Dottorato di Ricerca in Informatica
Università di Bologna e Padova

Decidable and Computational properties of Cellular Automata

Pietro Di Lena

December 2006

Coordinatore:
Prof. Özalp Babaoğlu

Tutore:
Prof. Luciano Margara

Contents

List of Figures

Chapter 1

Introduction

In this thesis we investigate *decidable* and *computational* properties of *Cellular Automata*. This investigation is intended to be a contribute to the study of the more general theory of *Complex Systems*.

A central interest in the sciences of complex systems is to understand the laws by which a global complex behavior can emerge for the collective interaction of simple components. *Computation Theory* and *Dynamical System Theory* provide a general framework for understanding and describing the behavior of such systems. Since Cellular Automata offer a very large and diverse dynamical behavior as well as a wide variety of possible computational models, they represent an ideal subject to investigate the possible relations between dynamics and computation.

In this chapter we provide an overview of the current approaches on this subject and next an overview of our results.

1.1 Wolfram's classification

Wolfram proposed two different approaches to investigate Cellular Automata. The following two sentences are quoted from [60].

> In the first approach, Cellular Automata are viewed as discrete dynamical systems, or discrete idealizations of partial differential equations. The set of possible (infinite) configurations of a cellular automaton forms a Cantor set; cellular automaton evolution may be viewed as a continuous mapping on this Cantor set. Quantities such as entropies, dimensions and Lyapunov exponents may then be considered for Cellular Automata.
>
> In the second approach, Cellular Automata are instead considered as information-processing systems, or parallel-processing computers of simple construction. Information represented by the initial configuration is processed by the evolution of the cellular automaton. The results of this information processing may then be characterized in terms of the types of formal languages generated. (Note that the mechanisms for information processing in natural system appear to be much closer to those in Cellular Automata than in conventional serial-processing computers: Cellular Automata may, therefore, provide efficient media for practical simulations of many natural systems.)

Adopting this approach, in [61] Wolfram proposed an heuristic classification of Cellular Automata based on the qualitative observed behavior of Cellular Automata by performing computer simulations of the evolution starting from random configurations. According to his observations every cellular automaton falls in one of the following classes:

1. Evolution leads to a homogeneous state (i.e. a fixed point for the shift map);

2. Evolution leads to a set of separated simple stable or periodic structures;

3. Evolution leads to a chaotic pattern;

4. Evolution leads to complex localized structures, sometimes long-lived.

Wolfram suggested that the different behavior of automata in his classes seems to be related to the presence of different types of attractors. For instance, the first class seems to be related to the presence of fixed point attractors, the second class to the presence of periodic attractors while the third class of chaotic attractors. Moreover he conjectured that Cellular Automata in class 4 must be capable of universal computation. There have been several attempts to formalize Wolfram's classification using concepts both from dynamical systems theory [30, 24] and from formal language theory [37, 11]. In all these classifications it is not clear how the dynamical properties are related to the computational properties of Cellular Automata except for the connection with Wolfram's empirical classes.

1.2 Computation at the edge of chaos

In [41] Langton tried to make a quantitaive analysis of Wolfram's classification by introducing the λ parameter, a statistical value computable from the local rule of Cellular Automata. Langton studied the *average dynamics* by performing Monte Carlo samples of two-dimensional Cellular Automata in an attempt to characterize such *average behavior* as a function of λ. According to his observation, as the value λ increases starting from 0, the average behavior of the automaton passes through the four different classes of behavior:

$$\text{fixed point} \rightarrow \text{periodic} \rightarrow \text{complex} \rightarrow \text{chaotic}.$$

This four classes roughly correspond to Wolfram's classes $1, 2, 4, 3$ respectively. Langton observes that as λ value increases there is a phase transition between *highly ordered* and *highly disordered* dynamics. Class 4 (complex) behavior seems to be related to a *phase transition* between such ordered and chaotic behavior and seems to be associated to a critical λ_c value. Langton hypothesizes that Cellular Automata computational capability are related to the average behavior which is in turn related to the λ values. Thus, Cellular Automata capable of perform nontrivial computation, in particular universal computation, are most likely to be found near λ_c values. This is the orgin of the notion of *computation at the edge of chaos.*

Both Wolfram's and Langton's studies deal with a generic or average behavior and they don't provide any kind of qualitative measure of the computational capability of Cellular Automata. This question was partially adressed by Packard in [49] which used genetic algorithm to evolve Cellular Automata to perform some specific computational task. His experiment was meant to test two hypothesis:

1. Cellular Automata able to perform complex computations are most likely to be found near λ_c values.
2. When Cellular Automata are evolved to perform a complex computation, the evolution will tend to select rule near λ_c values.

The results of Packard investigation seem support Langton's thesis. However, while trying to replicate Packard's results, Mitchell and colleagues found results which contradict those of Packard [45].

1.3 Computation with dynamical systems

What lacks in two previous approaches is a meaningful notion of computation for dynamical system. In particular, there's 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 (see, for example, [52]) or some other system which is know to be computationally universal (see, for example, [14]).

While it is generally accepted to interpret the evolution of a dynamical system as a process of computation, it is much more less clear 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 [8] 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 [16]. 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 if and only if it enters into all clopen invariant sets whose omega limits coincide with such attractor. The authors of [16] 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, not equicontinuos and does not satisfy the shadowing property. Moreover they conjecture that a universal dynamical system must have an infinite number of subsystems.

1.4 Overview of the dissertation

In Chapter 2, we review briefly computation theory, formal language theory and symbolic dynamics theory which are necessary frameworks for our investigation. Chapter 3 provides a detailed overview of Cellular Automata, mostly in the contest of dynamical systems. In Chapter 4, we investigate the class of regular Cellular Automata. We are mostly interested in decidable properties of regular Cellular Automata. We show that regularity is an undecidable property, i.e. there is no algorithm which can decide if some cellular automaton is regular. Despite this negative result, the dynamics of regular Cellular Automata is, in some sense, predictable. A fact which supports this argument is that some of the topological properties which are in general undecidable for general Cellular Automata are decidable if we restrict only to the class of regular Cellular Automata. This suggests that regularity is a property which cannot be related to computational universality. In Chapter 5, we introduce a measure of *computational complexity* for Cellular Automata. We follow an approach very close to the one reviewed in Section 1.3. We consider the process of computation in Cellular Automata as a a flow toward a subshift attractor. The basins of attraction of subshift attractors are dense open sets. We characterize such basins of attraction by using formal language theory and we show that deciding whether some Turing machine halts on some input word is equivalent to decide if some basin of attraction contains some open set. We can then have arbitrarly high basin languages complexity. We introduce a classification of Cellular Automata related to such basin languages complexity. In our classification the computational power of Cellular Automata is explicitly related to a topological property. We can then explore the intersection classes between our classification and other topological classification of Cellular Automata. From the emptiness of some intersection classes we can easily derive some necessary dynamical conditions for the universality. In particular we show that, according to our model, regular Cellular Automata cannot be universal.

Chapter 2

Theoretical frameworks

Here we provide a very brief introduction on the subject of computation theory, formal languages and symbolic dynamics. The main motivation of this chapter is to introduce notations and basic results rather than open problems and research directions on the subjects. For an introduction on Computation Theory and Formal Language refer to [29]and refer to [42] for and introduction on Symbolic Dynamics.

2.1 Computation Theory and Formal Languages

The *theory of computation* is the branch of computer science whose central question is addressing the limits of computing devices by understanding the class of problems which can be solved on a computer. In order to perform a rigorous study, computer scientists work with mathematical abstractions of computers called *models of computation*. There are several formulations in use, but the most commonly examined is the *Turing Machine*. A Turing machine is an idealization of a computer with an infinite memory capacity. Even given arbitrarily vast computational resources, it is possible to show clear limits to the ability of computers to solve even simple problems. The goal of *Computation Theory* is to answer the question whether it is possible to define a formal sense in which we can understand how hard it's to solve a particular problem on a computer. To explore these areas, computer scientists usually address the ability of a computer to answer the question: given a formal language, and a word, is the word a member of that language?

Formally a language is a collection of finite length words on a finite alphabet. Let A be a finite alphabet. For $n > 0$, $A^n = \{a_1...a_n \mid a_i \in A, 1 \leq i \leq n\}$ is the set of blocks on A of length n while $A^0 = \{\epsilon\}$ is the set containing just the empty word ϵ. The set of finite words on A is defined as $A^* = \cup_{n>0} A^n$ and $A^+ = A^* \setminus \{\epsilon\}$. A *language* $\mathcal{L}$ on finite alphabet A is defined as a subset of A^*, i.e. $\mathcal{L} \subseteq A^*$.

In order to begin to answer the central question of computability theory, it is necessary to define in a formal way what a computer is. There are a number of useful models of computation. In the following sections we formally define some models of computation and the languages they accept.

2.1.1 Recursively Enumerable languages and Turing machines

A *Turing machine* is an hypothetical machine defined in 1935 by Alan Turing [55]. It consists of an infinitely long *tape* with symbols (chosen from some finite set) written at regular intervals. A pointer marks the current position and the machine is in one of a finite set of *internal states*. At each step the machine reads the symbol at the current position on the tape. For each combination of current state and symbol read, a program specifies the new state and either a symbol to write to the tape or a direction to move the pointer (left or right) or to halt.

Definition 2.1. *A Turing machine M is a 7-tuple $(Q, A, b, I, \delta, q_0, F)$ where*

- *Q is a finite set of states;*
- *A is a finite set of the tape alphabet/symbols;*
- *b is the blank symbol (the only symbol allowed to occur on the tape infinitely often at any step during the computation);*
- *$I = A \setminus \{b\}$ is the set of input symbols;*

- $\delta : Q \times A \to Q \times A \times \{L, R\}$ *is a partial function called the transition function, where* L *is left shift,* R *is right shift;*
- $q_0 \in Q$ *is the initial state;*
- $F \subseteq Q$ *is the set of final or accepting states.*

The *halting problem* is one of the most famous problems in computer science, because it has deep implications on the theory of computability and in how we use computers in everyday practice. The problem can be phrased as follows. Given a description of a Turing machine and its initial input, determine whether the program, when executed on this input, ever halts (completes). The alternative is that it runs forever without halting. That is, the only general way to know for sure if a given program will halt on a particular input in all cases is simply to run it and see if it halts. If it does halt, then you know it halts. If it doesn't halt, however, you may never know if it will eventually halt. The historical importance of the halting problem lies in the fact that it was one of the first problems to be proved undecidable [55].

The language recognized by a Turing machine generally can be only *enumerated*, then the class of languages accepted by a Turing machine is called *recursively enumerable.*

Definition 2.2. *The language* $\mathcal{L}_M$ *accepted by* $M = (Q, A, b, I, \delta, q_0, F)$ *is defined as*

$$\mathcal{L}_M = \{w \in I^* \mid M \text{ on input } w \text{ halts in an accepting state } q \in F\}.$$

It is possible to construct languages which are not even recursively enumerable, however. For instance the complement $\overline{L} = A^* \setminus L$ of a strictly recursively enumerable language $L \subset A^*$ cannot be recursively enumerable. It is not so difficult to see that if both L and $\overline{L}$ are recursively enumerable it is possible to build a Turing machine M which works in parallel to check if some word $w \in A^*$ is in L or $\overline{L}$. Then the Turing machine M halts on every word $w \in L$ thus L cannot be strictly recursively enumerable.

The language accepted by a Turing machine which halts on every input is called *recursive.*

Definition 2.3. *A language $L \subseteq A^*$ is* recursive *if there is a Turing machine M such that $L = \mathcal{L}_M$ and such that M halts on every word $w \in A^*$.*

This type of language was not defined in the Chomsky hierarchy [12]. Examples of recursive languages are *contex-sensitive* languages which coincide with the class of languages which can be recognized by a Turing machine which works only on a portion of the tape whose length is exactly the length of the input word. Since the portion of the tape is bounded, there is always an upper bound on the number of steps the machine must do before to in a loop. Thus it is always possible to decide if a word is accepted or not.

2.1.2 Regular languages and Finite State Automata

A *Finite State Automaton*, or finite state machine, is a simpler and less powerful model of computation than Turing machines.

Definition 2.4. *A* Finite State Automaton *(FA) is a graph determined by a 5-tuple (Q, A, δ, q_0, F) where*

- Q *is a* finite set of states*;*
- A *is a* finite alphabet*;*
- $\delta : Q \times A \to \mathcal{P}(Q)$ *is a partial* transition function *to the set $\mathcal{P}(Q)$ of subsets of Q;*
- $q_0 \in Q$ *is the* initial state*;*
- $F \subseteq Q$ *is the set of* accepting *states.*

The automaton is deterministic *(DFA) if the transition function is of the form* $\delta : Q \times A \to Q$*, i.e. for every state in* $q \in Q$ *and every symbol* $a \in A$ *there is at most one state (possibly none)* $q' = \delta(q, a)$.

We can represent a FA $M = (A, Q, \delta, q_0, F)$ as a graph with a vertex for every state in Q. There is an edge labeled $a \in A$ between vertices $p, q \in Q$ if and only if $\delta(p, a) = q$. We use the notation $p \xrightarrow{w} q$ to denote the existence of a path in starting at state $p \in Q$ and terminating at state $q \in Q$ such that the labels of the edges in the path generate the word $w \in A^*$.

The language *accepted* or *recognized* by a FA is the set of words identified by paths starting from the initial vertex and ending in the terminal vertex. Such class of languages is called *regular.*

Definition 2.5. *The language* accepted *by a finite state automaton* $M = (A, Q, \delta, q_0, F)$ *is defined as*

$$\mathcal{L}_M = \{w \in A^* \mid \exists q \in F, q_0 \xrightarrow{w} q\}.$$

Example 2.6. In Figure 2.1 is represented the smallest DFA recognizing the language $\mathcal{L} = \{w \in \{0,1\}^* \mid 0 \sqsubseteq w, \forall k > 1, 01^{2k}0 \not\sqsubseteq w\}$, i.e. the language of blocks on $\{0, 1\}$ which contain at least one 0 and which do not contain any even sequences of 1s between two consecutive 0s. The initial state is the state q_0. The accepting states are are signed with a double circle.

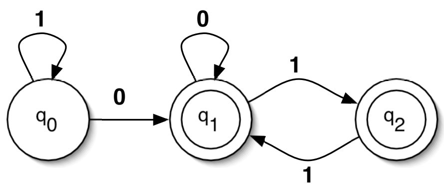


Figure 2.1: Example of smallest Deterministc Finite State Automaton

Regular languages have been widely studied in literature. The class of regular languages recognized bu FA coincides with the class of regular languages recognized by DFA. In particular, for every regular language exists a unique minimal (in the number of states) finite state automaton up to graph isomorphism and state renaming which recognizes it.

2.2 Symbolic Dynamics Theory

Symbolic dynamics originated as a method to study general dynamical systems. A dynamical system can be modelled as a space consisting of *infinite sequences of symbols* where each symbol corresponds to a state of the system. The dynamics of the systems is represented by *shifting* the sequences of symbols. Here we define *symbolic spaces*, the *shift operator* on a symbolic spaces and introduce some class of *symbolic dynamical systems*.

Let A denote a finite set with the discrete topology. For $n > 0$, A^n is the set of blocks on A of length n and $A^0 = \{\epsilon\}$ is the set containing just the empty word ϵ. The set of finite words on A is defined as $A^* = \cup_{n>0} A^n$ and $A^+ = A^* \setminus \{\epsilon\}$. For $a \in A$ we denote with a^∞ the biinfinite sequence $...aaaa...$. The *concatenation* of words $u, v \in A^*$ is denoted with uv. We say that $u \in A^*$ is a *subsword* of $v \in A^*$ if there exist $a, b \in A^*$ such $v = aub$. We use the shortcut $u \sqsubseteq v$ to denote that u is a subword of v.

By *symbolic space* we mean the product space $A^{\mathbb{Z}}$ with the product topology. An element of $A^{\mathbb{Z}}$ is a doubly infinite sequence of symbols in A:

$$x = ...x_{-1}x_0x_1 = (x_i)_{i\in\mathbb{Z}} \text{ where } x_i \in A, \forall i \in \mathbb{Z}.$$

The space $A^{\mathbb{Z}}$ is compact, metrizable and a metric compatible with the topology is defined by

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

By distance d, two sequences are close if they coincide on a large interval around the zero coordinate. If $[i, j] \subset \mathbb{Z}$ is an interval and $x \in A^{\mathbb{Z}}$ we denote with $x_{[i,j]}$ the word $x_i x_{i+1} ... x_j \in A^*$. Given a finite word $u \in A^+$, the set

$$[u]_i = \{x \in A^{\mathbb{Z}} \mid x_{[i,|u|-1]} = u\}$$

is called *cylinder set*. The cylinder sets are clopen (closed and open) sets and they are a basis for the product topology on $A^{\mathbb{Z}}$.

The *shift map* $\sigma : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ is the homeomorphism defined by $\sigma(x)_i = x_{i+1}$. The *symbolic dynamical system* $(A^{\mathbb{Z}}, \sigma)$ is called the *full shift*. A *subshift* $S \subseteq A^{\mathbb{Z}}$ is a closed and σ-invariant (i.e. $\sigma(S) = S$), subset of a full shift. Sometimes we will consider also the *one-sided full shift* $A^{\mathbb{N}}$ and *one-sided subshifts* $S \subseteq A^{\mathbb{N}}$. Note that the shift map σ on $A^{\mathbb{N}}$ is not biiective.

A subshift S is *mixing*, if for any $x, y \in S$ and any $\epsilon > 0$ there exists $m > 0$ such that for every $k \geq m$ there exists $z \in S$ such that $d(x,z) < \epsilon$ and $d(y, \sigma^k(z)) < \epsilon$.

Let S_1, S_2 be two different subshifts. A *factor map* $\varphi : S_A \to S_B$ is a continuous onto map such that $\varphi\sigma = \sigma\varphi$. If the factor map is also injective it is called *conjugacy*. Factor maps between shift spaces can be characterized in a very concrete way as the class of continuous mappings induced by block maps (Curtis-Lyndon-Hedlund Theorem). A *block code* $\varphi : S_A \to S_B$ between shift spaces $S_A \subseteq A^{\mathbb{Z}}$ and $S_B \subseteq B^{\mathbb{Z}}$ is a continuous σ-commuting function induced by some *block mapping* $f : A^{l+r+1} \to B$, $l, r \geq 0$:

$$\forall x \in S_A, F(x)_i = f(x_{i-l}, ..., x_{i+r}).$$

Theorem 2.7. *(Curtis-Lyndon-Hedlund [26]) Every continuous σ-commuting map between shift spaces is a block code.*

Example 2.8. The shift map $\sigma : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ itself is a block map induced by the 2-block mapping $f : A^2 \to A$ defined by $f(a_1, a_2) = a_2, \forall a_1, a_2 \in A$.

A shift space S can be conveniently recoded according a factor map of the form $\varphi : S \to S^k$ such that $\varphi(x)_i = x_{[i,i+k-1]}$. The shift space S^k is called the *higher k-block presentation* of S and it is topologically conjugated to S.

A subshift $S \subseteq A^{\mathbb{Z}}$ can be characterized by the *language* of words which occur in its sequences.

Definition 2.9. *Let $S \subseteq A^{\mathbb{Z}}$ be a subshift. The set of words of length $k > 0$ of S is denoted as*

$$\mathcal{L}_k(S) = \{w \in A^k \mid \exists x \in S, x_{[1,k]} = w\}.$$

The language *of S is defined as*

$$\mathcal{L}(S) = \cup_{k>0} \mathcal{L}_k(S).$$

The language of a subshift S is:

- *factorial*: $\forall u \in \mathcal{L}(S)$ and $\forall v \sqsubseteq u, v \in \mathcal{L}(S)$.
- *extendable*: $\forall u \in \mathcal{L}(S), \exists v_1, v_2 \in \mathcal{L}(S)$ such that $v_1 u v_2 \in \mathcal{L}(S)$.

A susbhift S is univocally determined by its language $\mathcal{L}(S)$ [5].

The exponential growth rate of words in a subshift S is a topological invariant of S. This quantity is called *topological entropy.*

Definition 2.10. *The* topological entropy *of a subshift S is defined as*

$$H(S) = \lim_{n\to\infty} \frac{\log |\mathcal{L}_n(S)|}{n}$$

Example 2.11. Consider the full shift $(A^{\mathbb{Z}}, \sigma)$ on $N = |A|$ symbols. For every $k > 0$, $\mathcal{L}_k(A^{\mathbb{Z}}) = A^k$ and $\mathcal{L}(A^{\mathbb{Z}}) = A^*$. The topological entropy of $(A^{\mathbb{Z}}, \sigma)$ is

$$H(A^{\mathbb{Z}}) = \lim_{n\to\infty} \frac{\log |\mathcal{L}_n(A^{\mathbb{Z}})|}{n} = \lim_{n\to\infty} \frac{\log N^n}{n} = \log N.$$

There are several classes of subshifts. Here we are interested essentially in *shifts of finite type* (SFT) and *sofic shifts*. Shifts of finite type were introduced by Parry [50]. The class of sofic shifts was introduced by Weiss [56] as the smallest class which is closed under factors and which contains shifts of finite type. A sofic shift can defined as the image of a shift of finite type under a factor map. The topological entropy of sofic shift is always computable (see, for example, [42]).

Since a subshift $S \subseteq A^{\mathbb{Z}}$ is a closed subset of a full shift $A^{\mathbb{Z}}$, its complement $A^{\mathbb{Z}} \setminus S$ is open and it is thus a countable union of cylinder sets. A cylinder set is univocally identified by some word in A^*. Then any subshift may be defined by forbidding a countable collection of words. If a subshift can be defined by forbidding a finite collection of words then it is a shift of finite type.

Definition 2.12. *A subshift $S_F \subseteq A^{\mathbb{Z}}$ is a* shift of finite type *iff there exists a finite collection of words $F \subset A^*, |F| < \infty$ such that $S_F = \{x \in A^{\mathbb{Z}} \mid \forall i < j, x_{[i,j]} \notin F\}$.*

Example 2.13. Consider the 2-full shift $(\{0,1\}^{\mathbb{Z}}, \sigma)$ and the set of forbidden blocks $F = \{11\}$. The shift of finite type $S_{\{11\}} = \{x \in \{0,1\}^z \mid \forall i \in \mathbb{Z}, x_{[i,i+1]} \neq 11\}$ is known as *golden mean* shift. The language of the golden mean shift is

$$\mathcal{L}(S_{\{11\}}) = \{\epsilon, 0, 1, 00, 01, 10, 000, 001, 010, 100, 101, 0000, ...\}.$$

Definition 2.14. *A subshift S is a* sofic shift *if and only if exist a shift of finite type T and a factor map $\varphi : T \to S$.*

Equivalently, a subshift is *sofic* if and only if it can be represented by means of a *labeled graph*. A labeled graph is a finite state automaton such that every vertex is initial and such that every state is accepting. The language of a sofic shift is always regular. We review the representation of a sofic shift as the shift space defined by the labeling of *vertex shift* of a labeled graph.

Definition 2.15. *A* graph *G is a pair $(\mathcal{V}_G, \mathcal{E}_G)$ where $\mathcal{V}_G$ is a finite set of* vertices *and $\mathcal{E}_G$ is a finite set of* edges. *Every edge $e \in \mathcal{E}_G$ identifies a* starting vertex *$s(e) \in \mathcal{V}_G$ and a* terminal vertex *$t(e) \in \mathcal{V}_G$.*

For notational convenience, when it is clear from the context, we denote vertex and edge set of graph G simply as $\mathcal{V}$ and $\mathcal{E}$, respectively.

Definition 2.16. *Let $G = (\mathcal{V}, \mathcal{E})$ be a graph. The* vertex shift Σ *of G is defined as*

$$\Sigma = \{v = (v_i)_{i\in\mathbb{Z}} \in \mathcal{V}^{\mathbb{Z}} \mid \forall i \in \mathbb{Z}, \exists e \in \mathcal{E}, s(e) = v_i, t(e) = v_{i+1}\}.$$

The vertex shift of a graph is always of finite type.

Definition 2.17. *A* labeled graph $\mathcal{G}$ *is a pair (G, ζ), where G is a graph equipped with a labeling function $\zeta : \mathcal{V}_G \to A$ which maps vertices into a finite alphabet A.*

Every sofic shift can be presented by labeling of vertices of a graph.

Definition 2.18. *Let $\mathcal{G} = (G, \zeta)$ be a labeled graph with $\zeta : \mathcal{V} \to A$. The sofic shift* S presented *by $\mathcal{G}$ is defined by*

$$S = \{a = (a_i)_{i\in\mathbb{Z}} \in A^{\mathbb{Z}} \mid \exists v = (v_i)_{i\in\mathbb{Z}} \in \Sigma_{\mathcal{V}}, \zeta(v_i) = a_i, \forall i \in \mathbb{Z}\}.$$

Note that a labeled graph (G, ζ) *identifies* a 1-block mapping $\varphi : \Sigma \to S$.

Example 2.19. Consider the golden mean shift of Example 2.13. The (strictly) sofic shift obtained by the 2-block mapping $f(0,0) = 0, f(0,1) = f(1,0) = 1$ is called *even shift*. In figure 2.2 we can see a labeled graph presenting the 2-block presentation of the golden mean as a vertex shift and the even shift presented by the labeling of the vertices.

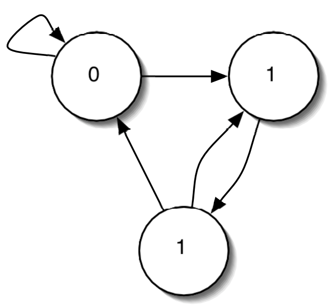


Figure 2.2: Golden mean and Even shift.

Chapter 3

Cellular Automata

Cellular automata (CA) were introduced by Von Neumann in the fifties [48] as a simple mathematical model capable of *universal* computation and *self-reproduction* like in biological systems.

According to the original definition, a cellular automaton consists of an infinite, regular grid of cells. The grid can be in any finite number of dimensions (Von Neumann's cellular automaton is bidimensional). Each cell can be in one of a finite number of possible states. Time is also discrete, and the state of a cell at time t is a function of the states of a finite number of cells (called its neighborhood) at time $t-1$. Every cell has the same rule for updating.

The best well known example of cellular automaton is the Conway's *Game of Life* which made its first public appearance in the October 1970 issue of Scientific American [23]. The mathematician John Conway was interested in simplifying Von Neumann's model (his original cellular automaton consists of 29 states per cell) and succeed to find a simple example of cellular automaton (with just two state per cell) capable of universal computation. Since its publication, Game of Life has attracted much interest because of the surprising ways the patterns can evolve. It is an example of emergence and self-organization. It is interesting for physicists, biologists, economists, mathematicians, philosophers and others to observe the way that complex patterns can emerge from very simple rules.

At Conway time there was no high availability of fast and cheap computers so the earliest results in the Game of Life were obtained without the use of computers. With the increase of processors speed and availability of cheap computers, Cellular Automata found many applications in the field of simulation of natural processes. Since here we are not interested in Cellular Automata as simulation models, we cite just two of the most remarkable examples of use of Cellular Automata for modeling. Among the most famous application there is the HPP lattice gas model [28]. The HPP dynamics was initially planned as a theoretical model to study the fundamental statistical properties of a gas of interacting particles and next it found some practical applications. A case of industrial application of HPP is the simulation of water percolation process occurring in a porous medium: ground and toasted coffee [10]. This work has been developed within the cellular automata for percolation processes (CAPPs) transfer technology project [4]. Traffic control is another application area that involves CA models and systems. An overview of the main results in this area can be found in [51]. The main applications concern both urban and extra-urban traffic, and the CA approach allows the knowledge of the traffic state to be explicitly represented in the model in order to simulate crucial situations (i.e. traffic jams).

The computational capabilities of Cellular Automata have been studied extensively since the beginning and it was well known since than that Cellular Automata have the same computational capabilities of Turing Machines (see, for example, [54, 13, 2, 9]). There's 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 or some other system which is known to be computationally universal. For example, the Game of Life was proved to be computationally universal by using some special patterns in Life (known as *gliders* and *guns*) to implement logical gates [3]. In [52], the author shows that any Turing machine with m symbols and n states can be simulated by a one-dimensional cellular automaton with $m + 2n$ states. More recently, in [14] the author proves that a very simple one-dimensional cellular automaton with just two states is universal by showing it is possible to use the rule to emulate another computational model, the cyclic tag system, which is also universal.

Mathematical theory of Cellular Automata was developed by Hedlund [26] about two decades later Von Neumann's work. Hedlund studied Cellular Automata in the context of symbolic dynamics as homomorphisms of the full shift. Hedlund's work is not related directly to Cellular Automata but with the current problems in symbolic dynamics. However, despite the differences of objectives, symbolic dynamics theory provides many useful tools even for the investigation of computational properties of Cellular Automata. This is actually the theoretical framework in which we study Cellular Automata.

Most of the research on Cellular Automata from the dynamical systems point of view was instead stimulated in the eighties by Wolfram's studies on dynamical and computational aspects of Cellular Automata [57, 58, 59]. In [61], Wolfram proposed an heuristic classification of Cellular Automata based on the qualitative observed behavior of a meaningful class of Cellular Automata by performing computer simulations of the evolution of the automata starting from random configurations. Wolfram suggested that the different behavior of automata in his classes seems to be related to the presence of different types of attractors. There have been several attempts to formalize Wolfram's classification using concepts both from dynamical systems theory [30, 24] and from formal language theory [37, 11]. The most well known are Equicontinuity, Attractors and Languages classifications (see **??**).

The rest of the chapter is organized as follows. In Section 3.1, we provide a formal definition of Cellular Automata while in Section 3.2, we provide a detailed introduction of Cellular Automata in the context of Dynamical System theory. In particular, in Sections 3.2.2, 3.2.3 and 3.2.4 we review respectively Equicontinuity, Attractors and Languages classifications.

3.1 Definition

We consider only the class of one-dimensional Cellular Automata defined as endomorphisms of full shits.

Definition 3.1. *Let A be a finite alphabet. A couple $(A^{\mathbb{Z}}, F)$ is a* cellular automaton *if there exists two positive integers $m \geq 0$ (*memory*) $a \geq 0$ (anticipation) and a* local rule $f : A^{m+a+1} \to A$ *such that*

$$\forall x \in A^{\mathbb{Z}}, \forall i \in \mathbb{Z}, F(x)_i = f(x_{i-m}, ..., x_{i+a}).$$

The value $r = \max\{m, a\}$ is called *radius* of the automaton.

According to Curtis-Hedlund-Lyndon theorem [26], $(A^{\mathbb{Z}}, F)$ is a cellular automaton if and only if F is a continuous and σ-commuting function.

Definition 3.2. *A cellular automaton is* one-sided *if the local rule has memory $m = 0$ (equivalently if it has anticipation $a = 0$). A one-sided cellular automaton is generally denoted as $(A^{\mathbb{N}}, F)$.*

Example 3.3. The shift map $\sigma : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ is a cellular automaton. It is also one-sided because the local rule $f : A^2 \to A$ is defined as $f(a, b) = b$.

3.2 Cellular Automata as Dynamical Systems

Cellular Automata can be considered as symbolic discrete dynamical system. As dynamical system they have a very rich and diverse behavior.

In this section we review known results about the dynamical properties of Cellular Automata. We don't provide a complete introduction on the subject but we focus our attention only on the aspects that will be relevant for our investigation. We are essentially interested in those properties of Cellular Automata which can provide measures of complexity of the dynamics and on decidability questions related to such properties.

3.2.1 Limit Set and Topological Entropy

A measure of the complexity of a cellular automaton is given by its *limit set*. It was introduced by Wolfram for studying the long-term behavior of Cellular Automata [59] and consists of all the configurations that can occur after arbitrarily many iterations.

Definition 3.4. *The* limit set *of* $(A^{\mathbb{Z}}, F)$ *is defined as* $\Lambda(F) = \cap_{i\in\mathbb{N}} A^{\mathbb{Z}}$.

Definition 3.5. *A cellular automaton* $(A^{\mathbb{Z}}, F)$ *is called* stable *if there exist* $n \in \mathbb{N}$ *such that* $\Lambda(F) = F^n(A^{\mathbb{Z}})$. *It is called* unstable *otherwise.*

Example 3.6. (Stable limit set) The cellular automaton $(A^{\mathbb{Z}}, F)$ defined by $\forall x \in A^{\mathbb{Z}}, F(x) = 0^\infty$ where $0 \in A$ is stable since $F(A^{\mathbb{Z}}) = F^2(A^{\mathbb{Z}}) = \{0^\infty\}$.

The great attention the limit set of Cellular Automat has received in literature was stimulated by the question 13 posed by Wolfram in [58]: what limit sets can cellular automata produce?

This question was first addressed in [31, 32] where the author studies the complexity of limit sets by using formal language theory showing that the language complexity of a limit set can be arbitrary high. It's not difficult to see that limit set of stable Cellular Automata are always mixing sofic shifts (then of regular language complexity). In [43], Maass attempts to characterize the class of sofic shifts which can be limit sets of Cellular Automata. A shift of finite type cannot be the limit set of an unstable cellular automaton [33] and also of a larger class of sofic systems [43]. There are also non-sofic systems which cannot be at all limit sets of Cellular Automata [44]. Kari [35] shows that all non trivial properties of limit sets of Cellular Automata are not decidable. In general, it is not possible to decide even when the limit set of a cellular automaton consists only of a single configuration. Such Cellular Automata are called *nilpotent*.

Definition 3.7. *A cellular automaton* $(A^{\mathbb{Z}}, F)$ *is* nilpotent *if* $|\Lambda(F)| = 1$.

Example 3.8. The stable cellular automaton $(A^{\mathbb{Z}}, F)$ of Example 3.6 is nilpotent since $\Lambda(F) = F(A^{\mathbb{Z}}) = \{0^\infty\}$.

Nilpotent Cellular Automata are stable. The following result, due to Kari, is used extensively to prove that other topological properties are undecidable.

Theorem 3.9. *[34] It is undecidable whether a cellular automaton is nilpotent.*

The topological entropy of a cellular automaton $(A^{\mathbb{Z}}, F)$ is also a measure of the complexity of the mapping F. It was introduced for general dynamical systems in [1]. For Cellular Automata it has a simpler definition and it can be computed from the entropy of *column factors*.

Definition 3.10. *The* column factor *of width* $k > 0$ *of* $(A^{\mathbb{Z}}, F)$ *is the set of one-sided infinite sequences* $\Sigma_k = \{y \in (A^K)^{\mathbb{N}} \mid \exists x \in A^{\mathbb{Z}}, \forall n \geq 0, F^n(x)_{[0,k)} = y_n\}$.

If the automaton is one sided, i.e. the local rule has memory zero, the column factor Σ_{1+a} is denoted simply with Σ and is called *canonical factor*. Since the column factors are symbolic factors of the automaton, the topological entropy of every column factor is a lower bound to the topological entropy of the automaton.

Definition 3.11. *The* topological entropy *of* $(A^{\mathbb{Z}}, F)$ *is defined as*

$$H(F) = \lim_{k \to \infty} H(\Sigma_k).$$

Example 3.12. The nilpotent cellular automaton of Example 3.8 has $H(F) = 0$ since for all $k > 0$, $\Sigma_k = \{x \in (A^k)^{\mathbb{N}} \mid x_0 \in A^k \text{ and } x_i = 0^k, i > 0\}$ and $H(\Sigma_k) = 0$.

More generally the topological entropy of nilpotent Cellular Automata is always zero. In general, it is not possible to decide if a cellular automaton has strictly positive topological entropy. This leads to the following result.

Theorem 3.13. *[27] The topological entropy of Cellular Automata is not computable.*

For one-sided Cellular Automata the topological entropy has a nicer characterization since it coincides with the entropy of the canonical factor.

Theorem 3.14. *[6] Let* $(A^{\mathbb{N}}, F)$ *be a one-sided CA. Then* $H(F) = H(\Sigma)$.

It is an open question whether Theorem 3.14 can be extended to general Cellular Automata.

Question 3.2.1. [19] For every cellular automaton $(A^{\mathbb{Z}}, F)$ is there a (computable) number $k > 0$ such that $H(F) = H(\Sigma_k)$?

3.2.2 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.

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

Definition 3.15. *A point $x \in A^{\mathbb{Z}}$ is an* equicontinuity point *for $(A^{\mathbb{Z}}, F)$ if*

$$\forall \epsilon > 0, \exists \delta > 0, \forall y \in \mathcal{B}_\epsilon(x), \forall n \geq 0, d(F^n(x), F^n(y)) < \epsilon$$

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

Definition 3.16. *(Equicontinuity) $(A^{\mathbb{Z}}, F)$ is* equicontinuous *if*

$$\forall x \in A^{\mathbb{Z}}, \forall \epsilon > 0, \exists \delta > 0, \forall y \in \mathcal{B}_\epsilon(x), \forall n \geq 0, d(F^n(x), F^n(y)) < \epsilon$$

The following theorem characterizes equicontinuous Cellular Automata.

Theorem 3.17. *[37] For a cellular automaton $(A^{\mathbb{Z}}, F)$ the following conditions are equivalent:*

1. *$(A^{\mathbb{Z}}, F)$ is equicontinuous;*

2. *there exist $m \geq 0$, $n > 0$ such that for every $x \in A^{\mathbb{Z}}$, and for every $i \geq m$ we have $F^{i+n}(x) = F^i(x)$.*

From Theorem 3.17 follows that every equicontinuous cellular automaton is also stable.

Example 3.18. The identity cellular automaton $(A^{\mathbb{Z}}, Id)$ defined by $Id(x) = x$ is equicontinuous since for every $x \in A^{\mathbb{Z}}$ and for every $k > 0, F([x_{[-k,k]}]) = [x_{[-k,k]}]$.

Example 3.19. The nilpotent cellular automaton of Example 3.8 is equicontinuous.

A cellular automaton is *almost equicontinuous* if it has at least one equicontinuity point.

Definition 3.20. *(Almost Equicontinuity)* $(A^{\mathbb{Z}}, F)$ *is* almost equicontinuous *if*

$$\exists x \in A^{\mathbb{Z}}, \forall \epsilon > 0, \exists \delta > 0, \forall y \in \mathcal{B}_{\epsilon}(x), \forall n \geq 0, d(F^n(x), F^n(y)) < \epsilon$$

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

Definition 3.21. *A word* $u \in A^+$ *with* $|u| \geq k > 0$ *is* k-blocking *for* $(A^{\mathbb{Z}}, F)$ *if there exists* $p \in [0, |u| - s]$ *such that*

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

Theorem 3.22. *[36] For a cellular automaton* $(A^{\mathbb{Z}}, F)$ *the following conditions are equivalent:*

1. $(A^{\mathbb{Z}}, F)$ *is almost equicontinuous;*
2. $(A^{\mathbb{Z}}, F)$ *has a blocking word.*

Example 3.23. Let $(A^{\mathbb{Z}}, F)$ with $A = \{0, 1\}$ be the *product* cellular automaton defined by $F(x)_i = x_{i-1}x_ix_{i+1}$ is almost equicontinuous because the word 0 is 1-blocking.

A cellular automaton is *sensitive* when for every point x, in every neighborhood of x there exists a point y whose orbit separate from the orbit of x.

Definition 3.24. *(Sensitivity)* $(A^{\mathbb{Z}}, F)$ *is* sensitive *if*

$$\exists \epsilon > 0, \forall x \in A^{\mathbb{Z}}, \forall \delta > 0, \exists y \in \mathcal{B}_{\epsilon}(x), \exists n \geq 0, d(F^n(x), F^n(y)) \geq \epsilon.$$

While this does not hold for general dynamical systems, for Cellular Automata sensitivity implies not almost equicontinuity.

Theorem 3.25. *[36] For a cellular automaton* $(A^{\mathbb{Z}}, F)$ *the following conditions are equivalent:*

1. $(A^{\mathbb{Z}}, F)$ *is sensitive;*
2. $(A^{\mathbb{Z}}, F)$ *does not have a blocking word.*

Example 3.26. The *shift* cellular automaton $(A^{\mathbb{Z}}, \sigma)$ is sensitive.

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.

Definition 3.27. *(Positively expansiveness)* $(A^{\mathbb{Z}}, F)$ *is* positively expansive *if*

$$\exists \epsilon > 0, \forall x, \forall y \neq x, \exists n \geq 0, d(F^n(x), F^n(y)) \geq \epsilon.$$

There is an interesting class of positively expansive Cellular Automata.

Definition 3.28. *Let* $(A^{\mathbb{Z}}, F)$ *be defined by the local rule* $f : A^{m+a+1} \to A$.

- $(A^{\mathbb{Z}}, F)$ *is* left permutive *if* $\forall u \in A^{m+a-1}, \forall b \in A, \exists! a \in A$ *s.t.* $f(au) = b$
- $(A^{\mathbb{Z}}, F)$ *is* right permutive *if* $\forall u \in A^{m+a-1}, \forall b \in A, \exists! a \in A$ *s.t.* $f(ua) = b$
- $(A^{\mathbb{Z}}, F)$ *is* bipermutive *if it is left and right permutive.*

The following proposition shows that permutive Cellular Automata are positively expansive.

Proposition 3.29. *Let* $(A^{\mathbb{Z}}, F)$ *be a cellular automaton with memory* m *and anticipation* a, $m < 0 < a$.

- *If* $(A^{\mathbb{Z}}, F)$ *is bipermutive then* $(A^{\mathbb{Z}}, F)$ *is conjugated to the* $|A|^{m+a}$*-full shift* (Σ_{m+a}, σ).

- *If* $(A^{\mathbb{Z}}, F)$ *is right permutive then* $(A^{\mathbb{N}}, F)$ *is conjugated to the* $|A|^{a}$*-full shift* (Σ_{a}, σ) *(the case left permutive is symmetric).*

Proof. Let x be a sequence of the one-sided $|A|^{m+a}$-full shift. Since $(A^{\mathbb{Z}}, F)$ is bipermutive there exists exactly one sequence $y \in A^{\mathbb{Z}}$ s.t. $\forall i \in \mathbb{N}, F^i(y)_{[0,m+a-1]} = x_i$.

Equivalently, let x be a sequence of the one-sided $|A|^{a}$-full shift, since $(A^{\mathbb{Z}}, F)$ is right permutive, there exists exactly one $y \in A^{\mathbb{N}}$ s.t. $\forall i \in \mathbb{N}, F^i(y)_{[0,a-1]} = x_i$. □

Example 3.30. Let $(A^{\mathbb{Z}}, F)$ be defined by $F(x)_i = [(x_{i-1} + x_{i+1}) \mod |A|]$ Then $(A^{\mathbb{Z}}, F)$ is bipermutive with $-m = a = 1$ and Σ_2 is the 4-full shift.

There exists also positively expansive cellular automata which are not permutive. The characterization of Proposition 3.29 holds for arbitrary positively expansive Cellular Automata.

Theorem 3.31. *[37, 47] Let* $(A^{\mathbb{Z}}, F)$ *be a positively expansive cellular automaton of radius* r*. Then* $(A^{\mathbb{Z}}, F)$ *is conjugated to the shift of finite type* (Σ_{2r+1}, σ).

Since every positively expansive Cellular Automata is conjugated to Σ_{2r+1} it is easy to see that there exists a σ-commuting conjugacy $\varphi : \Sigma_{2r+1} \to \Sigma_{2r+1}$ such that the two dynamical systems $(A^{\mathbb{Z}}, \sigma)$ and (Σ_{2r+1}, φ) are conjugated.

The following classification of Cellular Automata is Kůrka's modification [36] of Gilman's Equicontinuity classification [24]. Gilman's classification is based on measure-theoretic concepts, while Kůrka's one uses only topological concepts.

Corollary 3.32. *[36] Every* $(A^{\mathbb{Z}}, F)$ *falls exactly in one of the following classes:*

E1 $(A^{\mathbb{Z}}, F)$ *is equicontinuous;*

E2 $(A^{\mathbb{Z}}, F)$ *is almost equicontinuous but not equicontinuous;*

E3 $(A^{\mathbb{Z}}, F)$ *is sensitive but not positively expansive;*

E4 $(A^{\mathbb{Z}}, F)$ *is positively expansive.*

Since positively expansive Cellular Automata do not exist in any dimension greather than 1 (see [53]), Equicontinuity classification can be formulated only for one-dimensional Cellular Automata.

It is easy to see that equicontinuity is not a decidable property. Assume it is. Then, since equicontinuous Cellular Automata are stable it would be possible to compute the limit set and then the nilpotency wuold be decidable contradicting Theorem 3.9. More generally, it undecidable if a Cellular Automaton is almost equicontinuous which implies that sensitivity is also undecidable.

Theorem 3.33. *[15] It is undecidable if a cellular automaton has a blocking word.*

It is actually unknown if positively expansiveness is a decidable property.

Question 3.2.2. Is positively expansiveness a decidable property?

3.2.3 Attractors classification

In dynamical systems, an *attractor* is a set toward which the system evolves after a long enough time. For the set to be an attractor, trajectories that get close enough to the attractor must remain close even if slightly perturbed. To define mathematically the concept of attractor of $(A^{\mathbb{Z}}, F)$ we need to define the ω-limit of a set.

Definition 3.34. *The* ω-limit *of a set* $U \subseteq A^{\mathbb{Z}}$ *is* $\omega(U) = \cap_{n>0}\overline{\cup_{m>n}F^n(U)}$.

Definition 3.35. *A nonempty set* $Z \subseteq A^{\mathbb{Z}}$ *is an* attractor *if there exists an* F*-invariant clopen set* $U \subseteq A^{\mathbb{Z}}$ *such that* $\omega(U) = Z$*. A nonempty set is a* quasi-attractor *if it is the countable intersection of attractors. An attractor is* minimal *if it doesn't contain any proper subset which is also an attractor.*

Every $(A^{\mathbb{Z}}, F)$ has at least the maximal attractor $\Lambda(F) = \omega(A^{\mathbb{Z}})$.

Definition 3.36. *The* basin of attraction *of an attractor* Z *is defined as the set*

$$\mathcal{B}(Z) = \{x \in A^{\mathbb{Z}} \mid \omega(x) \subseteq Z\}.$$

The basin of attraction is always an open F-invariant set.

This following classification is Kůrka's refinement of Hurley's Attractor classification for Cellular Automata [30].

Corollary 3.37. *[36] Every* $(A^{\mathbb{Z}}, F)$ *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* $\omega(A^{\mathbb{Z}})$.

A4 *There exists a unique minimal attractor* $\omega(A^{\mathbb{Z}}) \neq A^{\mathbb{Z}}$.

A5 *There exists a unique minimal attractor* $\omega(A^{\mathbb{Z}}) = A^{\mathbb{Z}}$.

We list some examples to show that Attractor classes are not empty.

Example 3.38. The identity cellular automaton of Example 3.18 has two disjoint attractors $\omega([0]_0)$ and $\omega([1]_0)$ then it belongs to class **A1**.

Example 3.39. The *Hurley* cellular automaton $(\{0,1\}^{\mathbb{Z}}, F)$ defined by $F(x)_i = x_i x_{i+1}$ has unique minimal quasi-attractor $\{0^\infty\} = \cap_k \omega([0^k])$ (see [30] or [37]) and it belongs to class **A2**.

Example 3.40. The cellular automaton of Example 3.23 has just two attractors $\omega(A^{\mathbb{Z}}) = \{x \in A^{\mathbb{Z}} \mid 10^+1 \not\sqsubseteq x\}$ and $\{0^\infty\} \neq \omega(A^{\mathbb{Z}})$. Obviously $\{0^\infty\}$ is a minimal attractor.

Every nilpotent cellular automaton is in class **A4** while every positively expansive one is in class **A5** (see [37]).

We don't know if it is decidable the membership in some of the Attractor classes.

Question 3.2.3. Is the membership in Attractor classes decidable?

An interesting class of attractors for Cellular Automata is the class of subshift attractors.

Definition 3.41. *A* subshift attractor *is a σ-invariant attractor.*

Subshift attractors have been considered in [39] and [22]. They are generated by clopen invariant sets which are also *spreading*.

Definition 3.42. *A clopen F-invariant set $U \subseteq A^{\mathbb{Z}}$ is* spreading *if $F^k(U) \subseteq \sigma^{-1}(U) \cap U \cap \sigma(U)$ for some $k > 0$.*

The following proposition characterizes clopen sets whose omega limits are subshift attractors.

Proposition 3.43. *[22] Let $(A^{\mathbb{Z}}, F)$ be a cellular automaton and $U \subseteq A^{\mathbb{Z}}$ a clopen F-invariant set. Then $\omega(U)$ is a subshift attractor if and only if U is spreading.*

Every cellular automaton $(A^{\mathbb{Z}}, F)$ has at least one subshift attractor $\omega(A^{\mathbb{Z}})$ but it can have also an infinite number of subshift attractors [22]. For instance, Kůrka [39] shows that, for surjective cellular automata, the full space is the unique subshift attractor. In general a cellular automaton can have an infinite number of attractors and just one subshift attractor.

Example 3.44. The Hurley cellular automaton of Example 3.39 has unique minimal quasi-attractor 0^∞ and unique subshift attractor $\omega(A^{\mathbb{Z}}) = \{x \in A^{\mathbb{Z}} \mid 10^+1 \not\sqsubseteq x\}$ (see [22]).

The cellular automaton of example 3.39 is unstable and it has just one subshift attractor while the one in Example 3.40 has two distinct subshift attractors and it is also unstable. The cellular automaton with an infinite number of subshift attractors of Example 6 in [22] is also unstable. We are not aware of the existence of stable Cellular Automata with an infinite number of subshift attractors or simply with two distinct subshift attractors.

Question 3.2.4. Is there a stable cellular automaton with an infinite number of subshift attractors?

3.2.4 Languages classification

The complexity of the languages of the column factors is a measure of the complexity of Cellular Automata. This measure was introduced by Kůrka for general dynamical systems ([38]). Given some column factor Σ_k of a cellular automaton, the language $\mathcal{L}(\Sigma_k)$ is always context sensitive, since it is always possible to decide in a bounded amount of time if a block $w \in (A^k)^*$ is also in $\mathcal{L}(\Sigma_k)$ (see [25]). There are Cellular Automata whose column factors languages are strictly context sensitive.

Example 3.45. [25] For the cellular automaton $(\{0,1\}, F)$ where $F(x)_i = x_{i+1}x_{i+2}$ the language of the column factor Σ_1 is context sensitive since $1^n 0^m 1 \in \mathcal{L}(\Sigma_1)$ if and only if $m > n$.

Other classes of complexities arise naturally.

Definition 3.46. *A cellular automaton* $(A^{\mathbb{Z}}, F)$ *is* bounded periodic *if* $\forall k > 0, \exists m > 0, \exists n > 0$ *such that* $\forall x \in \Sigma_k, \forall i \geq m, \sigma^i(x) = \sigma^{i+n}(x)$.

The class of bounded period Cellular Automata coincides with the class of equicontinuous Cellular Automata.

Theorem 3.47. *[37] A cellular automaton is bounded periodic iff is equicontinuous.*

A dynamical system is regular when all of its factor subshifts are sofic [38]. This definition simplifies for Cellular Automata to have sofic column factors.

Definition 3.48. *A cellular automaton is* regular *if* $\forall k > 0$, Σ_k *is a sofic shift.*

Example 3.49. The product cellular automaton of Example 3.23 is regular. Note that for all $k > 0$, for every $x \in \Sigma_k$, $\sigma^k(x) = (0^k)^{\mathbb{N}}$.

Regular Cellular Automata are in some sense approximable systems.

Definition 3.50. *An* ϵ*-chain of* $(A^{\mathbb{Z}}, F)$ *from* $x_0 \in A^{\mathbb{Z}}$ *to* $x_n \in A^{\mathbb{Z}}$ *is a sequence of configurations* $x_i \in A^{\mathbb{Z}}$ *such that* $d(f(x_i), x_{i+1}) < \epsilon$ *for* $0 \leq i \leq n$.

An ϵ-chain is an approximation of an orbit. While such approximation works in general for a short number of steps, there are dynamical systems whose orbits can be approximated for a large number of steps.

Definition 3.51. *A point* $x \in A^{\mathbb{Z}}$ *ϵ-shadows in* $(A^{\mathbb{Z}}, F)$ *a sequence* $x_0, ..., x_n \in A^{\mathbb{Z}}$ *if* $d(F^i(x), x_i) < \epsilon$ *for* $0 \leq i \leq n$.

Definition 3.52. *A cellular automaton* $(A^{\mathbb{Z}}, F)$ *has the* shadowing property *if for every* $\epsilon > 0$ *there exists a* $\delta > 0$ *such that every* ϵ*-chain is* δ*-shadowed by some point.*

The orbits of a dynamical system with the shadowing property are approximable.

Proposition 3.53. *[37] Every cellular automaton with the shadowing property is regular.*

The converse of Proposition 3.53 is in general not true (see Example 5.78 in [36]).

The following classification is Kůrka's Language classification of Celular Automata according to the language complexity of column factors.

Corollary 3.54. *[36] Every* $(A^{\mathbb{Z}}, F)$ *falls exactly in one of the following classes:*

L1 $(A^{\mathbb{Z}}, F)$ *is bounded periodic.*

L2 $(A^{\mathbb{Z}}, F)$ *is regular not bounded periodic.*

L3 $(A^{\mathbb{Z}}, F)$ *is not regular.*

Since bounded periodic Cellular Automata coincide with equicontinuous Cellular Automata, it follows that the membership in **L1** is undecidable. In Section 4.3 we will show that regularity is also an undecidable property which implies that the membership in all Languages classes is undecidable.

The intersections classes between the tree classifications are shown in figures 3.1, 3.2 and 3.3.

	A1	A2	A3	A4	A5
E1	X			X	
E2	X	X	X	X	X
E3	X	X	X	X	X
E4					X

Figure 3.1: Equicontinuity and Attractors classifications.

	A1	A2	A3	A4	A5
L1	X			X	
L2	X	X	X	X	X
L3	X	X	X	X	X

Figure 3.2: Languages and Attractors classifications.

	E1	E2	E3	E4
L1	X			
L2		X	X	X
L3		X	X	

Figure 3.3: Languages and Equicontinuity classifications.

Chapter 4

Regular Cellular Automata

In this chapter we investigate *regular* Cellular Automata (see Definition 3.48).

We show that regularity is an undecidable property. Moreover, we show that if we know that a cellular automaton is regular then we can decide if it is nilpotent or equicontinuous or positively expansive and, if the automaton is also one-sided, we can compute its topological entropy.

In Section 4.1 we show some examples of regular Cellular Automata. In particular we show that additive Cellular Automata are regular. In Section 4.2 we investigate a subclass of regular Cellular Automata: the class of one-sided Cellular Automata whose canonical factors are shifts of finite type. We show that in general it is not possible to decide if the canonical factor of a one-sided cellular automaton is a shift of finite type (results of this section are collected in [17]). From this result doesn't follow the undecidability of the regularity property which is investigated in Section 4.3 (results of this section are collected in [18]).

4.1 Examples of regular Cellular Automata

In Section 3.2.4 we saw that the class of regular Cellular Automata is large. In particular, equicontinuous and positively expansive Cellular Automata are regular.

Here we investigate the regularity for the class of *additive* Cellular Automata.

Definition 4.1. *Let denote with* $X_n = \{1, 2, ..., n\}^{\mathbb{Z}}$ *the set of biinfinite sequences on alphabet* $\{1, ..., n\}$*,* $n > 0$*.*

Definition 4.2. *An* additive *cellular automaton is a an automaton* (X_n, F) *such that the local rule* $f : \{1, ..., n\}^{2r+1} \to \{1, ..., n\}$ *is of the form*

$$f(x_{-r}, ..., x_r) = [\Sigma_{i=-r}^{r} a_i x_i \mod n] \text{ for } a_{-r}, .., a_r \in \mathbb{N}.$$

It is easy to check if a cellular automaton is additive. Moreover, almost all properties which are in general undecidable are decidable for additive Cellular Automata (see [46]).We show that additive Cellular Automata are regular.

The following theorems provide respectively an useful property of additive Cellular Automata and a strong characterization of additive sensitive Cellular Automata

Definition 4.3. *Let denote* $F_n = F \mod n$*.*

Theorem 4.4. *[21] Let* (X_{pq}, F) *be an additive cellular automaton with* $gcd(p, q) = 1$*. Then* (X_{pq}, F) *is conjugated to the additive cellular automaton* $(X_p \times X_q, F_p \times F_q)$*.*

Theorem 4.5. *[46] Let* (X_n, F) *be an additive cellular automaton with local rule* $f(x_{-r}, ...x_r) = [\Sigma_{i=-r}^{r} a_i x_i \mod n]$*. Then* (X_n, F) *is sensitive if and only if there exists a prime p such that*

$$p \mid n \text{ and } p \not| gcd(a_{-r}, ..., a_{-1}, a_1, ..., a_r).$$

The following lemma shows that for every additive sensitive CA (X_{p^k}, F) with p prime, there exists a $h > 0$ such that the local rule of the additive CA (X_{p^k}, F^h) is permutive in its rightmost and leftmost variable.

Lemma 4.6. *[21] Let* (X_{p^k}, F) *be an additive CA with p prime and local rule* $f(x_{-r}, ...x_r) = [(a_{-r}x_{-r} + ... + a_r x_r) \mod p^k]$*. Suppose* (X_{p^k}, F) *is sensitive and let* a_i *such that* $gcd(a_i, p) = 1$*. Moreover let*

$$L = \min\{j : gcd(a_j, p) = 1\}, R = \max\{j : gcd(a_j, p) = 1\}.$$

Then there exists $h \geq 1$ such that the local rule f^h associated to F^h has the form

$$f^h(x_{-hr},...,x_{hr}) = [\Sigma_{i=-hL}^{hR} b_i x_i \mod p^k] \text{ with } gcd(b_{hL},p) = gcd(b_{hR},p) = 1.$$

By Proposition 3.29, a bi-permutive cellular automaton is regular. Then, by Lemma 4.6, we can conclude that any sensitive additive cellular automaton (X_{p^k}, F) with p prime is regular.

Theorem 4.7. *Additive Cellular Automata are regular.*

Proof. By Theorem 4.4, any additive cellular automaton can be decomposed in the product of a finite number of additive Cellular Automata

$$(X_{p_1^{n_1}} \times ... \times X_{p_k^{n_k}}, F_{p_1^{n_1}} \times ... \times F_{p_k^{n_k}})$$

with $p_1 \neq ... \neq p_k$ prime numbers. An additive cellular automaton can be either equicontinuous or sensitive [46]. In both cases $(X_{p_i^{n_i}}, F_{p_i^{n_i}})$ is regular for all $1 \leq i \leq k$. Then their product must be regular. □

To conclude, it could be interesting to know what is the class of one-sided sofic shifts which rise from column factors of Cellular Automata. So we raise the following question.

Question 4.1.1. Is every one-sided sofic shift conjugated to some column factor of some cellular automaton?

We can provide only a partial answer.

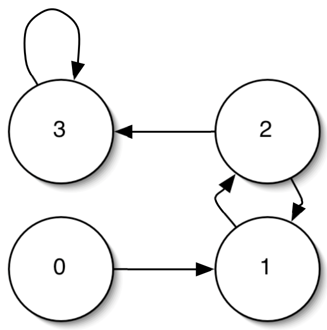
Proposition 4.8. *Every one-sided shift of finite type is conjugated to the canonical factor of some one-sided cellular automaton.*

Proof. Let X be a one-sided SFT of order $K > 0$ and let $Y = X^K$ be the higher K-block presentation of X. We describe a procedure to define a CA $(A^{\mathbb{Z}}, F)$ with radius $r = 1$ such that $\Sigma = \Sigma_1(A^{\mathbb{Z}}, F) = Y$. Let $A = \mathcal{L}_1(Y)$ and let $f : A^2 \to A$ be the local rule of $(A^{\mathbb{Z}}, F)$ defined by

$$f(a,b) = \begin{cases} b & if\ ab \in \mathcal{L}_2(Y) \\ c & otherwise,\ for\ some\ ac \in \mathcal{L}_2(Y) \end{cases}$$

By definition, $\mathcal{L}_2(\Sigma) = \mathcal{L}_2(Y)$ then $\Sigma \subseteq Y$. Conversely, let $x \in Y$ and let $y \in A^{\mathbb{Z}}$ such that $y_{[0,\infty)} = x$. By definition, $\forall i \geq 0, F^i(x)_0 = x_i$ then $x \in \Sigma$ and $Y \subseteq \Sigma$. □

Example 4.9. Let X be the a SFT on alphabet $A = \{0, 1, 2, 3\}$ defined by the following list of allowed blocks $\{01, 12, 21, 23, 33\}$. In figure 4.1 it is possible to see the graph representation of X and the block map defined by the procedure described in Proposition 4.8. Note that blocks 20 and 22 can be mapped indifferently to either 1 or 3 without changing the symbolic factor of width 1.



00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33
1	1	1	1	2	2	2	2	1|3	1	1|3	3	3	3	3	3

Figure 4.1: Cellular automaton with shift of finite type canonical factor.

4.2 One-sided Cellular Automata with SFT canonical factors

In this section we provide a characterization for one-sided Cellular Automata whose canonical factors are shifts of finite type (Lemma 4.20). From such characterization we can easily derive the property that given a CA $(A^{\mathbb{N}}, F)$ and $k > 0$, it is possible to decide if Σ is a SFT of order k (Theorem 4.21).

The immediate consequences of Theorem 4.21 are that the topological entropy is computable for any one-sided CA $(A^{\mathbb{N}}, F)$ whose canonical factor Σ is a shift of finite type (Proposition 4.22) and that it is in general undecidable if Σ is a SFT (Proposition 4.23).

Definition 4.10. *The* SFT k-approximation *(or simply* k-approximation*) of a one-sided subshift X is the one-sided SFT $X(k)$ such that $x \in X(k)$ if and only if $x_{[i,i+k-1]} \in \mathcal{L}_k(X)$, $\forall i \in \mathbb{N}$.*

If Σ is a SFT of order $K > 0$, it happens that for any $k \geq K$, $\Sigma(k) = \Sigma$. However, in general, if $\Sigma(k) = \Sigma(k+1) = ... = \Sigma(k+i)$ for some $k, i > 0$, we cannot conclude that Σ has order k. The following example shows that, in general, if a finite number of increasing and successive SFT approximations of Σ coincide with the same SFT X, we cannot conclude that $\Sigma = X$.

Example 4.11. For any $n > 0$, let $X_n = \{a, b, c_1, ..., c_n\}$ be an alphabet and let $(X_n^{\mathbb{N}}, F_n)$ be the one-sided CA whose local rule $f_n : X_n^2 \to X_n$ is represented in figure 4.2. For any $n > 0$, the sequence of SFT approximations of Σ starting from order 2 up to order n coincide with the SFT of order 2 defined by the set of allowed blocks $\{aa, ab, ba, c_1c_2, c_2c_3, ..., c_{n-2}c_{n-1}, c_{n-1}c_n, c_nb\}$. This shift of finite type is represented as edge shift in figure 4.2. It is easy to verify that $a...a = a^{n+1} \notin \mathcal{L}(\Sigma)$ which implies that Σ is not an SFT of order n.

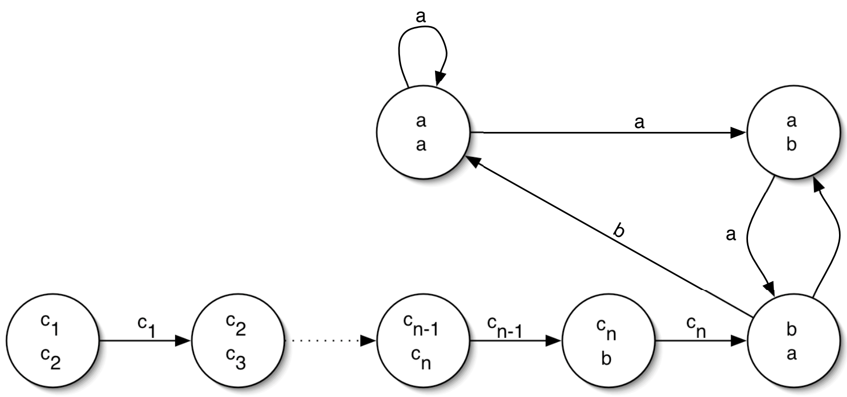
We show that there's an effective algorithmic way to decide if $\Sigma(k) = \Sigma$ for some $k > 0$. In order to see this, we need to introduce some preliminary properties.

Definition 4.12. *Let $(A^{\mathbb{N}}, F)$ be a CA and let $b \in \mathcal{L}_t(\Sigma_k)$, $t, k > 0$. We can see b either as a sequence of blocks $b'_1...b'_t$ where $b'_1, .., b'_t \in A^k$ or as a sequence of blocks $b''_1...b''_k$ where $b''_1, .., b''_k \in A^t$. Let*

$$\lambda_{t,k}^t : \mathcal{L}_t(\Sigma_k) \to (A^k)^t \text{ and } \lambda_{t,k}^k : \mathcal{L}_t(\Sigma_k) \to (A^t)^k$$

denote the 1*-to-*1 *onto mappings such that*

$$\lambda_{t,k}^t(b) = b'_1...b'_t \in A^k \text{ and } \lambda_{t,k}^k(b) = b''_1...b''_k \in A^t.$$



aa	ab	ac_1	...	ac_n	$b \cdot X_n$	$c_1 \cdot X_n$	$c_2 \cdot X_n$	...	$c_{n-2} \cdot X_n$	$c_{n-1} \cdot X_n$	$c_n \cdot X_n$
b	a	a	a	a	a	c_2	c_3	...	c_{n-1}	c_n	b

Figure 4.2: One-sided CA whose canonical factor is not a SFT of order n

Definition 4.13. *Let $(A^{\mathbb{N}}, F)$ be a one-sided cellular automaton.*
1. *Let $t > 0$, $k > 1$ and let $x, y \in \mathcal{L}_t(\Sigma_k)$. Suppose that $\lambda^k_{t,k}(x) = x_1 ... x_k$ and $\lambda^k_{t,k}(y) = y_1 ... y_k$ where $x_2 = y_1, ..., x_k = y_{k-1}$. Then we say that x, y are* horizontally compatible blocks *and we denote their* horizontal overlapping concatenation *with $x \odot y$ where $\lambda^k_{t,k}(x \odot y) = x_1 ... x_k y_k$.*
2. *Let $t > 1$, $k > 0$ and let $x, y \in \mathcal{L}_t(\Sigma_k)$. Suppose that $\lambda^t_{t,k}(x) = x_1 ... x_t$ and $\lambda^t_{t,k}(y) = y_1 ... y_t$ where $x_2 = y_1, ..., x_t = y_{t-1}$. Then we say that x, y are* vertically compatible blocks *and we denote their* vertical overlapping concatenation *with $x \oplus y$ where $\lambda^t_{t,k}(x \oplus y) = x_1 ... x_t y_t$.*

Example 4.14. For the cellular automaton $(\{0,1\}^{\mathbb{Z}}, \sigma^2)$, $a = \begin{matrix} 00 \\ 01 \end{matrix} \in \mathcal{L}_2(\Sigma_2)$ and $b = \begin{matrix} 01 \\ 10 \end{matrix} \in \mathcal{L}_2(\Sigma_2)$ are both horizontally and vertically compatible blocks. For instance

$$a \odot b = \begin{matrix} 001 \\ 010 \end{matrix} \text{ and } a \odot b = \begin{matrix} 00 \\ 01 \\ 10 \end{matrix}.$$

In general, if $x_1, ..., x_s \in \mathcal{L}_t(\Sigma_k)$ is a sequence of blocks such that x_i, x_{i+1} are horizontally (resp. vertically) compatible for $1 \leq i < s$ we say that $x_1, ..., x_s$ are horizontally (resp. vertically) compatible and we denote with $x_1 \odot ... \odot x_s$ (resp. $x_1 \oplus ... \oplus x_s$) their horizontal (resp. vertical) overlapping concatenation.

Definition 4.15. *Let $(A^{\mathbb{N}}, F)$ be a CA with radius r. For $t, k > 0$, let*

$$\varrho_{t,k} : A^{k+r(t-1)} \to \mathcal{L}_t(\Sigma_k)$$

be the onto mapping defined in the following way: $\forall a \in A^{k+r(t-1)}$, $\varrho_{t,k}(a) = b \in \mathcal{L}_t(\Sigma_k)$ if and only if $\exists x \in A^{\mathbb{N}}$ such that $x_{[0,k+r(t-1))} = a$ and $F^i(x)_{[0,k)} = b_{i+1}$, $0 \leq i < t$ where $\lambda^t_{t,k}(b) = b_1...b_t$.

Remark 4.2.1. Let $(A^{\mathbb{N}}, F)$ be a CA with radius r and let $k, t > 0$. Note that, the block $b \in \mathcal{L}_t(\Sigma_k)$ is *completely determined* by the set of blocks $\varrho^{-1}_{t,k}(b)$. This means that if $b_1, b_2 \in \mathcal{L}_t(\Sigma_k)$ and $b_1 \neq b_2$ then $\varrho^{-1}_{t,k}(b_1) \cap \varrho^{-1}_{t,k}(b_2) = \emptyset$.

Moreover, the set of blocks $\mathcal{L}_t(\Sigma_{k+r})$ *completely determines* the set of blocks $\mathcal{L}_{t+1}(\Sigma_k)$. That is, $\varrho^{-1}_{t,k+r}(\mathcal{L}_t(\Sigma_{k+r})) = \varrho^{-1}_{t+1,k}(\mathcal{L}_{t+1}(\Sigma_k))$.

The following lemma shows a very useful property.

Lemma 4.16. *Let $(A^{\mathbb{N}}, F)$ be a CA with radius r. Let $x_1, .., x_k \in \mathcal{L}_t(\Sigma_{r+1})$ be horizontally compatible blocks, $t > 0, k > 1$. Then $x_1 \odot ... \odot x_k \in \mathcal{L}_t(\Sigma_{r+k})$.*

Proof. Let $\lambda^t_{t,r+1}(x_i) = x^i_1...x^i_t$, $1 \leq i \leq k$. Let $b \in \varrho^{-1}_{t,r+1}(x_k)$ and let $a \in A^{rt+k}$ be such that $a_{[i,i+r]} = x^i_1$, $1 \leq i \leq k$ and $a_{[k,rt+k]} = b$. Then, it is easy to check that $\varrho_{t,r+k}(a) = x_1 \odot ... \odot x_k$ which implies that $x_1 \odot ... \odot x_k \in \mathcal{L}_t(\Sigma_{r+k})$. □

Note that Lemma 4.16 doesn't work if we consider $x, y \in \mathcal{L}_t(\Sigma_k)$ where $k \leq r$. In this case, as the following example shows, even if x, y are two horizontally compatible blocks, we cannot assure that $x \odot y$ is a legal block of $\mathcal{L}_t(\Sigma_{k+1})$.

Example 4.17. For the cellular automaton $(\{0,1\}^{\mathbb{Z}}, \sigma^2)$ of Example 4.14, $a \odot b = \begin{matrix} 001 \\ 010 \end{matrix}$ is not a legal block of $\mathcal{L}_2(\Sigma_3)$.

A one-sided CA is regular if and only if $\mathcal{L}(\Sigma)$ is a regular language [7]. We provide an equivalent and useful characterization for the case in which Σ is supposed to be a SFT.

Proposition 4.18. *Let $(A^{\mathbb{N}}, F)$ be a CA with radius r. The following conditions are equivalent:*

1. $\exists t \geq r$, $\Sigma_t = \Sigma_t(K)$
2. $\Sigma = \Sigma(K)$
3. $\forall t \geq r$, $\Sigma_t = \Sigma_t(K)$

Proof. $(1 \Rightarrow 2)$ Let $t > 0$ and suppose Σ_t is a SFT of order K. Let $x, y \in \mathcal{L}_K(\Sigma)$ be two vertically compatible blocks. We have to show that $x \oplus y \in \mathcal{L}_{K+1}(\Sigma)$. Let $x' \in \mathcal{L}_K(\Sigma_t)$ such that $x' = x'_0 \odot ... \odot x'_{t-r}$ where $x'_0, .., x'_{t-r} \in \mathcal{L}_K(\Sigma)$ and $x'_{t-r} = x$. Equivalently, let $y' \in \mathcal{L}_K(\Sigma_t)$ such that $y' = y'_0 \odot ... \odot y'_{t-r}$ where $y'_0, .., y'_{t-r} \in \mathcal{L}_K(\Sigma)$ and $y'_{t-r} = y$. Moreover, let $\lambda^K_{K,t}(x') = a_1..a_K$ and $\lambda^K_{K,t}(y') = b_1..b_K$. Since $(A^{\mathbb{N}}, F)$ is (right) one-sided, we can choose y' such that $b_1 = a_2$. Then, by definition, x' and y' are vertically compatible blocks which implies that $x \oplus y \in \mathcal{L}_{K+1}(\Sigma)$.
$(2 \Rightarrow 3)$ Suppose Σ is a SFT of order $K > 0$. Let $t > 0$ and let $x, y \in \mathcal{L}_K(\Sigma_t)$ be two vertically compatible blocks. We have to show that $x \oplus y \in \mathcal{L}_{K+1}(\Sigma_t)$. Let $x_0, .., x_{t-r}, y_0, .., y_{t-r} \in \mathcal{L}_K(\Sigma)$ such that $x = x_0 \odot ... \odot x_{t-r}$ and $y = y_0 \odot ... \odot y_{t-r}$. By hypothesis, $z = x_{t-r} \oplus y_{t-r} \in \mathcal{L}_{K+1}(\Sigma)$. Let $\lambda^K_{K,t}(x) = a_1..a_K$, $b \in \varrho^{-1}_{K+1,r}(z)$ and let $c \in A^{t+rK}$ be such that $c_{[1,t]} = a_1$ and $c_{[t+1,t+rK]} = b_{[r+1,r+rK]}$. Then, it is easy to verify that $\varrho_{K+1,t}(c) = x \oplus y$ which implies that $x \oplus y \in \mathcal{L}_{K+1}(\Sigma_t)$.
$(3 \Rightarrow 1)$ Trivial. □

Definition 4.19. *Let A be a finite alphabet. Let $t \geq 1$ and let $[i, j] \subseteq [1, t]$ be an integer interval. Let*

$$\Phi_{[i,j]} : (A^t)^{\mathbb{N}} \to (A^{j-i+1})^{\mathbb{N}}$$

denote the projection map induced by the one-block factor map

$$\varphi_{[i,j]} : A^t \to A^{j-i+1}$$

defined by $\varphi_{[i,j]}(a_1...a_t) = a_i a_{i+1}...a_j, \forall a_1 a_2 ... a_t \in A^t$.

Remark 4.2.2. Let $(A^{\mathbb{N}}, F)$ be a CA. Since F is σ-commuting, $\forall k > 0$ and $1 \leq i \leq k+1$, the projection obtained by restricting $\Phi_{[i,i+k]}$ to Σ_{2k+1} is Σ_{k+1}. That is, $\forall k > 0$, $1 \leq i \leq k+1$, $\Phi_{[i,i+k]}(\Sigma_{2k+1}) = \Sigma_{k+1}$.

The following lemma shows a strong property Σ must have in order to be a shift of finite type of order K. For instance, the canonical factor Σ of a one-sided cellular automaton is a shift of finite type of order K if and only if the K-approximation of Σ_{2r+1} is invariant under projections.

Lemma 4.20. *Let* $(A^{\mathbb{N}}, F)$ *be a CA with radius* r. *Let* $K > 0$ *and let* $\Sigma_{2r+1}(K)$ *be the SFT* K*-approximation of* Σ_{2r+1}. *Moreover, let*

$$X_1 = \Phi_{[1,r+1]}(\Sigma_{2r+1}(K)), ..., X_{r+1} = \Phi_{[r+1,2r+1]}(\Sigma_{2r+1}(K))$$

be the projections of $\Sigma_{2r+1}(K)$ *obtained by restricting* $\Phi_{[i,i+r]}$ *to* $\Sigma_{2r+1}(K)$. *Then*

$$\Sigma = \Sigma(K) \text{ if and only if } X_1 = ... = X_{r+1}.$$

Proof. Suppose $\Sigma = \Sigma(K)$. Then, by Proposition 4.18, it follows that $\Sigma_{2r+1} = \Sigma_{2r+1}(K)$ and $\Sigma_{r+1} = \Sigma_{r+1}(K)$ which implies that $\Sigma_{r+1} = X_1 = ... = X_{r+1}$. Conversely, suppose $X_1 = ... = X_{r+1}$. By Proposition 4.18, it is sufficient to show that $\Sigma_{2r+1} = \Sigma_{2r+1}(K)$. Trivially $\Sigma_{2r+1} \subseteq \Sigma_{2r+1}(K)$, then we prove by induction on $t > 0$ that $\mathcal{L}_t(\Sigma_{2r+1}(K)) \subseteq \mathcal{L}_t(\Sigma_{2r+1})$.

1. (Base Case) By hypothesis, $\forall t \leq K$, $\mathcal{L}_t(\Sigma_{2r+1}(K)) \subseteq \mathcal{L}_t(\Sigma_{2r+1})$.

2. (Inductive Case) Let $t \geq K$ and suppose $\mathcal{L}_t(\Sigma_{2r+1}(K)) = \mathcal{L}_t(\Sigma_{2r+1})$. We have to show that $\mathcal{L}_{t+1}(\Sigma_{2r+1}(K)) \subseteq \mathcal{L}_{t+1}(\Sigma_{2r+1})$.

   First of all, observe that, since $\mathcal{L}_t(\Sigma_{2r+1}(K)) = \mathcal{L}_t(\Sigma_{2r+1})$, it follows that $\mathcal{L}_{t+1}(X_1) = ... = \mathcal{L}_{t+1}(X_{r+1}) = \mathcal{L}_{t+1}(\Sigma_{r+1})$. This is a consequence of the fact that $\varrho^{-1}_{t,2r+1}(\mathcal{L}_t(\Sigma_{2r+1})) = \varrho^{-1}_{t+1,r+1}(\mathcal{L}_{t+1}(\Sigma_{r+1}))$. Let $b_1 \odot ... \odot b_{r+1} \in \mathcal{L}_{t+1}(\Sigma_{2r+1}(K))$ where $b_1 \in \mathcal{L}_{t+1}(X_1), ..., b_{r+1} \in \mathcal{L}_{t+1}(X_{r+1})$. Then, $b_1, ..., b_{r+1} \in \mathcal{L}_{t+1}(\Sigma_{r+1})$ and, by Lemma 4.16, it follows that $b_1 \odot ... \odot b_{r+1} \in \mathcal{L}_{t+1}(\Sigma_{2r+1})$. □

Now we are ready to show that given a CA $(A^{\mathbb{N}}, F)$ and $k > 0$ it is possible to decide if $\Sigma = \Sigma(k)$. Note that this implies that the set of Cellular Automata whose canonical factor is a SFT is recursively enumerable. The proof relies essentially on the fact that the condition imposed by Lemma 4.20 is algorithmically checkable.

Theorem 4.21. *Let consider a one-sided cellular automaton $(A^{\mathbb{N}}, F)$ and let $k > 0$. Then, it is decidable whether $\Sigma = \Sigma(k)$.*

Proof. Let r be the radius of the CA. It is easy to see that it is always possible to compute a FSA recognizing $\mathcal{L}(X)$ where $X = \Sigma_{2r+1}(k)$. Moreover, given a FSA recognizing $\mathcal{L}(X)$, it is easy to build $r+1$ FSAs recognizing $\mathcal{L}(\Phi_{[1,r+1]}(X)), ..., \mathcal{L}(\Phi_{[r+1,2r+1]}(X))$.

Then, since a subshift is completely determined by its language, by Lemma 4.20, the decidability of whether Σ equals $\Sigma(k)$ comes from the decidability of the equivalence between finite state automata. □

From Theorem 4.21 follows that there exists an algorithm which, given a CA, computes a graph representation of Σ provided Σ is a SFT, otherwise works forever. Thus, as immediate consequence of Theorem 4.21, the topological entropy is computable for the class of one-sided CAs whose canonical factors are SFTs. In contrast, it comes also out that it is undecidable if a CA is in this class.

Proposition 4.22. *Let $(A^{\mathbb{N}}, F)$ be a regular CA and suppose Σ is a SFT. Then $H(F)$ is computable.*

Proof. Since the topological entropy of a sofic shift is computable, the proof follows from Proposition 3.14 and Theorem 4.21. □

A natural question is to ask if we can extend Proposition 4.22 to the whole class of one-sided regular Cellular Automata. In the next section we provide a positive answer to this question.

Proposition 4.23. *Let $(A^{\mathbb{N}}, F)$ be a CA. It is undecidable whether Σ is a SFT.*

Proof. The proof is a reduction from the nilpotency problem: if it is possible to decide if Σ is a SFT then it is possible to decide if $(A^{\mathbb{N}}, F)$ is nilpotent.

By definition, $(A^{\mathbb{N}}, F)$ is nilpotent if $\exists x \in A^{\mathbb{N}}, \sigma(x) = x$ and $\exists N > 0$ such that $\forall n \geq N, F^n(A^{\mathbb{N}}) = x$. If $(A^{\mathbb{N}}, F)$ is nilpotent then Σ is a SFT. In particular, $(A^{\mathbb{N}}, F)$ is nilpotent if and only if Σ is a one-sided subshift such that $\forall n \geq N, \sigma^n(\Sigma) = x_{[1,r]}$. Given a labeled graph representation of Σ, this last condition is algorithmically checkable.

Suppose it is decidable if Σ is a SFT. If it is not a SFT then $(A^{\mathbb{N}}, F)$ is not nilpotent otherwise, by Theorem 4.21, Σ is computable then it is possible to check if there exists $N > 0$ and $a \in A^r$ such that $\forall n \geq N, \sigma^n(\Sigma) = a$. □

4.3 Undecidability of regularity

In this section we study general regular Cellular Automata. In the previous section (Section 4.2) we saw that it is undecidable whether a one-sided cellular automaton has as a shift of finite type as a canonical factor. Here we show that regularity is an undecidable property (Corollary 4.39). Moreover, we show that if we know that a cellular automaton is regular then we can decide a lot of useful properties which are undecidable for general Cellular Automata. For instance, we show that if we restrict to regular Cellular Automata we can decide nilpotency, equicontinuity and positively expansiveness (Proposition 4.38) and, as we anticipated in the previous section, we show that the topological entropy is computable for the whole class of one-sided regular Cellular Automata (Proposition 4.37). The undecidability of regularity is a negative consequence of these decidability results.

Most of our effort here will be devoted to show that if $S \subseteq (A^{2r+1})^{\mathbb{N}}$ is a sofic shift and $(A^{\mathbb{Z}}, F)$ is a CA with radius r, it is possible to decide whether $S = \Sigma_{2r+1}$ (Theorem 4.35). This strong result has a lot of consequences. The most relevant one is that for regular CA it is possible to compute column subshifts of every given width (Theorem 4.36). All our decidability results for regular Cellular Automata easily follows from this property.

In order to show our fundamental decidability result (Theorem 4.35) we need to define the *cellular automaton extension* of a sofic shift and to show some basic properties.

Definition 4.24. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r. Let $\mathcal{G} = (V, E, \zeta)$ be a labeled graph with $\zeta : V \to A^{2r+1}$. For $t > 0$, let the* (F,t)-extension *of $\mathcal{G}$ be the labeled graph $\mathcal{G}_{(F,t)} = (V_t, E_t, \zeta_t)$, with $\zeta_t : V_t \to A^{2r+t}$, defined in the following way (see figure 4.3):*

- vertex set:

$$V_t = \{(v_1, .., v_t) \in V^t \mid \exists a \in A^{2r+t}, \zeta(v_i) = a_{[i,2r+i]}, 1 \leq i \leq t\}$$

- edge set:

$$E_t = \{(e_1, .., e_t) \in E^t \mid \exists v, v' \in V_t, i(e_j) = v_j, t(e_j) = v'_j, f(\zeta(v_j)) = \zeta(v'_j)_{r+1}\}$$

- labeling function:

$$\forall v = (v_1, ..., v_t) \in V_t, \zeta_t(v) = a \text{ where } a_{[i,2r+i]} = \zeta(v_i), 1 \leq i \leq t.$$

Definition 4.25. *Let $x, y \in \Sigma_k$ such that $x = x_1..x_k, y = y_1...y_k$ where $x_i, y_i \in A^{\mathbb{N}}$ and $x_{i+1} = y_i, 1 \leq i < k$. We say that x, y are* compatible sequences *and, abusing the notation of Definition 4.13, we denote with $x \odot y = x_1...x_k y_k$ their* overlapping concatenation.

We can extend Lemma 4.16 to infinite sequences.

Lemma 4.26. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r. Let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be a sofic shift and let $\mathcal{G}$ be a labeled graph presentation of S. Let $x, y \in S_{\mathcal{G}_{(F,1)}}$ be compatible sequences. Then $x \odot y \in S_{\mathcal{G}_{(F,2)}}$.*

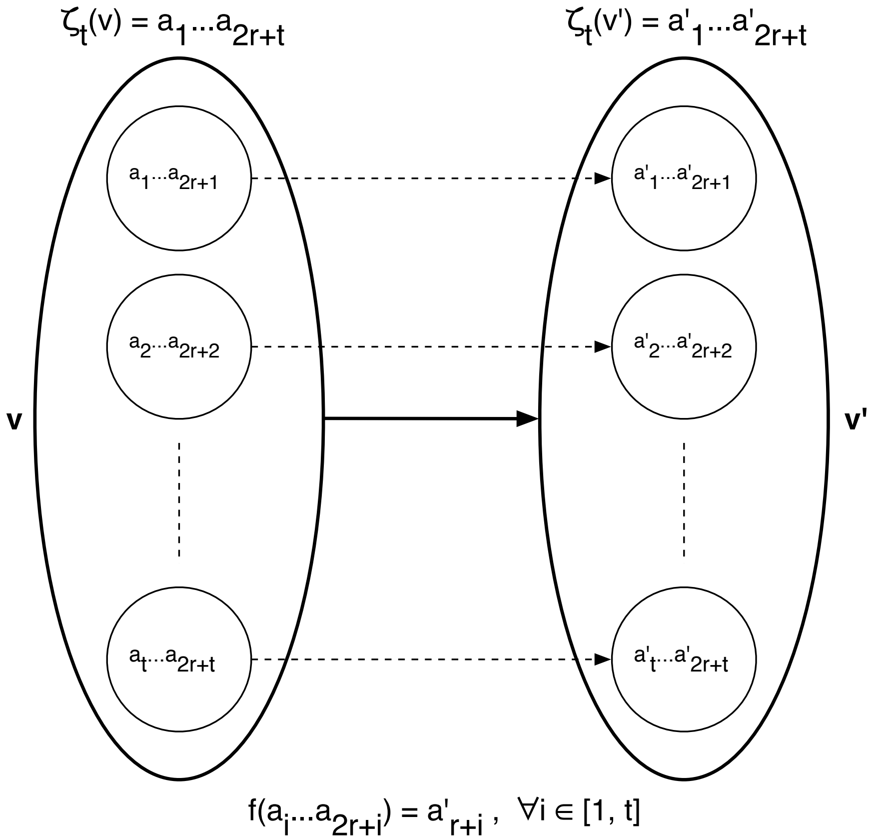


Figure 4.3: A legal edge $v \to v'$ of an (F,t)-extended graph $\mathcal{G}_{(F,t)}$.

Proof. Since, by hypothesis, $x = (x_i)_{i \in \mathbb{N}}, y = (y_i)_{i \in \mathbb{N}} \in S_{\mathcal{G}_{(F,1)}}$, there exist two paths $u_1 \to u_2 \to \ldots$ and $v_1 \to v_2 \to \ldots$ in $\mathcal{G}$ such that $\zeta(u_i) = x_i$ and $\zeta(v_i) = y_i$, $i \in \mathbb{N}$. Then, $(u_1, v_1) \to (u_2, v_2) \to \ldots$ is a legal path in $\mathcal{G}_{(F,2)}$ which implies that $x \odot y \in S_{\mathcal{G}_{(F,2)}}$. □

The following proposition shows that the sofic shift presented by the (F,t)-extension $\mathcal{G}_{(F,t)}$ of a labeled graph $\mathcal{G}$ doesn't depend on $\mathcal{G}$ but only on the sofic shift presented by $\mathcal{G}$.

Proposition 4.27. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r and let $\mathcal{G}, \mathcal{G}'$ be two distinct labeled graph presentations of the same sofic shift $S = S_{\mathcal{G}} = S_{\mathcal{G}'} \subseteq (A^{2r+1})^{\mathbb{N}}$. Then, for any $t > 0$, $S_{\mathcal{G}_{(F,t)}} = S_{\mathcal{G}'_{(F,t)}}$.*

Proof. We show that $S_{\mathcal{G}_{(F,t)}} \subseteq S_{\mathcal{G}'_{(F,t)}}$. The proof for the converse inclusion can be obtained by exchanging $\mathcal{G}$ with $\mathcal{G}'$.

First of all, note that, by definition of $(F,1)$-extension, $S_{\mathcal{G}_{(F,1)}} = S_{\mathcal{G}'_{(F,1)}}$. Let $x \in S_{\mathcal{G}_{(F,t)}}$ and let $x_1, ..., x_t \in S$ such that $x = x_1 \odot ... \odot x_t$. Then, $x_1, ..., x_t \in S_{\mathcal{G}'_{(F,1)}}$ and, by Lemma 4.26, it follows that $x \in S_{\mathcal{G}'_{(F,t)}}$. □

Thanks to Proposition 4.27 we can refer directly to the extension of a sofic shift S rather than to the extension of a labeled graph presentation of S.

Definition 4.28. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r. Let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be a sofic shift and let $\mathcal{G}$ be a labeled graph presentation of S. For $t > 0$, let denote with $S_{(F,t)} = S_{\mathcal{G}_{(F,t)}}$ the* (F,t)-extension *of the sofic shift S.*

We now show some useful properties of the (F,t)-extensions of sofic shifts.

Lemma 4.29. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r. Let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be a sofic shift. Then $\forall t > 0$,*

- **a.** *if $\Sigma_{2r+1} \subset S$ then $\Sigma_{2r+t} \subseteq S_{(F,t)}$,*
- **b.** *if $\Sigma_{2r+1} = S$ then $\Sigma_{2r+t} = S_{(F,t)}$,*
- **c.** *if $\Sigma_{2r+1} \supset S$ then $\Sigma_{2r+t} \supset S_{(F,t)}$.*

Proof.

- **a.** Let $x \in \Sigma_{2r+t}$ such that $x = x_1 \odot .. \odot x_t$ where $x_i \in \Sigma_{2r+1}$, $1 \leq i \leq t$. Then, $x_i \in S_{(F,1)}$, $1 \leq i \leq t$ and, by Lemma 4.26, $x_1 \odot .. \odot x_t \in S_{(F,t)}$.
- **b.** By point a, $\Sigma_{2r+t} \subseteq S_{(F,t)}$, thus we just have to show that $S_{(F,t)} \subseteq \Sigma_{2r+t}$ or, equivalently, that $\mathcal{L}(S_{(F,t)}) \subseteq \mathcal{L}(\Sigma_{2r+t})$. Let $k > 0$ and let $a \in \mathcal{L}_k(S_{(F,t)})$. Let $a_1, ..., a_t \in \mathcal{L}_k(S)$ be such that $a_1 \odot ... \odot a_t = a$. By hypothesis, $a_1, ..., a_t \in \mathcal{L}_k(\Sigma_{2r+1})$ then, by Lemma 4.16, it follows that $a_1 \odot ... \odot a_t \in \mathcal{L}_k(\Sigma_{2r+t})$.
- **c.** Since $\Sigma_{2r+1} \supset S$, applying the same argument of point b, it is possible to conclude that $\Sigma_{2r+t} \supseteq S_{(F,t)}$. We have just to show that the inclusion is strict. Since $\Sigma_{2r+1} \supset S$, there exists a block $b_1 \in \mathcal{L}(\Sigma_{2r+1})$ such that $b_1 \notin \mathcal{L}(S)$. Then, let $b \in \mathcal{L}(\Sigma_{2r+t})$ such that $b = b_1 \odot b_2 \odot ... \odot b_t$ for some $b_2, ..., b_t \in \mathcal{L}(\Sigma_{2r+1})$. Trivially, $b \notin \mathcal{L}(S_{(F,t)})$. □

The following theorem easily follows from Lemma 4.29 and provides a strong characterization for regular CA. It is a two-sided extension of a theorem proved by Blanchard and Maass for one-sided CA [7].

Theorem 4.30. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r. Then $(A^{\mathbb{Z}}, F)$ is regular if and only if Σ_{2r+1} is a sofic shift.*

Proof. The necessary implication is trivial. Then, suppose Σ_{2r+1} is a sofic shift. For every $d < 2r+1$, Σ_d is a factor of Σ_{2r+1} then it is a sofic shift. For every $d > 2r+1$, by Lemma 4.29 point b, Σ_d can be represented by a labeled graph then it is a sofic shift. □

In general, if Σ_d is a sofic shift for $d < 2r+1$ it is not possible to conclude that the CA is regular (see [40]).

In Section 4.2 we saw that for one-sided Cellular Automata Σ is a shift of finite type of order k if and only if the k-approximation of Σ_{2r+1} is invariant under projections. Here the scenario is a bit more complicated. To decide if some sofic shift $S = \Sigma_{2r+1}$ we build some (F,t)-extension of S and we check if the extended sofic shift $S_{(F,t)}$ respects two trivial necessary conditions:

- $S_{(F,t)}$ is invariant under projections
- $\mathcal{L}_k(S_{(F,t)}) = \mathcal{L}_k(\Sigma_{2r+t})$ for some sufficiently large $k > 0$.

Remark 4.3.1. Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r and let $\mathcal{G}_{(F,t)}$ be the (F,t)-extension of $\mathcal{G}$. Then for every $i \in [1,t]$, $\Phi_{[i,2r+i]}(S_{\mathcal{G}_{(F,t)}}) \subseteq S_{\mathcal{G}}$ where $\Phi_{[i,2r+i]} : (A^{2r+t})^{\mathbb{N}} \to (A^{2r+1})^{\mathbb{N}}$. is the projection map of Definition 4.19.

We say that a sofic shift S is F-extendible, if every (F,t)-extension of S is invariant under projections.

Definition 4.31. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r and let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be a sofic shift. S is F-*extendible *if*

$$S = \Phi_{[i,2r+i]}(S_{(F,t)}), \forall t > 0, \forall i \in [1,t].$$

Note that for a sofic shift to be F-extendible is a necessary condition in order to be equal to Σ_{2r+1}. The property of being F-extendible is decidable.

Proposition 4.32. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r and let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be a sofic shift. Then, S is F-extendible iff $S = \Phi_{[1,2r+1]}(S_{(F,2)}) = \Phi_{[2,2r+2]}(S_{(F,2)})$.*

Proof. The necessary implication is trivial. Let $S = \Phi_{[1,2r+1]}(S_{(F,2)}) = \Phi_{[2,2r+2]}(S_{(F,2)})$. Note that this implies $S = S_{(F,1)}$. Let $t > 2$, we have to show that $S = \Phi_{[i,2r+i]}(S_{(F,t)})$ for $1 \leq i \leq t$. Let $z \in S$ and let $k \in [1,t]$. To reach the proof it is sufficient to show that $z \in \Phi_{[k,2r+k]}(S_{(F,t)})$. Since $S = \Phi_{[1,2r+1]}(S_{(F,2)}) = \Phi_{[2,2r+2]}(S_{(F,2)})$, there exists $x_1, .., x_{t-1} \in S_{(F,2)}$ such that $\Phi_{[2,2r+2]}(x_i) = \Phi_{[1,2r+1]}(x_{i+1})$, $1 \leq i < t-1$ and $\Phi_{[2,2r+2]}(x_{k-1}) = \Phi_{[1,2r+1]}(x_k) = z$. Then, $x_1, .., x_{t-1}$ are compatible and by Lemma 4.26, it follows that $x_1 \odot ... \odot x_{t-1} \in S_{(F,t)}$ and $\Phi_{[k,2r+k]}(x_1 \odot ... \odot x_{t-1}) = z$. □

If a sofic shift $S \subseteq (A^{2r+1})^{\mathbb{N}}$ is F-extedible then it mut be contained in Σ_{2r+1}.

Proposition 4.33. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r and let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be a sofic shift. Suppose S is F-extendible then $S \subseteq \Sigma_{2r+1}$.*

Proof. Since S is F-extendible, $S = \Phi_{[1,2r+1]}(S_{(F,2r+1)}) = ... = \Phi_{[2r+1,4r+1]}(S_{(F,2r+1)})$. We prove by induction on $k > 0$ that $\mathcal{L}_k(S) \subseteq \mathcal{L}_k(\Sigma_{2r+1})$.

1. (Base Case) By definition, $\mathcal{L}_1(S) \subseteq \mathcal{L}_1(\Sigma_{2r+1}) = A^{2r+1}$.

2. (Inductive Case) Suppose $\mathcal{L}_k(S) \subseteq \mathcal{L}_k(\Sigma_{2r+1})$ for $k > 0$. We have to show that $\mathcal{L}_{k+1}(S) \subseteq \mathcal{L}_{k+1}(\Sigma_{2r+1})$.

   Since the radius of the CA is r, the set of blocks $\mathcal{L}_{k+1}(\Sigma_{2r+1})$ is completely determined by the set of blocks $\mathcal{L}_k(\Sigma_{4r+1})$ and $\mathcal{L}_{k+1}(\Phi_{[r+1,3r+1]}(S_{(F,2r+1)}))$ is completely determined by the set of blocks $\mathcal{L}_k(S_{(F,2r+1)})$. Thus, showing that $\mathcal{L}_k(S_{(F,2r+1)}) \subseteq \mathcal{L}_k(\Sigma_{4r+1})$ we can reach the conclusion $\mathcal{L}_{k+1}(S) \subseteq \mathcal{L}_{k+1}(\Sigma_{2r+1})$.

Let $x \in \mathcal{L}_k(S_{(F,2r+1)})$. Since S is F-extendible, there exist $x_1, .., x_{2r+1} \in \mathcal{L}_k(S)$ such that $x = x_1 \odot ... \odot x_{2r+1}$. By inductive hypothesis, $x_1, ..., x_{2r+1} \in \mathcal{L}_k(\Sigma_{2r+1})$ then, by Lemma 4.16, $x \in \mathcal{L}_k(\Sigma_{4r+1})$. □

Proposition 4.34. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r and let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be an F-extendible sofic shift. Let n be the number of states of the smallest DFA recognizing $\mathcal{L}(S)$ and let $N = (n \cdot |A|^{2r+1})^{2r+1}$. Assume that $\mathcal{L}_N(\Sigma_{4r+1}) = \mathcal{L}_N(S_{(F,2r+1)})$. Then $\Sigma_{2r+1} = S$.*

Proof. Let $M = (A^{2r+1}, Q, \delta, q_0, T)$ be the smallest DFA recognizing $\mathcal{L}(S)$. Let consider the graph $G = (V, E)$ obtained from M in the following way: the set of vertices V is the set of couples $(q, a) \in Q \times A^{2r+1}$ such that $\delta(q, a) \in Q$ and there is an edge between vertices $(q, a), (q', a')$ if and only if $\delta(q, a) = q'$. A labeling $\zeta : V \to A^{2r+1}$ for G is $\zeta((q, a)) = a$.
We show by induction on $k > 0$ that $\mathcal{L}_k(\Sigma_{4r+1}) = \mathcal{L}_k(S_{(F,2r+1)})$.

a. (Base Case) By hypothesis, $\mathcal{L}_N(\Sigma_{4r+1}) = \mathcal{L}_N(S_{(F,2r+1)})$. Moreover, since the language of a subshift is factorial, $\mathcal{L}_k(\Sigma_{4r+1}) = \mathcal{L}_k(S_{(F,2r+1)})$, $\forall k \leq N$.

b. (Inductive Case) Suppose $\mathcal{L}_K(\Sigma_{4r+1}) = \mathcal{L}_K(S_{(F,2r+1)})$, $K \geq N$. We have to show that $\mathcal{L}_{K+1}(\Sigma_{4r+1}) = \mathcal{L}_{K+1}(S_{(F,2r+1)})$.

Let $a \in \mathcal{L}_{K+1}(\Sigma_{4r+1})$ and let $a^1, ..., a^{2r+1} \in \mathcal{L}_{K+1}(\Sigma_{2r+1})$ the unique blocks such that $a = a^1 \odot ... \odot a^{2r+1}$. By inductive hypothesis, $\mathcal{L}_K(\Sigma_{4r+1}) = \mathcal{L}_K(S_{(F,2r+1)})$ and the set of blocks $\mathcal{L}_K(\Sigma_{4r+1})$ determines the set of blocks $\mathcal{L}_{K+1}(\Sigma_{2r+1})$. Then it follows that $\mathcal{L}_{K+1}(\Sigma_{2r+1}) = \mathcal{L}_{K+1}(S)$ and that $a^1, ..., a^{2r+1} \in \mathcal{L}_{K+1}(S)$. By definition of $\mathcal{G}$, for every such a^i there is in $\mathcal{G}$ a unique legal path

$$(q_0, a_0^1) \to (q_1^1, a_1^1) \to ... \to (q_K^1, a_K^1)$$
$$...$$
$$(q_0, a_0^{2r+i}) \to (q_1^{2r+i}, a_1^{2r+i}) \to ... \to (q_K^{2r+i}, a_K^{2r+i}).$$

We show that there exists $x \in S_{(F,2r+1)}$ such that $x_{[0,K]} = a$. Let $y \in S_{(F,2r+1)}$ such that $y_{[0,K-1]} = a_0 ... a_{K-1}$. One such y exists since, by inductive hypothesis, $\mathcal{L}_K(\Sigma_{4r+1}) = \mathcal{L}_K(S_{(F,2r+1)})$. Then there exists an unique path in $\mathcal{G}_{(F,2r+1)}$

$$v_0 \to v_1 \to v_2 \to ...$$

such that $v_0 = ((q_0, a_0^1), .., (q_0, a_0^{2r+i})), ..., v_{K-1} = ((q_{K-1}^1, a_{K-1}^1), .., (q_{K-1}^{2r+i}, a_{K-1}^{2r+i}))$ and $\zeta(v_i) = y_i$. Since $K > N$ there exist $0 \leq i < j < K$ such that $v_i = v_j$. Then, let consider the legal paths in $\mathcal{G}$:

$$(q_0, a_0^1) \to ... \to (q_i^1, a_i^1) \to (q_{j+1}^1, a_{j+1}^1) \to ... \to (q_K^1, a_K^1)$$
$$...$$
$$(q_0, a_0^{2r+1}) \to ... \to (q_i^{2r+1}, a_i^{2r+1}) \to (q_{j+1}^{2r+1}, a_{j+1}^{2r+1}) \to ... \to (q_K^{2r+1}, a_K^{2r+1})$$

and the related labeling

$$\overline{a}^1 = a_0^1 ... a_i^1 a_{j+1}^1 ... a_K^1$$
$$...$$
$$\overline{a}^{2r+1} = a_0^{2r+1} ... a_i^{2r+1} a_{j+1}^{2r+1} ... a_K^{2r+1}.$$

Since S is F-extendible, by Proposition 4.33, $\overline{a}^1, ..., \overline{a}^{2r+1} \in \mathcal{L}(\Sigma_{2r+1})$. Then, by Lemma 4.16, $\overline{a} = \overline{a}^1 \odot ... \odot \overline{a}^{2r+1} \in \mathcal{L}(\Sigma_{4r+1})$ and, by inductive hypothesis, $\overline{a} \in \mathcal{L}(S_{(F,2r+1)})$. Then there exists $\overline{y} \in S_{(F,2r+1)}$ such that $\overline{y}_{[0,|\overline{a}|-1]} = \overline{a}$ and a unique path in $\mathcal{G}_{(F,2r+1)}$:

$$\overline{v}_0 \to \overline{v}_1 \to \overline{v}_2 \to ...$$

such that $\overline{v}_0 = v_0, .., \overline{v}_i = v_i$, $\overline{v}_{i+1} = v_j$ and $\zeta(\overline{v}_i) = \overline{y}_i$. Then there exists also the path in $\mathcal{G}_{(F,2r+1)}$:

$$\overline{v}_0 \to ... \to \overline{v}_i \to v_{i+1} \to ... \to v_{j-1} \to \overline{v}_{i+1} \to \overline{v}_{i+2} \to ....$$

Thus the sequence $x = \overline{y}_0 ... \overline{y}_i y_{i+1} ... y_{j-1} \overline{y}_{i+1} ... \in S_{(F,2r+1)}$ and $x_{[0,K]} = a$. □

Now we are ready to state our main result and next to show the most immediate consequences.

Theorem 4.35. *Let $(A^{\mathbb{Z}}, F)$ be a CA with radius r and let $S \subseteq (A^{2r+1})^{\mathbb{N}}$ be a sofic shift. Then it is decidable whether $S = \Sigma_{2r+1}$.*

Proof. By Proposition 4.32, it is decidable if S is F-extendible. Then, the proof follows from the decidability of the condition of Proposition 4.34. ☐

We now explore some important consequences of Theorem 4.35 related to regular Cellular Automata.

Theorem 4.36. *Let $(A^{\mathbb{Z}}, F)$ be regular. Then $\forall t > 0$, Σ_t is computable.*

Proof. Let r be the radius of the CA. By Theorem 4.35, given a sofic shift $S \subseteq (A^{2r+1})^{\mathbb{N}}$, it is possible to decide if $S = \Sigma_{2r+1}$. We can enumerate all labeled graph representing all sofic shifts contained in A^{2r+1}. Then there exists an algorithm that iteratively generates graphs in the enumeration and checks if the shift represented is Σ_{2r+1}. Since $(A^{\mathbb{Z}}, F)$ is regular, Σ_{2r+1} will be eventually generated and recognized. This proves that, if $(A^{\mathbb{Z}}, F)$ is regular, Σ_{2r+1} is computable.

In general, if $t < 2r+1$, we can compute Σ_t by simply taking the projection $\Phi_{[1,t]}(\Sigma_{2r+1})$ otherwise, if $t > 2r+1$, by Lemma 4.29 point b, we can compute Σ_t by computing the $(F, t-2r)$-extension of Σ_{2r+1}. ☐

The following proposition extends Proposition 4.22.

Proposition 4.37. *The topological entropy of one-sided regular CA is computable.*

Proof. Since the entropy of sofic shifts is computable, the conclusion follows from Theorem 3.14 and Theorem 4.36. ☐

The general question whether the topological entropy is computable for the class of regular Cellular Automata remains open (see [19]).

Question 4.3.2. Is the topological entropy computable for regular Cellular Automata?

The following following shows that if we restrict to the class of regular CA, it is possible to provide answers to questions which are undecidable in the general case.

Proposition 4.38. *Let $(A^{\mathbb{Z}}, F)$ be a regular CA. Then the following topological properties are decidable.*

1. *Nilpotency*

2. *Equicontinuity*

3. *Positively Expansiveness*

Proof. By Theorem 4.36, given $(A^{\mathbb{Z}}, F)$, it is possible to compute Σ_{2r+1}.

1. It is easy to see that $(A^{\mathbb{Z}}, F)$ is nilpotent if and only if there exists $a \in A^{2r+1}$ and $N > 0$ such that $\forall n \geq N, \forall x \in \Sigma_{2r+1}$, $\sigma^n(x) = a$. Given a labeled graph representation of Σ_{2r+1}, this last condition is trivially algorithmically checkable.

2. It is easy to see that $(A^{\mathbb{Z}}, F)$ is equicontinuous if and only if $\mathcal{L}(\Sigma_{2r+1})$ is a bounded periodic language and that, given a labeled graph representation of Σ_{2r+1}, it is algorithmically checkable if $\mathcal{L}(\Sigma_{2r+1})$ is bounded periodic.

3. Every positively expansive CA is conjugated to (Σ_{2r+1}, σ). If we can compute Σ_{2r+2} we can also check if there is some k-block automorphism between $\Phi_{[1,2r+1]}(\Sigma_{2r+2})$ and $\Phi_{[1,2r+1]}(\Sigma_{2r+2})$. □

Nilpotency and equicontinuity are in general undecidable properties. It is actually unknown if positively expansiveness is a decidable property for general Cellular Automata. Since nilpotency is undecidable, from Proposition 4.38 follows the undecidability of regularity.

Corollary 4.39. *Regularity is an undecidable property.*

To conclude, we remark that, as a consequence of Corollary 4.39, the membership in class **L3** of Kůrka's Language classification is undecidable.

Corollary 4.40. *The membership in Kůrka's Languages classes is undecidable.*

Question 4.3.3. Is sensitivity a decidable property for regular Cellular Automata?

Chapter 5

Computational Complexity of Cellular Automata

In this chapter we study the intersection between computational and dynamical properties of Cellular Automata (the results in this chapter are collected in [20]).

We interpret the process of computation in Cellular Automata as a flow toward a subshift attractor. We show that it is possible to restate the halting problem as the problem to decide if the omega limit of some clopen set converges to an *halting* subshift attractor (that is, as the problem to decide if the orbits of all sequences contained in some clopen set converge to some attractor eliged as halting set). We say that the computational complexity of a cellular automaton $(A^{\mathbb{Z}}, F)$ 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 a word in A^*, we can characterize the complexity of a basin of attraction by using formal language theory. We propose a classification of Cellular Automata according to the complexity of *basin languages* (Section 5.1). A cellular automaton with highest computational complexity has at least one subshift attractor whose basin language is strictly recursively enumerable.

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, Equicontinuity and Languages classifications (Section 5.2). From the intersection classes we can provide necessary conditions for a cellular automaton to be universal (Section 5.3).

5.1 Basin Language classification

In this section we are interested in the basins of attraction of subshift attractors. We study the *complexity* of such basins by using formal language theory.

First, we show that the basin of attraction of a subshift attractor is always a dense open set.

Proposition 5.1. *The basin of every subshift attractor is a dense open set.*

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

A *qualitative characterization* of basins of attraction is provided by formal language theory. By Proposition 5.1, the basin $\mathcal{B}(Z)$ 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^*. Considering basins of subshift attractors offers some advantages respect to basins of general attractors. Since the basin of a subshift attractor is σ-invariant, we don't need to take care of the coordinate of the cylinder in the space $A^{\mathbb{Z}}$. This means that if a cylinder $[u]_i$ is contained in the basin of some subshift attractor Z, then for every $j \in \mathbb{Z}$, $[u]_j$ is contained in $\mathcal{B}(Z)$ (this implies that the orbit of every configuration which contains the word u will converge to Z).

Definition 5.2. *Let denote with*

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

the basin language *of the subshift attractor Z of $(A^{\mathbb{Z}}, F)$.*

Note that, since $\mathcal{B}(Z)$ is open and σ-invariant, $A^{\mathbb{Z}} \setminus \mathcal{B}(Z)$ is either a subshift or it is empty. The language complexity of $\mathcal{L}_Z$ is a qualitative measure of the *complexity* of $\mathcal{B}(Z)$. We show that the language $\mathcal{L}_Z$ can be at most recursively enumerable. Next we show that $\mathcal{L}_Z$ can be strictly recursively enumerable.

Lemma 5.3. *Let $(A^{\mathbb{Z}}, F)$ be a cellular automaton. Let $V \subseteq A^{\mathbb{Z}}$ be a clopen F-invariant spreading set and let $U \subseteq A^{\mathbb{Z}}$ 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, $\overline{V} = A^{\mathbb{Z}} \setminus V$ is clopen and compact. For $n \in \mathbb{N}$, let define $X_n = \{x \in U \mid F^n(x) \notin V\} \subseteq U \cap \overline{V}$. 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, $X = \cap_{n\in\mathbb{N}} X_n \subseteq U \cap \overline{V}$ is not empty and $\omega(X) \cap \overline{V} \neq \emptyset$ which is a contradiction. □

Proposition 5.4. *Let Z be a subshift attractor of $(A^{\mathbb{Z}}, F)$. Then $\mathcal{L}_Z$ is r.e.*

Proof. Let $U \subseteq A^{\mathbb{Z}}$ be a clopen F-invariant spreading set such that $\omega(U) = Z$. By Lemma 5.3, for every $u \in A^*$, $[u] \in \mathcal{B}(Z)$ 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 $[u] \subseteq \mathcal{B}(Z)$ is a semidecidable question. Then $\mathcal{L}_Z$ is at most recursively enumerable. □

The following proposition shows that every r.e. language recognition problem is Turing-reducible to the basin language recognition problem for some cellular automaton. In particular we show that the *halting problem* for Turing Machines can be rephrased in terms of *reachability of a subshift attractor* for Cellular Automata. For instance, we show that the question:

> *does the Turing Machine M halt on input $u \in B^*$?*

can be restated as

> *is $\omega([\varphi(u)]) \subseteq Z$?*

where $\varphi : B^* \to A^*$ is an injective computable mapping and Z is a subshift attractor of some cellular automaton $(A^{\mathbb{Z}}, F)$.

Proposition 5.5. *Let $\mathcal{L} \subseteq B^*$ be a r.e. language. Then there is a cellular automaton $(A^{\mathbb{Z}}, F)$ with a subshift attractor Z and an injective computable mapping $\varphi : B^* \to A^*$ such that $u \in \mathcal{L}$ if and only if $\varphi(u) \in \mathcal{L}_Z$.*

Proof. Let $M = (B, Q, \delta, q_0, F)$ be a Turing machine recognizing language L. Let define $(A^{\mathbb{Z}}, F)$ 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 encounters S and/or R: in that case generates a S particle. 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 some erroneous step occurs (unknown transition, two states collide, ..) then it is generated a particle S. If a final state is reached, then it is generated a particle S. Note that S^∞ is a subshift attractor.

Let define the computable mapping $\varphi : B^* \to A^*$ by $\varphi(u_1 ... u_n) = Lq_0u_1...u_nR$. It is easy to see that if $a \in B$ is some tape symbol of the Turing Machine then $\omega(...aaaLq_0u_1...u_nRaaa...) = S^\infty$ if and only if $u = u_1...u_n \in \mathcal{L}$. Then u is accepted by M if and only if $\omega([Lq_0uR]) = S^\infty$. □

We can classify Cellular Automata according to basin languages complexity.

Corollary 5.6. *Every $(A^{\mathbb{Z}}, F)$ falls exactly in one of the following classes:*

B1 $\exists Z, \mathcal{L}_Z = A^*$

B2 $\forall Z, \mathcal{L}_Z \neq A^*$ *is recursive*

B3 $\exists Z, \mathcal{L}_Z$ *is strictly r.e.*

According to the above Basin Language classification, Cellular Automata capable of universal computation are in class **B3**. By the existence of intermediate Turing degrees we cannot affirm that all Cellular Automata in class **B3** are universal so if we can provide some characterization for class **B3** we just have necessary conditions for the universality. Several natural questions easily arise.

Question 5.1.1. Is the membership in Basin Language classes decidable?

Is it possible to characterize classes $\mathbf{B1}, \mathbf{B2}, \mathbf{B3}$ in terms of the cardinality of subshift attractors? For instance, every cellular automaton in **B1** has just one subshift attractor.

Question 5.1.2. Is there some cellular automaton with an infinite number of subshift attractors in **B2**?

Question 5.1.3. Is there some cellular automaton with a finite number of subshift attractors in **B3**?

5.2 Classes comparison

In this section we compare Basin Language classification with Attractors (Section 3.2.3), Equicontinuity (Section 3.2.2) and Language (Section 3.2.4) classifications. First we show two techniques to build Cellular Automata with nice properties. These two constructions will be useful to investigate the intersection classes.

The first construction is the *product cellular automaton.*

Definition 5.7. *The* product cellular automaton $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G)$ *of* $(A^{\mathbb{Z}}, F)$ *with* $(B^{\mathbb{Z}}, G)$ *is defined by* $\forall (x, y) \in A^{\mathbb{Z}} \times B^{\mathbb{Z}}, (F \times G)(x, y) = (F(x), G(y))$.

The proof of the following lemmas are trivial.

Lemma 5.8. *Let* $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G)$ *be a product cellular automaton. Then* $(Z', Z'') \subseteq A^{\mathbb{Z}} \times B^{\mathbb{Z}}$ *is a (subshift) attractor of* $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G)$ *if and only if* Z' *and* Z'' *are (subshift) attractors of* $(A^{\mathbb{Z}}, F)$ *and* $(B^{\mathbb{Z}}, G)$*, respectively.*

Lemma 5.9. *Let* $(A^{\mathbb{Z}}, F) \in \mathbf{A}i$ *and let* $(B^{\mathbb{Z}}, G) \in \mathbf{A}j$ *for* $1 \leq i, j \leq 5$*. Then* $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G) \in \mathbf{A}k, k = Min\{i, j\}$.

Lemma 5.10. *Let* $(A^{\mathbb{Z}}, F) \in \mathbf{E3}$*. Then* $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G) \in \mathbf{E3}$ *for every cellular automaton* $(B^{\mathbb{Z}}, G)$.

Lemma 5.11. *Let* $(A^{\mathbb{Z}}, F) \in \mathbf{L3}$. *Then* $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G) \in \mathbf{L3}$ *for every cellular automaton* $(B^{\mathbb{Z}}, G)$.

Lemma 5.12. *Let* $(A^{\mathbb{Z}}, F) \in \mathbf{B}i$ *and let* $(B^{\mathbb{Z}}, G) \in \mathbf{B}j$ *for* $1 \leq i, j \leq 3$. *Then* $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G) \in \mathbf{B}k, k = Max\{i, j\}$.

Proof. By Lemma 5.8, the language $\mathcal{L}_Z$ of the subshift attractor $Z = (Z', Z'')$ of $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G)$ is $\mathcal{L}_Z = \mathcal{L}_{Z'} \times \mathcal{L}_{Z''}$. Then, since $\mathcal{L}_Z$ can be at most recursively enumerable, the language complexity of $\mathcal{L}_Z$ is trivially the highest between the complexities of languages $\mathcal{L}_{Z'}$ and $\mathcal{L}_{Z''}$. □

The second construction consists in adding a spreading state to a cellular automaton.

Definition 5.13. *Let* $(A^{\mathbb{Z}}, F)$ *be a CA and let* $s \notin A, A_s = A \cup \{s\}$. *Then, let* $(A_s^{\mathbb{Z}}, F_s)$ *denote the CA whose local rule* $f_s : A_s^{2r+1} \to A_s$ *is defined by*

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

Lemma 5.14. *Let* $(A^{\mathbb{Z}}, F)$ *be a cellular automaton and let* $s \notin A$. *Let consider* $(A_s^{\mathbb{Z}}, F_s)$. *Then* $(A_s^{\mathbb{Z}}, F_s) \in \mathbf{E2} \cap \mathbf{A3} \cap (\mathbf{B2} \cup \mathbf{B3})$. *Moreover,* $(A_s^{\mathbb{Z}}, F_s) \in \mathbf{B2}$ *if and only if* $(A^{\mathbb{Z}}, F) \in \mathbf{B1} \cup \mathbf{B2}$.

Proof. By definition, s is a blocking word. Moreover, $Z_s = \{s^\infty\} \neq \omega(A_s^{\mathbb{Z}})$ is a fixed point attractor. Then $(A_s^{\mathbb{Z}}, F_s) \in \mathbf{E2} \cap \mathbf{A3}$ and $(A_s^{\mathbb{Z}}, F_s) \notin \mathbf{B1}$. We now show that adding a spreading state doesn't affect the complexity of the basin languages of $(A^{\mathbb{Z}}, F)$. The basin of attraction of Z_s consists of the set of all biinfinite sequences which contain at least one occurrence of s, that is $\mathcal{B}(Z_s) = \{x \in A_s^{\mathbb{Z}} \mid \exists i \in Z, x_i = s\}$. Then, the basin language $\mathcal{L}_{Z_s} = \{w \in A_s^* \mid \exists i, w_i = s\}$ is recursive. It is easy to see that Z is a subshift attractor of $(A_s^{\mathbb{Z}}, F_s)$ if and only if $Z = \omega(U \cup [s])$ where $U \subseteq A^{\mathbb{Z}}$ is a clopen F-invariant spreading set for $(A^{\mathbb{Z}}, F)$. Let $Z' = \omega(U) \subset A^{\mathbb{Z}}$ ba a subshift attractor of $(A^{\mathbb{Z}}, F)$. Then $\mathcal{L}_Z = \mathcal{L}_{Z'} \cup \mathcal{L}_{Z_s}$ and $\mathcal{L}_{Z'} \cap \mathcal{L}_{Z_s} = \emptyset$ which implies that $\mathcal{L}_Z$ is strictly recursively enumerable if and only if $\mathcal{L}_{Z'}$ is strictly recursively enumerable. □

5.2.1 Comparison with Language classification

By Theorem 3.47, 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 5.15. *Every equicontinuous cellular automaton has a unique subshift attractor which is a mixing shift of finite type.*

Proof. Since $(A^{\mathbb{Z}}, F)$ is stable, then $Z = \omega(A^{\mathbb{Z}}) = F^n(A^{\mathbb{Z}})$ for some $n \in \mathbb{N}$. Then Z is a mixing sofic shift. We show that Z is actually a SFT. Since $(A^{\mathbb{Z}}, F)$ is equicontinuous, there exists $p > 0$ such that $\forall x \in Z, \forall i \in \mathbb{N}, F^{ip}(x) = x$. (see [36]). Let r be the radius of $(A^{\mathbb{Z}}, F)$ and let consider the shift of finite type defined by $Z^{(2rp+1)} = \{x \in A^{\mathbb{Z}} \mid \forall i \in \mathbb{Z}, x_{[i,2rp+i]} \in \mathcal{L}_{2rp+1}(Z)\}$, i.e. the shift of finite type identified by the set of legal $(2rp+1)$-blocks of Z. Obviously, $Z \subseteq Z^{(2rp+1)}$. Moreover, F^p is the identity on $Z^{(2rp+1)}$, then $Z^{(2rp+1)} \subseteq Z$.
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 $U \neq Z$, $U \cap Z \neq \emptyset$ and Z is mixing, there exists $y \in Z$ and $m \in \mathbb{Z}$ such that $y \in U$ and $\sigma^m(y) \notin U$. Then, for every $i \in \mathbb{N}$, $F^{ip}(\sigma^m(x)) = \sigma^m(x) \notin U$ 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 5.16. *If* $(A^{\mathbb{Z}}, F)$ *is regular then* $\forall Z, \mathcal{L}_Z$ *is recursive.*

Proof. We show that for every $u \in A^*$ the question $[u] \subseteq \mathcal{B}(Z)$ is decidable.

Let $U \subseteq A^{\mathbb{Z}}$ 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 $(A^{\mathbb{Z}}, F)$ is regular, by Theorem 4.36, it is possible to compute a labeled graph representation $\mathcal{G}$ of its column factor Σ_N where $N = \max\{k, |v|\}$. Then $\omega([u]) \not\subseteq Z$ if and only if there exists in $\mathcal{G}$ an infinite path $q_1 \xrightarrow{w_1} q_2 \xrightarrow{w_2} q_3 ...$ such that $u \sqsubseteq w_1$ and $[w_i] \not\subseteq U, \forall i \in \mathbb{N}$. Given a labeled graph $\mathcal{G}$ this property is easily decidable. □

Corollary 5.17. **L1** $\subset$ **B1***,* **L2** $\cap$ **B1** $\neq \emptyset$*,* **L3** $\cap$ **B1** $\neq \emptyset$*.*

Proof. Since every surjective cellular automaton is in $\mathbf{B1}$, the proof follows from the nonemptiness of the intersection classes $\mathbf{L}i \cap \mathbf{A5} \neq \emptyset, 1 \leq i \leq 3$ (see [37]) and from $\mathbf{L1} = \mathbf{E1} \subset \mathbf{B1}$ (see Theorem 3.47 and Proposition 5.15). □

Corollary 5.18. $\mathbf{L2} \subset \mathbf{B1} \cup \mathbf{B2}$

Proof. The automaton of Example 3.49 has two subshift attractors and it is regular. Then $\mathbf{L2} \cap \mathbf{B2} \neq \emptyset$. The conclusion follows from Proposition 5.16. □

Corollary 5.19. $\mathbf{L3} \cap \mathbf{B2} \neq \emptyset$, $\mathbf{B3} \subset \mathbf{L3}$.

Proof. Let $(A^{\mathbb{Z}}, F) \in \mathbf{L3} \cap \mathbf{B1}$ and let $(B^{\mathbb{Z}}, G) \in \mathbf{L2} \cap \mathbf{B2}$. Then, by Lemma 5.11 and Lemma 5.12, $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G) \in \mathbf{L3} \cap \mathbf{B2}$. The inclusion $\mathbf{B3} \subset \mathbf{L3}$ follows from Corollary 5.18. □

	L1	L2	L3
B1	X	X	X
B2		X	X
B3			X

Figure 5.1: Basin Language and Languages classifications.

5.2.2 Comparison with Equicontinuity classification

Corollary 5.20. $\mathbf{E1} \subset \mathbf{B1}$, $\mathbf{E2} \cap \mathbf{B1} \neq \emptyset$, $\mathbf{E3} \cap \mathbf{B1} \neq \emptyset$, $\mathbf{E4} \subset \mathbf{B1}$.

Proof. By Proposition 5.15, $\mathbf{E1} \subset \mathbf{B1}$. Moreover $\mathbf{E4} \subset \mathbf{A5} \subset \mathbf{B1}$. For the other two cases, the proof follows from the nonemptiness of the intersection classes $\mathbf{E}i \cap \mathbf{A5} \neq \emptyset, 2 \leq i \leq 4$ (see [37]). □

Corollary 5.21. $\mathbf{E2} \cap \mathbf{B2} \neq \emptyset$, $\mathbf{E2} \cap \mathbf{B3} \neq \emptyset$.

Proof. Let $(A^{\mathbb{Z}}, F) \in \mathbf{B}i, 2 \leq i \leq 3$, and let $s \notin A$. Then, by Lemma 5.14, $(A_s^{\mathbb{Z}}, F_s) \in \mathbf{E2} \cap \mathbf{B}i$. □

Corollary 5.22. $\mathbf{E3} \cap \mathbf{B2} \neq \emptyset$, $\mathbf{E3} \cap \mathbf{B3} \neq \emptyset$.

Proof. Let $(A^{\mathbb{Z}}, F) \in \mathbf{E3} \cap \mathbf{B1}$ and let $(B^{\mathbb{Z}}, G) \in \mathbf{E2} \cap \mathbf{B}i, 2 \leq i \leq 3$. Then, by Lemma 5.10 and Lemma 5.12, $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, F \times G) \in \mathbf{E3} \cap \mathbf{B}i$. □

	E1	E2	E3	E4
B1	X	X	X	X
B2		X	X	
B3		X	X	

Figure 5.2: Basin Language and Equicontinuity classifications.

5.2.3 Comparison with Attractor classification

Corollary 5.23. $\mathbf{A1} \cap \mathbf{B1} \neq \emptyset$, $\mathbf{A1} \cap \mathbf{B2} \neq \emptyset$, $\mathbf{A1} \cap \mathbf{B3} \neq \emptyset$.

Proof. The identity cellular automaton $(\{0,1\}^{\mathbb{Z}}, I)$ has disjoint attractors $\omega([0]), \omega([1])$ and, since it is surjective its unique subshift attractor is the full space. Then $\mathbf{A1} \cap \mathbf{B1} \neq \emptyset$. Let $(B^{\mathbb{Z}}, G) \in \mathbf{B}i, 1 \leq i \leq 3$. Then, by Lemma 5.9 and Lemma 5.12, $(A^{\mathbb{Z}} \times B^{\mathbb{Z}}, I \times G) \in \mathbf{A1} \cap \mathbf{B}i$. □

Corollary 5.24. $\mathbf{A2} \cap \mathbf{B1} \neq \emptyset$, $\mathbf{A2} \cap \mathbf{B2} \neq \emptyset$, $\mathbf{A2} \cap \mathbf{B3} \neq \emptyset$.

Proof. Let $(A^{\mathbb{Z}}, F) \in \mathbf{A2} \cap \mathbf{B1}$ be the Hurley cellular automaton of Example 3.39. Let $(B^{\mathbb{Z}}, G) \in \mathbf{B}i, 2 \leq i \leq 3$ and let $s \notin B$. By Lemma 5.14, $(B_s^{\mathbb{Z}}, G_s) \in \mathbf{A3} \cap \mathbf{B}i$. Then, by Lemma 5.9 and Lemma 5.12, $(A^{\mathbb{Z}} \times B_s^{\mathbb{Z}}, F \times G_s) \in \mathbf{A2} \cap \mathbf{B}i$. □

Corollary 5.25. $\mathbf{A3} \cap \mathbf{B1} = \emptyset$, $\mathbf{A3} \cap \mathbf{B2} \neq \emptyset$, $\mathbf{A3} \cap \mathbf{B3} \neq \emptyset$.

Proof. If $(A^{\mathbb{Z}}, F) \in \mathbf{A3}$ then it has at least two subshift attractors: the minimal attractor and the maximal attractor. Then $\mathbf{A3} \cap \mathbf{B1} = \emptyset$. Let $(A^{\mathbb{Z}}, F) \in \mathbf{B}i, 2 \leq i \leq 3$ and $s \notin A$. Then, by Lemma 5.14, $(A_s^{\mathbb{Z}}, F_s) \in \mathbf{A3} \cap \mathbf{B}i \neq \emptyset$. □

To conclude, since a cellular automaton in $\mathbf{A4} \cup \mathbf{A5}$ has only one attractor, we can easily derive the intersection classes for **A4** and **A5**.

Corollary 5.26. $\mathbf{A4} \cup \mathbf{A5} \subset \mathbf{B1}$.

	A1	A2	A3	A4	A5
B1	X	X		X	X
B2	X	X	X		
B3	X	X	X		

Figure 5.3: Basin Language and Attractors classifications

5.3 Necessary conditions for universality

In Section 5.1, we classified Cellular Automata according to the complexity of the languages rising from the basins of attraction of subshift attractors (see Corollary 5.6). According to our classification, Cellular Automata capable of universal computation are in our highest complexity class. In Section 5.2, we investigated the intersection classes between our classification and Languages, Equicontinuity and Attractors classifications (see figure 5.4). By exploring intersection classes we can provide necessary conditions for Cellular Automata to be universal. Like in [16], 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). Several questions remain open:

1. Is there some stable cellular automaton with an infinite number of subshift attractors?

2. Is the membership in our classes decidable?

3. Is there some cellular automaton with an infinite number of subshift attractors in class **B2**?

4. Is there some cellular automaton with a finite number of subshift attractors in class **B3**?

	A1	A2	A3	A4	A5	E1	E2	E3	E4	L1	L2	L3
B1	X	X		X	X	X	X	X	X	X	X	X
B2	X	X	X				X	X			X	X
B3	X	X	X				X	X				X

Figure 5.4: Classes comparison.

References

[1] R. Adler, A.Konheim, M. McAndrew. Topological entropy. Trans. Amer. Math. Soc., 114, 309–319 (1965).

[2] E.R. Banks. Information Processing and Transmission in Cellular Automata. Ph.D. thesis, 1971 (MIT, Dep't of Mechanical Engineering).

[3] E.R. Berlekamp, J.H. Conway, R.K. Guy. Winning Ways for your Mathematical Plays. Academic Press, 1982.

[4] S. Bandini, G. Erbacci, G. Mauri. Implementing Cellular Automata Based Models on Parallel Architectures: The CAPP Project. PaCT 1999: 167–179.

[5] F. Blanchard, G. Hansel. Languages and subshifts. Automata on infinite words (Le Mont-Dore, 1984), 138–146, Lecture Notes in Comput. Sci., 192, Springer, Berlin, 1985.

[6] F. Blanchard, A. Maass. Dynamical properties of expansive one-sided cellular automata. Israel J. Math. 99, 149–174 (1997).

[7] F. Blanchard, A. Maass. Dynamical Behavior of Coven's Aperiodic Cellular Automata. Theoret. Comput. Sci. 163 (1996), no. 1 & 2, 291–302 .

[8] A. Ben-Hur, H. Siegelmann, S. Fishman. A theory of complexity for continuous time systems. J. Complexity 18 (2002), no. 1, 51–86.

[9] A.W. Burks. Essays on Cellular Automata. University of Illinois Press (1970).

[10] R. Cappuccio, G. Cattaneo, G. Erbacci, U. Jocher. A parallel implementation of a cellular automata based model for coffee percolation. Cellular automata: from modeling to applications (Trieste, 1998). Parallel Comput. 27 (2001), no. 5, 685–717.

[11] K. Culik II, L. P. Hurd, S. Yu. Computation theoretic aspects of cellular automata. Cellular automata: theory and experiment (Los Alamos, NM, 1989). Phys. D 45 (1990), no. 1-3, 357–378.

[12] N. Chomsky. Three models for the description of language. IRE Transactions on Information Theory (2): 113–124 (1956).

[13] E.F. Codd. Cellular Automata. Academic Press (1968).

[14] M. Cook. Universality in Elementary Cellular Automata. Complex Systems 15, 1–40, 2004.

[15] B. Durand, E. Formenti, G. Varouchas. On undecidability of equicontinuity classification for cellular automata. Discrete models for complex systems, DMCS '03 (Lyon), 117–127. Discrete Math. Theor. Comput. Sci. Proc., AB, Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2003.

[16] J.C. Delvenne, P. Kůrka, V. Blondel. Decidability and universality in symbolic dynamical systems. Machines, computations, and universality, 104–115, Lecture Notes in Comput. Sci., 3354, Springer, Berlin, 2005.

[17] P. Di Lena. On Computing the Topological Entropy of one-sided Cellular Automata. To appear International Journal of Unconventional Computing, 2007.

[18] P. Di Lena. Decidable properties for Regular Cellular Automata. In preceedings of IFIP/TCS conference 22-24 August 2006, Santiago, Chile.

[19] P. Di Lena, L. Margara. Row Subshifts and Topological Entropy of Cellular Automata. To appear in International Journal of Unconventional Computing, 2007.

[20] P. Di Lena, L. Margara. Computational complexity of Dynamical Systems: the case of Cellular Automata. To appear in proceedings of 1st International Conference on Languages and Automata Theory and Application (LATA2007), March 29 - April 4, 2007, Tarragona, Spain.

[21] M.D'amico, G. Manzini, L. Margara. On computing the entropy of cellular automata. Theoret. Comput. Sci. 290 (2003), no. 3, 1629–1646.

[22] E. Formenti, P. Kůrka. Subshift attractors of cellular automata. Nonlinearity 20 (2007), 105–117.

[23] M. Gardner. Mathematical games. Scientific American, October 1970.

[24] R.H. Gilman, Robert H. Classes of linear automata. Ergodic Theory Dynam. Systems 7, no. 1, 105–118 (1987).

[25] R.H. Gilman. Notes on Cellular Automata. Preprint (1988).

[26] Hedlund, G. A. Endormorphisms and automorphisms of the shift dynamical system. Math. Systems Theory 3, 320–375 (1969).

[27] L.P. Hurd, J. Kari, K. Culik. The topological entropy of cellular automata is uncomputable. Ergodic Theory Dynam. Sys. 12, no. 2, 255–265 (1992).

[28] J. Hardy, O. de Pazzis, and Y. Pomeau. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. Physical Review A, 13(5):1949–1961, May 1976.

[29] J. Hopcroft, J.D. Ullman. Introduction to automata theory, languages, and computation. Addison-Wesley Series in Computer Science. Addison-Wesley Publishing Co., Reading, Mass. (1979).

[30] M. Hurley. Attractors in cellular automata. Ergodic Theory Dynam. Systems 10 (1990), no. 1, 131–140.

[31] L. Hurd, Formal language characterizations of cellular automaton limit sets, Complex Systems 1 (1987) 69–80.

[32] L. Hurd, The application of formal language theory to the dynamical behaviour of cellular automata, Ph.D. Thesis, Princeton University, Princeton, NJ (1988).

[33] L. Hurd, Recursive cellular automata invariant sets, Complex Systems 4 (1990) 119–129.

[34] J. Kari. The nilpotency problem of one-dimensional cellular automata. SIAM J. Comput. 21, no. 3, 571–586 (1992).

[35] J. Kari. Rice's theorem for the limit sets of cellular automata. Theoret. Comput. Sci. 127 (1994), no. 2, 229–254.

[36] P. Kůrka. Topological and symbolic dynamics. Cours Spécialisés [Specialized Courses], 11. Société Mathématique de France, Paris (2003).

[37] P. Kůrka. Languages, equicontinuity and attractors in cellular automata. Ergodic Theory Dynamical Systems 17, no. 2, 417–433 (1997).

[38] P. Kůrka. Zero-dimensional dynamical systems, formal languages, and universality. Theory Comput. Syst. 32, no. 4, 423–433 (1999).

[39] P. Kůrka. On the measure attractor of a cellular automaton. Discrete Contin. Dyn. Syst. 2005, suppl., 524–535.

[40] Z.S. Jiang, H.M. Xie. Evolution complexity of the elementary cellular automaton rule 18. Complex Systems 13, no. 3, 271–295 (2001).

[41] C.G. Langton. Computation at the edge of chaos: phase transition and emergent computation. Physica D, 42:12–37, 1990.

[42] D. Lind, B. Marcus. An introduction to symbolic dynamics and coding. Cambridge University Press, Cambridge (1995).

[43] A. Maass, On the sofic limit sets of cellular automata, Ergodic Theory Dynamical Systems 15 (1995) 663–684.

[44] A. Maass. Some coded systems that are not unstable limit sets of cellular automata. Cellular automata and cooperative systems (Les Houches, 1992), 433–449, NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., 396, Kluwer Acad. Publ., Dordrecht, 1993.

[45] M. Mitchell, J.P. Crutchfield, P.T. Hraber. Evolving cellular automata to perform computations: mechanisms and impediments. Physica D, 75:361–369, 1994.

[46] G.Manzini, L.Margara. A complete and efficiently computable topological classification of D-dimensional linear cellular automata over Z_m. Automata, languages and programming (Bologna, 1997), 794–804, Lecture Notes in Comput. Sci., 1256, Springer, Berlin, 1997.

[47] M. Nasu, Textile Systems for Endomorphisms and Automorphisms of the Shift. Volume 546 of Mem. Amer. Math. Soc., 1995.

[48] J. von Neumann. Theory of self-reproducing automata. Univ. of Illinois Press, Urbana (1966).

[49] N.H. Packard. Adaptation toward the edge of chaos. In J.A.S. Kelso, A.J. Mandell and M.F. Shlesinger, editors. Dynamic Patterns in Complex Systems, pages 293–301. World Scientific, 1988.

[50] W. Parry, Intrinsic Markov chains, Trans. Amer. Math. Soc. 112 (1964), 55–66.

[51] Schreckenberg. D.E. Wolf (Eds.). Traffic and Granular Flow'97. World Scientific. Singapore, 1998.

[52] A.R. Smith III. Simple computation-universal cellular spaces. J. Assoc. Comput. Mach. 18 (1971), 339–353.

[53] M.A. Shereshevsy. Expansiveness, entropy and polynomial growth for groups acting on subshifts by automorphisms. Indag. Math. 4 (1993) 203–210.

[54] J. W. Thatcher. Universality in the Von Neumann cellular model. In Burks, pages 132–186 (Essay Five).

[55] A. Turing. On computable numbers, with an application to the Entscheidungsproblem, Proceedings of the London Mathematical Society, Series 2, 42 (1936), pp 230–265.

[56] B. Weiss. Subshifts of finite type and sofic systems. Monatsh. Math. 77, 462–474 (1973).

[57] S. Wolfram. Theory and Application of Cellular Automata. World Scientific, Singapore, 1986.

[58] S. Wolfram. Twenty problems in the theory of cellular automata. Physics of chaos and related problems (Graftåvallen, 1984). Phys. Scripta 1985, Vol. T9, 170–183.

[59] S. Wolfram. Computation theory of cellular automata. Comm. Math. Phys. 96 (1984), no. 1, 15–57.

[60] S. Wolfram. Cellular automata as models of complexity. Nature, Vol. 311, No. 4, 1984, pp. 419-424.

[61] S. Wolfram. Universality and complexity in cellular automata, Physica D, 10:1–35, (1984).