. Meanwhile, the space between L and R is used to simulate the Turing machine and is composed of symbols from A and exactly one symbol from $(A \times Q)$, which denotes the position of the head. When L or R symbols meet each other, then a spreading $Error$ symbol is produced, that erases everything.

This cellular automaton is again universal, because the (clopen-set-to-clopen-set) halting problem is r.e.-complete. Indeed, there is an orbit from the cylinder $[L,$ initial data of the Turing machine, $R]$ to the cylinder [halting state] (both cylinders centered at cell of index zero) if and only if the universal Turing machine halts on the initial data.

Tag systems.

Tag systems were introduced by Post in 1920. A *tag system* is a transformation rule acting on finite binary words. At each step, a fixed number of bits is removed from the beginning of the word and, depending on the values of these bits, a finite word is appended at the end of the word. Minsky [22] proved that there is a so-called universal tag system, for which checking whether a given word will end up to the empty word when repeating the transformation is an r.e.-complete problem.

We can extend the rule of tag systems to infinite words, by just removing from them a fixed number of bits. Thus we have a dynamical system on the compact space $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$ of finite and infinite words, in which finite words are clopen sets. Again, if the tag system is universal for the word-to-word definition, then it is universal for Definition 6 with the halting problem on clopen sets of $\{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$.

Collatz functions.

We can also apply our definition to functions on integers. Let $\mathbb{N} \cup \{\infty\}$ be the topological space with the metric $d(n,m) = |2^{-n} - 2^{-m}|$. This is effectively homeomorphic to the set $\{1^n 0^\omega | n \in \mathbb{N}\} \cup \{1^\omega\}$. Then some functions on integers may be extended to infinity. For instance, the famous $3n + 1$ function sends even n's to $n/2$, odd n's to $3n+1$ and ∞ to ∞. Whether this map is decidable is unsettled. But Conway [4] proved that similar functions, called Collatz functions, can be universal.

Counter machines.

A k-counter machine is a composed of k counters, each containing a nonnegative integer, and a head that can test which counters are at zero and can increment or decrement every counter (with the convention $0-1=0$). Thus a counter machine is a map $f : Q \times \mathbb{N}^k \to Q \times \mathbb{N}^k$, where Q is the finite set of states of the head. There exists such a machine f for which given two configurations $x, y \in Q \times \mathbb{N}^k$, the problem whether the trajectory of x reaches y is r.e.-complete; see Minsky [22].

The map f is easily extended to the compact space $Q \times (N \cup \{\infty\})^k$, with the convention $\infty \pm 1 = \infty$. Here again, the points of $Q \times \mathbb{N}^k$ are clopen sets of $Q \times (N \cup \{\infty\})^k$, hence f is universal for the halting problem.

More examples.

In Section 7 we give an example of a universal system that is chaotic, and for which the usual halting problem is decidable, but not the variant expressed by logical formula (2). In Section 6.4 we build a system which is neither decidable nor universal. In the setting of point-to-point properties, it was proved by Sutner [32] that there exist cellular automata with a halting problem of an intermediate degree between decidability and r.e.-completeness. The same kind of examples for Turing machines have been known for long time (Friedberg-Muchnik theorem, see for instance [13]). However we have not been able to build a system for which finite-automata properties of trajectories are undecidable, but not r.e.-complete.

6. Sufficient conditions for decidability

It has been highlighted in the Introduction that some attempts have been made to link computational capabilities of a system to its dynamical properties. This is also the purpose of this section. Most results proved in this section are in fact sufficient conditions of decidability and can thus be interpreted as necessary conditions for universality. For instance, minimal systems are decidable, thus universal systems are not minimal.

The following constructions and propositions are useful in several proofs.

Given an effective system $f : X \to X$, a clopen partition $\mathcal{A} = \{A_1, \ldots, A_N\}$ of X and a transition function $\Delta : Q \times \mathcal{A} \to Q$, we construct the observation system $f_\Delta : X \times Q \to X \times Q$ by

$$f_\Delta(x, q) = (f(x), \Delta(q, A_i)), \text{ where } x \in A_i$$

Clearly f_Δ is an effective system, and the projection $\pi_X : X \times Q \to X$ is an effective factor of f_Δ to f.

Definition 7. We say that a dynamical system $f : X \to X$ has clopen basins, if for every clopen set $V \subseteq X$, its basin $\mathcal{B}(V) = \bigcup_{n \geq 0} f^{-n}(V)$ is a clopen set.

Proposition 4. If $f : X \to X$ is an effective system with clopen basins, then the operation $V \mapsto \mathcal{B}(V)$ is computable.

Proof:
If V and $\mathcal{B}(V)$ are clopen sets, then by compactness there exists $m > 0$ such that

$$\mathcal{B}(V) = \bigcup_{n<m} f^{-n}(V) = \bigcup_{n<m+1} f^{-n}(V)$$

Given V we can determine m effectively so the operation $\mathcal{B}(V)$ is effective too. There exists a computable function $k : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{B}(P_n) = P_{k(n)}$. □

Proposition 5. Any effective system with clopen basins is decidable.

Proof:
We show first that for every clopen partition $\mathcal{A}$ and for every transition function $\Delta : Q \times \mathcal{A} \to Q$, the system $f_\Delta : X \times Q \to X \times Q$ has clopen basins too. Let $V \subseteq X \times Q$ be a clopen set. For any $q \in Q$, the set $V_q = \{x \in X : (x, q) \in V\}$ is clopen too, so there exists $m > 0$ such that for every $q \in Q$, $\mathcal{B}(V_q) = \bigcup_{n<m} f^{-n}(V_q)$. Since $V = \bigcup_{q \in Q} V_q \times \{q\}$, we get $\mathcal{B}(V) = \bigcup_{n<m} f_\Delta^{-n}(V)$. This proves that f_Δ has clopen basins.

Assume now that $V \subseteq X \times Q$ is clopen, so that $\mathcal{B}(V)$ is clopen and the index of $\mathcal{B}(V)$ can be computed from the index of V. Moreover $I(V) = \mathcal{B}(\mathcal{B}(V)^c)^c$, where ^{c} denotes the complement, is a clopen set too and its index can be again computed from that of V. A point (x, q) belongs to $I(V)$ iff the trajectory of (x, q) passes through V infinitely often. Given $q_0, q_1 \in Q$, then $(X \times \{q_0\}) \cap I(X \times \{q_1\})$ is again a computable clopen set, so the set $\{x \in X : \forall n, \exists m > n, f_\Delta^m(x, q_0) = q_1\}$ is computable as well. It follows that for a family $\mathcal{F}$ of subsets of Q, the set

$$\{x \in X : \{q \in Q : \forall n, \exists m > n, f_\Delta^m(x, q_0) = q\} \in \mathcal{F}\}$$

is computable too. In particular, whether this set is empty can be decided algorithmically. □

6.1. Minimality

A *minimal* dynamical system is a system with no subsystem (except the empty set and itself). It is characterized by the property that all orbits are dense. By compactness, any minimal system has clopen basins. From Proposition 5 we get

Proposition 6. Any symbolic minimal system is decidable.

We strenghten this result and prove that an undecidable system must have a ‘thin’ subsystem.

Proposition 7. A symbolic system whose every non-empty subsystem has a non-empty interior is decidable.

Proof:
We show that such a system $f : X \to X$ must have clopen basins.

Given a clopen set $V \subseteq X$, we show that the border $\partial\mathcal{B}(V) \setminus \mathcal{B}(V)$ is f-invariant. Indeed $f(\mathcal{B}(V) \setminus V) \subseteq \mathcal{B}(V)$ and, since V is clopen, $\partial\mathcal{B}(V) = \partial(\mathcal{B}(V) \setminus V)$. If $y \in \partial\mathcal{B}(V)$ and U is a neighbourhood of $f(y)$, then $f^{-1}(U)$ is a neighbourhood of y that contains a point $z \in \mathcal{B}(V) \setminus V$. Thus $f(z) \in U \cap \mathcal{B}(V)$.

Since U is arbitrary, this proves that $f(y) \in \partial\mathcal{B}(V)$. But $f(y) \notin \mathcal{B}(V)$, because $y \notin \mathcal{B}(V) \setminus V$. This proves the invariance of $\partial\mathcal{B}(V)$.

Since $\partial\mathcal{B}(V)$ is a subsystem with empty interior, it must be empty, so $\mathcal{B}(V)$ is clopen. By Proposition 5, the system f is decidable. □

Any dynamical system has a minimal subsystem, thanks to Zorn's lemma and compactness. In particular, any point comes arbitrarily close to a minimal system, since the closed orbit of the point is itself a dynamical system. Suppose that the symbolic system is not minimal but consists of one minimal subsystem attracting the whole space of configurations. In other words, the limit set is minimal. The limit set of a dynamical system $f : X \to X$ is the set $\bigcap_{n\geq 0} f^n(X)$. Then such a system is again decidable. The following proposition is more general.

Proposition 8. A symbolic system whose limit set is the union of finitely many minimal systems is decidable.

Proof:
Suppose that the limit set is $Y_1 \cup \cdots \cup Y_p$, where Y_i are minimal subsystems, so that $Y_i \cap Y_j = \emptyset$ for $i \neq j$. Let $V \subseteq X$ be a clopen set. If $V \cap Y_i = \emptyset$, then $\mathcal{B}(V) \cap Y_i = \emptyset$. If $V \cap Y_i \neq \emptyset$, then for some $m > 0$, $Y_i \subseteq V_m = \bigcup_{n<m} f^{-n}(V)$. Thus there exists $m > 0$ such that for all i either $Y_i \subseteq V_m$ or $Y_i \cap V_m = \emptyset$. Then $W_m = f^{-m}(V) \setminus V_m$ is a clopen set disjoint from the limit set. From compactness there exists $k > 0$ such that $f^{-k}(W_m) = \emptyset$, so $\mathcal{B}(W_m)$ is a clopen set. It follows that $\mathcal{B}(V) = V_m \cup \mathcal{B}(W_m)$ is a clopen set too, so f is decidable by Proposition 5. □

A stronger statement is suggested by the intuition that an undecidable system (and especially a universal system) is likely to be able to 'simulate' many other systems.

Conjecture 1. A universal symbolic system has infinitely many minimal subsystems.

6.2. Equicontinuity

A system $f : X \to X$ is *equicontinuous* if for every $\epsilon > 0$ there exists a $\delta > 0$ such that $d(x, y) < \delta$ implies $d(f^t(x), f^t(y)) < \epsilon$, for any points x, y and $t \in \mathbb{N}$. Note that equicontinuity in symbolic systems is a topological property not just a metric one. Instead of 'For every $\epsilon > 0$, there is a $\delta \ldots$' we could say 'For every clopen partition, there is a finer clopen partition such that if two points are in the same subset of the finer partition, then they generate the same infinite word in the coarser partition.'

Proposition 9. An equicontinuous symbolic system is decidable.

Proof:
We prove that an equicontinuous system has clopen basins. Let V be a clopen set. Then from equicontinuity there exists a δ such that any two points distant of less than δ either both eventually reach V or both never reach V. Hence $\mathcal{B}(V)$ is the union of balls of radius δ, and is a clopen set. It follows by Proposition 5 that the system is decidable. □

We say that a point x of a dynamical system f is *sensitive* if there is an $\epsilon > 0$ such that for every $\delta > 0$ there is a point y with $d(x, y) < \delta$ and a non-negative time t such that $d(f^t(x), f^t(y)) > \epsilon$. It is easy to show that an equicontinuous dynamical system is exactly a system with no sensitive point. Hence, Proposition 9 implies that an undecidable symbolic system must have a sensitive point. Equicontinuity in the case of cellular automata has been given a combinatorial characterization in [16]. It is also proved that equicontinuous cellular automata are eventually periodic, thus confirming in this particular case that equicontinuity is incompatible with computational universality.

6.3. Regular Systems

A subshift is called sofic, if its language is regular. A symbolic system is called *regular*, if all its induced subshifts are sofic (see [15, 18]). Can a regular system be universal? We consider first a closely related question. We say that an effective system is *effectively regular* if it is regular and there is an algorithm that builds from a given clopen partition the finite automaton recognizing the regular language induced by the partition.

Proposition 10. An effectively regular system is decidable.

Proof:
It is well-known that the intersection of two ω-regular languages is an ω-regular language. The Muller automaton of the intersection is constructed as the product of the two automata. Moreover, the problem whether the language accepted by a given Muller automaton is empty is algorithmically decidable too. If a subshift is sofic, then the subshift itself is an ω-regular set.

Suppose that we are given an effective system, a clopen partition $\mathcal{A}$ of the space and a Muller automaton over the alphabet $\mathcal{A}$. Then we construct another Muller automaton that accepts exactly the subshift induced by $\mathcal{A}$ and verify whether the language accepted by the product of these two Muller automata is empty or not. Hence the system is decidable. □

If the system is regular but not effectively regular, then the argument of the proof fails.

Proposition 11. There exists a symbolic system that is regular and universal.

Proof:
Let X_n be the subshift of $2^{\mathbb{N}}$ whose forbidden words are words of the form 10^t1, where t is less than the (possibly infinite) halting time of the universal Turing machine launched on data n. If the Turing machine does not halt, then X_n is the sofic subshift $\{0^*10^\omega, 0^\omega\}$. If the Turing machine halts in k steps, then X_n is the subshift of finite type with forbidden words 11, 101, 1001, …, $10^{k-1}1$. So all subshifts are sofic, but we cannot effectively build the automaton recognizing the language, for it would allow to solve the halting problem.

Now consider the product of all X_n. This product is again an effective symbolic system X, and all its induced subshifts are sofic, due to the fact that the finite product of sofic subshifts is a sofic subshift and the induced subshift of a sofic subshift is again sofic. Thus the system is regular, but not effectively regular. Finally, it is r.e.-complete to check whether there is a trajectory starting from $\pi_n^{-1}([1])$ which eventually reaches $\pi_n^{-1}([01])$. Here $\pi_n : X \to X_n$ is the projection. □

6.4. Shadowing property

Definition 8. Let $f : X \to X$ be a symbolic dynamical system. A *δ-pseudo-orbit* is a (finite or infinite) sequence of points $(x_n)_{n\geq 0}$ such that $d(f(x_n), x_{n+1}) < \delta$ for all n. A point x *ϵ-shadows* a (finite or infinite) sequence $(x_n)_{n\geq 0}$ if $d(f^n(x), x_n) < \epsilon$ for all n. A dynamical system is said to have the *shadowing property* if for every $\epsilon > 0$ there is a $\delta > 0$ such that any δ-pseudo-orbit is ϵ-shadowed by some point. If moreover such a rational δ can be effectively computed from a rational ϵ then we say that the system has the *effective shadowing property*.

For example, the one-sided and two-sided shifts have the shadowing property for $\delta = \epsilon$. It is easy to see that the effective shadowing property is invariant under effective conjugacies. We can give the following interpretation to the effective shadowing property. Suppose that we want to compute numerically the trajectory of x such that at every step numerical errors are bounded by δ. The resulting sequence of points is a δ-pseudo-orbit, and the shadowing property ensures that this pseudo-orbit is ϵ-close to an actual trajectory of the system, ensuring that the result of the numerical computation is not meaningless.

Proposition 12. A symbolic system (effective or not) with the shadowing property is regular. An effective symbolic system with the effective shadowing property is effectively regular.

Proof:
The proof generalizes Proposition 5.69 of [19] about cellular automata. Consider a symbolic system $f : X \to X$ with the shadowing property and a clopen partition $\mathcal{A} = \{A_1, \dots, A_N\}$. There exists an ϵ such that all clopen sets of the partition are finite unions of balls of radius ϵ. By the shadowing property, there exists δ such that every δ-pseudo-orbit is ϵ-shadowed. We may suppose without loss of generality that $\delta \leq \epsilon$. Let $\mathcal{B} = \{B_1, \dots, B_M\}$ the clopen partition where each B_i is a ball of radius δ. Then the set of of all infinite words induced by all δ-pseudo-orbits through $\mathcal{B}$ is a subshift of finite type (that is, a subshift characterized by a finite set of finite forbidden words): the word B_iB_j is forbidden iff $B_i \cap f^{-1}(B_j) = \emptyset$, i.e., we cannot go from B_i to B_j in one step. But the partition $\mathcal{A}$ is coarser than $\mathcal{B}$, so the subshift induced by $\mathcal{A}$ is a factor of a subshift of finite type, hence sofic. If the system has the effective shadowing property, then we can effectively find δ, effectively describe the subshift of finite type and effectively build the sofic subshift. □

Proposition 13. A symbolic system that has the effective shadowing property is decidable.

Proof:
By Propositions 12 and 10. □

In particular, the shift is decidable. Proposition 13 is stronger than Proposition 9, as we now show.

Proposition 14. An equicontinuous effective symbolic system has the effective shadowing property.

Proof:
Let $f : X \to X$ be an equicontinuous system.Then for every $\epsilon > 0$, there is a δ such that any two points distant of less than δ have ϵ-close trajectories. We show that any δ-pseudo-orbit is ϵ-shadowed by some point.

Let $x_0, x_1, x_2, \ldots$ be a δ-pseudo-orbit. We show by induction on m that $d(f^n(x_m), f^{n+m}(x_0)) < \epsilon$ for every m and n. The case $m = 0$ is obvious. If it is true for m then in particular $d(f^{n+1}(x_m), f^{n+m+1}(x_0)) < \epsilon$. But $d(x_{m+1}, f(x_m)) < \delta$ implies $d(f^n(x_{m+1}), f^{n+1}(x_m)) < \epsilon$. From ultrametric inequality we have $d(f^n(x_{m+1}), f^{n+m+1}(x_0)) < \epsilon$.

It is now enough to prove that a δ is computable from ϵ, i.e. an equicontinuous symbolic system is always 'effectively' equicontinous. Take the partition $\mathcal{B}_0$ of all balls of radius ϵ. For $n = 0, 1, 2, \ldots$, let $\mathcal{B}_{n+1}$ be the coarsest partition finer than $\mathcal{B}_n$ and $f^{-1}(\mathcal{B}_n)$. From equicontinuity, this sequence of finer and finer partitions must stabilize to some $\mathcal{B}_n = \mathcal{B}_{n+1} = \mathcal{B}_{n+2} = \ldots$. To check that we have reached this point it is enough to check that $\mathcal{B}_n = \mathcal{B}_{n+1}$. We choose δ so that the clopen sets of $\mathcal{B}_n$ can be expressed as balls of radius δ. □

We also have the following result.

Proposition 15. A symbolic system that has the shadowing property is not universal.

Proof:
Let $f : X \to X$ be a symbolic system with the shadowing property. Given a finite automaton observing the system through a given clopen partition, the problem is to check whether there exists a finite word induced by the clopen partition that is accepted by the automaton. As we have noticed after stating Definition 6, this problem is recursively enumerable. We show that it is also co-recursively enumerable. This will prove that the problem is in fact decidable and that f is not universal.

Let $\mathcal{A} = \{A_1, \ldots, A_N\}$ be a clopen partition and $\Delta : Q \times \mathcal{A} \to Q$ the transition function of a finite automaton. We must essentially prove that the halting problem is decidable for the observation system $f_\Delta : X \times Q \to X \times Q$.

But f_Δ is an effective symbolic system with the shadowing property. Indeed for an $\epsilon > 0$, choose an $\epsilon' \leq \epsilon$ such that any A_i can be written as a union of balls of radius ϵ'. Then the shadowing property for f yields a corresponding δ'. Choose a $\delta \leq \delta'$ such that δ is strictly smaller than the distance between any two sets $X \times \{q\}$ and $X \times \{q'\}$. Then it is easy to see that any δ-pseudo-orbit of f_δ is ϵ-shadowed by some point of $X \times Q$.

Take two clopen sets $U, V \subseteq X \times Q$. There exists an orbit from U to V iff for every $\delta > 0$ there exists a δ-pseudo-orbit from U to V (see Proposition 2.15 of [19]). If there is no orbit starting in U that reaches V, then there exists a δ such that no δ-pseudo-orbit goes from U to V, and we can algorithmically check it. Thus the halting problem for f_Δ is decidable.

In particular if $U = X \times \{q_0\}$ (where q_0 is the initial state of the automaton) and $V = X \times F$ (where $F \subseteq Q$ is the set of final states of the automaton), then we can algorithmically check whether there exists a point of X which induces through the clopen partition a word that is accepted by the automaton. □

The following proposition shows that the effective shadowing property is stronger than the shadowing property.

Proposition 16. There exists an undecidable symbolic system that has the shadowing property, but not the effective shadowing property.

Proof:
Let X_n be the subshift with forbidden words 0^t, where the universal Turing machine stops on data n in at

most t steps. If the Turing machine does not halt on n, then X_n is the full shift; if it stops in k steps, then the forbidden word is 0^k. All these subshifts are effective, but we cannot compute their set of forbidden words.

The product X of all X_n is an effective system. Whether there is a point that remains for ever in $\pi_n^{-1}[0]$ is co-r.e.-complete (where $\pi_n : X \to X_n$ is the projection). This property has been shown in Figure 3 to be expressible in terms of Muller automata. Hence the system is undecidable.

By a theorem of Walters, a subshift of finite type has the shadowing property (see [19] for a proof). We show that the countable product of subshifts that have the shadowing property also has the shadowing property. A ball of radius ϵ in the product system may be expressed as the products of balls of radius ϵ' in a finite number of constituent subshifts. We choose the smallest of the corresponding δ' given by shadowing property in the subshifts. The product of balls of radius δ' may be expressed as union of balls of radius δ; this is the δ corresponding to ϵ. Hence the system has the shadowing property but not the effective shadowing property, since it is undecidable. □

As the shadowing property implies non-universality, it also proves that universality is stronger than undecidability.

Corollary 1. There exists a symbolic system that is neither decidable nor universal.

Note also that Turing machines that satisfy the shadowing property have been given a combinatorial characterization in [17]; in particular, the proof shows that the link between ϵ and δ (see Definition 8) is linear. Hence the effective shadowing property is not stronger than the shadowing property in the case of Turing machines.

7. A universal chaotic system

According to Devaney [8], a system is *chaotic* if it is infinite, topologically transitive and has a dense set of periodic points. One can prove that such a system is sensitive [2]. It is not difficult to construct a universal subshift. Indeed, in $\{0,1\}^{\mathbb{N}}$ consider all forbidden words of the form 01^n00^t1, where the universal Turing machine launched on data n does not halt in less than t steps. Then the subshift of all configurations avoiding this set of words is effective and universal: the halting problem is r.e.-complete. Modifying this construction, we get the following result:

Proposition 17. There exists an effective system on the Cantor space that is chaotic and universal.

Proof:
Consider a subshift $X \subset \{0,1,\S\}^{\mathbb{N}}$ whose forbidden words are all 01^n00^t1, where the universal Turing machine launched on data n does not halt in less than t steps. Denote by $L \subset \{0,1\}^*$ the language of binary words satisfying the constraint. Then the language of X consists of words $w_1\S w_2\S\ldots\S w_n$, where $w_i \in L$. We show that X is a universal chaotic system.

First note that X is a perfect subshift, so it is effectively conjugated to a system on the Cantor space. Then X has dense periodic points: if $w \in L$, then $(w\S)^\omega$ is in X. Finally X is topologically transitive: for any two finite words v, w of the language we can go from $[v]$ to $[w]$ with the point $v\S w\ldots$. Thus X is chaotic.

Moreover, given n it is undecidable whether there is a point of $[01^n0]$ that eventually reaches $[001]$ without passing through $[\S]$ meanwhile. This property can be expressed by the finite automaton constructed in Figure 2. Thus X is universal. $\square$

Note that the system built in the proof is a one-sided subshift, hence it is expansive: there is an ϵ such that any two points are eventually separated by at least ϵ. The central idea of the 'edge of chaos' is that a system that has a complex behavior should be neither too simple nor chaotic. There are several ways to understand that. Here we interpret 'complex system' by 'universal symbolic system'. Then 'too simple' could refer to the situation treated in Proposition 8: one or several attracting minimal subsystems. This includes of course the case of a globally attracting fixed point. If we take 'chaotic' as meaning 'Devaney-chaotic', then computational universality need not be on the 'edge of chaos', since we have just constructed a chaotic system that is universal.

However, many examples of chaotic systems (whatever the exact meaning given to 'chaotic', and for symbolic systems as well as for analog ones), have the shadowing property. For instance the shift and Smale's horseshoe (present in some physical systems), as well as hyperbolic systems, satisfy the shadowing property.

Thus we suggest that the term 'edge of shadowing property' would be more appropriate (at least for symbolic systems), although not so thrilling.

Note nevertheless that the 'edge of chaos' has been mostly studied in cellular automata, and we don't know whether an example of a chaotic universal cellular automaton exists.

8. Discussion of universality

Turing [33] justified the form of his machine along the following lines. A human operator applying an algorithmic procedure can be supposed to be at every step of time in a unique mental state. He can be supposed to have finitely many possible mental states, and to have at his disposal a pencil and as much paper as needed, on which he may write out letters or digits. In a finite time he may read or write only finitely many symbols on the paper. Paper is modelled by the tape and the human by a kind of finite automaton that is able to read, write or shift the tape.

Now suppose that the human operator has no paper or pencil, but can observe a (physical realization of) a symbolic dynamical system, without controlling it. The system can serve as a 'universal computer' if with its help, the human operator is able solve all problems he could also solve with paper and pencil. As the human operator has finitely many possible mental states, at every step he can distinguish only finitely many configurations of the system. If we group together all points that are undistinguishable between them, we obtain a partition of the system state space. We suppose that this partition is clopen, because clopen partitions express in a natural way that finitely many symbols are observed from the system at every step of time, analogously to Turing's assumption.

A symbolic system can be used as a 'computer', if it is computationally universal. When we observe the system using a given finite automaton acting on a clopen partition, then deciding whether the finite automaton can reach a final state from an initial state is at least as difficult as deciding the halting problem for a universal Turing machine. This means that when we look for the answer to a recursively enumerable problem, then we can obtain the answer by observing the system, provided we are lucky and wait long enough.

Our definition of universality perhaps differs in several ways from what we could expect at first glance from a generalization of Turing machine universality. We give now various arguments to support the present definition against seemingly more obvious attempts. In particular, we justify the use of *set-to-set* properties, observed by *finite automata*, on systems defined by a *computable* map.

8.1. Set-to-set properties

Davis [6] proposed the following definition: a Turing machine is universal if the relation 'x_n is in the orbit of x_m' is r.e.-complete, where x_m and x_n are arbitrary finite configurations. This definition has the advantage to bypass the need for a description of a way to encode the input and decode the output of a computation. Many definitions of universality for particular systems (cellular automata, for instance) propose to observe point-to-point properties.

Hemmerling [11] proposes a definition for an effective metric space; the basic idea is to endow a metric space with a countable dense set of points. Examples include the reals with rational points, the Cantor space with ultimately constant configurations, the Cantor space with ultimately periodic configurations. This seems to provide a suitable framework to generalize Davis' definition. Let us say that a metric space endowed with a dense set of points $(x_n)_{n\in\mathbb{N}}$ is universal if the property 'x_n is in the trajectory of x_m' is r.e.-complete.

However, as remarked in [9], this leads to conclude that the shift is universal; a consequence that is counter-intuitive. Indeed, consider the set of all configurations with primitive recursive digits. This set is countable and dense. Then we take as an initial configuration the sequence of states of the head of a universal Turing machine during a computation. And we only have to shift it to know whether the halting state will ever appear. It sounds unreasonable to classify the shift among universal systems, because it does not compute anything but just reads the memory.

The definition presented in Section 5 overcomes this problem in a simple manner: the user needs only to specify a finite number of bits as an initial condition. Instead of initial *configurations* we should rather talk about initial *sets*, which may be seen as 'fuzzy points', points defined with finite accuracy. This solution is also more satisfactory from the point of view of physical realizability. Indeed, we expect the set of configurations of a physical system to be uncountable in general, and specifying an initial point for the computation means *a priori* that we must give an infinite amount of information. Preparing a physical system to be in a very particular configuration is likely to be impossible, because of the noise or finite precision inherent to every measure.

8.2. Finite automata

What kind of property are we going to test on clopen sets (or, equivalently, on induced subshifts)? Here again, we must avoid trivialities. Suppose that we look at identity on the Cantor space. We now choose to observe the following property: a clopen set satisfies the property if and only if its index (i.e., the integer describing the clopen set) satisfies some r.e.-complete property on $\mathbb{N}$. Then we find that the identity is computationally universal, which is a result not to be desired. The complexity of computation is artificially hidden in the decoding.

On the other hand, we see no reason to restrict ourselves to the sole halting property: 'there is a trajectory from this clopen set to that clopen set'. Any observable property could *a priori* be used as a basis for computation. For instance, the chaotic system built in Section 7 is universal but not for the halting property. So we must precisely define a class of observable properties of clopen sets, not too large and not too restricted. Finite automata used to express properties of finite words have been extensively studied in the literature. They also agree with Turing's idea of modelling a human operator as having finitely many possible mental states. We do not use the powerful setting of Muller automata, because it may need an infinite time to check that a trajectory has the required property, which goes against the idea that a successful computation should end in a finite time.

8.3. Effectiveness

Finally, the following example shows that it is useful to add an effectiveness structure on dynamical systems. Fix an r.e.-complete set $H \subset \mathbb{N}$ of integers and consider the symbolic system $f : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ such that $f(1^\omega) = 1^\omega$ and $f(1^n 0 x_0 x_1 x_2 \ldots) = 1^m 0 x_0 x_1 x_2 \ldots$, where m depends on n. If $n \in H$, then m is the largest integer strictly smaller than n such that $m \in H$ or 0 if no such number exists. If $n \notin H$, then $m = n$. Suppose now that $13 \in H$. Then the relation 'the clopen set $[1^n 0]$ will eventually reach $[1^{13} 0]$' is r.e.-complete, because H is.

On the other hand, if we were provided with an actual implementation of $f : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$, we could decide an undecidable problem (namely, H) by observing the trajectories. So there is a discrepancy between the computational complexity of properties of clopen sets and the actual possibilities of the machine. This is because we cannot compute even a single step of f: it is a 'non-simulable' system. We therefore restrict ourselves to systems such that the inverse image of a clopen set is computable. Note that for instance in [30] Siegelmann allows neural networks with non-recursive weights, leading to a non-computable maps and to super-Turing capabilities.

9. Conclusions and future work

We provided a definition of decidability and universality for a symbolic systems, and established some links between decidability and the dynamical properties of the system. We also constructed a chaotic system that is universal. These results are summed up in Figure 4. We have already formulated some open problems. Is there a cellular automaton that is chaotic and universal? Do undecidable system have infinitely many disjoint subsystems? But many more questions are yet to be solved. For instance, can we find sufficient conditions of universality? Which simplicity criteria proposed in [15] are sufficient conditions for decidability? Are the Game of Life and the automaton 110 universal? Can a linear cellular automaton be universal?

It also remains to extend in the definitions and results to systems in $\mathbb{R}^n$ in discrete time or even continuous time. The resulting definition of universality could then be compared to existing definitions, for instance [31, 3, 28, 26]. Then, results like those of Section 6 could hopefully be adapted. For instance, are minimal systems capable of universal computation? Such results could then be applied to physical systems. What systems that can be found in Nature are able to compute? For instance, hyperbolic dynamical systems are known to have the effective shadowing property. This would suggest that hyperbolic systems are not universal.

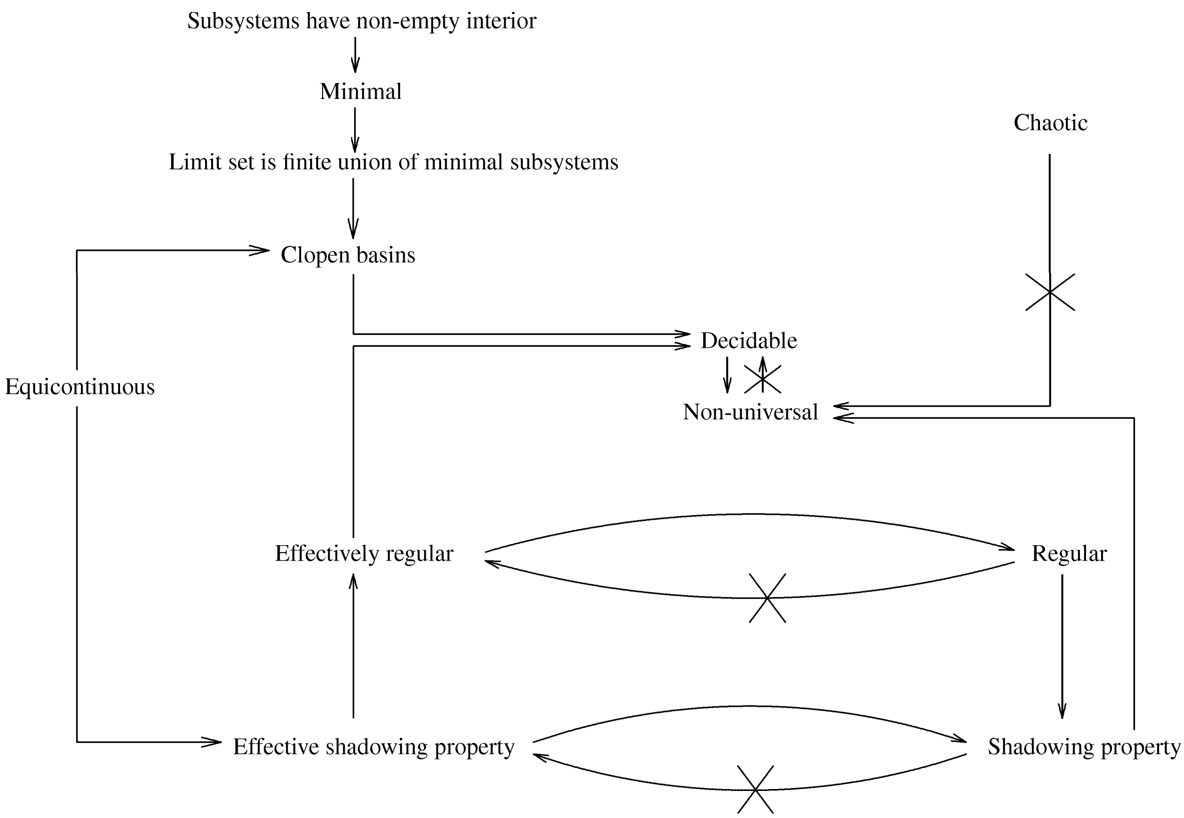

Figure 4. Summary of the results. Arrows read ‘implies’, crossed arrow read ‘does not imply’.

10. Acknowledgements

We warmly thank Eugene Asarin and Cris Moore for fruitful discussions and comments. This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. The scientific responsibility rests with its authors. J.-Ch. D. holds a FNRS fellowship (Belgian Fund for Scientific Research).

References

[1] E. Asarin and A. Bouajjani. Perturbed Turing machines and hybrid systems. In *Proceedings of the 6th IEEE Symposium on Logic in Computer Science (LICS’01)*, Boston, USA, June 2001. IEEE.

[2] J. Banks, J. Brooks, G. Cairns, G. Davis, and P. Stacey. On Devaney’s definition of chaos. *American Mathematics Monthly*, 99:332–334, 1992.

[3] O. Bournez and M. Cosnard. On the computational power of dynamical systems and hybrid systems. *Theoretical Computer Science*, 168:417–459, 1996.

[4] J.H. Conway. Unpredictable iterations. In *Proceedings of the 1972 Number Theory Conference*, pages 49–52, Boulder, Colorado, 1972.

[5] J.P. Crutchfield and K. Young. Computation at the onset of chaos. In W. Zurek, editor, *Complexity, Entropy and the Physics of Information*, pages 223–269. Addison-Wesley, 1989.

[6] M.D. Davis. A note on universal Turing machines. In C.E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 167–175. Princeton University Press, 1956.

[7] J.-Ch. Delvenne, P. Kůrka, and V. D.Blondel. Computational universality in symbolic dynamical systems. In M.Margenstern, editor, *MCU 2004*, number 3354 in Lecture Notes in Computer Science, pages 104–115. Springer-Verlag, 2005.

[8] R. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, 1989.

[9] B. Durand and Z. Róka. The game of life: universality revisited. In M. Delorme and J. Mazoyer, editors, *Cellular Automata: a Parallel Model*, volume 460 of *Mathematics and its Applications*, pages 51–74. Kluwer Academic Publishers, 1999.

[10] P. Gacs. Reliable cellular automata with self-organization. In *IEEE Symposium on Foundations of Computer Science*, pages 90–99, 1997.

[11] A. Hemmerling. Effective metric spaces and representations of the reals. *Theoretical Computer Science*, 284(2):347–372, 2002.

[12] E. M. Clarke Jr., O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, 1999.

[13] H. Rogers Jr. *Theory of recursive functions and effective computability*. MIT Press, Cambridge, MA, second edition, 1987.

[14] R. Kaivola. *Using automata to characterise fixed point temporal logics*. PhD thesis, University of Edinburgh, 1996.

[15] P. Kůrka. Simplicity criteria for dynamical systems. *Analysis of Dynamical and Cognitive Systems*, pages 189–225, 1993.

[16] P. Kůrka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory & Dynamical Systems*, 17:417–433, 1997.

[17] P. Kůrka. On topological dynamics of Turing machines. *Theoretical Computer Science*, 174(1-2):203–216, 1997.

[18] P. Kůrka. Zero-dimensional dynamical systems, formal languages, and universality. *Theory of Computing Systems*, 32(4):423–433, 1999.

[19] P. Kůrka. *Topological and Symbolic Dynamics*, volume 11 of *Cours Spécialisés*. Société Mathématique de France, 2004.

[20] C.G. Langton. Computation at the edge of chaos. *Physica D*, 42:12–37, 1990.

[21] W. Maass and P. Orponen. On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10(5):1071–1095, 1998.

[22] M. L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Ann. of Math. (2)*, 74:437–455, 1961.

[23] M. Mitchell, P.T. Hraber, and J.P. Crutchfield. Dynamic computation, and the "edge of chaos": A re-examination. In G. Cowan, D. Pines, and D. Melzner, editors, *Complexity: Metaphors, Models, and Reality*, Santa Fe Institute Proceedings, Volume 19, pages 497–513. Addison-Wesley, 1994. Santa Fe Institute Working Paper 93-06-040.

[24] C. Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64(20):2354–2357, 1990.

[25] C. Moore. Generalized shifts: Unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4:199–230, 1991.

[26] C. Moore. Dynamical recognizers: real-time language recognition by analog computers. *Theoretical Computer Science*, 201:99–136, 1998.

[27] C. Moore. Finite-dimensional analog computers: Flows, maps, and recurrent neural networks. In C.S. Calude, J. Casti, and M.J. Dinneen, editors, *Unconventional Models of Computation*. Springer-Verlag, 1998.

[28] P. Orponen. A survey of continuous-time computation theory. In D.-Z. Du and K.-I Ko, editors, *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Kluwer Academic Publishers, 1997.

[29] D. Perrin and J.-E. Pin. *Infinite Words (Automata, Semigroups,Logic and Games)*, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004.

[30] H.T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Progress in Theoretical Computer Science. Springer-Verlag, 1999.

[31] H.T. Siegelmann and S. Fishman. Analog computation with dynamical systems. *Physica D*, 120:214–235, 1998.

[32] K. Sutner. Cellular automata and intermediate degrees. *Theoretical Computer Science*, 296(2):365–375, 2003.

[33] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936.

[34] K. Weihrauch. *Computable analysis. An introduction*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000.

[35] S. Wolfram. *A new kind of science*. Wolfram Media, Inc., Champaign, IL, 2002.