Computational complexity of dynamical systems: The case of cellular automata

P. Di Lena *, L. Margara
Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy

ARTICLE INFO

Article history:
Received 9 July 2007
Revised 16 February 2008
Available online 5 June 2008

Keywords:
Cellular Automata
Universality
Classification

Computational complexity of dynamical systems: The case of cellular automata

Cellular Automata can be considered discrete dynamical systems and at the same time a model of parallel computation. In this paper we investigate the connections between dynamical and computational properties of Cellular Automata. We propose a classification of Cellular Automata according to the complexities which rise from the basins of attraction of subshift attractors and investigate the intersection classes between our classification and other three topological classifications of Cellular Automata. From the intersection classes we can derive some necessary topological properties for a cellular automaton to be computationally universal according to our model.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

The concept of computation and Computation theory itself are strictly related to Turing Machines. In recent years, however, a new trend of investigation attempts to find connections between Dynamical System theory and Computation theory. Cellular Automata can be considered discrete dynamical systems and at the same time a model of parallel computation. It is well known that they have the same computational power of Turing Machines. There is no general agreement on the concept of universality for Cellular Automata. The universality of a cellular automaton is generally proved by showing that such automaton can simulate a universal Turing Machine [21] or some other system which is known to be computationally universal [3]. A different approach was taken by Wolfram in [23] where the author classifies empirically Cellular Automata in four classes according to the observed (by computer simulation) evolution of the automata on random configurations. He suggested that Cellular Automata in the last of his classes must be capable of universal computation. Several authors have offered formalization to Wolfram classes. We cite just few of them. Gilman [8] proposed a classification based on the concept of equicontinuity while Hurley [11] proposed a classification based on the concept of attractors. Kurka [13] refined the Equicontinuity and Attractor classifications by using purely topological definitions and investigated the intersection classes between the two classifications and a third one based on the complexity of the languages rising from the column factors of Cellular Automata. All three classifications are based uniquely on topological concepts and it is not evident how these dynamical properties are related to computational properties of Cellular Automata except for the connection with Wolfram empirical classification.

While it is generally accepted to interpret the evolution of a dynamical system as a process of computation, it is much more less evident how to interpret the input and the output of the computation in the evolution of the system. A possible
approach is to see the process of computation in a dynamical system as a flow toward an attractor. The attractor is considered the halting state of the computation. One such approach has been taken in [2] to develop a complexity theory for the set of continuous time dynamical systems defined by differential equations. A more general approach has been taken recently in [5]. The authors rephrase the halting problem as the problem to decide if there exists at least one configuration from some initial set whose orbit reaches some halting set. Initial and halting sets are intended to be clopen (closed and open) sets of a Cantor space so that they can be described by means of finite information. It is easy to see how these two approaches are related: in a compact metric space the orbit of some configuration converges to an attractor $Z$ if and only if it enters into all clopen invariant sets whose omega limits coincide with $Z$. The authors of [5] propose a definition of universality which applies to general discrete symbolic (i.e. defined on a Cantor space) dynamical systems and they provide necessary conditions for the universality. According to their model, a universal symbolic dynamical system is not minimal (i.e. it contains at least one proper subsystem), not equicontinuous and does not satisfy the shadowing property. Moreover they conjecture that a universal dynamical system must have an infinite number of subsystems.

Here we interpret the process of computation in Cellular Automata as a flow toward a subshift attractor. A subshift attractor is an attractor which is invariant under the shift map. Subshift attractors have been investigated in [14] and [7]. We show that it is possible to restate the halting problem as the problem to decide if the omega limit of some clopen set is contained in a halting subshift attractor (that is, as the problem to decide if the orbits of all sequences contained in some clopen set converge to the attractor). We say that the computational complexity of a cellular automaton $(A^Z, F)$ with respect to the halting subshift attractor $Z$ is defined as the complexity of clopen sets contained in the basin of attraction of $Z$. Since a basin of attraction is the countable union of cylinder (clopen) sets and a cylinder set can be univocally described by some word in $A^*$, we can characterize the complexity of basins of attraction by using Formal Language theory. We propose a classification of Cellular Automata according to the complexity of basin languages (Section 3). A cellular automaton with highest computational complexity has at least one subshift attractor whose basin language is recursively enumerable complete.

Since our classification is based on purely topological concepts it is easy to explore the intersection classes with other well known topological classifications of Cellular Automata such as Attractors, Languages and Equicontinuity classifications (Section 4). From the intersection classes we can provide necessary conditions for a cellular automaton to be universal (Section 5). Even in our model a universal cellular automaton is not minimal, not equicontinuous, does not have the shadowing property and, in particular, it is not regular. It is open also in our case the question whether a universal cellular automaton must have an infinite number of subsystems.

2. Notation and definitions

In this section, we introduce the notation and the basic concepts that will be necessary to understand the rest of the paper. Cellular Automata as dynamical systems were first considered by Hedlund in the late sixties who studied this formalism in the context of Symbolic Dynamics as endomorphisms of full shifts [10]. In this paper we will adopt Symbolic Dynamics terminology. For an introduction in Symbolic Dynamics the reader can refer to [19] and for an introduction on Topological Dynamics to [15]. In the following, we will assume that the reader is familiar with Computation theory and Formal Language theory (see, for example, [12]).

Let $A$ be a finite alphabet with at least two elements. With $A^Z$ and $A^N$ we denote, respectively, the set of sequences $(x_i)_{i\in\mathbb{Z}}$ and $(x_i)_{i\in\mathbb{N}}$ where $x_i \in A$. For $x \in A^Z$, let $x_{[i,j]} \in A^{j-i+1}$ denote the word $x_i x_{i+1} \ldots x_j$. We use the shortcut $w \sqsubset x$ to say that $w \in A^+$ is a subword of $x \in A^Z$. Let us define a metric $d$ on $A^Z$ by

$$d(x,y) = 2^{-n} \text{ where } n = \min(|i| \mid x_i \neq y_i).$$

The set $A^Z$ endowed with metric $d$ is a compact metric space. For $u \in A^*$ and $i \in \mathbb{Z}$, let

$$[u]_i = \left\{ x \in A^Z \mid x_{i+j+|u|-1} = u \right\}$$

denote a cylinder set. Sometimes we will refer to the cylinder set $[u]_i$ simply with $[u]$. A cylinder set is a clopen (closed and open) set in $A^Z$. Every clopen set in $A^Z$ is a finite union of cylinder sets. The shift maps $\sigma : A^Z \rightarrow A^Z$, $\sigma : A^N \rightarrow A^N$ are defined by

$$\sigma(X) = X_{i+1}.$$  

The shift map is a continuous function and it is bijective on $A^Z$ while it is not on $A^N$. The dynamical system $(A^Z, \sigma)$ is called full shift. A shift space or subshift is a non-empty closed subset $\Sigma \subseteq A^Z$ which is also strongly shift invariant, i.e. $\sigma(\Sigma) = \Sigma$. A subshift is one-sided if it is a closed subset $\Sigma \subseteq A^N$ and it is $\sigma$-invariant, i.e. $\sigma(\Sigma) \subseteq \Sigma$. Usually we will denote the shift dynamical system $(\Sigma, \sigma)$ simply with $\Sigma$. A subshift $\Sigma$ is mixing if for all clopen sets $U, V \subseteq \Sigma$, there exists $n_0 > 0$ such that for all $n \geq n_0$ $\sigma^n(U) \cap V \neq \emptyset$. The language associated to a subshift $\Sigma$ is defined as

$$L(\Sigma) = \{ w \in A^* \mid \exists x \in \Sigma, w \sqsubset x \}.$$

Any subshift $\Sigma$ is completely determined by the set of its forbidden words $A^* \setminus L(\Sigma)$ (see [19]). A shift of finite type (SFT) is a subshift which can be defined by a finite set of forbidden words. The language $L(\Sigma)$ of a subshift $\Sigma$ is bounded periodic
if there exist integers \( m \geq 0, n > 0 \) such that for all \( x \in \mathcal{L}(\Sigma) \) and all \( x_i = x_{i+n} \). A subshift \( \Sigma \), with bounded periodic language \( \mathcal{L}(\Sigma) \) finite, i.e. \( |\Sigma| < \infty \) and in particular it is a shift of finite type. A generalization of SFTs are sofic shifts. Sofic shifts are factors of SFTs, i.e. a subshift \( S \) is sofic if and only if there is a shift of finite type \( \Sigma \) and a continuous \( \sigma \)-commuting function \( \varphi : \Sigma \to S \) such that \( \varphi(\Sigma) = S \) [22]. A subshift \( S \) is sofic if and only if its language \( \mathcal{L}(S) \) is regular so sofic shifts can be graphically represented by labeled graphs and in particular by finite state automata.

A cellular automaton is a dynamical system \((A^Z,F)\) where \( F : A^Z \to A^Z \) is a continuous and \( \sigma \)-commuting function, i.e. \( F\sigma = \sigma F \). According to Curtis–Hedlund–Lyndon Theorem [10], \((A^Z,F)\) is a cellular automaton if and only if there exists some local function \( f : A^{2d+1} \to A \) of radius \( r > 0 \) such that

\[
\forall x \in A^Z, F(x)_1 = f(x_{-r}, \ldots x_{i+r}).
\]

In the following sections we review Attractor, Equicontinuity and Language classifications for Cellular Automata. The intersections classes between the three classifications are shown in Figs. 1–3 (see [13]).

### 2.1. Equicontinuity classification

We review some topological properties of Cellular Automata based on the concept of equicontinuity point. These topological properties can be formulated for arbitrary dynamical systems.

**Definition 2.1.** For \( x \in A^Z, \epsilon > 0 \), let \( B_\epsilon(x) = \{ y \in A^Z \mid d(x,y) < \epsilon \} \) be the open ball of radius \( \epsilon \) centered in \( x \).

A point \( x \in A^Z \) is an equicontinuity point for \((A^Z,F)\) if the orbit of every point in every neighborhood of \( x \) stays forever close to the orbit of \( x \).

**Definition 2.2.** A point \( x \in A^Z \) is an equicontinuity point for \((A^Z,F)\) if

\[
\forall \epsilon > 0, \ \exists \delta > 0, \ \forall y \in B_\delta(x), \ \forall n \geq 0, \ \ d(F^n(x),F^n(y)) < \epsilon.
\]

A cellular automaton is equicontinuous if all its points are equicontinuity points.

**Definition 2.3** (Equicontinuity), \((A^Z,F)\) is equicontinuous if

\[
\forall x \in A^Z, \ \forall \epsilon > 0, \ \exists \delta > 0, \ \forall y \in B_\delta(x), \ \forall n \geq 0, \ \ d(F^n(x),F^n(y)) < \epsilon.
\]

The following theorem characterizes equicontinuous Cellular Automata.

**Theorem 2.4** [14]. For \((A^Z,F)\) the following conditions are equivalent:

1. \((A^Z,F)\) is equicontinuous;
2. there exist \( m \geq 0, n > 0 \) such that for every \( x \in A^Z \), and for every \( i \geq m \) we have \( F_i^+x = F^i(x) \).

A cellular automaton is almost equicontinuous if it has an equicontinuity point.

**Definition 2.5** (Almost Equicontinuity), \((A^Z,F)\) is almost equicontinuous if

\[
\exists x \in A^Z, \ \forall \epsilon > 0, \ \exists \delta > 0, \ \forall y \in B_\delta(x), \ \forall n \geq 0, \ \ d(F^n(x),F^n(y)) < \epsilon.
\]

By definition, every equicontinuous cellular automaton is also almost equicontinuous. Almost equicontinuous Cellular Automata are characterized by the presence of blocking words.

**Definition 2.6.** A word \( u \in A^+ \) with \( |u| \geq k > 0 \) is \( k \)-blocking for \((A^Z,F)\) if there exists \( p \in [0,|u| - k] \) such that

\[
\forall x,y \in [u]_0, \ \forall n \geq 0, \ F^n(x)_{[p,p+k-1]} = F^n(y)_{[p,p+k-1]}.
\]

**Theorem 2.7** [15]. \((A^Z,F)\) is almost equicontinuous iff it has a blocking word.

A cellular automaton is sensitive if for every point \( x \), in every neighborhood of \( x \) there exists a point \( y \) whose orbit separates from the orbit of \( x \). While this does not hold for general dynamical systems, for Cellular Automata sensitivity implies not almost equicontinuity.
Definition 2.8 (Sensitivity). \((A^Z,F)\) is sensitive if
\[ \exists \epsilon > 0, \forall x \in A^Z, \forall \delta > 0, \exists y \in B_\delta(x), \exists n \geq 0, \ d(F^n(x), F^n(y)) \geq \epsilon. \]

Theorem 2.9 [15]. \((A^Z,F)\) is sensitive iff it is not almost equicontinuous.

Positively expansiveness is a stronger form of sensitivity. A cellular automaton is positively expansive if the orbits of every two distinct points eventually separate under the evolution. Positively expansive Cellular Automata do not exist in any dimension greater than 1 [20].

Definition 2.10 (Positively expansiveness). \((A^Z,F)\) is positively expansive if
\[ \exists \epsilon > 0, \forall x, \forall y \neq x, \exists n \geq 0, \ d(F^n(x), F^n(y)) \geq \epsilon. \]

The following classification of Cellular Automata is Kůrka’s modification [13] of Gilman’s Equicontinuity classification [8]. Gilman’s classification is based on measure-theoretic concepts, while Kůrka’s one uses only topological concepts.

Corollary 2.11 [13]. Every \((A^Z,F)\) falls exactly in one of the following classes:

- **E1** \((A^Z,F)\) is equicontinuous;
- **E2** \((A^Z,F)\) is almost equicontinuous but not equicontinuous;
- **E3** \((A^Z,F)\) is sensitive but not positively expansive;
- **E4** \((A^Z,F)\) is positively expansive.

The membership is undecidable for most of the Equicontinuity classes.

Theorem 2.12 [4]. Equicontinuity, almost equicontinuity and sensitivity are undecidable properties for Cellular Automata.

It is actually unknown whether positive expansiveness is a decidable property.

Question 2.13. Is positive expansiveness a decidable property?

2.2. Language classification

The complexity of the languages of the column factor subshifts is a measure of the complexity of Cellular Automata. This measure was introduced by Kůrka for general dynamical systems [17].

Definition 2.14. The column factor subshift of width \(k > 0\) of \((A^Z,F)\) is the set of one-sided infinite sequences
\[ \Sigma_k(F) = \{ y \in A^{k \times \mathbb{N}} | \exists x \in A^Z, \forall n \geq 0, F^n(x)_{(0,k)} = y_n \}. \]

When it is clear from the context, we will drop the dependence from \(F\) in \(\Sigma_k(F)\). The languages of column factors of Cellular Automata are context sensitive [9]. It is possible to define classes of complexity for Cellular Automata according to the complexity of column factors. The lowest complexity class consists of Cellular Automata whose column factors give rise to only bounded periodic languages.

Definition 2.15. \((A^Z,F)\) is bounded periodic if \(\forall k > 0, \mathcal{L}(\Sigma_k)\) is bounded periodic.

From Theorem 2.4, it immediately follows that the class of equicontinuous Cellular Automata is included in the class of bounded periodic Cellular Automata. Actually, the two classes coincide.

Proposition 2.16 [13]. \(L_1 = E_1\).

A cellular automaton is called regular if all its column factors are sofic shifts. A bounded periodic cellular automaton is also regular.

Definition 2.17. \((A^Z,F)\) is regular if \(\forall k > 0, \mathcal{L}(\Sigma_k)\) is regular.
Theorem 2.18 [6]. $(A^Z, F)$ of radius $r$ is regular if and only if $\Sigma_{2r+1}$ is sofic.

Obviously, if $\Sigma_k$ is sofic for some $k > 2r + 1$ then it follows that $(A^Z, F)$ is regular, since $\Sigma_{2r+1}$ is a factor of $\Sigma_k$. On the contrary, $\Sigma_k$ can be sofic for some $k < 2r + 1$ and $(A^Z, F)$ be not regular (see [18]). The following classification is Kůrka’s Language classification of Cellular Automata according to the language complexity of column factors.

Corollary 2.19 [13]. Every $(A^Z, F)$ falls exactly in one of the following classes:

1. $L_1$ $(A^Z, F)$ is bounded periodic;
2. $L_2$ $(A^Z, F)$ is regular not bounded periodic;
3. $L_3$ $(A^Z, F)$ is not regular.

The membership in the Language classification is undecidable. The undecidability of class $L_1$ follows from Proposition 2.16 and Theorem 2.12. The undecidability of regularity has been shown in [6].

Theorem 2.20 [6]. Regularity is an undecidable property for Cellular Automata.

Regularity is actually a semidecidable property. If we know in advance that a cellular automaton is regular then we can compute its column factors, i.e. there exists an algorithm that given the local rule of a cellular automaton and an integer $k > 0$, it returns a finite state automaton representation of $\Sigma_k$.

Theorem 2.21 [6]. If $(A^Z, F)$ is regular then for any $k > 0$, $\Sigma_k$ is computable.

To conclude this section, we review the shadowing property for Cellular Automata.

Definition 2.22. An $\epsilon$-chain of $(A^Z, F)$ from $x_0 \in A^Z$ to $x_n \in A^Z$ is a sequence of configurations $x_i \in A^Z$ such that $d(f(x_i), x_{i+1}) < \epsilon$ for $0 \leq i \leq n$.

An $\epsilon$-chain is an approximation of an orbit. While such approximation works in general for a short number of steps, there are Cellular Automata whose orbits can be approximated for a large number of steps.

Definition 2.23. A point $x \in A^Z$ $\epsilon$-shadows in $(A^Z, F)$ a sequence $x_0, \ldots, x_n \in A^Z$ if $d(F^i(x), x_i) < \epsilon$ for $0 \leq i \leq n$.

Definition 2.24. A cellular automaton $(A^Z, F)$ has the shadowing property if for every $\epsilon > 0$ there exists a $\delta > 0$ such that every $\epsilon$-chain is $\delta$-shadowed by some point.

The definition of shadowing property holds for general dynamical systems. The orbits of a dynamical system with the shadowing property are approximable. Approximable Cellular Automata are a subclass of regular Cellular Automata.

Proposition 2.25 [13]. If $(A^Z, F)$ has the shadowing property then it is regular.

The converse of Proposition 2.25 is in general not true (see Example 5.78 in [15]).

2.3. Attractor classification

In dynamical systems, an attractor is a set toward which the system evolves after a long enough time. The orbits that get close enough to an attractor must remain close to it even if slightly perturbed. To define mathematically the concept of attractor we need to define the $\omega$-limit of a set.

Definition 2.26. The $\omega$-limit of a set $U \subseteq A^Z$ is $\omega(U) = \cap_{n > 0} \overline{\bigcup_{m=-n}^{m=\infty} F^m(U)}$.

Definition 2.27. A nonempty set $Z \subseteq A^Z$ is an attractor if there exists an $F$-invariant clopen set $U \subseteq A^Z$ such that $\omega(U) = Z$. A nonempty set is a quasi-attractor if it is the countable intersection of attractors but it is not an attractor. An attractor (quasi-attractor) is minimal if it does not contain any proper subset which is also an attractor (quasi-attractor).

Every $(A^Z, F)$ has at least the maximal attractor $\omega(A^Z)$ which is called limit set. A cellular automaton is stable if the limit set can be attained after a finite number of steps, i.e. if there exists some $n > 0$ such that $F^n(A^Z) = \omega(A^Z)$. 
Definition 2.28. The basin of attraction of an attractor \( Z \subseteq A^2 \) is defined as
\[
B(Z) = \{ x \in A^2 \mid \omega(x) \subseteq Z \}.
\]
The basins of attraction are open \( F \)-invariant sets (see, for example, [15]).

The following classification is Kürka’s refinement of Hurley’s Attractor classification for Cellular Automata [11].

Corollary 2.29. [13]. Every \((A^2,F)\) falls exactly in one of the following classes.

- **A1** There exist two disjoint attractors;
- **A2** There exists a unique minimal quasi-attractor;
- **A3** There exists a unique minimal attractor different from \( \omega(A^2) \);
- **A4** There exists a unique attractor \( \omega(A^2) \neq A^2 \);
- **A5** There exists a unique attractor \( \omega(A^2) = A^2 \).

We do not know whether the membership is decidable in some Attractor classes.

**Question 2.30.** Is the membership in Attractor classes decidable?

A natural class of attractors for Cellular Automata is the class of attractors which are also subshifts.

**Definition 2.31.** An attractor \( Z \) is a subshift attractor if \( \sigma(Z) = Z \).

Subshift attractors have been considered in [16] and [7]. They are characterized as the \( \omega \)-limit of clopen spreading sets.

**Definition 2.32.** A clopen \( F \)-invariant set \( U \subseteq A^2 \) is spreading if
\[
F^k(U) \subseteq \sigma^{-1}(U) \cap U \cap \sigma(U) \text{ for some } k > 0.
\]

**Proposition 2.33.** [7] Let \((A^2,F)\) be a cellular automaton and \( U \subseteq A^2 \) a clopen \( F \)-invariant set. Then \( \omega(U) \) is a subshift attractor if and only if \( U \) is spreading.

Since the limit set is actually a subshift, Cellular Automata have at least one subshift attractor but they can have also an infinite number of subshift attractors [16]. Kürka shows that, for surjective Cellular Automata, the full space is the unique subshift attractor [14]. Hurley shows that for Cellular Automata a minimal attractor is always a subshift attractor [11].

To conclude this section we show three examples which will be useful later. The first example shows a stable cellular automaton with a countable number of disjoint attractors and just one subshift attractor. The second example shows an unstable cellular automaton with a countable number of not disjoint attractors and with just one subshift attractor. The last example shows an unstable regular cellular automaton with just two \( \sigma \)-invariant attractors.

**Example 2.34.** The identity cellular automaton \((A^2,F)\) is defined by \( F(x) = x \) for every \( x \in A^2 \). For every \( u,v \in A^* \) such that \( u \subset v \) and \( v \subset u \) the two attractors \( \omega([u]) = [u] \) and \( \omega([v]) = [v] \) are disjoint. Moreover, since the identity cellular automaton is surjective, its unique subshift attractor is the full space.

**Example 2.35.** The Hurley cellular automaton, whose local rule \( f : \{0,1\}^2 \to \{0,1\} \) is defined by \( f(a,b) = ab \) has unique minimal quasi-attractor \( \{\infty0\infty\} \) (see [11] or [15]) and unique subshift attractor \( \omega(A^2) = \{ x \in A^2 \mid 10^4 \not\subseteq x \} \) (see [7]).

**Example 2.36.** The product cellular automaton of radius 1 on alphabet \( \{0,1\} \), whose local rule \( f : \{0,1\}^3 \to \{0,1\} \) is defined by \( f(x,y,z) = xyz \), has just two shift invariant attractors \( Z = \omega(A^2) = \{ x \in A^2 \mid 10^4 \not\subseteq x \} \) and \( Z'' = \{\infty1\infty\} \) whose basins of attraction are \( B(Z) = A^2 \) and \( B(Z'') = A^2 \setminus \{\infty1\infty\} \), respectively. By Theorem 2.18, a cellular automaton is regular if and only if \( \Sigma_{r+1} \) is a sofic shift. In this example, it is easy to see that \( \Sigma_3 \) is the one-sided sofic shift defined by the \( \sigma \)-closure of the sequences \((111)^* x(000)^\infty\), where \( x = (110) \mid (110)(100) \mid (011) \mid (011)(001) \mid (010) \).

### 3. Basin language classification and computational complexity of Cellular Automata

In this section we are interested in the basins of attraction of subshift attractors. We characterize the complexity of such basins by using formal language theory.
First we show that the basin of attraction of a subshift attractor is a dense open set. The property to be dense is in general not true for general attractors.

**Proposition 3.1.** The basin of a subshift attractor is a dense open set.

**Proof.** Let $Z$ be a subshift attractor of $(A^Z, F)$. Then $B(Z)$ is open so we just need to show that every $x \in A^Z$ belongs to the closure of $B(Z)$. Consider a clopen set $V \subseteq B(Z)$ and let $\epsilon > 0$. Since $(A^Z, \sigma)$ is mixing, there exists $n > 0$ such that $\emptyset \neq \sigma^n(B_\epsilon(x)) \cap V \subseteq \sigma^n(B_\epsilon(x)) \cap B(Z)$. Since $Z$ is a subshift, for all $n \in \mathbb{Z}, \sigma^{-n}(V) \subseteq B(Z)$ and $\emptyset \neq B_\epsilon(x) \cap \sigma^{-n}(V) \subseteq B_\epsilon(x) \cap B(Z)$. Then $x$ is in the closure of $B(Z)$. □

*A qualitative characterization* of basins of attraction is provided by formal language theory. By Proposition 3.1, the basin $B(Z)$ of a subshift attractor $Z$ is defined by the countable union of cylinder sets. A cylinder set can be (univocally) identified by some word in $A^*$. The collection of all such words is a language on $A$. Since the basin of a subshift attractor is $\sigma$-invariant, we do not need to take care of the coordinates of the cylinder in the space $A^Z$. This means that if a cylinder $[u]_j$ is contained in the basin of some subshift attractor $Z$, then for every $j \in \mathbb{Z}$, $[u]_j$ is contained in $B(Z)$ (equivalently, this implies that the orbit of every configuration which contains the word $u$ will converge to $Z$).

**Definition 3.2.** Let

$$L_Z = \{u \in A^* \mid [u] \subseteq B(Z)\} = A^* \setminus \mathcal{L}(A^Z \setminus B(Z))$$

denote the *basin language* of the subshift attractor $Z$ of $(A^Z, F)$.

Note that, since $B(Z)$ is open and $\sigma$-invariant, $Z^* = A^Z \setminus B(Z)$ is either a subshift or it is empty. In particular, the set $Z^*$ is also $F$-invariant and it is called *repeller* of $Z$ in Conley index theory (see, for example, [1]). The basin language of $Z$ is exactly the
set of forbidden words of the repeller \( Z^* \). We now show that basin languages are recursively enumerable (r.e. for short) and that there are r.e.-complete basin languages.

**Lemma 3.3.** Let \( (A^Z,F) \) be a cellular automaton. Let \( V \subseteq A^Z \) be a clopen \( F \)-invariant spreading set and let \( U \subseteq A^Z \) be a clopen set such that \( \omega(U) \subseteq V \). Then \( \exists n \in \mathbb{N} \) such that \( F^n(U) \subseteq V \).

**Proof.** Since \( V \) is clopen, \( \nabla = A^Z \setminus V \) is clopen and compact. For \( n \in \mathbb{N} \), let us define \( X_n = \{ x \in U \mid F^n(x) \notin V \} \subseteq U \cap \nabla \). Since \( U \) is clopen, every \( X_n \) is clopen. Moreover, since \( V \) is \( F \)-invariant, \( \forall n \in \mathbb{N}, X_{n+1} \subseteq X_n \). Assume for absurd that, \( \forall n \in \mathbb{N}, X_n \neq \emptyset \).

Then, by compactness, \( X = \cap_{n \in \mathbb{N}} X_n \subseteq U \cap \nabla \) is not empty and \( \omega(X) \cap \nabla \neq \emptyset \) which is a contradiction. \( \square \)

**Proposition 3.4.** Let \( Z \) be a subshift attractor of \( (A^Z,F) \). Then \( \mathcal{L}_Z \) is r.e.

**Proof.** Let \( U \subseteq A^Z \) be a clopen \( F \)-invariant spreading set such that \( \omega(U) = Z \). By Lemma 3.3, for every \( u \in A^* \), \( \{u\} \in B(Z) \) if and only if \( \exists n \in \mathbb{N} \) such that \( F^n(\{u\}) \subseteq U \). Since \( U \) is a finite union of cylinder sets, given some \( v \in A^* \) and \( k \in \mathbb{N} \), the property \( F^k(\{v\}) \subseteq U \) is decidable. This implies that \( \{u\} \in B(Z) \) is a semidecidable question. Then \( \mathcal{L}_Z \) is recursively enumerable. \( \square \)

The following proposition shows that every r.e. language recognition problem is Turing-reducible to the basin language recognition problem for some cellular automaton. For instance, we show that the question:

*does the Turing Machine \( M \) halt on input \( u \in B^* \)?*

can be restated as

\[
\text{is } \omega(\{\psi(u)\}) \subseteq Z?
\]

where \( \psi : B^* \rightarrow A^* \) is an injective computable mapping and \( Z \) is a subshift attractor of some cellular automaton \( (A^Z,F) \).

**Proposition 3.5.** Let \( \mathcal{L} \subseteq B^* \) be a r.e. language. Then there is a cellular automaton \( (A^Z,F) \) with a subshift attractor \( Z \) and an injective computable mapping \( \psi : B^* \rightarrow A^* \) such that \( u \in \mathcal{L} \) if and only if \( \psi(u) \in \mathcal{L}_Z \).

**Proof.** Let \( M = (B,Q,\delta,q_0,F) \) be a Turing machine recognizing the language \( \mathcal{L} \). Consider \( (A^Z,F) \) where \( A = B \cup Q \cup \{S,L,R\} \). The particle \( S \) is a spreading state. The particle \( L \) moves to left one step at time and erases everything on its path except when it collides with an \( R \) or with a \( q \in Q \): in that case an \( S \) is generated. The \( R \) particle behaves exactly like \( L \) but it moves on the right. The other particles simulate the computation of the Turing machine \( M \) (the tape alphabet symbols are always quiescent). When there is some erroneous step in the simulation of the computation, for example when two states collide, a particle \( S \) is generated. When the computation terminates in some accepting state then an \( S \) is generated. When the computation terminates in a non accepting state then the state is left unchanged. Note that \( \{^{\infty}S^{\infty}\} \) is a subshift attractor. Now, let us define the computable mapping \( \psi : B^* \rightarrow A^* \) by \( \psi(u) = Lq_0uK \). Then, by construction, it is easy to see that the word \( u \) is accepted by the Turing machine \( M \) if and only if \( \omega(Lq_0uK)) = ^{\infty}S^{\infty} \). \( \square \)

We can classify Cellular Automata according to basin languages complexities.

**Corollary 3.6.** Every \( (A^Z,F) \) falls exactly in one of the following classes:

- **B1** \( \forall Z, \mathcal{L}_Z = A^* \)
- **B2** \( \exists Z, \mathcal{L}_Z \neq A^* \) and \( \forall Z, \mathcal{L}_Z \) is recursive
The following lemmas show useful properties of product Cellular Automata.

**Proof.** By Lemma 4.2, $Z_{\mathcal{L}}$ is strictly r.e. and $\forall Z, Z_{\mathcal{L}}$ is not r.e.-complete.

**B3** $\exists Z, Z_{\mathcal{L}}$ is strictly r.e. and $\forall Z, Z_{\mathcal{L}}$ is not r.e.-complete

**B4** $\exists Z, Z_{\mathcal{L}}$ is r.e.-complete.

It is easy to see that basin languages classes **B1,B2,B4** are not empty. For instance, class **B1** coincides with the class of Cellular Automata with just one subshift attractor. The product cellular automaton of Example 2.36 is in class **B2**. By Proposition 3.5, class **B4** is not empty and it contains Cellular Automata with the computational complexity of universal Turing machines. We do not know whether class **B3** is empty or not and we do not know whether the membership is decidable for Basin languages classification.

**Question 3.7.** Is class **B3** empty?

**Question 3.8.** Is the membership in Basin Language classes decidable?

4. Classes comparison

In this section we compare Basin Language classification with Attractors, Equicontinuity and Languages classifications. We said in the previous section that we do not know whether class **B3** is empty or not. In the following we investigate intersection classes for **B3** under the assumption that it is not empty. This does not affect the properties of the other basin languages classes. First we show two techniques to build Cellular Automata with nice properties: the product Cellular Automata and Cellular Automata with a spreading state. These two constructions will be useful to investigate the intersection classes.

4.1. Cellular Automata extensions

A product cellular automaton is simply the product of two Cellular Automata.

**Definition 4.1.** The product cellular automaton $(A^x \times B^x, F \times G)$ of $(A^x, F)$ with $(B^x, G)$ is defined by $\forall(x, y) \in A^x \times B^x, (F \times G)(x, y) = (F(x), G(y))$.

The following lemmas show useful properties of product Cellular Automata.

**Lemma 4.2.** Let $(A^x \times B^x, F \times G)$ be a product cellular automaton. Then $Z = Z' \times Z'' \subseteq A^x \times B^x$ (with $Z' \neq \emptyset$ and $Z'' \neq \emptyset$) is a (subshift) attractor of $(A^x \times B^x, F \times G)$ if and only if $Z'$ and $Z''$ are (subshift) attractors of $(A^x, F)$ and $(B^x, G)$, respectively.

**Proof.** Let $U' \subseteq A^x, U'' \subseteq B^x$ be nonempty, respectively, $F, G$-invariant clopen (spreading) sets such that $\omega(U') = Z'$ and $\omega(U'') = Z''$. Then $U = U' \times U''$ is a $(F \times G)$-invariant clopen (spreading) set of $(A^x \times B^x, F \times G)$ and $\omega(U) = Z$. Conversely, assume that $U = U' \times U'' \subseteq A^x \times B^x$, with both $U'$ and $U''$ nonempty, is a $(F \times G)$-invariant clopen (spreading) set such that $\omega(U) = Z$. Then $U'$ and $U''$ must be, respectively, $F, G$-invariant (spreading) clopen.

**Lemma 4.3.** Let $(A^x, F) \in A_i$ and let $(B^x, G) \in A_j$ for $1 \leq i, j \leq 5$. Then $(A^x \times B^x, F \times G) \in A_k, k = \text{Min}(i, j)$.

**Proof.** By Lemma 4.2, $Z' \times Z''$ is an attractor of $(A^x \times B^x, F \times G)$ if and only if $Z', Z''$ are attractors of $(A^x, F)$ and $(B^x, G)$, respectively. If $(A^x, F) \in A_1$ then there exist two disjoint attractors $Z_1, Z_2 \subseteq A^x \times B^x, Z_1 \times Z_2 \times Z''$ are disjoint attractors of $(A^x \times B^x, F \times G)$. If $Z'' = \cap_i Z_i''$, $Z' = \cap_i Z_i'$ are the non-empty countable intersections of all attractors of $(A^x, F)$ and $(B^x, G)$, respectively, then the countable intersection of all attractors of $(A^x \times B^x, F \times G)$ is non-empty and it is defined as $Z = \cap_i (Z_i' \times Z_i'') = Z' \times Z''$. Then it easily follows that $Z$ is a quasi-attractor if and only if at least one of $Z'$ and $Z''$ is a quasi-attractor. If neither $Z'$ nor $Z''$ are quasi-attractors then $Z \neq \omega(A^x \times B^x)$ if and only if $Z' \neq \omega(A^x)$ or $Z'' \neq \omega(B^x)$. To conclude, it is sufficient to note that $Z = A^x \times B^x$ if and only if both $Z' = A^x$ and $Z'' = B^x$.

**Lemma 4.4.** Let $(A^x, F) \in E_3$. Then $(A^x \times B^x, F \times G) \in E_3$ for every cellular automaton $(B^x, G)$.

**Proof.** By Theorem 2.7 and Theorem 2.9, since $(A^x, F) \in E_3$, $\forall u \in A^x, u$ is not a blocking word for $(A^x, F)$. Then $\forall (u, v) \in A^x \times B^x, (u, v)$ is not a blocking word for $(A^x \times B^x, F \times G)$ which implies $(A^x \times B^x, F \times G) \in E_3$.

**Lemma 4.5.** Let $(A^x, F) \in L_3$. Then $(A^x \times B^x, F \times G) \in L_3$ for every $(B^x, G)$.

**Proof.** Let $r_1, r_2 > 0$ be the radius of $(A^x, F)$ and $(B^x, G)$, respectively, and let $r = \text{Max}(r_1, r_2)$ be the radius of $(A^x \times B^x, F \times G)$. Let $\Sigma_{2r}^{1}(F)$ and $\Sigma_{2r}^{1}(G)$ be the column factors of width $2r + 1$ of $(A^x, F)$ and $(B^x, G)$, respectively. By Theorem 2.18, the
subshift $\Sigma_{2r+1}(F)$ is not sofic. Then the $2r+1$ column factor $\Sigma_{2r+1} = \Sigma_{2r+1}(F) \times \Sigma_{2r+1}(G)$ of $(A^2 \times B^2,F \times G)$ is also not sofic and, by Theorem 2.16, it follows that $(A^2 \times B^2,F \times G) \in I_3$. □

**Lemma 4.6.** Let $(A^2,F) \in B_i$ and let $(B^2,G) \in B_j$ for $1 \leq ij \leq 4$. Then $(A^2 \times B^2,F \times G) \in B_k$ for $k = \max(i,j)$.

**Proof.** By Lemma 4.2, the language $L_2$ of the subshift attractor $Z = Z' \times Z''$ of $(A^2 \times B^2,F \times G)$ is $L_2 = L_{Z'} \times L_{Z''}$. Then the language complexity of $L_2$ is trivially the highest between the complexities of languages $L_{Z'}$ and $L_{Z''}$. □

Given a cellular automaton, we can easily extend it by adding a spreading state.

**Definition 4.7.** Let $(A^2,F)$ be of radius $r$ and let $A_s = A \cup \{s\}$ where $s \notin A$. Let $(A^2,F_s)$ denote the CA whose local rule $f_s : A_i^{2r+1} \rightarrow A_s$ is defined by

$$f_s(x_{r-1},...,x_r) = s \text{ if } \exists x_i = s \text{ and } f_s(x_{r-1},...,x_r) = f(x_{r-1},...,x_r) \text{ otherwise.}$$

Adding a spreading state can change the dynamical properties of the cellular automaton but it does not change the complexity of its basin languages.

**Lemma 4.8.** Consider $(A^2,F)$ and let $s \notin A$. Then $(A^2,F_s) \notin B_1$ and $(A^2,F_s) \in E_2 \cap A_3$. Moreover, $(A^2,F) \in B_2$ if and only if $(A^2,F_s) \in B_1 \cup B_2$ and $(A^2,F_s) \in B_i$ if and only if $(A^2,F) \in B_i, 3 \leq i \leq 4$.

**Proof.** By definition, $s$ is a blocking word. Moreover, $Z_s = (\infty s \infty) \neq \omega(A^2)$ is a fixed point attractor. Then $(A^2,F_s) \notin B_1$ and $(A^2,F_s) \in E_2 \cap A_3$. We now show that adding a spreading state does not affect the complexity of the basin languages of $(A^2,F)$. The basin of attraction of $Z_s$ consists of the set of all biinfinite sequences which contain at least one occurrence of the symbol $s$, that is $B(Z_s) = \{x \in A^2 | \exists i \in Z : x_i = s\}$. Then, the basin language $L_{Z_s}$ is recursive. It is easy to check that $Z$ is a subshift attractor of $(A^2,F_s)$ if and only if $Z = \omega(U \cup \{s\})$ where $U \subseteq A^2$ is a clopen $F$-invariant spreading set for $(A^2,F)$. Let $Z' = \omega(U) \subseteq A^2$ be a subshift attractor of $(A^2,F)$. Then $L_{Z} = L_{Z'} \cup L_{Z_s}$ and $L_{Z'} \cap L_{Z_s} = \emptyset$ which implies that $L_{Z}$ is strictly r.e. if and only if $L_{Z'}$ is strictly r.e. and, in particular, $L_{Z_s}$ is strictly r.e.-complete if and only if $L_{Z'}$ is strictly r.e.-complete. □

4.2. Comparison with attractor classification

In this section we explore the intersection classes between Attractors and Basin languages classifications (see Fig. 4).

**Corollary 4.9.** $A_1 \cap B_1 \neq \emptyset$, $A_1 \cap B_2 \neq \emptyset$, $A_1 \cap B_3 \neq \emptyset$, $A_1 \cap B_4 \neq \emptyset$.

**Proof.** The identity cellular automaton $(A^2,F)$ shown in Example 2.34 belongs to $A_1 \cap B_1$. Let $(B^2,G) \in B_i, 1 \leq i \leq 4$. Then, by Lemma 4.3 and Lemma 4.6, $(A^2 \times B^2,F \times G) \in A_1 \cap B_i$. □

**Corollary 4.10.** $A_2 \cap B_1 \neq \emptyset$, $A_2 \cap B_2 \neq \emptyset$, $A_2 \cap B_3 \neq \emptyset$, $A_2 \cap B_4 \neq \emptyset$.

**Proof.** Let $(A^2,F) \in A_2 \cap B_1$ be the Hurley cellular automaton of Example 2.35. Let $(B^2,G) \in B_i, 2 \leq i \leq 4$ and let $s \notin B$. By Lemma 4.8, $(B^2,G) \in A_3 \cap B_i$. Then, by Lemma 4.3 and Lemma 4.6, $(A^2 \times B^2,F \times G) \in A_2 \cap B_i$. □

**Corollary 4.11.** $A_3 \cap B_1 = \emptyset$, $A_3 \cap B_2 \neq \emptyset$, $A_3 \cap B_3 \neq \emptyset$, $A_3 \cap B_4 \neq \emptyset$.

**Proof.** If $(A^2,F) \in A_3$ then it has at least two subshift attractors: the minimal attractor and the maximal attractor. Then $A_3 \cap B_1 = \emptyset$. Let $(A^2,F) \in B_i$ for $2 \leq i \leq 4$ and let $s \notin A$. Then, by Lemma 4.8, $(A^2,F_s) \in A_3 \cap B_i \neq \emptyset$. □

To conclude, since a cellular automaton in $A_4 \cup A_5$ has only one attractor, we can easily derive the intersection classes for $A_4$ and $A_5$.

**Corollary 4.12.** $A_4 \cup A_5 \subseteq B_1$.

4.3. Comparison with Language classification

In this section we explore the intersection classes between Languages and Basin languages classifications (see Fig. 5).
By Proposition 2.16, the class L1 of bounded periodic Cellular Automata coincides with the class E1 of equicontinuous Cellular Automata. We show that every equicontinuous cellular automaton has exactly one subshift attractor.

**Proposition 4.13.** Every equicontinuous cellular automaton has a unique subshift attractor which is a mixing shift of finite type.

**Proof.** By Theorem 2.4, $Z = \omega(A^Z) = F^n(A^Z)$ for some $n \in \mathbb{N}$ and in particular there exists $m \geq 0$, such that for every $x \in A^Z$, and for every $i \geq m$ we have $F^{i+n}(x) = F^i(x)$. Then $Z$ is the image of the mixing shift of finite type $A^Z$ under a continuous $\sigma$-commuting function. Since the mixing property is preserved under factors, it follows that $Z$ is a mixing sofic shift. We show that $Z$ is actually a shift of finite type. Let $r$ be the radius of $(A^Z,F)$ and consider the shift of finite type defined by $Z^{2r+1} = \{ x \in A^Z \mid \forall i \in \mathbb{Z}, x|(2r+1) \in L_{2r+1}(Z) \}$, i.e. the shift of finite type identified by the set of legal $(2r+1)$-blocks of $Z$. Obviously, $Z \subseteq Z^{2r+1}$. Moreover, $F^r$ is the identity on $Z^{2r+1}$, hence $Z^{2r+1}$ is regular. Now, assume for absurd that there exists a subshift attractor $Z' \subset Z$. Let $U$ be a clopen spreading set such that $\omega(U) = Z'$. Since $Z' \neq Z$, $U \cap Z' \neq \emptyset$ and $Z$ is mixing, there exists $y \in Z$ and $m \in Z$ such that $y \in U$ and $\sigma^m(y) \notin U$. Then, for every $i \in \mathbb{N}$, $F^i(\sigma^m(x)) = \sigma^i(x) \notin U$ contradicting the fact that $U$ is spreading.

More generally, the basins of attraction of regular Cellular Automata give rise only to recursive basin languages.

**Proposition 4.14.** If $(A^Z,F)$ is regular then $\forall Z, L_Z$ is recursive.

**Proof.** We show that for every $u \in A^*$ the question $|u| \subseteq B(Z)$ is decidable. Let $U \subseteq A^Z$ be a clopen $F$-invariant spreading set such that $\omega(U) = Z$. Let $k = \max|u| \mid |u| \subseteq U$ and let $v \in A^*$. Since $(A^Z,F)$ is regular, by Theorem 2.21, it is possible to compute a finite state automaton representation $M$ of its column factor $\Sigma_N$ where $N = \max|k|^{|v|}$. Then $\omega(|u|) \subseteq Z$ if and only if there exists in $M$ an infinite path $q_1 \rightarrow q_2 \rightarrow q_3 \ldots$ such that $u \subseteq w_1$ and $|w_i| \subseteq U, \forall i \in \mathbb{N}$. Given a finite state automaton $M$, this property is easily decidable.

**Corollary 4.15.** $L_1 \subseteq B_1$, $L_2 \cap B_1 \neq \emptyset$, $L_3 \cap B_1 \neq \emptyset$.

**Proof.** Since, by Corollary 4.12, $A_5 \subseteq B_1$, the proof follows from the nonemptiness of the intersection classes $L_i \cap A_5 \neq \emptyset, 1 \leq i \leq 3$ (see Fig. 2) and from $L_1 = E_1 \subseteq B_1$ (see Proposition 2.16 and Proposition 4.13).

**Corollary 4.16.** $L_2 \subseteq B_1 \cup B_2$.

**Proof.** The cellular automaton of Example 2.36 has two subshift attractors and it is regular. Then $L_2 \cap B_2 \neq \emptyset$. The conclusion follows from Proposition 4.14 and Corollary 4.15.

**Corollary 4.17.** $L_3 \cap B_2 \neq \emptyset$, $B_3 \subseteq L_3$, $B_4 \subseteq L_3$.
Proof. Let \((A^Z,F) \in L_3 \cap B_1\) and let \((B^Z,G) \in L_2 \cap B_2\). Then, by Lemma 4.5 and Lemma 4.6, \((A^Z \times B^Z,F \times G) \in L_3 \cap B_2\). The inclusions \(B_3 \subset L_3\) and \(B_4 \subset L_3\) follow from Corollary 4.16. □

4.4. Comparison with equicontinuity classification

In this section we explore the intersection classes between Equicontinuity and Basin languages classifications (see Fig. 6).

**Corollary 4.18.** \(E_1 \subset B_1\), \(E_2 \cap B_1 \neq \emptyset\), \(E_3 \cap B_1 \neq \emptyset\), \(E_4 \subset B_1\).

**Proof.** By Proposition 4.13, \(E_1 \subset B_1\). Moreover \(E_4 \subset A_5 \subset B_1\) (see Fig. 1 and Corollary 4.12). For the other two cases, the proof follows from Corollary 4.12 and from \(E_i \cap A_5 \neq \emptyset, 2 \leq i \leq 3\) (see Fig. 1). □

**Corollary 4.19.** \(E_2 \cap B_2 \neq \emptyset\), \(E_2 \cap B_3 \neq \emptyset\), \(E_2 \cap B_4 \neq \emptyset\).

**Proof.** Let \((A^Z,F) \in B_i, 2 \leq i \leq 4\), and let \(s \notin A\). Then, by Lemma 4.8, \((A^Z,F) \in E_2 \cap B_i\). □

**Corollary 4.20.** \(E_3 \cap B_2 \neq \emptyset\), \(E_3 \cap B_3 \neq \emptyset\), \(E_3 \cap B_4 \neq \emptyset\).
5. Conclusions

We investigated the connections between dynamical and computational properties of Cellular Automata. We classified Cellular Automata according to the complexity of the languages rising from the basins of attraction of subshift attractors (see Corollary 3.6). According to our classification, Cellular Automata with the computational power of always-halting Turing machines are contained in class $B_2$. Cellular Automata capable of universal computation are in our highest complexity class $B_4$. This does not mean that all Cellular Automata capable of universal computation are in class $B_4$. For instance, there are invertible Cellular Automata which are universal but, according to our classification, invertible Cellular Automata are contained in the lowest complexity class $B_1$. We further investigated the intersection classes between our classification and Attractors, Languages and Equicontinuity classifications (see Figs. 4–6). The non regularity property permits us to distinguish Cellular Automata in class $B_2$ from those in higher complexity classes. There is no topological property which permits us to distinguish Cellular Automata in class $B_4$ from those in class $B_3$ and, actually, we do not know whether class $B_3$ is empty or not. However, by exploring intersection classes, we can provide necessary conditions for Cellular Automata to be universal. Like in $[5]$, according to our model, a universal cellular automaton is not regular (then it is not equicontinuous, not positively expansive and does not satisfy the shadowing property) and is not minimal (minimal Cellular Automata cannot have two distinct subshift attractors so they belong to our lowest complexity class). It is actually open whether a cellular automaton in class $B_4$ must have an infinite number of subsystems. Several other questions remain open, first of all the decidability of the membership in Basin language classes. Other questions regard the possibility to derive some other properties of class $B_4$ such as stability or cardinality of the number of subshift attractors. For instance:

1. Is there some cellular automaton with a finite number of subshift attractors in class $B_4$?
2. Is there some stable cellular automaton in class $B_4$?

Acknowledgments

We thank the anonymous referees for their useful comments and remarks.

References